

January 2021

Secure Automatic Speaker Verification Systems

Muteb Aljasem
Wayne State University

Follow this and additional works at: https://digitalcommons.wayne.edu/oa_dissertations



Part of the [Electrical and Computer Engineering Commons](#)

Recommended Citation

Aljasem, Muteb, "Secure Automatic Speaker Verification Systems" (2021). *Wayne State University Dissertations*. 3529.

https://digitalcommons.wayne.edu/oa_dissertations/3529

This Open Access Dissertation is brought to you for free and open access by DigitalCommons@WayneState. It has been accepted for inclusion in Wayne State University Dissertations by an authorized administrator of DigitalCommons@WayneState.

SECURE AUTOMATIC SPEAKER VERIFICATION SYSTEM

by

MUTEB ALJASEM

DISSERTATION

Submitted to the Graduate School,

of Wayne State University,

Detroit, Michigan

in partial fulfillment of the requirements

for the degree of

DOCTOR OF PHILOSOPHY

2021

MAJOR: ELECTRICAL ENGINEERING

Approved By:

Advisor

Date

DEDICATION

To my parents, wife, children, my grand mother, and friends.

ACKNOWLEDGEMENTS

I would like to express my sincere appreciation to Professor Malik Hafiz, professor Mohammad Mehrmohammadi and professor Aun Irtaza, who contributed tremendous time to my research. Appreciation is also due to Professor Lubna Alazzawi, Professor Mumtaz Usmen, and Professor Caisheng Wang for their constructive comments and valuable suggestions. Special thanks to my parents, my brothers, my sister and friends especially Bryan Terrace for their love and continuous support. Most of all, I would like to express my appreciation to my wife and my three lovely children for their love and encouragement.

TABLE OF CONTENTS

Dedication	ii
Acknowledgements	iii
List of Tables	vii
List of Figures	viii
Chapter 1 Introduction	1
1.1 Problem Statement	3
1.1.1 Research Questions	3
1.2 Research Objectives	4
1.3 Research Contributions	6
1.4 Thesis Organization	8
Chapter 2 Background and Literature Review	10
2.1 Automatic Speaker Verification Systems	10
2.2 ASV Spoofing Challenge	11
2.2.1 ASVspoof 2015	13
2.2.2 ASVspoof 2017	15
2.2.3 ASVspoof 2019	16
2.3 Machine learning and Speech recognition	18
2.4 Ensemble Classifications	20
2.4.1 Bagging	21
2.4.2 Boosting	21
2.4.3 Stacking	22
2.4.4 Random Subspace	22

2.5	Machine Learning Algorithms in Python with scikit-learn	23
2.5.1	Scikit-learn	23
2.6	ASV System Operations and Feature Extraction Methods	24
2.6.1	The Automated Speaker Verification System (ASV) Operations	24
2.7	Feature Extraction	25
2.8	Feature Extraction Methods	26
2.8.1	Mel Frequency Cepstral Coefficients (MFCCs)	26
2.8.2	Linear Prediction Cepstral Coefficients (LPCC)	31
2.8.3	Perceptual Linear Prediction (PLP)	33
2.9	Literature Review	34
2.10	Details of Specific Approaches	37
2.11	Limitations of the Existing Approaches	41
Chapter 3	Architecture of Secure Automatic Speaker Verification System (SASV) .	42
3.1	Aims of SASV System	42
3.2	Overview of SASV Framework	43
3.3	Operational Contributions	44
3.4	Operational Architecture	46
Chapter 4	Proposed Method	48
4.1	Overview of ALTP features	48
4.2	Limitations of ALTP Features	50
4.3	Motivation for the sm-ALTP Features	51
4.4	sm-ALTP Features	51
4.5	Classifier Comity Learning for Ensembles	54

4.5.1	Training-Phase—Asymmetric Bagging and Subspace Sampling	55
4.5.2	Weighted Normalized Voting Rule (wNVR)	56
4.5.3	Testing Phase	57
4.6	Overcoming the Limitations of Existing Approaches	57
Chapter 5	Experiments and Results	59
5.1	Dataset	59
5.2	Experiment I—Performance Evaluation for Speaker Verification	60
5.3	Experiment II—Audio Synthesis Algorithm Detection	63
5.4	Experiment III—Performance Evaluation for Compromised Speaker Identification	63
5.5	Cross-Dataset Evaluation	66
5.6	Replay Attack Detection	70
5.6.1	Replay and Cloned Replay Patterns	70
5.6.2	Replay and Cloned Replay Attack Detection	71
5.7	Comparison Against Other Feature Extraction Approaches	72
5.8	Comparison Against State-of-the-art Methods	74
Chapter 6	Conclusion and Future Work	76
6.1	Conclusion	76
6.2	Future Work	77
References	79
Abstract	91
Autobiographical Statement	93

LIST OF TABLES

Table 1	Number of non-overlapping target speakers and utterances in the ASVspoof 2019 database’s training and development sets.	60
Table 2	Details of Voice Spoofing Detection Corpus (VSDC)	61
Table 3	Performance of the proposed method for bonafide Speaker Verification	62
Table 4	Performance evaluation of the proposed method for synthetic algorithm recognition	65
Table 5	Speaker identification whose voices are used to attack the system with a certain audio synthesis algorithm	66
Table 6	By training on the LA-training set and testing on the LA-development and LA-evaluation sets, performance evaluation for unseen speakers and seen/unseen algorithms may be achieved.	68
Table 7	Cross dataset validation using unseen algorithms of the LA-evaluation set.	69
Table 8	The PA-evaluation set of ASVspoof 2019, as well as the VSDC dataset, were used to assess performance for replay- and cloned replay attack detection.	70
Table 9	Comparison against other feature extraction approaches using VSDC, LA- and PA-training sets of ASVspoof 2019.	71
Table 10	Comparison against state-of-the-art method on LA and PA evaluation sets of ASVspoof 2019.	72
Table 11	ASVspoof 2019 top 10 teams in LA and PA scenarios are compared to the suggested approach.	73

LIST OF FIGURES

Figure 1	Procedure to compute MFCC features.	27
Figure 2	The original Signal in the Time Domain.	28
Figure 3	The Signal in the Time Domain after Pre-Emphasis.	29
Figure 4	Hamming Window in the time domain and frequency domain.	29
Figure 5	Processing steps involved in LPCC computation.	32
Figure 6	Processing steps involved in PLP computation.	34
Figure 7	Original (a) and cloned (b) utterances are analyzed spectrally. Vertical lines that only occur in copied audios can be used as a possible cloning attack hint. By using neighborhood statistics, these lines may be recorded.	40
Figure 8	Block diagram of Secure ASV (SASV) system.	43
Figure 9	Detailed architecture of SASV system.	45
Figure 10	sm-ALTP representation (eq. (4.21)) for genuine and cloned audios.	56
Figure 11	Effect of the dynamic threshold over the audio frames.	58
Figure 12	Confusion matrix analysis for voice cloning and voice algorithm detection using 70-30 ratio	64
Figure 13	Confusion matrix analysis for voice cloning and voice algorithm detection using 30-70 ratio	64

CHAPTER 1 INTRODUCTION

With the emergence of smart speakers such as Amazon Alexa, Google Home, Google Assistant etc., and Internet of Things (IoT), many voice-enabled technologies are becoming a main part of our lives. The voice-enabled technologies are used in smart homes, consumer electronics, automotive industry, e-commerce, etc. On the one hand, these technologies have eased consumers/customers' lives and on the other hand they increased the revenue for the businesses. For instance, due to ease of use through the voice-based searching product, searches became more targeted thus, contributing to results in the form of more profit for businesses. In the meantime, using these devices to store very private and sensitive information has placed them under direct threat and made them an ideal target for attackers as well. The voice-based technologies authenticate users through the voice signatures by capturing the unique vocal profile of the users [45].

The value of the global voice biometrics market is expected to reach \$3.91 billion by 2026 [45]. All voice-biometric applications such as phone banking, voice-enabled devices, credit card usage, and multimedia forensics, etc. consider the automatic speaker verification (ASV) as a mandatory component of their applications. While speaker identification systems identify the speaker's voice amongst a dataset of other speaker voices, automatic speaker verification systems verify the speaker's voice by matching it to one voice print [45]. However, these systems are in continuous threat due to various audio spoofing attacks i.e., audio synthesis (voice cloning), voice replays, voice morphing, and voice mimicry attacks [69].

The ASV systems need to be robust against any intentional attacks that can either

be launched through voice cloning (logical access or LA) or replays (physical access or PA) [86]. Attacks through text-to-speech synthesis [69] and voice conversion [9, 16] are classified as voice cloning attacks. Whereas, voice replay is used as a common mean for impersonation [21]. Voice cloning is the process of generating the voice of a target user by using artificial intelligence algorithms (i.e., machine learning and deep learning models) through the pre-recorded voice samples. Afterwards, the text-to-speech conversion is performed to synthesize the voice of the target user to read any text. Voice conversion is to modify the acoustical signal of anyone's voice in the form of the target person's voice by transforming the source signal properties. The converted voice appears as if it was spoken by the target speaker and it can make the ASV systems vulnerable to spoofing attacks [24].

A replay attack is to use a prerecorded speech of the target speaker to deceive the ASV systems to penetrate in the system [24, 56]. As the voice samples belong to the real target speaker, therefore, the ASV systems are unable to distinguish them as a spoofing attack. However, the microphone characteristics and microphone artifacts can be identified to determine the sample as a genuine or replayed voice. Thus, due to the spoofing attacks the maximum application benefits of the ASV systems are initially far from reach.

Several solutions for combating voice spoofing attacks have been proposed in this research. In this context, three ASVspoof community-led challenges were established : ASVspoof2015, ASVspoof2017 and ASVspoof2019 in order to encourage the development of countermeasures to defend ASV systems against such attacks [71]. The proposed defenses systems have been designed to integrate countermeasures with ASV in an embedded manner, this can be accomplished by placing the step of the countermeasure followed by ASV, or vice versa, or in parallel [35]. Spoofing detection was carried out in all of

these systems using various characteristics and classifier integration, with spoofing detection being treated as a binary classification issue [85]. These methods begin by generating audio representations using various combinations of features. The binary classifiers then determine if the input audio is bonafide or spoofed.

1.1 Problem Statement

The spoofing detection approaches rely on classification outcome that usually comes through two different modules i.e. voice cloning detection, and replay attack detection. In this regard, voice cloning detection modules ignore altogether the channel artifacts that are natural to come in real-time settings and can cause the spoofing detection a failed task. Similarly, the voice replay detection modules can also be deceived by playing a cloned audio before a microphone in real time. In this regard, the second-order non-linearity will not become the part of replayed voice, thus voice replay detection will also fail. Moreover, the heterogeneous design of spoofing detection modules is also not effective as these modules cannot stand alone to prevent spoofing attacks. Additionally, there is no research effort has been done so far to obtain the clues about a potential counterfeiter.

1.1.1 Research Questions

In order to overcome the above-mentioned problems, following research questions are explored in this dissertation:

1. Can a comprehensive framework based on a single model be developed to detect various types of spoofing attacks?
2. In case of LA attacks, is it possible to analyze the voice signal to obtain the clue about the underlying voice cloning algorithms?

3. If voice cloning algorithms are detectable then can this information be utilized for the counterfeiter identification?
4. Can a single model based framework be developed in a way to perform reliably in the real-time environments in presence of enhanced attack vectors?

1.2 Research Objectives

In this dissertation, the research work was organized to address following objectives:

- **Development of a comprehensive anti-spoofing framework** - A comprehensive anti-spoofing framework based on a single model is required to address the requirements of a real-time ASV system. The real-time ASV system serves as a backbone of many voice-enabled devices and critical applications. As conventional anti-spoofing approaches ignores the dynamics of a real attack, consequently, results in form of an unreliable anti-spoofing system, which may fail during application.
- **Audio representation through novel feature extraction approach**- A powerful feature extraction approach for audio representation capable of capturing speaker as well as attack specific attributes ensures a reliable ASV system. • Attaining a more secure and reliable ASV system requires a powerful and robust feature extraction approach competent in seizing the distinct attributes of both authentic speaker and spoofing attacks. Therefore, in this research a novel feature extraction approach was emphasized that reliably discriminates between bonafide and spoofed cases.
- **Enhancement of the attack vector**- For a robust anti-spoofing system, the attack vectors is further grown as per the requirements of a real-time system. In this re-

gard, cloned replay attack detection is described in this research. The cloned replay attack detection considers real scenarios when cloned voices are passed through a microphone instead of transmitting them directly to the ASV system. The replay detection module may consider such audios as bonafide as they will not contain the second order non-linearity consequently comes through the harmonic distortions by playing a recorded audio against a microphone and serves a clue of a replay attack. Moreover, as second order non linearity traces are different than the cloning algorithm artifacts, therefore, the cloning detection module may not detect them.

- **Voice cloning algorithm detection-** The aim of the algorithm of the voice cloning detection is to analyze how artifacts are induced in the cloned audio signals. Later on this information can be utilized to identify commercial solutions which serves as a common mean for cloned audio generation. Through this discovery counterfeiter can be identified at least in the cases where a commercial solution is used for fake audio generation.

To fulfill these objectives in a way to overcome security breaches, vulnerabilities, and develop a robust and secure ASV system, a novel sign-modified acoustic local ternary pattern (sm-ALTP) feature extraction is developed and used along with asymmetric bagging based on SVM classifier that ensembles with enhanced attack vector. The machine learning algorithms were utilized to automatically secure the ASV system by only allowing identified voices and authentic speaker identities and protect the system from probable unauthorized intrusions. Features extracted from speaker data are conceived by these algorithms to be used for classifier training [48].

1.3 Research Contributions

The focus of this research work was to develop a secure ASV system that is robust enough against the various attacks mentioned above. The intent of this study was to introduce a novel feature extraction approach, i.e., sm-ALTP features for audio representation and to also improve the existing classification approaches. This study aimed to analyze the signal properties to identify the artifact traces and to capture them by observing the artifact patterns and signal properties in terms of local neighborhood. Moreover, we also aimed to analyze the vocal tract profiles of different speakers; and based on this information, the novel sm-ALTP feature extraction approach were developed. This sm-ALTP feature extraction is a development and an extension of the ALTP feature extraction that allows seizing the user's speech vocal features. In addition, the sm-ALTP uses local correlation scores to discover signal non-linearity that exists as a consequence of voice cloning or recording artifacts.

For speaker verification, and attack detection, robust classification models were developed. The support vector machine (SVM) classifier ensemble is proposed in this study to determine the vitality of the voice. The SVM-based classifier is a powerful technique for solving binary classification problems. The SVM-based classifier ensemble was created using across the feature repository, asymmetric bagging and random subspace sampling to generate a stable classifier by integrating the outputs of a group of fragile classifiers using the weighted normalized voting rule (wNVR). The new created model was used for speaker verification to detect and track down attacks of voice cloning, cloning algorithm employed to perform an attack, voice replays-, and cloned voice replay attacks

(also a novel concept proposed in this study) using the ASVSpooof 2019 dataset, and voice spoofing detection corpus (VSDC) via rigorous experimentation over standard benchmark datasets i.e., ASVspooof-2019, and voice spoofing corpus against state-of-the-art methods, and effectiveness of the proposed approach were justified. Additionally, this research also aimed to improve the effectiveness and reliability of the ASV system by identifying the scannerios and reasons that cause ASV systems to fail. Moreover, dimensionality reduction approaches and feature optimization approaches were explored to develop a reliable ASV system.

Behavior analysis of voice cloning algorithm was performed through algorithm detection to fortify countermeasures designed to offset and prevent the commercial +cloned audios. Through algorithm detection, it was clearly possible to classify and recognize the culprits based upon the severity of the attack. Furthermore, the proposed solution detects cloned replay attacks, which is a unique idea introduced in this research. By putting synthetic speech samples in front of the microphone, these cloned replays consist of recorded voice samples. Some applications, such as voice-controlled devices, are vulnerable to cloned replays, such application include: Google Home, Amazon Alexa, and Apple Siri (Baumann et al., 2021). In cloned replays, the attacker uses a recorded voice for impersonation without possessing the speaker's prerecorded voice samples. The ASV system proposed in this study was strengthened through model evaluation over the enhanced attack vector to effectively counter all possible security breaches. In summary, this study involves:

- Creating a secure ASV system capable of counteracting a multitude of audio spoofing

attacks.

- Reinforcing the ASV systems by broadening the attack vector through both cloning algorithm and cloned replay attack detections.
- Developing a novel feature extraction model for audio representation competent in seizing the distinct attributes of both authentic speaker and spoofing attacks.
- The outcome of this work has been published in one of the top IEEE journals [?].

1.4 Thesis Organization

Rest of the thesis is organized as follows:

Chapter 2—Background and Related Work: In chapter 2, we first provided the required background on ASV, which details about the ASV spoofing challenges, various feature extraction approaches, application of machine learning in speaker recognition, and ensemble based classification. Afterwards, the relevant literature on ASV spoofing detection is provided by describing the strengths and weaknesses of the existing approaches. The chapter is finally concluded by providing the details about the common limitations of the existing approaches. These limitations are then addressed by the proposed method.

Chapter 3—Architecture of Secure Automatic Speaker Verification System (SASV): In this chapter architecture of the proposed SASV system is presented. The chapter starts by first highlighting the aims of the SASV system. Afterwards, overview of the SASV framework is presented. The framework overview describes a practical scenario in which a spoofing attack can be launched and how our method will counter that attack. After the framework overview, operational contributions are discussed in which various novelties

that are the attribute of the proposed SASV system are discussed. Finally the chapter describes the processing details of the proposed SASV system.

Chapter 4—Proposed Method: This chapter starts with the discussion of the baseline ALTP features. The ALTP features suffers from several limitations that make them unreliable for the countermeasure development task. These limitations are described in the next subsection. After establishing that the ALTP features are not suitable for the ASV and countermeasure development task, proposed sm-ALTP features are described. Moreover, the justification about sm-ALTP as a reliable feature extraction approach is also provided here. After feature extraction details, the proposed ensemble-based classifier committee learning approach is described. The chapter concludes on how the proposed approach overcomes the limitations of the existing state-of-the-art approaches.

Chapter 5—Experiments and Results: This chapter provides the experimentation details that were carried out to prove the reliability of the proposed SASV system against the standard approaches. The chapter first provides the details of the datasets used for performance evaluation purposes. Afterwards, conventional attacks as well as the advanced attacks are analyzed. Through rigorous experimentation against standard approaches it is proved that the proposed method outperforms most of the approaches. Furthermore, amongst 74 teams participated in ASVspoof-2019 challenge, our method ranks amongst top 4 teams.

Chapter 6—Conclusion and Future Work: This is the final chapter of the dissertation, where our research findings and future directions are described.

CHAPTER 2 BACKGROUND AND LITERATURE REVIEW

2.1 Automatic Speaker Verification Systems

The need to verify the identity of the speaker for security issues became mandatory and eminent. Accordingly, speaker verification has been broadly utilized and implemented in many control system applications such as smart phone fraud prevention, telephone banking, and computer login [48]. The duty of the speaker verification is to determine whether the claimed identity of the speaker matches that of a specific speaker model - voice print - validated by the system [85], [48]. Voice print depends mainly on the special physical and learned characteristics of the speaker's unique type of speech that differentiate one speaker from another. The physical component is distinguished by the shape of the vocal tract that include the organs responsible for creating the speech. The learning component include the speaker's dialect, the speed of speaking, and the prosodic features. Speaker verification can be achieved through speech signal analysis, speaker features and pattern recognition algorithm. For possible verification, the users are required to store their voice prints to be used later in the speaker verification process. The automatic speaker verification system is capable of analyzing the speech signal produced from a microphone to allow the discovery of the speaker's identity and whether to reject or accept their claim [?, ?, 15]. The use of the ASV is broken down into two categories, text-dependent and text-independent. Mainly, authentication systems utilize text-dependent verification by allowing the speakers to use a ready-made text to record their voice. However, text-independent verification is appropriate for use in surveillance activities and it does not depend on a previously defined text. The ASV is based on using various technologies such as the Gaussian Mixture

Model (GMM), and Hidden Markov Model (HMM), the SVM and Artificial Neural Network (ANN). In addition, the feature extraction methods used include Mel-Frequency Cepstral Coefficients (MFCC), Linear Predictive Cepstral Coefficients (LPCC), and Perceptual Linear Predictive (PLP) [50].

2.2 ASV Spoofing Challenge

While the ASV systems are crucial for speaker's identity verification on many applications and should be reliable, they are vulnerable to the threat of spoofing. Spoofing diminishes confidence and gives advantage to illegitimate intervention. Thus, it jeopardizes credibility by not allowing a user to differentiate between an authentic and unauthentic voice. Accordingly, it became obligatory to adopt reliable countermeasures for detecting and halting such unwarranted and unjustified access [79]. As previously mentioned, there are two types of intentional attacks or spoofing threats the logical access (LA) through speech synthesis [18,69] and voice conversion [4,5,7] and the physical access (PA) through voice replay [18,21] and impersonation [18,78]. A study was conducted to detect the vulnerability of the ASV system to be attacked by speech synthesis through generating synthetic speech using HMM-based synthesizer. In this study two automated speaker verification systems, Gaussian Mixture Model-Universal Background Model (GMMUBM) and support vector machine (SVM) using Gaussian supervectors were evaluated. Both systems have shown a very low equal error rate at the presence of synthetic speech. GMM-based synthetic speech classifier (SSC) was also used to detect synthetic speech based on the relative phase shift features. The relative phase shift was used to eliminate the linear phase component and allow the phase structure to be easily clarified. This method could detect

synthetic speech up to 88% while 4.2% of the bonafide speech was incorrectly classified as synthetic [21]. Another study addressed the threat of voice conversion to automatic speaker verification (ASV) systems to overcome this threat and to enhance the security of ASV. The features derived by the phase spectrum in speech perception were used in the experiment under three different training situations. In the First situation, only GMM-based converted speech data are available, so the natural speech data was not presented in this phase. In the second situation, only unit-selection based converted speech data are presented. The third situation had natural speech data to train the converted speech model. Using the features derived from the phase spectrum showed a better performance comparing to Mel frequency central coefficients (MFCCs) features. The equal error rate (EER) was reduced from 20.20% of MFCCs to 2.35% in synthesis/converted speech detection [78]. To fight and go up against the spread of this threat, researchers and experts used their expertise and effort to find reliable solutions and develop countermeasures capable of identifying and classifying the spoofed speech of unauthorized users and distinguishing it from genuine speech of authorized users [79]. The sequential ASVspoof challenge editions were intended to find the best error free spoofing countermeasures. These editions intend to provide countermeasures through the effort of gathering and disseminating standard datasets involving a group of spoofing attacks with a multitude and multifold algorithms and a chain of well-studied evaluations [79]. For example, the ASVspoof 2015 focused on speech synthesis and voice conversion and the ASVspoof 2017 focused on the replay speech. However, ASVspoof 2019 was able to deal with all these spoofing threats and separately dealing with logical access (LA) and physical access (PA) [71].

2.2.1 ASVspoof 2015

ASVspoof 2015 was the first spoofing countermeasure of the series of the ASVspoof challenges that was designed to only detect spoofed speech in a way to reduce entry costs and optimize participation. Participation was encouraged to verify this countermeasure by developing detection algorithms and making yielding outcomes of standard dataset and protocol freely accessible [79]. To evaluate the ASVspoof 2015 unmodified, noise free spoofed speech and genuine speech were used from 106 participants. Voice conversion and speech synthesis spoofing algorithms were used to create spoofed speech. The ASVspoof 2015 dataset was divided into three subdivisions that include: training, development, and evaluation [79].

Several studies have used the development and evaluation datasets to evaluate various systems. For example, [59] performed experiments on ASVspoof 2015 using 19 speech features of voice conversion and speech synthesis spoofing. The performance of these features was evaluated using two classifiers: the Gaussian mixture model (GMM) and the support vector machine (SVM). [59]. Another study by [5] was conducted using the ASVspoof 2015 challenge evaluation and development test dataset. The spoofing attack detection model used in this study employed a system that consisted of amplitude, phase, the linear prediction residual, and combined amplitude - phase-based. In addition, various features were used in this experiment that include Mel-frequency cepstral coefficients (MFCC), product spectrum-based cepstral coefficients, modified group delay cepstral coefficients, weighted linear prediction group delay cepstral coefficients, the linear prediction residual cepstral coefficients, cosine normalized phase-based cepstral features (CNPCC), and

a combination of MFCC-CNPCC [5]. The Gaussian Mixture Model (GMM) classifier was used to differentiate bonafide and spoofed speech signals. The performance of using the different features showed that the PS-MFCC and MFCC outperformed the other models on unseen with EER the average equal error rate (EER) of 0.041% on seen spoofing attacks, 5.347% on unseen spoofing attacks, and 2.69% on unseen and seen spoofing attacks [5].

Moreover, an experiment to evaluate the Speech Technology Center (STC) systems using ASVspoof Challenge 2015 was conducted by examining various acoustic feature spaces. The aim was to build an effective model for detecting unknown spoofing attacks. The systems consist of three main components i.e. Acoustic feature extractor, TV i-vector extractor, and classifiers. For feature extraction, different methods were investigated to find the most reliable and robust countermeasures against spoofing attacks. The molar frequency cepstral coefficient (MFCC) features, Mel-Frequency Principal Coefficients (MFPC) features, the cos-phase features extracted from phase spectrum obtained by Fourier Transform and features based on applying the multiresolution wavelet transform were investigated [51]. The standard TV-JFA approach for probability modeling in spoofing detection systems was used. The linear support vector machine (SVM) classifier with a linear kernel and nonlinear Deep Belief Network (DBN) classifier with softmax output units and stochastic binary hidden units were used and their performances were compared. The fused TV systems with the combination of feature extractors based on SVM and DBN classifiers showed that the SVM-based system provided a better result than the DBN-based system with EER of 0.03% on the development dataset and on the evaluation dataset was 1.965% EER for all spoofing attacks [51].

2.2.2 ASVspoof 2017

ASVspoof 2017 was the second spoofing countermeasure designed to detect replay speech attacks. Detection of replay speech can be troublesome if associated with inconsistent and erratic quality of the replay attack and probably deceitfully recorded and collected speech that may be twisted with noise or other additives. The identification of these attacks could be constricted to a channel or background noise obstacle. However, high-level quality speech recordings with soft acoustic surroundings can be recognized as genuine. In addition, any genuine digital recordings that are copied and implanted into the ASV system can hardly be detected. While speech synthesis and voice conversion attacks are mostly performed by an experienced person, replay attacks can be performed with the least expertise possible by the means of a recording device. Accordingly, it was important to develop an effective replay attack countermeasure such as the ASVspoof 2017 capable of evaluating the drawbacks in replay threat detection and enhancing replay attack countermeasure development [36]. The ASVspoof 2017 countermeasure was utilized by [9] for constructing a replay attack detection system based on the blind estimation of the magnitude of channel responses. The model used a Gaussian mixture model (GMM) of RASTA filtered Mel-Frequency cepstral Coefficients (MFCCs) that trained on clean speech to predict the log-spectrum average of the clean speech signal. The predicted log-spectrum average of the clean signal was subtracted from the log-spectrum of the observed signal to obtain the magnitude response of the channel to estimate variations in the spectrum because the replay attack signal could be affected by different factors e.g. environment, recording device, and playback device. In this experiment, the TIMIT dataset was used for

training the log-spectrum average of the clean signal and ASVspoof 2017 challenge dataset was used during Automatic Speaker Verification Spoofing and Countermeasures [9]. Moreover, in this study, principal component analysis (PCA) was also utilized for dimensionality reduction which yields to the improvement of the system performance. A GMM classifier was used to distinguish between bonafide speech and spoofed speech. This approach was compared with two different benchmarks. These are the discrete Fourier transform power spectral (DFTspec) and the constant Q cepstral coefficients (CQCCs). This system outperformed the two methods with an equal error rate (EER) of 6.87% when testing on the development dataset and EER of 11.28% on the evaluation set [9].

2.2.3 ASVspoof 2019

The ASVspoof 2019 is the third version of the countermeasure development process to secure and safeguard the ASV systems from spoofing attacks. This new edition deals with logical access (LA) involving text-to-speech synthesis and voice conversion attacks as well as physical access (PA) involving replay spoofing attacks. The ASVspoof 2019 challenge embraces the tandem decision cost function (t-DCF) as a cost-based evaluation metric to certify that the attained scores and ranks credit the comparative effect of the spoofing attacks and countermeasures on ASV reliability and robustness [71]. According to [18], it was found that some models on the physical access dataset surpass the performance of other models, because unauthorized recorded replay speech was detected to have prolonged periods of silence than its authorized counterpart. Difficulty is experienced in the duty of physical access after eliminating such models and the t-DCF of the optimal model increases from 0.1672 to 0.5018 with an increase in the equal error rate (EER) from 5.98% to 19.8% regarding the development dataset [18].

A study was conducted to depict the outcomes of the ASVspoof database, protocols, and challenge through utilizing the effect of the LA and PA scenarios on the performance of ASV. The authors found that the detection of text-to-speech synthesis and voice conversion attacks is possible if more than one classifier is used in the countermeasure. Evaluating and controlling the replay speech threat and its countermeasure was possible using simulation. The simulation reflects on some factors such as the quality of the replay audio device, changes in echo time and room size, and the physical difference between genuine speakers and suspicious recordings on one side and the speakers and the microphone of the ASV system on the other side. ASVspoof 2019 was proven to achieve great success [71]. Another study was conducted to address the ASVspoof 2019 challenge for both physical and logical access scenarios for ASV. To counter the physical access (PA) attacks i.e. voice replays, two VGG networks were fused and trained over the power spectrogram, and constant Q-transform (CQT) features. Similarly, for logical access (LA) attacks i.e. voice conversion, and voice synthesis, the VGG network was fused with the SincNet architecture and raw audio files were used as inputs. For PA the results were good where the model got 86% improvement as compared to the baseline method. The primary reason for the PA attack detection is that the replays introduce the nonlinearities and the model was good to capture those nonlinearities. Whereas, in the case of LA, the model significantly failed to discriminate against the synthetic audios generated through the neural Waveform based synthetic audio generation algorithms [86].

2.3 Machine learning and Speech recognition

Human voice is unique and no two persons sound the same. Voice variations among speakers depend on three aspects: 1. Speaking style such as the accent of the speaker, 2. The unique vocal tract shape and vocal cords of each speaker, and 3. The method used by the speaker to convey a certain message. The speaker uses a large number of words, phrases, and syntactic sentence structures that are difficult to count or control in an experiment. The automatic speaker recognition systems can only use words and phrases to inspect the acoustic properties of a speaker's signal [25]. The two main roles of speech recognition are to verify and identify the speaker. This is achieved through a task similar to the brain's function which is the capability of the recognition system to accept speech signals, recognize and identify the speaker and remember the speaker for future recognition. The process of speech recognition starts after receiving a speech by performing three tasks which are: acoustic processing, feature extraction, and then recognizing the speaker [6]. To verify a speaker is to ascertain that the heard voice belongs to a special enrolled speaker and in this case, the speaker has to claim an identity which is validated by the system. To identify a speaker is to link an unspecified voice to one of the speakers in the enrolled group and in this case a voice sample is provided by the speaker without the need for a claimed identity. The system will then specify the speaker related to the voice sample from an identified group of enrolled speakers [25].

Both automatic speech recognition and machine learning paradigms affect each other. Studies demonstrate that automatic speech recognition researchers tend to use machine learning to support theoretical results with mathematical calculation and applications

[23]. Machine learning approaches allow controlling huge speech databases to handle variations with high quality details and improve performance. Thus, machine learning is vital for developing the automatic speech recognition technology and enhancing its accuracy. Machine learning use automatic speech recognition as a platform for evaluating the strength of developed techniques to solve problems associated with speech sequential properties [4]. The development of automatic speech recognition is always associated with advancement in machine learning methodologies that are successful in modeling profound and dynamic structures of speech. These advanced methodologies are capable of controlling complex interference of acoustic environmental factors with speech and handling sequential data. Accordingly, the role of machine learning is to enhance the capability of the automatic speech recognition systems to generalize through recognizing previous perceived examples using functional dependencies between random input and output domains. Automatic speech recognition is a basic machine learning problem utilized to convert the speaker's speech sequence data into linguistic fabric [23]. This is made possible by using inputs of continuous acoustic sequence data such as sound waves and outputs of categorical label sequence such as words or phrases to identify the new output sequence from the input used. If the temporal segment boundaries of the output labels are identified, this task is known as phonetic classification, and if not, then this task is known as phonetic recognition. Phonetic classification contains phone boundaries in training and testing data, however the absence of these boundaries in phonetic recognition makes it much tougher [23, 48].

2.4 Ensemble Classifications

In the realm of machine learning and artificial intelligence, the ensemble systems known as multiple classifier systems have received special attention over the past three decades. These ensemble systems have demonstrated effectiveness, flexibility, and adaptability in a wide array of problem areas and real-world applications. The purpose for using an ensemble system is to maximize the accuracy of an automated system by decreasing its variance. The ensemble system is effectively used to solve many of the machine learning problems such as that associated with a feature to be selected or one that is missing, class-imbalanced data, or errors to be corrected [87].

Ensemble systems are meant to create a group of classifiers with comparatively fixed or even similar bias and integrate their outputs to decrease variance as much as possible. This is because, bias which is the extent of a classifier's accuracy and variance which is the classifier's precision when trained by various training sets, are the two components of any classification error and they have an inverse relationship. There are several types of ensemble-based algorithms, but usually, ensemble members are used in either classifier fusion or classifier selection settings. Classifier selection involves training classifiers as local experts in some local neighborhood of the feature space, while classifier fusion involved training all classifiers across all of the feature space and then joining them to create a compound classifier that operates with reduced variance and consequently with less error [87]. There are various methods for constructing ensembles using classifier fusion, however, bagging, boosting, stacking, and random subspace are the most used techniques [75].

2.4.1 Bagging

The word ‘bagging’ was coined by Leo Breiman and it is the short term of Bootstrap Aggregation [13]. Bagging is an ensemble-based method that proved to be effective for individual high-variance classifiers [72]. Bagging trains the same algorithm several times by sampling with replacement and allows the aggregation of the decisions of these classifiers [52,75]. The variations within the bootstrapped replicas guarantee the diversity in the ensemble [87]. Each one of the produced samples is individually trained forming a number of separate decisions, and classification is carried out by integrating these decisions together using multiple voting to give one output prediction [37, 75]. Through bagging, classification accuracy is developed and improved due to the reduction of classification error variance [75]. Accuracy can be remarkably enhanced through bagging if a significant change can result in the created predictor when the learning set is perturbed [13,75].

2.4.2 Boosting

Boosting was introduced by Robert Schapire’s in his study ‘The strength of weak learnability’ for the 30th annual symposium on foundations of computer science [87]. In Schapire’s study, the boosting technique was used for converting an ensemble of weak classifiers to an extremely strong one; thus, leading to a boost in the accuracy of the algorithm and arbitrarily reduction in training error [65, 87] . Similar to bagging, boosting uses majority voting to integrate an ensemble of weak classifiers. However, while bagging involves using bootstrapped replicas or instances of training data to train individual classifiers; thus, giving the opportunity to each instance to be in each training dataset, boosting is related to the focus of training datasets for every classifier on fixing prediction error by prior gen-

erated classifiers. Boosting is effective for high-bias classifiers that adapt to new data very slowly and it is designed for problems of binary classification [72, 87].

2.4.3 Stacking

Stacking is an ensemble technique that involves making stacks of machine learning models or base learners by putting one on top of the other. In other means, it is a method used to train several models together, then a meta-classifier is trained to get a final prediction output learner [37, 75]. This technique can result in some errors. In stacking, the outcome of the first individual predictions forms the first or the base layer of machine learning models and is used as the following training data. Another layer is stacked on top of the base layer forming a second layer that is called a meta learner and so on [37].

2.4.4 Random Subspace

The random subspace ensemble method is a parallel learning algorithm that depends on using various feature subsets and the dimension of the subspaces is a parameter of this method [30, 52, 87]. Random sampling resulting from the original high dimensional feature vector creates a set of low dimensional subspaces that allows discrepancy reduction. Moreover, the multiple classifiers built in this method are integrated together in the final decision [73]. This method is suitable for parallel implementation for fast learning and it is based on training each member with all of the training examples, however with a subset of the attributes. The resulted prediction of the random subspace method is the average of all predictions involved [52].

2.5 Machine Learning Algorithms in Python with scikit-learn

Python is one of the most popular, object-oriented, non-compiled programming languages that was created by Guido van Rossum and released in 1991. Python is the most attractive option when it comes to algorithmic development and data analysis and the most used by data scientists and software developers in academic as well as industry settings. That is due to its extremely interactive nature and its developing and trustworthy ecosystem of scientific libraries [54]. A large number of applications that are related to the use of Python include the development of software, websites, games, scientific computation, and graphical user interface among others. The 2.x and 3.x are the main versions of Python series. While the 2.x version was widely used, it was intended to end in 2020. However, the 3.x version, which is developed from the 2.x version, is considered the future version of Python. Because Python is not a compiled language, its interpreter converts the script to binary in real time while the code is implemented. All numerical computation used by Python are provided through core external packages that are broadly accepted by the Python community such as NumPy, SciPy, Pandas, Matplotlib, and Scikit-learn. Moreover, the Jupyter Notebook is used for Python as a user-friendly interface suitable for most data analysis. Python, its core packages, and the Jupyter Notebook can be acquired and installed via the anaconda website that possesses installers for Linux, MacOS, and Windows [28].

2.5.1 Scikit-learn

Scikit-learn, an open-sourced Python programming language for machine learning, aims to develop up-to-date implementations of a large number of distinguished machine-

learning algorithms and at the same time keep a user-friendly interface [54]. The many features of scikit-learn made it highly beneficial and the mostly required software for applications related to machine learning. One of these features is that the Scikit-learn package has an extensive coverage of machine learning methods that are mainly based on compiled binary libraries that were programmed in C, C++, and Fortran. In addition, Scikit-learn can enhance the machine learning algorithm for computation efficiency through its binary-based implementations. Scikit-learn is powerfully supported by the community for issues such as bug monitoring, documentation, quality assurance. Moreover, Scikit-learn provides a unified input/output data usage and a steady model fitting procedure allowing easy switching from one method to another [28, 54].

2.6 ASV System Operations and Feature Extraction Methods

2.6.1 The Automated Speaker Verification System (ASV) Operations

Automatic speaker verification (ASV) system has two main phases of operation. The first phase is the speaker voice enrollment. At this phase, the speaker voice signal is acquired through microphone then acoustical features are extracted from the speech input. These features are used to create a speaker model. The speaker model is stored into database for later use at verification phase. The second phase is the verification phase which involves the speech signal of a speaker that is given to the system through a microphone then feature extraction is done. The acoustical feature of the speech signal is then compared with the same speaker model and score of similarity is computed. If the score is within a chosen threshold, the access is granted, otherwise, the ASV system will reject the access. In the first step feature extraction is performed, where the raw speaker acoustic

signal is converted into a sequence of acoustic feature vectors carrying characteristic information of the speaker's voice. The most commonly used feature extraction methods are Mel Frequency Cepstral Coefficients (MFCC) [20, 31], Perceptual Linear Prediction Cepstral (PLPC) Coefficients [29], and Linear Prediction Cepstral Coefficients (LPCC) [44]. These methods are based on spectral information and will be discussed in more detail later. The second step of ASV consists of the classification of the acoustic feature vectors to make the final decision. There are different classifiers that can be used such as ANN, SVM and Logistic Regression.

2.7 Feature Extraction

The first step in the ASV system is to acquire new acoustic vector features from the original audio signal i.e., processing the speech signal to identify the informative features and discard all uninformative components by removing noise. Feature extraction aims to interpret and exhibit a speech signal using a previously established set of the signal components. By doing so, the significant information is extracted, while the irrelevant, ineffective, and unmanageable information can be removed for an easy identification task. The execution of feature extraction is achieved through the front-end signal processing that modifies the speech waveform into a type of parametric representation for later development and analysis. This process converts the speech signal to a more compressed, reliable, and distinguishable representation which significantly affects the quality of the subsequent features i.e., pattern matching and speaker modeling and contribute to acceptable classification. Feature extraction is used in the ASV systems to detect a representation that is relatively reliable for all different forms of the speech signal. This representation can

maintain the part that carries the characteristics of the information in the speech signal, regardless of the changes in the environmental factors or speaker [6].

It is apparent that feature extraction represents the most significant part of speaker recognition and it decreases the magnitude of the speech signal without affecting its power. Through the means of feature extraction, a multidimensional feature vector can be generated for each speech signal. There is a variety of techniques that parametrically represent the speech signal for the recognition and verification process such as mel frequency cepstral coefficients (MFCC) [20, 31], linear prediction cepstral coefficients (LPCC) [44], and perceptual linear prediction cepstral (PLPC) coefficients [29] and they will be discussed in the following section. All these techniques have been extensively investigated and proven to be reliable and accepted in various applications. A variety of modifications have been performed in many studies on these techniques to deliver more robust, time-effective, and noise-free outcomes. While it is not easy to rank these techniques in terms of the superiority of one over the other, selecting a technique depends mainly on the area of application used [6].

2.8 Feature Extraction Methods

The feature extraction techniques described in this section are Mel Frequency Cepstral Coefficients (MFCC) [20], Linear Prediction Cepstral Coefficients (LPCC) [44], and Perceptual Linear Prediction (PLP) [29].

2.8.1 Mel Frequency Cepstral Coefficients (MFCCs)

Mel Frequency Cepstral Coefficients (MFCCs) technique is commonly used in automatic speaker verification (ASV) and automatic speech recognition (ASR) [31]. (MFCC)

is one of the widely used feature extraction techniques that was presented by Davis and Mermelstein in the 1980s and has been primarily recommended for spotting monosyllabic words in continuously spoken sentences. This technique is popularity for its effective computation and nature of the Mel scale that resembles the functions of the human auditory system [6,62]. The features of this technique simulate the variation of the human auditory system's bandwidths with filters spaced linearly when frequencies are low and spaced logarithmically when frequencies are high. These filters are used to maintain the phonetically vital properties of the speech signals which incorporate tones with different frequencies and computed subjective pitch with the Mel scale. The Mel scale exhibit linear frequency spacing when the frequency is 1000 Hz and logarithmic spacing when the frequency is above 1000 Hz [6,57].

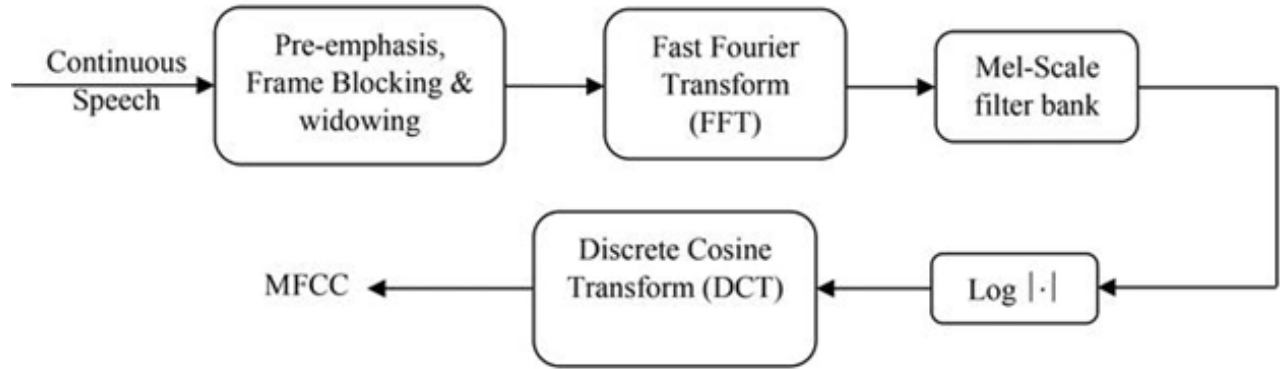


Figure 1: Procedure to compute MFCC features.

The MFCC diagram illustrated in figure 1 shows the processes associated with obtaining the required coefficients. MFCC is capable of representing low frequency regions more effectively than high frequency regions. Therefore, computing formants in low frequency range can be achieved along with the vocal tract resonances [6].

The first step of MFCC is to apply a pre-emphasis filter on the audio signal using the

first-order filter in equation 1 to amplify the high frequencies and balance the frequency spectrum as shown in figure 3, because the magnitudes of high frequencies are usually smaller than lower frequencies.

$$y(t) = x(t) - \alpha x(t) \quad (2.1)$$

Where α is the filter coefficient and its typical value is 0.95 or 0.97. Figure 3 depicts the original audio signal in the time domain:

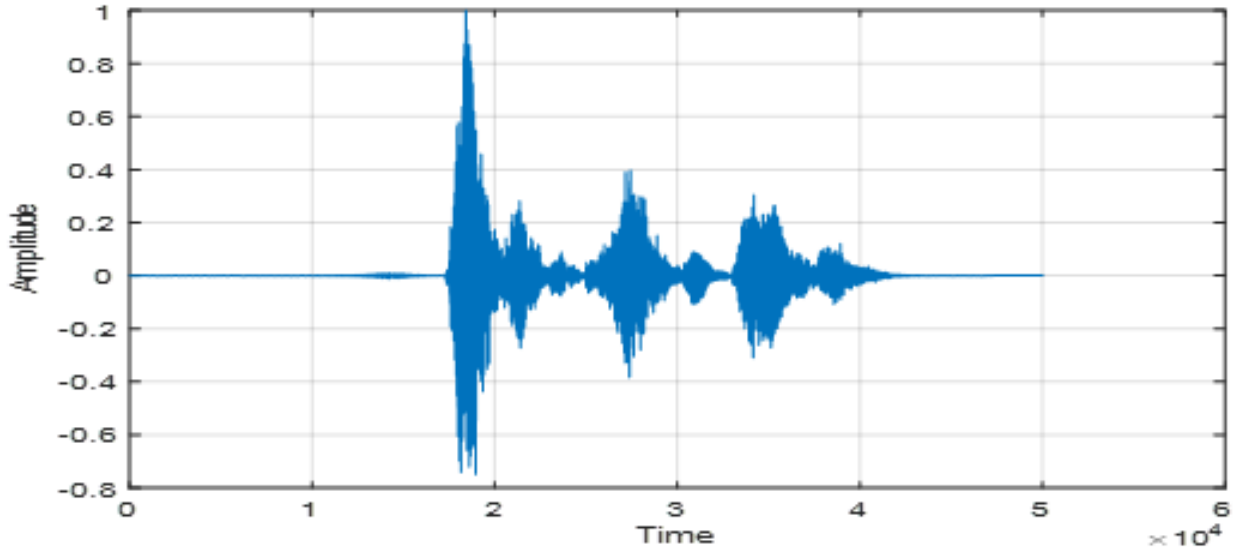


Figure 2: The original Signal in the Time Domain.

After pre-emphasis, the audio signal is split into short-time frames because the frequencies of the signal constantly and statistically change over time and if the Fast Fourier Transform is directly applied to the entire signal, the frequency contours of the signal over time will be lost.

After framing the signal into short time frames, a window function such as the Hamming window is applied to each frame to smooth the signal transition and minimize the

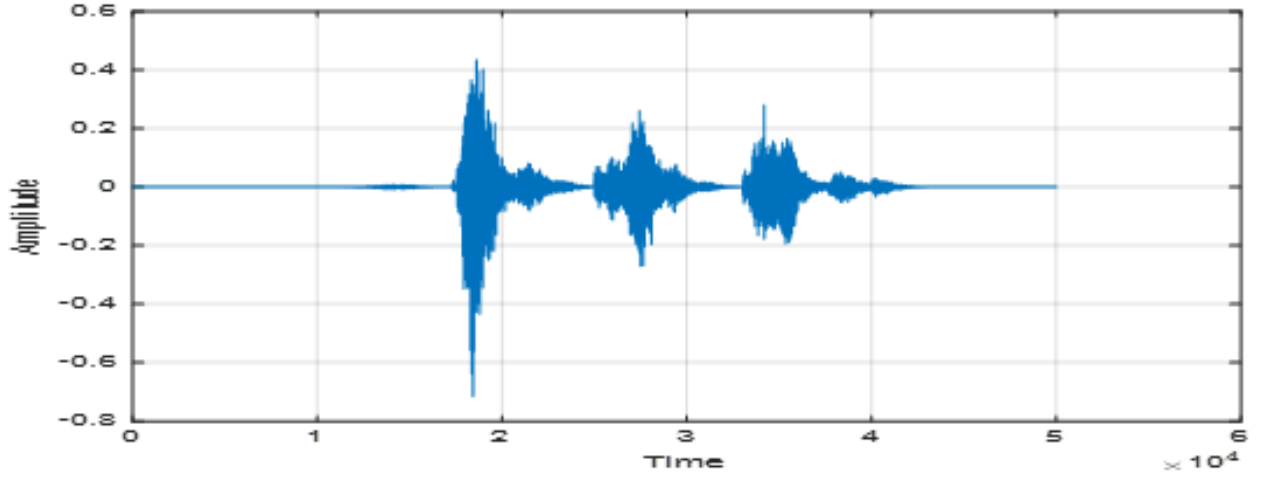


Figure 3: The Signal in the Time Domain after Pre-Emphasis.

impacts of FFT over non-integer values. The following equation is the Hamming window formula:

$$w(n) = 0.54 - 0.46\cos\left(\frac{2\pi n}{N}\right), \quad 0 \leq n \leq N \quad (2.2)$$

The window length is $N+1$. The plot (figure 4) below is the result of the plotting equation 2.2. Fast Fourier Transform is applied to every single frame to estimate the frequency

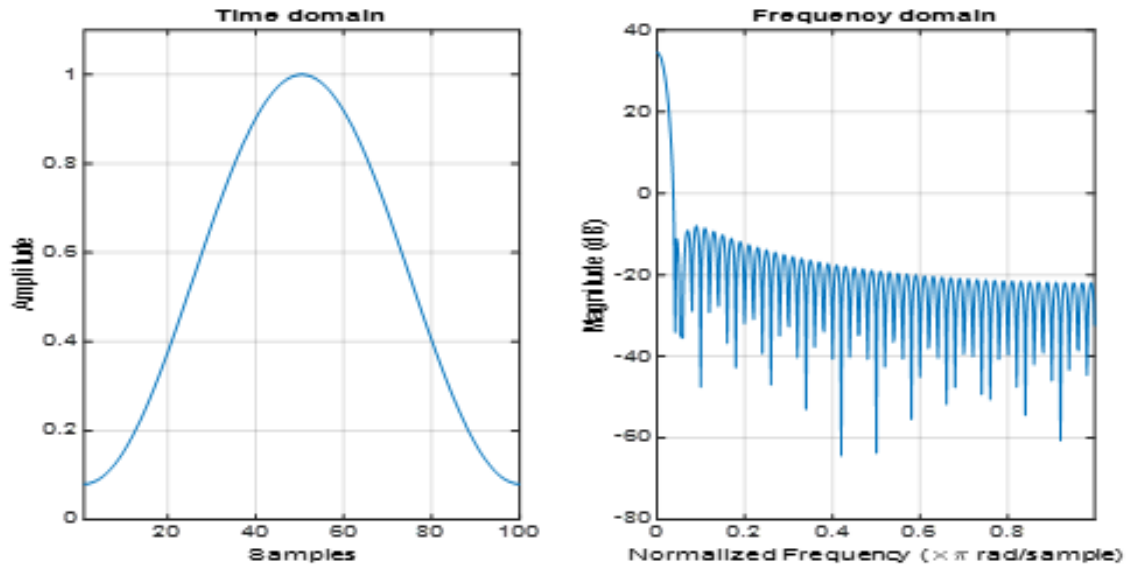


Figure 4: Hamming Window in the time domain and frequency domain.

spectrum, and then the power spectrum or also called Periodogram estimate of the power spectrum is computed by using equation 2.3.

$$P_{spec} = \frac{|FFT(X_I)|^2}{N} \quad (2.3)$$

The Mel-spaced filter banks (a set of 20-40 triangular filters) are applied on Mel-scale to the periodogram power spectral estimate to extract frequency bands. Equations 4 and 5 are used to convert Hertz (f) and Mel (m):

$$m = 2595 \log_{10} \left(1 + \frac{f}{700} \right) \quad (2.4)$$

$$f = 700 \left(10^{\frac{m}{2595}} - 1 \right) \quad (2.5)$$

Each filter in the filter bank is triangular which starts at the first point, reaches its peak, and linearly decreases to zero. Equation 2.6 is used to calculate the filter bank.

$$H_m(k) = \left\{ \begin{array}{ll} 0 & k < f(m-1) \\ \frac{k-f(m-1)}{f(m)-f(m-1)} & f(m-1) \leq k \leq f(m) \\ \frac{f(m+1)-k}{f(m+1)-f(m)} & k > f(m+1) \end{array} \right\} \quad (2.6)$$

Where M is the desired number of filters, and f() is $M + 2$ Mel-spaced frequencies.

Discrete Cosine Transform (DCT) is applied to the filter banks to decorrelate the filter bank coefficients because they are highly correlated, which could be a problem for some classifiers and then result in cepstral coefficients. 2-13 cepstral coefficients are kept, and the other coefficients are discarded because they quickly change in the filter bank coeffi-

cients and do not contribute to Automatic speaker verification (ASV).

Dynamics features namely differential (delta) and acceleration (delta delta) coefficients can be calculated by using equation 2.7. These features can improve ASV performance because the mfcc feature vector consists only of the power spectral envelope of a single frame.

$$d_t = \frac{\sum_{n=1}^N \left(c_{n+1} - c_t - n \right)}{\left(2 \sum_{n=1}^N n_2 \right)} \quad (2.7)$$

2.8.2 Linear Prediction Cepstral Coefficients (LPCC)

Linear Prediction Cepstral Coefficients (LPCC) is a widely used method that captures speaker's specific information by modeling vocal tract characteristics and is capable of eliminating excitation parameters in speech [47, 67]. Thus, the appropriate data size of LPCC allows speech compression through the digital channel. LPCC are derived from the linear prediction coefficients (LPC) through the Fourier transformation illustration of the logarithmic magnitude spectrum. LPC is a powerful speech analysis method known as a formant estimation method which is found effective in encoding high-quality speech at low bit rate [6]. In the area of speech processing and speaker verification, cepstral analysis is usually employed for its potential to thoroughly represent speech waveforms and characteristics with a fixed size of features [6].

Since the LPCC is used to estimate the parameters of an acoustic signal [22 26], so it can predict a sample as linear combination of past acoustic samples. Figure 6 illustrate the LPCC flow diagram: The acoustic signal is firstly pre-emphasized to boost up the energies in the high frequencies by equation 2.8 because the energies in low frequencies are

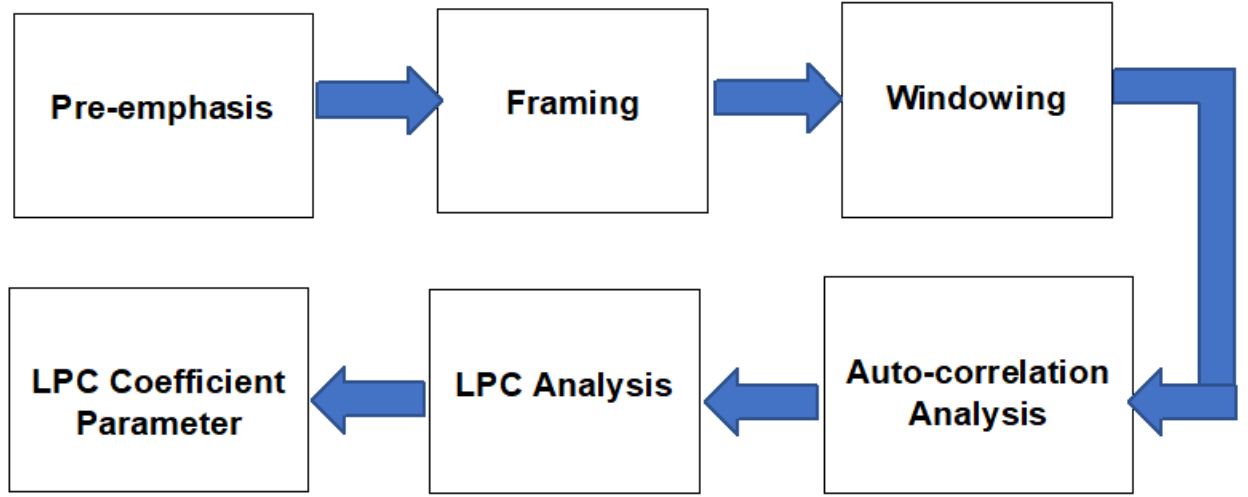


Figure 5: Processing steps involved in LPCC computation.

distributed more than the energies in high frequencies.

$$y(n) = c(n) - \alpha x(n - 1), \quad 0.9 < \alpha < 1.0 \quad (2.8)$$

where α is a constant of pre-emphasis filter. The signal is then framed into a number of frames and these frames are overlap, so no signal is lost. Each frame is windowed with some window function such as a hamming window function to minimize discontinuity of the signal frame from beginning to the end of each frame. Equation 2.9 gives a signal $y(n)$ resulted from windowing a frame and $w(n)$ is the window function:

$$y_i(n) = x_i(n)w(n) - \alpha x(n - 1), \quad \text{where } 0 \leq n \leq N - 1 \text{ and } i = 1, 2, 3, \dots, N \quad (2.9)$$

Next step is auto-correlation analysis toward each $y_i(n)$ using equation 2.10:

$$r_i(m) = \sum_{n=1}^N y_i(n)y(n+m), \text{ where } m = 0, 1, 2, \dots, p \quad (2.10)$$

P is an LPC order and has values within 8 and 16.

Next step is to convert each frame from $p+1$ auto-correlation into LPC parameter using equations 2.11 and 2.12:

$$k_m = \frac{\left\{ r(m) - \sum_{j=1}^{m-1} a_j^{m-1} r(|m-j|) \right\}}{E^{m-1}}, \quad 1 \leq m \leq p \quad (2.11)$$

$$a_j^m = a_j^{m-1} - k_m \alpha_{m-1}^{(m+1)}, \quad 1 \leq j \leq m-1 \quad (2.12)$$

where $\alpha_m = \alpha_m^p$ for $m = 1, 2, \dots, p$, $r(0)$ represent auto-correlation result, $E(m)$ represents an error a_j^m represents the coefficient's prediction and k_m is a rebound of the coefficient [61].

Finally, the LPCC is derived from LPC parameter using the following equations:

$$c_m = \alpha_m + \sum_{k=1}^{(m-1)} \left(\frac{k}{m} \right) c_k a_{m-k}, \quad 1 \leq m \leq p \quad (2.13)$$

$$c_m = \sum_{k=1}^{(m-1)} \left(\frac{k}{m} \right) c_k a_{m-k}, \quad m > p \quad (2.14)$$

2.8.3 Perceptual Linear Prediction (PLP)

In 1990, Herman Sky has introduced the Perceptual Linear Predictive (PLP) analysis to approximate voice features by an retrogressive all-pole mode [27]. This method is based

on the short-term spectrum of speech and uses three different psycho-acoustic of hearing concepts i.e., spectrum spectral resolution crucial band, equal loudness curve, and low intensity power [8, 38]. Figure 6 shows the flow of PLP. PLP is based on the nonlinear bark scale and is one of the techniques that have been prominently used in the tasks of speech and speaker recognition. PLP technique provides minimal resolution at high frequencies that represents an auditory filter bank-based application, as well as producing orthogonal outputs comparable to the cepstral analysis. This technique also integrates linear predictions and spectral analysis [6].

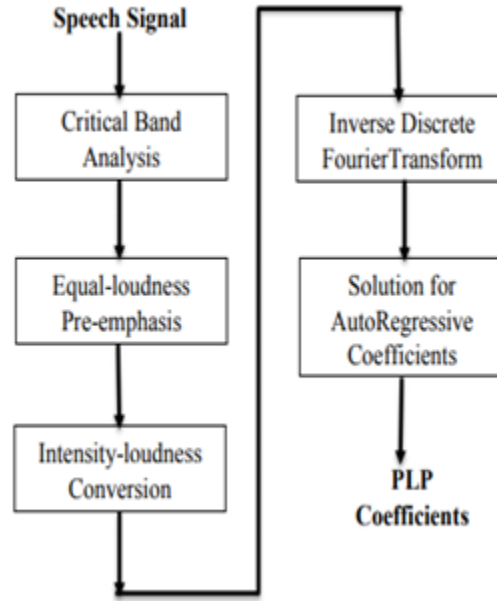


Figure 6: Processing steps involved in PLP computation.

2.9 Literature Review

As previously mentioned, reviewing the literature showed that detecting audio spoofing attacks is regarded as a binary classification problem. The goal of these spoofing attack detection methods is to use the various feature-classifier combinations in creat-

ing countermeasures [85]. The several types of cepstral coefficient features that include Mel-frequency cepstral coefficient (MFCC), constant-Q transform (CQT), Log-CQT, linear frequency cepstral coefficient (LFCC), constant-Q cepstral coefficient (CQCC), extended CQCC (eCQCC), and inverted CQCC (iCQCC) are extensively used in human voice recognition and verification problems [2, 14, 19, 71, 84].

The CQT-based features, have been instrumental and beneficial in speaker verification and anti-spoofing with the variable spectro-temporal resolution, thus seizing the signs and evidence of manipulation artifacts to specify spoofing threats [70]. The CQT-based features which is based on time-frequency analysis have the property of offering higher time resolution at high frequency regions, while offering higher frequency resolutions at low-frequency regions. Nevertheless, integrating traditional cepstral analysis with CQT-based features is challenging since postprocessing is mandatory for reaching a linear frequency scale. Moreover, the combination of multi-resolution analysis and extra post-processing constitutes a heavy computational burden [22]. However, it is possible to obtain more spectral details in low-frequency regions using the constant-Q cepstral coefficient (CQCC), a derivative of the constant-Q transform (CQT) based features, but high-frequency regions that give discriminative features are unaffected. Furthermore, while the LFCC utilizes discrete Fourier transform (DFT) to carry out the time-frequency analysis for the whole input signal, the spoofing data reside specifically on high and low frequency sub-bands [43]. As a consequence, it is impossible for the LFCC features to donate spectral details in the discriminative frequency bands [43].

In addition, some other well-known cepstral features such as MFCC are very sensitive to noise, the matter that negatively affects their performance towards detecting spoofing at-

tacks [12]. In the same manner, various spoofing detection studies have tackled and examined phase based features that include the relative phase shift, phase difference, modified group delay, and group delay and cosine normalized phase features [60, 80, 82]. Sound analysis suggests that during the analysis-synthesis stage of several speech-synthesis methods, phase information is replaced or lost, which allows differentiating between spoofed and bonafide speech. Practically, this previous knowledge is missing which does not warrant the effectiveness of these features to spoofing attacks with constant phase information [88]. Deep features or deep neural networks with hidden layers are other renowned features which results in competitive outcomes. Deep features are explored by many studies, but because they require costly retraining, they are not widely used in resource-constrained environments [19, 26, 64, 86].

Some of the extensively utilized approaches for classification include deep neural networks (DNN) [7, 19, 66, 71], the Gaussian mixture model (GMM) and classifier ensembles [2, 85, 86]. The GMM uses basic hypothesis test to reiterate spoofing detection, however, the likelihood ratio test is used to differentiate between spoofed and bonafide speech. Usually the GMM yields encouraging outcomes, however, when high dimensional features are employed, its achievement deteriorates [82, 89]. In comparison to the GMM, DNN classifiers are powerful when high dimensional features are used, however, more training data is required for these DNN classifiers. At the same time, classifier ensembles carry an array of ineffective classifiers on the subset of the data and by uniting the classification outputs, ensembles can create a consistent classifier [32]. When it comes to ensemble approaches, they barely overfit giving room to solutions that are difficult to attain using one hypothesis [58].

2.10 Details of Specific Approaches

In a study by Todisco et al. exploring methods for detecting spoofing threats, the CQCC features were employed to train the GMM classifier. The CQCC features can secure the input signal characteristics through delivering variable-resolution, time-frequency representation of the spectrum [69]. In this study, the input signal characteristics were employed, thus, allowing the CQCC features to effectively surpass previous methods in spoofing threat detection by a reasonable margin. Yet, a disparity between identified and unidentified spoofing attacks was obvious. Another study by Nagarsheth et al. was conducted to create the tandem features for detecting replay attacks employing both CQCC and HFCC features and implementing cepstral mean and variance normalization (CMVN) [49].

Other studies have revealed that CMVN has been effectively used for automatic speech recognition and has the ability to eliminate the effects of nuisance channels [27,77]. Eliminating the effects of nuisance channels allows maximizing the performance of different types of systems that allow speaker verification and speech recognition. Feature embeddings were engendered through supplying the tandem features to a DNN classifier. Additionally, these features were turned over to a SVM classifier to allow specifying the type of the replay attack. Moreover, using CMVN for the detection of replay spoofing attacks might seem illogical and impractical, however, the aggregation of more channel effects is similar to the recording and replaying speech in various acoustic situations and with the use of various devices. While the main goal of using the CMVN is weakening the effects of the nuisance channels, it can be used for the detection of replay attacks when bonafide speech was apprehended within a stable channel [27].

Spoofing attacks are not only common in audio signals but also in-vehicle communications where CAN protocol is used [?, ?, ?, ?, ?, ?, ?, ?, ?, ?]. Studies devoted for replay spoofing detection [27, 77] the GMM classifier is trained on different high frequency features and also transmission line cochlea (TLC) features were employed in combination with the GMM classifier [27]. Transmission line cochlea features successfully use amplitude modulation for replay attack detection and are precisely similar to the auditory system, but their signals which are the input signal and output signal look different in the same dynamic range. Accordingly, it is very complicated to seize the distinctive data displayed in the low-frequency regions when there are high-frequency regions in the input signal [27]. Another study conducted by Witkowaski et al. declared that in detecting replay attacks, it was found that spoofing attacks show spectral changes at high frequencies ranging from 6 kHz to 8 kHz [77]

Moreover, many other methods were used for replay attack detection that include the inverted-MFCC, LPCC, and LPCC residual features in conjunction with CQCC, MFCC, and Cepstrum features side by side with the GMM features. However, these methods were not very successful in alleviating the spoofing problem entirely, but they presented a significant development concerning the CQCC-GMM system employed in the ASVspoof-2017 challenge [63]. Other researchers used the attributes of the recording device and playback device to detect spoofing attacks [46, 63, 83]. For example, in a study by Saranya et al. for detecting replay attacks, the researchers were able to train the GMM classifier using CQCC, MFCC, and Mel-Filterbank-Slope (MFS) features. The outcome of this study revealed that the distinguishing information employed to classify a signal to whether it is an authentic speech or replayed speech is precisely divided into two sub-bands: the first is 0-1 kHz and

the second is 7-8 kHz [83].

Furthermore, a low-frequency frame-wise normalization method was used by Yang et al. [17], and others used deep learning models for voice replay attack detection. In another study detecting voice replay attack, CQCC and MFCC were used to train a combination of classifiers that included GMM, DNN, and ResNet. Although this approach resulted in decreased equal error rate (EER), it led to more computational expenses [10]. These high expenses were mitigated in one study for audio replay attack detection using a light-weight CNN model that was initially recommended for face recognition [40]. However, successful training of deep learning and CNN models involve great amounts of data. Additionally, Baker et al. detected replay spoofing by training a DNN classifier employing both MFCC and long-term average spectrum (LTAS) features. This study revealed that the outcomes resulting from combining the MFCC and LTAS with DNN classifier surpass the GMM classifier with CQCC in the ASVspoof-2017 challenge [40].

Many studies used the GMM classifier in audio replay spoofing detection. A study by Leon et al. trained the GMM classifier by employing the elicited relative phase shift characteristics derived from the incoming speech signal's harmonic phase. This approach concluded sound outcomes, although the number of test samples were limited to 283 only. The whole system was very responsive to vocoders utilized to synthesize audio signals. If these vocoders were employed to train the system effective performance can be reached [76]. In the same manner, a study by Wester et al. detecting voice cloning attacks, the MFCC and cosine-normalized phase features were utilized by applying the GMM-Universal Background Model (GMM-USM). The work of Wester et al. was the first of its kind to make a comparison between 100 native English listeners and a system's performance. The out-

comes of this study emphasized the performance of automatic detectors that surpassed all those English listeners except for one person. In addition, the study revealed that human countermeasures utilize certain cues that are different from that of automatic countermeasures to differentiate between authentic and spoofed audios [81].

In a study by Patel et al. for detecting spoofing attacks, GMM was trained by employing MFCC features along with cochlear filter cepstral coefficients and cochlear filter cepstral coefficients-instantaneous frequency features. This study concluded that, countermeasures are more dependent on powerful features than classifiers [53]. Another method was used by Janicki et al. for detecting voice cloning attacks includes training SVM using long term prediction residual signals. The prediction coefficients were employed to discriminate authentic from spoofed signals. These coefficients included prediction gains, energy of the prediction error, and temporal parameters among others. This approach depended on adjusting the temporal parameters which has the ability to negatively influence the generalization capabilities, the matter that allowed this approach to effectively perform on acknowledged attacks than on acknowledged ones [34].

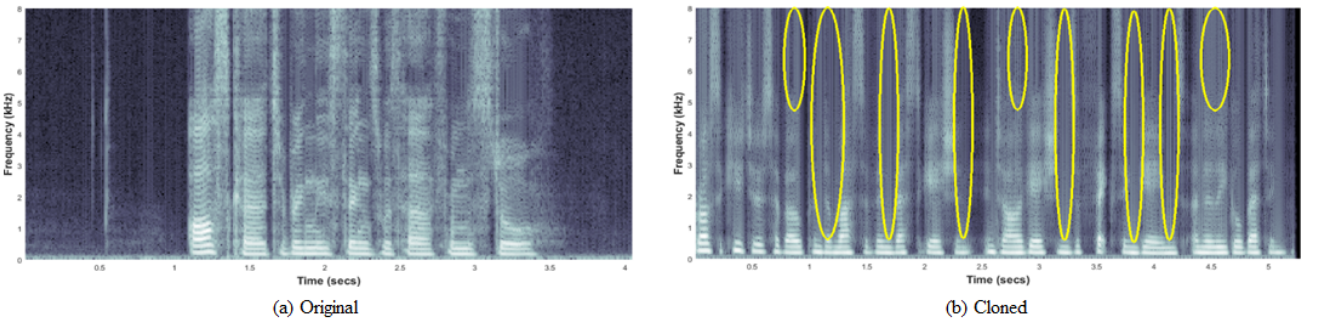


Figure 7: Original (a) and cloned (b) utterances are analyzed spectrally. Vertical lines that only occur in copied audios can be used as a possible cloning attack hint. By using neighborhood statistics, these lines may be recorded.

2.11 Limitations of the Existing Approaches

Since ASV systems are susceptible to voice cloning and replay attacks, some factors have to be examined during audio representation to gain a powerful countermeasure that include: (1) The intermodulation distortions allow the microphone to add a layer of non-linearity, which produce the detectable patterns. These patterns are identified through audio-fingerprinting using an audio representation approach that can distinguish genuine audios from replayed ones . (2) The commonly used successive recordings performed from the same recording in audio splicing, result in higher-order nonlinear functions and more discriminable audio signals. Accordingly, audio representation step should include pattern analysis of audio samples. (3) when deciding on audio representation approaches it is more likely to identify artifacts developed by voice cloning algorithms. When performing special analysis for both bonafide and cloned audio signals, the very fine lines that are observed in the spectral image shown in in (Figure 7), (3) serve as the voice cloning algorithmic artifacts. However, the spectral image of the bonafide audio signals does not include these fine lines. Due to the uniqueness if the artifacts belonging to the voice cloning algorithms, it is easy to distinguish each one of the cloned audios created by different cloning algorithms, in addition to bonafide audios. (4) For ASV systems, it is required that audio representation approach used for speaker verification should not be very sensitive to noise within all environments. (5) For real-time applications, a rapid retraining of the mode to integrate new users can be guaranteed when those features and classifier combinations are considered by the ASV systems.

CHAPTER 3 ARCHITECTURE OF SECURE AUTOMATIC SPEAKER VERIFICATION SYSTEM (SASV)

In this chapter basic architecture of the proposed framework is presented. The further details will be provided in the upcoming chapters.

3.1 Aims of SASV System

In this dissertation our emphasis was to develop a single model-based approach to simultaneously identify who the speaker is and safeguard the underlying ASV system against any possible spoofing attack. Our countermeasure approach is based on a comprehensive framework that perform speaker identification as a first step and then determine the liveness of the input audio to determine spoofing attack. The system was designed with an intention to be deployed in any environment accessible to multiple users. Furthermore, through the enhanced attack vector which is introduced in the proposed framework our system can even handle the advance spoofing attacks that may occur in practical scenarios. The conventional countermeasures consider only the replay and voice cloning attacks. In contrast, Our approach recognizes the methodology used for voice cloning in a LA attack, and the replay detection module detects cloned replay attacks. The cloning algorithm detection is also a novel concept that we introduced in our published work [?]. The voice cloning algorithm detection makes our approach more flexible by even providing the artifact level analysis. Later on this phenomenon can also help in counterfeiter identification who used a commercial solution for cloned audio generation. Furthermore, previous methods assume that the produced audio from a voice cloning process is sent directly to the anti-spoofing system, without the need for a physical channel. However, we have evaluated real-world LA attacks over PA attacks, where the LA attacks will be launched over a

physical channel. With this consideration the need to make replay and cloning detection module more powerful becomes apparent as this slight modification can fail both modules. The fundamental reason is dilution of the microphone and algorithmic artifacts and non linearity through simultaneous occurrence.

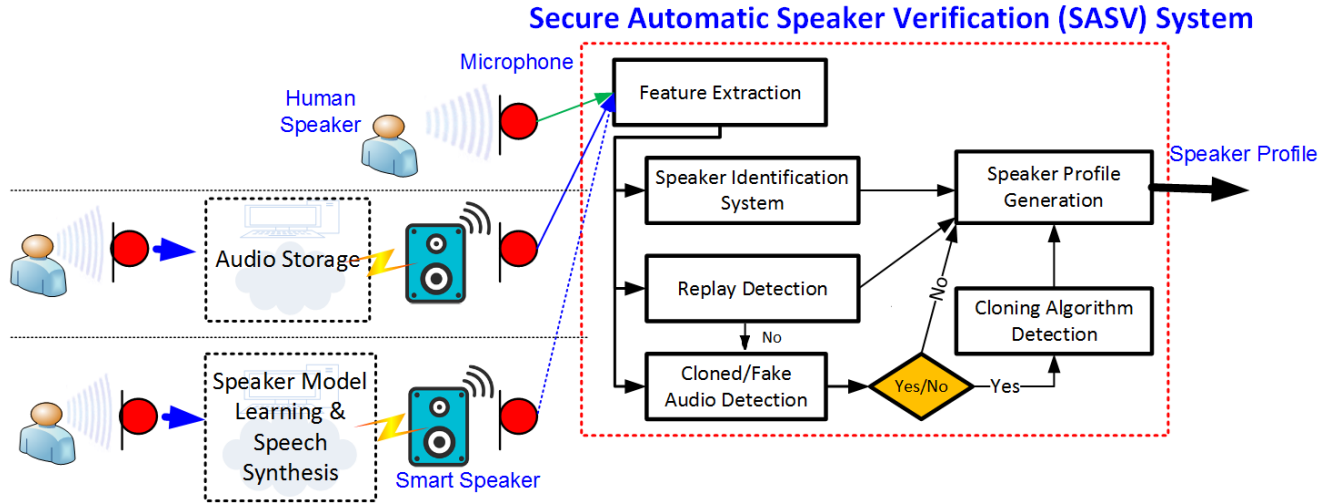


Figure 8: Block diagram of Secure ASV (SASV) system.

3.2 Overview of SASV Framework

As described in Figure 8, for each and every audible input signal, the proposed framework acts to identify the speaker who is communicating to the system following feature extraction and then sends the speaker ID to the module that creates speaker profiles. The SASV system has the ability to detect whether the system was invaded by a cloned audio or not and the binary decision information is then moved to the speaker profile generation module. In case of an authentic input audio, the system examines and analyzes it for potential voice cloning attacks that might be initiated through a microphone or a smart speaker. However, in case of a replayed audio, the system can specify the audio cloning algorithm which was employed to produce the cloned audios. Moreover, the speaker pro-

file generation module receives the audio cloning algorithm decisions giving access to the system to authentic speakers where the replay and cloned audios are absent from the system (represented by No in the figure). In support of the genuine audios, the information of each user including user's name, ID, account type and number etc., are presented in the speaker profile and accessed from the main stream databases according to what is required by the application. However, regarding the spoofed attack, the system will reveal the information of the attack such as the person who was attacked, the algorithm of solution used to create the cloned audios, and so on.

3.3 Operational Contributions

The SASV approach presented in this dissertation represents input audios through a novel feature extraction scheme. This novel feature extraction scheme is the sign modified acoustic local ternary pattern (sm-ALTP) features. This sign modified version is an extension of the ALTP features that overcomes inherent limitations (details in chapter 4). The sm-ALTP allows seizing the user's speech vocal features [3]. In addition, the sm-ALTP uses local correlation scores to discover signal non-linearity that exists as a consequence of voice cloning or recording artifacts. The SVM-based classifier ensemble is utilized to identify the vitality of a voice. In this study, this classifier ensemble uses a group of weak classifiers and integrate their outputs to create a more stable classifier. The new created model was used for speaker verification to detect and track down attacks of voice cloning, cloning algorithm employed to perform an attack, voice replays-, and cloned voice replay attacks (also a novel concept proposed in this study) against ASVSpooof 2019 and VSDC datasets.

Using the voice cloning algorithm detection of the generated system, it is possible to examine and analyze some of the problematic scenarios and situations that are difficult to handle and can impede the success of any available countermeasure. The availability of commercial solutions made it easy even for the amateurs to create voice cloning attacks. However, voice cloning algorithm detection can counteract the effect of these solutions and thus, it would be easy to identify attackers based on the seriousness of each case.

The cloned voice replays are recorded audio samples played before the microphone using fake voice samples. Applications used for cloned voice replays can be employed in cases where the attacker uses recorded voice for impersonation, however, this attacker does not have the speaker's prerecorded audio samples. Therefore, the created ASV system has the power to encounter different security breaches with the aid of the model evaluation over the strengthened attack vector. In addition, the proposed system can be highly adopted in various resource constrained environments since it has a lightweight nature.

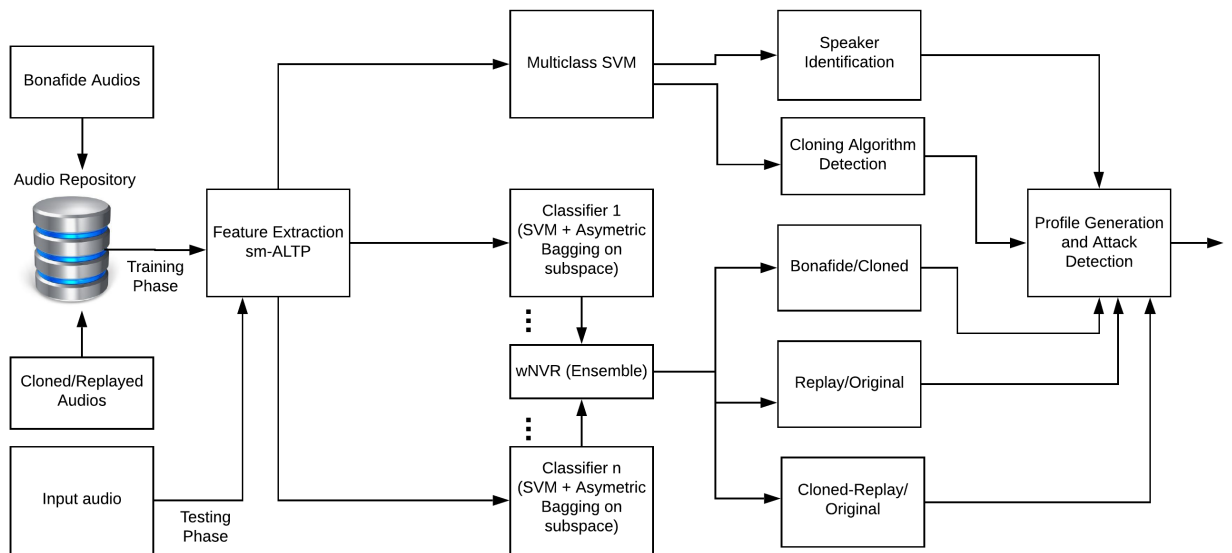


Figure 9: Detailed architecture of SASV system.

3.4 Operational Architecture

This study aims to create a profoundly secure ASV system capable of supporting and authenticating registered bonafide speakers and counteracting audio synthesis/cloning, voice replay, and other attacks concerning cloned voice replay. Furthermore, the developed ASV system can spot and classify the algorithm employed to create audio cloning attacks. The audio repository of this system is composed of the bona fide speakers' voices in addition to the replayed and cloned voices. Through advanced voice cloning algorithms, undistinguishable cloned voices from bona fide speakers' speeches are created. Accordingly, for m numbers of bona fide speakers and p numbers of voice cloning algorithms; thus, resulting in $(m \times p)$ cloned voice classes. While the proposed model might not be able to predict the type of cloning algorithm used, it can effectively detect and counteract the cloned audio samples used for attack and classify the audio input as cloned audio. In the same manner, through the detection of the replay attack, the input samples are classified as replayed/bona fide. Therefore, it is required to identify q , the number of speaker classes where $q = m + (m \times p) + 2 + 2$.

The architecture used for the proposed secure ASV system is illustrated in Figure 9. This figure shows that the feature extraction version was executed using the novel sm-ALTP features for both the spoofed and bona fide voice samples entering the audio repository. This step was followed by creating the SVM-based classifier ensembles using asymmetric bagging and random subspace sampling. Both asymmetric bagging and random subspace sampling help mitigate the effect of class imbalance associated with a small number of bona-fide samples compared to higher numbers of spoofed samples [32]. Furthermore,

in a way to overcome and counteract cloned voices and replay attacks, the weighted normalized voting rule (wNVR) was employed to integrate the outcome of the multiple SVM classifiers with that of the classifier ensembles. Using the speaker identification module in the architecture of the ASV system allowed identifying the registered speaker who is communicating with the system. The ASV system also included a module for cloning algorithm detection that is capable of identify the type of voice cloning algorithm employed to create fake audios. Due to the unique speech features of each speaker and the artifacts of the voice cloning method, the multi-class SVM classifier with polynomial kernel is used to perform both speaker identification and voice cloning algorithm detection. Input data can be verified after training all used models. Moreover, the system can be given access to the identified speaker once negative outcomes are received from the voice cloning and replay detection modules. More details about the proposed method are introduced in the following sections.

CHAPTER 4 PROPOSED METHOD

This chapter provides the details of the steps described in section 3.4. As a first step sm-ALTP features will be discussed by first establishing their difference with the baseline ALTP feature extraction approach. Afterwards, classifier committee learning approach will be discussed that is used to generate the ensemble-based countermeasure. Finally, the chapter will be concluded with the discussion on how proposed approach overcomes the limitations of the existing approaches mentioned in section 2.11.

4.1 Overview of ALTP features

N-sample input audio signal $Y[n]$ is divided into $i = \{1, 2, \dots, k\}$ non-overlapping frames/windows $F^{(i)}$ with length $l = 9$. c represents the center sample in a frame and has $z^{(j)}$ neighbors in each frame $F^{(i)}$, where j represents the neighbor index in the frame $F^{(i)}$. The difference between c and $z^{(j)}$ is determined by applying the parameter t_h around the sample c in order to compute the ALTP response. The parameter t_h has a value between 0 and 1 and it is acquired using a linear search process. The sample values in $F^{(i)}$ that fall within the range of width $\pm t_h$ around c are quantized to zero, whereas values above and below $c \pm t_h$ are quantized to 1 and -1 , respectively. As a result, we have a three-valued function as follows:

$$p(c, z^{(j)}, t_h) = \begin{cases} -1 & z^{(j)} - (c - t_h) \leq 0 \\ 0 & (c - t_h) < z^{(j)} < (c + t_h) \\ +1 & z^{(j)} - (c + t_h) > 0 \end{cases} \quad (4.1)$$

The function $p(c, z^{(j)}, t_h)$ is then split into two patterns classes, upper pattern $P^{up}(\cdot)$ and lower pattern $P^{lw}(\cdot)$ as follows:

$$P^{up}(c, z^{(j)}, t_h) = \begin{cases} 1 & p(c, z^{(j)}, t_h) = +1 \\ 0 & \text{Otherwise} \end{cases} \quad (4.2)$$

Similarly

$$P^{lw}(c, z^{(j)}, t_h) = \begin{cases} 1 & p(c, z^{(j)}, t_h) = -1 \\ 0 & \text{Otherwise} \end{cases} \quad (4.3)$$

Upper and lower ALTP representations are then generated using these upper and lower patterns. eq. 4.4 is used to calculate the upper-ALTP features A_U .

$$A_U = \sum_{j=0}^{j=l} P^{up}(c, z^{(j)}, t_h) * 2^j \quad (4.4)$$

whereas, lower-ALTP features A_L are computed through eq. 4.5.

$$A_L = \sum_{j=0}^{j=l} P^{lw}(c, z^{(j)}, t_h) * 2^j \quad (4.5)$$

Then, by using the Kronecker delta function $\delta(\cdot)$ as stated in eq. 4.6 and eq. 4.7, the histograms of A_U and A_L are calculated.

$$H^u(b) = \sum_{a=1}^{a=k} \delta(A_U^a, b) \quad (4.6)$$

$$H^l(b) = \sum_{a=1}^{a=k} \delta(A_L^a, b) \quad (4.7)$$

The bin is represented by b , while the frame index is represented by a . The ALTP characteristics are derived by concatenating ($||$) both histograms after computing $H^u(b)$ and $H^l(b)$.

$$H_A = [H^u(b) || H^l(b)] \quad (4.8)$$

4.2 Limitations of ALTP Features

The ALTP features were initially suggested for indoor applications, such as fall detection [3, 33]; and demonstrated excellent performance as a feature descriptor versus state-of-the-art feature extraction methods due to their noise tolerance. However, there are several flaws with ALTP that must be addressed before it can be used in ASV systems. As demonstrated in Figure 7, the spectrum analysis of the cloned audio indicates that the artifacts have a non-static repeating pattern, which may be recorded more effectively using a dynamic threshold mechanism. ALTP however, has only a static threshold, such as $\pm th$; hence, ALTP for ASV applications has space for improvement. (b) Signal volatility—To effectively capture the artifacts in cloned and replayed audios, It's crucial to understand how rapidly the signal changes in terms of artifacts to successfully capture them in cloned and replayed audios [1]. The ALTP features, on the other hand, lack this functionality. As a result, the performance suffers as compared to the faked audios. (c) Brute-force Optimizationtextemdash For threshold optimization in ALTP, a brute-force technique was necessary; as a result, in time-critical applications, error reduction was not assured. (d) Uniform noise ALTP was resistant to uniform noise that remained consistent in audio situations, such as indoor audios. In contrast, because noise is non-uniform in outdoor situations, static threshold-based feature extraction becomes inconsistent, necessitating a new

method to noise suppression.

4.3 Motivation for the sm-ALTP Features

sm-ALTP characteristics are proposed to overcome the limitations of ALTP features and to identify the liveliness of the voice more effectively. The dynamic optimizable threshold used by the sm-ALTP features successfully catches signal artifacts and provides distinct representations for spoofed and bonafide voices. As a consequence, a powerful CM approach emerges from the difference in representation for spoofed and bonafide voices. Furthermore, utilizing the vocal tract information, which was not present in the ALTP characteristics, can improve speaker identification and recognition.

4.4 sm-ALTP Features

By setting a dynamically optimizable threshold and recording the speaker's vocal tract, sm-ALTP features address the weaknesses of ALTP features. The three-valued function is computed in sm-ALTP as follows:

$$p(c, z^{(j)}, \sigma\alpha) = \begin{cases} -1 & z^{(j)} - (c - \sigma\alpha) \leq 0 \\ 0 & (c - \sigma\alpha) < z^{(j)} < (c + \sigma\alpha) \\ +1 & z^{(j)} - (c + \sigma\alpha) \geq 0 \end{cases} \quad (4.9)$$

where σ is the standard deviation of $F^{(i)}$ and α is the scaling factor, such as $(0 < \alpha < 1)$.

σ can be calculated as follows:

$$\sigma = \sqrt{\frac{\sum (z^{(j)})^2 - \left(\frac{\sum z^{(j)}}{l}\right)^2}{l - 1}} \quad (4.10)$$

We circumvent the constraints (a), (c), and (d) of the ALTP features (section 4.2), by

substituting t_h with $\sigma\alpha$, which requires the signal variance to be expressed in terms of neighborhood statistics. Another drawback of the ALTP functionality was that the t_h required brute-force linear search optimization. However, we may optimize the new threshold value, $\sigma\alpha$ by creating the following convex function.

$$J(\sigma) = \min \frac{\alpha}{2M} \sum_{q=1}^{q=M} \left(g \left(\theta^T \sigma(x^{(q)}) \right) - y^{(q)} \right)^2 \quad (4.11)$$

Where $J(\cdot)$ represents the cost function, θ represents the classification weights, $q = \{1, 2, \dots, M\}$ represents the total number of records in the training set, g represents the classification function used, such as *relu*, *sigmoid*, *tanh* etc., and $y^{(q)}$ represents the actual class-label of the audio record. The probabilistic meaning of the cost function is as follows:

$$p(y^{(q)}|x^{(q)}; \sigma) = \frac{1}{\sqrt{2\pi\sigma}} \exp \left(- \frac{y^{(q)} - x^{(q)}}{2\sigma^2} \right) \quad (4.12)$$

The parameter σ can then be optimized by applying the gradient descent algorithm as:

$$\sigma_{new} = \sigma - \alpha * \frac{\partial \sigma}{\partial z^{(j)}} \left(\sqrt{\frac{\sum (z^{(j)})^2 - (\frac{\sum z^{(j)}}{l})^2}{l-1}} \right) \quad (4.13)$$

where

$$\frac{\partial \sigma}{\partial z^{(j)}} = \left[\frac{\partial \sigma}{\partial z^{(1)}} \quad \frac{\partial \sigma}{\partial z^{(2)}} \quad \dots \quad \frac{\partial \sigma}{\partial z^{(l)}} \right] \quad (4.14)$$

thus

$$\frac{\partial \sigma}{\partial z^{(j)}} = \frac{1}{\sqrt{l-1}} * \frac{\partial}{\partial z^{(j)}} \left[\hat{A} + \hat{B} \right]^{1/2} \quad (4.15)$$

where

$$\hat{A} = (z^{(1)})^2 - \left(\frac{\sum z^{(j)}}{l}\right)^2 + \dots + (z^{(c-1)})^2 - \left(\frac{\sum z^{(j)}}{l}\right)^2 \quad (4.16)$$

and

$$\hat{B} = (z^{(c+l)})^2 - \left(\frac{\sum z^{(j)}}{l}\right)^2 + \dots + (z^{(l)})^2 - \left(\frac{\sum z^{(j)}}{l}\right)^2 \quad (4.17)$$

or in compact form we can write it as:

$$\frac{\partial \sigma}{\partial z^{(j)}} = \frac{1}{\sqrt{l-1}} * \frac{\partial}{\partial z^{(j)}} \left(\sum (z^{(j)})^2 - \left(\frac{\sum z^{(j)}}{l}\right)^2 \right)^{1/2} \quad (4.18)$$

thus, the partial derivative will return:

$$\frac{\partial \sigma}{\partial z^{(j)}} = \frac{1}{2\sqrt{l-1}} * \left(\sum (z^{(j)})^2 - \left(\frac{\sum z^{(j)}}{l}\right)^2 \right)^{-1/2} * \left(2z^{(j)} - \frac{2\sum z^{(j)}}{l^2} \right) \quad (4.19)$$

or

$$\frac{\partial \sigma}{\partial z^{(j)}} = \frac{1}{2\sqrt{l-1}} * \frac{1}{\sqrt{\left(\sum (z^{(j)})^2 - \left(\frac{\sum z^{(j)}}{l}\right)^2 \right)}} * \left(2z^{(j)} - \frac{2\sum z^{(j)}}{l^2} \right) \quad (4.20)$$

By replacing the eq. 4.2-4.5 with $\sigma\alpha$ we get the $H^u(b)$ and $H^l(b)$ using eq. 4.6 and 4.7 and generate feature representation as:

$$H = [H^u(b) || H^l(b)] \quad (4.21)$$

The H representation feature captures the patterns in the input signal, but it excludes the vocal tract information provided by the cepstral coefficients at Mel-scale [12]. For example, due to the phoneme representation attributed to that speaker's vocal structure, a speaker's cepstral coefficients always appear negative at 1000 Hz, and this frequency occurs frequently; in the case of sm-ALTP, a large positive histogram-spike will appear, but it will not provide any information regarding vocal behavior at this frequency. As a consequence, we utilized eq. 4.22 to investigate the sm-ALTP representation in further depth.

$$H_s = H \times \text{sgn}(\mu_t(C_\gamma(t))) \times \beta \quad (4.22)$$

The $C_\gamma(t)$ represents the t^{th} order MFCC of the γ^{th} frame (further details in [55]), μ_t is the frame-wise mean of $C_\gamma(t)$, and $t = \{1, 2, \dots, 20\}$. By calculating the frame energy $E(f)$ with index f as stated in eq., $C_\gamma(t)$ is applied 4.23.

$$C_\gamma(t) = \sum_{f=0}^{g-1} \log[E(f)] \cos\left[t\left(f - \frac{1}{2}\right)\frac{\pi}{q}\right] \quad (4.23)$$

For feature normalization in H_s , the parameter $\beta = 0.1$ in eq. 4.22 is utilized. The following is our final depiction of sm-ALTP features:

$$H_{sm} = [\mu_t(C_\gamma(t)) || H_s] \quad (4.24)$$

4.5 Classifier Comity Learning for Ensembles

The features of data in terms of data quality, data collecting methodology, and dataset size impact the classification performance in ASV systems, regardless of how effective a

feature extraction approach is. For example, if a training set has much fewer genuine representations than spoofed representations, a classifier may tend to favor the spoofed class. Higher classification accuracy in this example might be due to the classifier's bias towards the faked class; in actuality, the classifier is doing poorly for bonafide data, which is the fundamental aim of any ASV system. As a result, even better categorization accuracy will be irrelevant. It's critical to pinpoint the reasons why classifiers provide incorrect results. For cloning attack detection, we additionally identify the cloning algorithm utilized for faked audio production in order to achieve this goal. Classification models can be enhanced further by incorporating the connection between faked samples and the cloning technique. Furthermore, we have guaranteed that the testing procedure does not become so complex that the classification model becomes unsuitable for a real-time application.

4.5.1 Training-Phase—Asymmetric Bagging and Subspace Sampling

Asymmetric bagging and subspace sampling are used to create multiple classifiers [32]. Bootstrapping is performed over the faked class samples in asymmetric bagging since there are significantly more spoofed samples than genuine samples. In this manner, each classifier is trained on a balanced set that includes both genuine and faked data, resulting in improved unstable SVM classification performance. The SVM classifiers are stable and can thus distinguish between spoofed and bonafide data even when they aren't visible. However, if alternative data balancing approaches, such as up-sampling or down-sampling, are employed instead of asymmetric bagging, the classifier becomes either over-fit or under-fit. Following the asymmetric bagging, the weighted normalized voting rule (wNVR) is used to aggregate several classifiers across the development set..

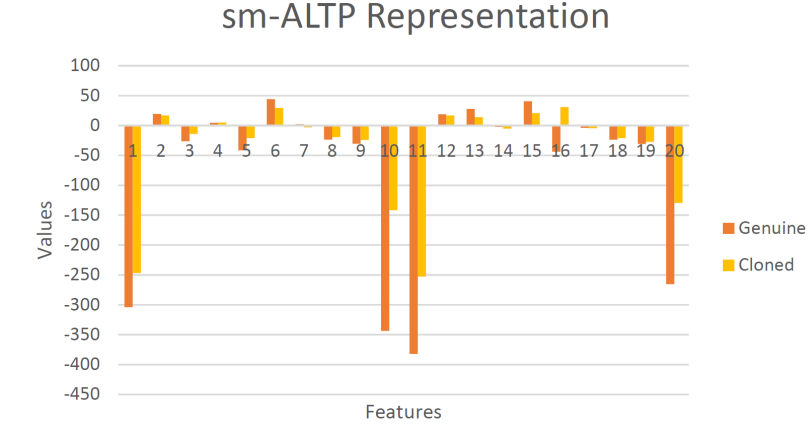


Figure 10: sm-ALTP representation (eq. (4.21)) for genuine and cloned audios.

4.5.2 Weighted Normalized Voting Rule (wNVR)

Following the training of several classifiers, wNVR is used to combine the results of all of these classifiers. The benefit of wNVR over majority voting rule (MVR) is that MVR cannot take use of accurate classifiers and gives equal weight to all classifiers [68].

By using the weighted cross-entropy function as stated in eq (4.25), $w = \{1, 2, \dots, Q\}$ classifiers are utilized to produce the ensemble classifier:

$$C(x) = \sum_{w=1}^Q \lambda_w \sum_{b=1}^M \sum_{k=1}^K [y_b = k] \log \frac{e^{\theta_k^T x_b}}{\sum_{v=1}^K e^{\theta_v^T x_b}} \quad (4.25)$$

Where $b = \{1, 2, \dots, M\}$ is the number of instances x_b in the development-set, and λ is the weight to take use of a more accurate classifier for $k = \{1, 2, \dots, K\}$ number of classes to be categorized. The eq. (4.26) is then used to produce the final class-label $C^*(x)$:

$$C^*(x) = \text{sgn} \left[C(x) - \frac{K-1}{2 \times s} \right] \quad (4.26)$$

The normalization factor s is used to limit the bias/variance impact.

4.5.3 Testing Phase

The trained model can be utilized for assessment purposes after training and model tuning. The evaluation set consists of instances with seen and unseen genuine speakers, as well as samples created by seen and unseen algorithms in the case of a voice-cloning assault. Any query audio sample may be supplied to the final model after model assessment, and it can execute ASV tasks in real-time settings.

4.6 Overcoming the Limitations of Existing Approaches

Existing methods, as mentioned in section 2.11, neglect several essential signal properties during feature extraction, lowering their performance. For example, the first three restrictions stress that intermodulation and algorithm artifact emerge during playback and voice cloning, exhibiting distinct patterns. The suggested method analyzes the input signal's pattern, successfully capturing these abnormalities and distinguishing faked signals from genuine signals. The genuine and cloned signals, for example, peak at the same feature locations, as shown in Figure 10, but owing to the difference in peaks, these signals are still readily identifiable. Furthermore, the spoofed and bonafide signals have opposing peaks at several feature locations, such as feature 16 in Figure 10. The discrepancy in feature values in Figure 10, illustrates that while the cloned audio looks to be close to the original, the fundamental signal components, such as pitch, loudness, and so on, are not properly duplicated. However, the suggested approach's lower level examination of the input signal quickly shows this discrepancy. Another drawback of the audio representation methods was that their resilience to noise was difficult to quantify. The proposed method, on the other hand, is noise-resistant, and we can readily test this claim. Take, for example,

the audio frame depicted in Figure 11. We can see that additive noise, which may raise or decrease the value of the central sample c in a frame $F^{(i)}$ and cause the incorrect code to be generated against c , becomes useless. The reason for this is that the sample value c now falls within a range of higher and lower threshold values, making it more tolerant to additive noise values. Furthermore, because of the fewer characteristics, quick model retraining is achievable, making our method suitable for applications that need continual user participation.

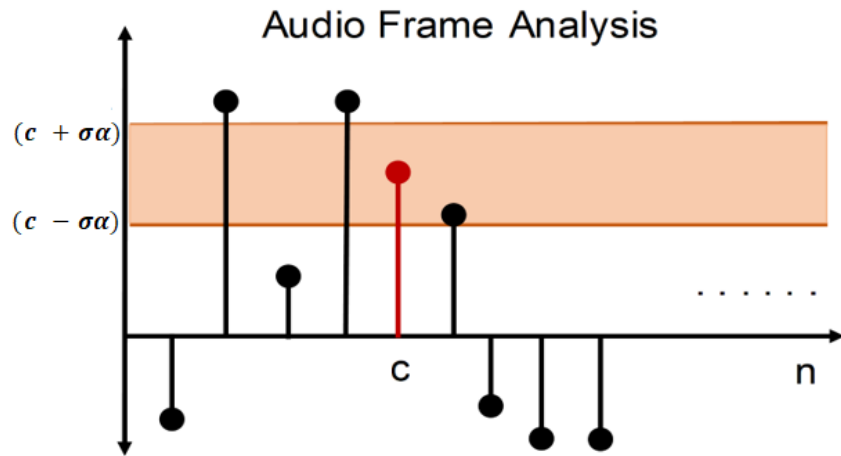


Figure 11: Effect of the dynamic threshold over the audio frames.

CHAPTER 5 EXPERIMENTS AND RESULTS

5.1 Dataset

The proposed method’s performance is assessed using the ASVSpooof 2019 [74] dataset and the VSDC dataset [11].

We used a large dataset which is called ASVSpooof 2019 (Table 1). The dataset we used in our evaluation is divided into two parts: the first part is for the logical access (LA) to detecting voice-cloning attacks and the second part physical-access (PA) to detecting replay attacks. Their dataset contains 25,380 samples for training and 24,844 samples for development, and 71,933 samples for assessment. Both training and development datasets contain voice samples from 20 different speakers, whereas the spoofed dataset contains cloned samples of the same speaker utterances generated by two voice-conversion and four speech synthesis algorithms, totaling 120 (20×6) cloned speakers in addition to algorithm classes. First, Neural networks and transfer-function-based methods are used in voice conversion techniques. Second, the voice synthesis techniques, on the other hand, are a mix of waveform concatenation and neural network based parametric speech synthesis utilizing source-filter vocoders , as well as neural network parametric speech synthesis using Wavenet. The evaluation set includes both spoofed and bonafide speech samples from 67 speakers, with the spoofed set containing examples created using 19 different techniques, including GAN and deep neural network approaches. Table 1 shows that the PA dataset that contains 54,000 for training, 33,534, for development and 1,53,522 for evaluation samples. At [74], you may learn more about the ASVspooof 2019 corpus.

The VSDC was created to detect replay and cloned replay attacks. The ASVspooof

Table 1: Number of non-overlapping target speakers and utterances in the ASVspoof 2019 database’s training and development sets.

Subset	#Speakers		#Utterances			
	Male	Female	Logical Access		Physical Access	
			Bonafide	Spoof	Bonafide	Spoof
Training	8	12	2,580	22,800	5,400	48,600
Development	8	12	2,548	22,296	5,400	24,300

cloning samples were used to create the replay samples in the same way as the bonafide voice recordings were done, where cloned replay attack is the recording of cloned voice samples. The samples in the collection vary in terms of environments, speaker genre, recording, settings, number of speakers and playback devices, and the samples in the collection are different (Table 2).

These samples contain noise and interference, to be more specific. Different playback devices were utilized to create the replays in order to counteract the influence of a certain playback device. Voice samples from 10 males and 9 females speakers who donated their services for data gathering are included in VSDC.

5.2 Experiment I—Performance Evaluation for Speaker Verification

The proposed method’s performance is assessed for bonafide speaker verification in this experiment. Any ASV system’s principal duty is to verify that the speakers are bonafide. The ASVspoof 2019 dataset was used to identify all 2580 audio samples matching to the 20 bonafide speakers for this experiment. 70% of the data which is 1806 records were used for model training, and the remaining 30% of the data which is 774 entries was utilized for testing.

The proposed approach produced on average 99 percent precision, recall, f1-score, and accuracy values, as shown in Table 3. The evaluation rates for the majority of the

Table 2: Details of Voice Spoofing Detection Corpus (VSDC)

Audio Samples		Rate	Environment	Microphone		Recording Device	Source	Recording Device	
Bonafide	4000	96K	Recording Chamber	Make	Model	Zoom R16	Male- 10	1st Order	2nd Order
Replay	4000		Kitchen Table	Audio-Technica	ST95MKII	Olympus		Zoom R16	Echo plus
Cloned	4000		Living Room	shure	SM58	LS-12		Laptop Asus GL504	Gen-2
Replay	4000		Office Desk	Behinger	ECM 8000		Female-9	GM-DS74	Echo plus
Total	12000		Dining Room	Electro-Voice	635 A/B			Ugreen 30521	Gen-3
Replay Samples			Vehicle Ground						

Table 3: Performance of the proposed method for bonafide Speaker Verification

Speaker ID	Precision	Recall	F1-score
LA_0079_bon	1.00	1.00	1.00
LA_0080_bon	0.97	0.94	0.96
LA_0081_bon	0.95	0.97	0.96
LA_0082_bon	1.00	1.00	1.00
LA_0083_bon	1.00	1.00	1.00
LA_0084_bon	1.00	1.00	1.00
LA_0085_bon	0.98	0.98	0.98
LA_0086_bon	0.95	1.00	0.97
LA_0087_bon	1.00	0.98	0.99
LA_0088_bon	1.00	0.97	0.99
LA_0089_bon	1.00	0.97	0.99
LA_0090_bon	1.00	1.00	1.00
LA_0091_bon	1.00	1.00	1.00
LA_0092_bon	1.00	0.98	0.99
LA_0093_bon	1.00	0.97	0.98
LA_0094_bon	1.00	1.00	1.00
LA_0095_bon	0.95	1.00	0.98
LA_0096_bon	1.00	1.00	1.00
LA_0097_bon	1.00	0.98	0.99
LA_0098_bon	0.95	1.00	0.97
Accuracy			0.99
Macro Avg	0.99	0.99	0.99
Weighted Avg	0.99	0.99	0.99

classes were 100%, while no class has more than one misclassified sample, and only 7 out of 774 testing samples were misclassified. Furthermore, even if the training and testing ratios were adjusted to 30-70 which means 774 records for training and the remaining for testing. Our solution still produced average precision, recall, f1-score, and accuracy values of 98 percent, indicating that our method successfully captures the unique vocal tract information of registered speakers; hence, our method is trustworthy for in-domain ASV tasks.

5.3 Experiment II—Audio Synthesis Algorithm Detection

By using the ASVspoof 2019 LA training dataset, we assessed the effectiveness of our solution for detecting synthetic audio production methods. As explained in section 5.1, the synthetic audio production methods include both voice conversion and speech synthesis algorithms. In this experiment, 70 percent of the data (15,874 samples) was used for model training to recognize six algorithm classes, while the remaining data (6,803 examples) was used for model testing. It can be seen from the data in Table 4 that our solution performs well in terms of all performance evaluation metrics, with a score of about 100 percent. After comparing the confusion matrices in both Figures Figure 12 and Figure 13, we can see that even when the testing samples are increased from 30% to 70%, and the training samples are decreased from 70% to 30%, the proposed method's algorithm detection performance remains sustained. As a result, our findings confirm that the algorithms produce unique properties in the generated cloned audios that are distinct from those produced by other audio generation algorithms; and that a good audio representation combined with an effective classification mechanism can exploit these artifacts to perform algorithm level detection, thereby increasing the reliability of the attack detection profile. This feature might also aid audio forensics applications by increasing credibility, which is important in court.

5.4 Experiment III—Performance Evaluation for Compromised Speaker Identification

The main goal of our experiment is to figure out which registered user voices have been hacked in order to attack the app. Additional security steps might be implemented

True Class	Neural-wave	1120	4	1	2	1	
	Vocoder-1	2	1102				
	Vocoder-2		1	1173	4		1
	Waveform	1	1	5	1037	2	10
	Vocoder-3		2			1171	
	Spectral				4		1159
		Neural-wave	Vocoder-1	Vocoder-2	Waveform	Vocoder-3	Spectral
		Predicted Class					

Figure 12: Confusion matrix analysis for voice cloning and voice algorithm detection using 70-30 ratio

True Class	Neural-wave	2594	28	8	19	10	
	Vocoder-1	7	2583	1	2	8	
	Vocoder-2	4	6	2618	5	1	3
	Waveform	12	10	25	2564	17	25
	Vocoder-3	6	9		4	2646	
	Spectral	1		2	11		2645
		Neural-wave	Vocoder-1	Vocoder-2	Waveform	Vocoder-3	Spectral
		Predicted Class					

Figure 13: Confusion matrix analysis for voice cloning and voice algorithm detection using 30-70 ratio

Table 4: Performance evaluation of the proposed method for synthetic algorithm recognition

Algo. ID	Algorithm	Precision	Recall	F1-score
A01	Neural waveform model	0.998	0.996	0.997
A02	Source filter vocoder-1	0.996	0.999	0.997
A03	Source filter vocoder-2	0.994	1.000	0.997
A04	Waveform concatenation	0.990	0.987	0.989
A05	Source filter vocoder-3	0.997	0.995	0.996
A06	Spectral filtering	0.998	0.997	0.998
Accuracy				0.996
Macro Avg		0.996	0.996	0.996
Weighted Avg		0.996	0.996	0.996

to safeguard the target users and their accounts if their user identification was stolen. As a result, we integrated the algorithm and speaker information in this experiment and used that knowledge to produce accurate labels for model evaluation. We labeled the algorithms from 'A01' through 'A06', as stated in Table 4, and we also labeled the users' IDs from 'LA00xx', with the spoof term in order to indicate that the audios are synthetic. We created 120 audio classes using 6 audio synthesis methods and 20 registered speakers from the ASVSpooof 2019 LA training dataset. We have given the results of 30 randomly selected classes in Table 5; from the data, we can see that our technique provides an accuracy of 97 percent. Similarly, the average value of all performance evaluation metrics is 97%. Table 4 and Table 5 have accuracy differences of less than 3%, which is due to the possibility of a sample's partial association with a particular output label; for example, a sample misclassified in terms of the real target speaker can still be associated with the correct voice cloning algorithm. Furthermore, because there were only 6 classes in algorithm identification (table 4), the margin of error was reduced. However, because our technique still performs well even when the drill down operation is used, we can conclude that our method dependably provides us with information about the compromised speakers, which

Table 5: Speaker identification whose voices are used to attack the system with a certain audio synthesis algorithm

Algo + Speaker ID	Precision	Recall	F1-score
A01_LA_0079_spoof	0.99	1.00	0.99
A01_LA_0080_spoof	0.96	0.98	0.97
A01_LA_0081_spoof	0.98	0.98	0.98
A01_LA_0082_spoof	1.00	0.99	0.99
A01_LA_0083_spoof	0.98	0.98	0.98
A02_LA_0084_spoof	1.00	0.98	0.99
A02_LA_0085_spoof	1.00	1.00	1.00
A02_LA_0086_spoof	0.98	1.00	0.99
A02_LA_0087_spoof	0.97	1.00	0.98
A02_LA_0088_spoof	0.98	1.00	0.99
A03_LA_0089_spoof	1.00	0.97	0.98
A03_LA_0090_spoof	0.97	0.95	0.96
A03_LA_0091_spoof	0.98	1.00	0.99
A03_LA_0092_spoof	1.00	1.00	1.00
A03_LA_0093_spoof	0.98	1.00	0.99
A04_LA_0094_spoof	0.99	0.99	0.99
A04_LA_0095_spoof	1.00	1.00	1.00
A04_LA_0096_spoof	1.00	0.97	0.98
A04_LA_0097_spoof	1.00	0.98	0.99
A04_LA_0098_spoof	0.98	0.94	0.96
A05_LA_0079_spoof	1.00	1.00	1.00
A05_LA_0080_spoof	1.00	1.00	1.00
A05_LA_0081_spoof	1.00	1.00	1.00
A05_LA_0082_spoof	1.00	1.00	1.00
A05_LA_0083_spoof	0.98	1.00	0.99
A06_LA_0094_spoof	0.94	0.94	0.94
A06_LA_0095_spoof	0.96	0.91	0.94
A06_LA_0096_spoof	0.86	0.89	0.87
A06_LA_0097_spoof	0.98	0.90	0.94
A06_LA_0098_spoof	0.88	0.83	0.85
Accuracy for 120 classes			0.97
Macro avg	0.97	0.97	0.97
Weighted avg	0.97	0.97	0.97

is also a unique feature of our method.

5.5 Cross-Dataset Evaluation

In our test, there are 76,236 previously unseen samples that were chosen for review.

A 9,902 among them are bonafide instances and the remaining are cloned examples. The

76,236 samples were made up of 5k instances from the ASVSpooof 2019 development set and the rest is made up from the evaluation set, which has not been used for training. All of these samples have unseen speakers where 20 of them made up from the development set and the remaining made up from the evaluation set, and 19 different voice conversion algorithms and voice cloning that include 6 algorithms from Table 4 and 13 algorithms from Table 7) are used to generate cloned audio for these 87 speakers. Our approach cannot anticipate algorithm labels since the algorithms used for voice cloning have not been utilized for training. As a result, we used the training set with two labels, bonafide and cloned, to train our model for this evaluation. The goal of this test was to see if our solution can distinguish between bonafide and cloned audios, regardless of who the speaker is or how the cloning is done. We can see from the presented results in Table 6, that our technique has an overall accuracy of 88%. By doing a drill down operation on this accuracy number, we discovered that the bonafide class's accuracy is 86%, while the cloned class's average accuracy is 90%, resulting in an overall accuracy of 88%. 20 speakers only are utilized for training, and those speakers are not taken into account for evaluating this test. Among these 87 speakers, the average accuracy remains above 90% for 72 speakers only, which is pretty high given that just 20 speakers are utilized for training and those speakers are not evaluated in this test. Similarly, as shown in Table 7, if we look at the 13 algorithms that were not utilized for training, we can observe that 8 of them have an accuracy of around 100%, while 2 of them have an accuracy of more than 90%. The most troublesome algorithms are A17-A19, which have a substantial reduction in accuracy. Table 7, on the other hand, shows that all of these algorithm classes have the smallest amount of samples. A17, which has the lowest accuracy, is only about 27% of A09, which

Table 6: By training on the LA-training set and testing on the LA-development and LA-evaluation sets, performance evaluation for unseen speakers and seen/unseen algorithms may be achieved.

Audio Label	Precision	Recall	F1-Score	EER	min t-dcf
Bonafide	0.67	0.91	0.81	5.22	0.132
Cloned	0.91	0.91	0.94		
Accuracy	0.88				

has the best accuracy (100%) and the greatest number of samples. As a result, we can conclude that model optimization has a positive relationship with sample size, and that, despite the lack of external algorithm labels, our model correctly identifies the correlation between the specific types of artifacts introduced by any synthetic algorithm and returns the correct output for the vast majority of samples.

A greater accuracy value is one of several prerequisites for a successful algorithm, which also includes algorithm performance in terms of recall, and f1-score and precision in class dependent scenarios. The main reason for the class dependent analysis is that even if a classifier overlooks the minor class in the event of unbalanced data, it will still offer greater overall accuracy and other performance assessment metrics. However, such higher assessment values are undesirable, because the minor class is generally the class of interest. We can observe that our approach has a 67% precision rate for the bonafide class and a 97% precision rate for the cloned class by looking at the results in Table 6. Because the precision measure also considers the false positive rate, the precision rate for the bonafide class drops when the data is highly imbalanced such as in our case, where the 13:87 ratio exists in both classes; however, because the false positives in the cloned class are lower, they do not have a significant negative impact on the cloned class's precision rate.

In the case of a recall, on the other hand, we only compared the properly categorized

Table 7: Cross dataset validation using unseen algorithms of the LA-evaluation set.

Algo. ID	Algorithm	No of Samples	Accuracy
A07	Vocoder+GAN	4823	0.98
A08	Neural waveform	4855	0.99
A09	Source filter vocoder-4	4893	1.00
A10	Neural waveform	4878	0.99
A11	Griffin lim	4882	0.99
A12	Neural waveform	4603	0.94
A13	waveform concatenation + waveform filtering	4908	1.00
A14	Source filter vocoder-5	4904	1.00
A15	Neural waveform	4747	0.97
A16	Waveform concatenation	4442	0.90
A17	Waveform filtering	1352	0.28
A18	Source filter vocoder-6	1855	0.38
A19	Spectral filtering	2345	0.48

instances in a class to all of the relevant examples for that class; as a result, the recall rates for the genuine class are 91%, which is around 24% greater than the accuracy rate. Similarly, recall rates for the cloned class fall by 6% and fall to 91%. As a result, our approach outperforms both the bonafide and cloned classes in terms of recall rate and accuracy rate. By combining the accuracy and recall rates using the f1-score, we get 81% and 94% for bonafide and cloned classes, respectively.

The f1-score differences suggest that our model requires a more extensive training set in order to correctly categorize the unseen bonafide cases. In real-world scenarios, however, because we only want our proposed SASV system to correctly classify the registered bonafide speakers over which the model has been trained as bonafide as shown in Table 3 and discussed in section 5.2), misclassifying the unregistered users, even if they are bonafide, it is an essential thing from a security standpoint. The system’s total EER is less than 6%, which is substantially lower when compared to the training and evaluation set sizes.

Table 8: The PA-evaluation set of ASVSpooof 2019, as well as the VSDC dataset, were used to assess performance for replay- and cloned replay attack detection.

Datasets	Sample Type	Precision	Recall	F1-Score	EER/ min t-dcf
VSDC	Bonafide	99	99	99	1.33 / 0.089
	Replay	98	98	98	
	Cloned Replay	98.9	98	98.4	–
ASVspooof	Bonafide	98	98	98	1.1 / 0.0335
	Replay	98	98	98	

5.6 Replay Attack Detection

Any bonafide speaker’s speech in a replay attack is pre-recorded and played again before the ASV systems. The artifacts that emerge during voice cloning are absent in the replay samples because the voice samples belong to actual speakers; consequently, the audio fingerprints match the bona fide speakers, and impersonation happens. However, a closer examination of the replay samples indicates that a recorded speech has non-linear components that can be utilized as a hint in the identification of replay attacks. To identify replay assaults, we must first define what a replayed sample is made up of:

5.6.1 Replay and Cloned Replay Patterns

A first-order speech replay attack may be described as a processing chain of microphone-speaker-microphone (MSM), which is equivalent to a cascade of three 2nd-order systems when the speakers act non-linearly. The processing chain depicting a first order replay assault is likely to produce higher order non-linearity due to the cascading of the MSM processing chain. As a result, higher-order harmonic distortions can be used to tell the difference between real and false audio. However, the voice cloning artifacts in cloned replays (added in the VSDC) also contain non-linear components and behave similarly to the MSM’s deeper chaining. Furthermore, cloned replays may be identified by con-

Table 9: Comparison against other feature extraction approaches using VSDC, LA- and PA-training sets of ASVspoof 2019.

Dataset	Features	EER/min t-dcf		
		Replay	Cloning	Cloned Replay
VSDC	MFCC-GTCC-Spectral	2.33/0.149	-	0.4/0.04
	ALTP-Spectral	2.5/0.164	-	1/0.061
	ALTP	2.9/0.194	-	1.2/0.072
	GTCC	7.5/0.497	-	4.1/0.29
	sm-ALTP	1.33/0.089	-	0.35/0.031
ASVspoof	MFCC-GTCC-Spectral	6.75/0.41	0.6/0.04	-
	ALTP-Spectral	1.5/0.091	0.8/0.053	-
	ALTP	3.4/0.24	0.9/0.06	-
	GTCC	8.4/0.561	6.1/0.42	-
	sm-ALTP	0.69/0.0169	0.5/0.037	-

currently collecting non-linear components and cloning artifacts using an efficient audio representation technique.

5.6.2 Replay and Cloned Replay Attack Detection

We evaluated the performance of the proposed technique for replay and cloned replay attack detection using the ASVspoof 2019 VSDC and PA-evaluation set. We can see from the findings in Table 8, that our technique performs admirably on both datasets when it comes to detecting audio replay attacks. On the VSDC and ASVspoof datasets, We got an F1-score of 98.4% and 99% , an EER of 1.33 and 1.1, and a min t-dcf score of 0.089 and 0.0335, respectively, with an average precision of 98.3% and 99%, a recall of 98.5% and 99%, and an F1-score of 98.4% and 99%, an EER of 1.33 and 1.1, and a min t-dcf score The results demonstrate that the suggested approach works somewhat better on the ASVspoof dataset than on the VSDC dataset, owing to the fact that VSDC samples are generated in more demanding and varied environments than the ASVspoof dataset. Our method beats the first-order replay attack in identifying cloned replay assaults in VSDC, confirming our findings that cloned signals become more distorted after replay than normal

Table 10: Comparison against state-of-the-art method on LA and PA evaluation sets of ASVspoof 2019.

Paper	Method	LA-Eval		PA-Eval	
		EER	min-tDCF	EER	min-tDCF
Baseline [74]	LFCC-GMM	11.96	0.212	13.54	0.3017
	CQCC-GMM	9.87	0.236	11.04	0.2454
ASSERT [39]	logSpec-SENet	11.75	0.216	1.29	0.036
	logspec-CQCC-SENet34-Mean-std-ResNet-SENet50-Dilated ResNet	6.70	0.155	0.59	0.016
STC [41]	LFCC-CMVN-LCNN	7.86	0.183	4.6	0.105
	FFT-LCNN	4.53	0.103	2.06	0.56
BUT-Omilia [86]	logSpec-VGG-SincNet 1-SincNet 2	8.01	0.208	1.51	0.0372
	SincNet with standard dropout	8.01	0.356	2.11	0.0527
	VGG 1-VGG 2	10.52	0.279	1.49	0.04
	SincNet with high dropout	22.99	0.381	2.31	0.0591
MFMT [42]	MFCC-CQCC-FBank-multi task learning	7.63	0.213	0.96	0.0266
DKU [14]	GD gram-ResNet	-	-	1.08	0.0282
Proposed	sm-ALTP-Asymmetric Bagging	5.22	0.132	1.1	0.0335

samples, making them more recognizable.

5.7 Comparison Against Other Feature Extraction Approaches

We evaluated our proposed sm-ALTP features to numerous acoustic characteristics for spoofing attack detection to better elucidate their efficacy. The chosen characteristics included a variety of MFCC, GTCC, ALTP, and spectral properties in various combinations. On the VSDC and ASVspoof 2019 LA and PA training datasets, the performance of various feature combinations was then assessed. According to the data in Table 9, , the proposed features beat all comparable features in terms of EER and min t-dcf scores for all types of spoofing assaults. As a consequence, the results of the comparison confirm the robustness of the suggested sm-ALTP characteristics.

Table 11: ASVspoof 2019 top 10 teams in LA and PA scenarios are compared to the suggested approach.

Position	Team	LA tdcf	LA Ranking	PA tdcf	PA Ranking	Average Ranking Score
1	T45	0.051	2	0.0122	2	2
2	T24	0.0953	4	0.0215	5	4.5
3	T05	0.0069	1	0.0672	12	6.5
3	T50	0.1118	5	0.035	8	6.5
4	Proposed	0.132	9	0.0335	9	9
4	T44	0.1554	15	0.0161	3	9
5	T60	0.0755	3	0.1492	21	12
6	T10	0.1829	23	0.0168	4	13.5
7	T02	0.1552	14	0.0614	12	13
8	T17	0.2129	30	0.0266	7	18.5
9	T53	0.2252	32	0.0219	6	19
9	T42	0.208	28	0.0372	10	19
10	T01	0.1409	12	0.2129	29	20.5
11	T58	0.1333	10	0.2767	40	25
12	T32	0.1239	8	0.281	43	25.5
13	T28	–	51	0.0096	1	26
14	T41	0.1131	6	0.5452	49	27.5
15	T39	0.1203	7	–	51	29
16	T04	0.1404	11	–	51	31
16	T07	–	51	0.057	11	31

5.8 Comparison Against State-of-the-art Methods

To see how successful the proposed method is in detecting spoofing attacks, we compared it to single-model alternatives such as [86], [74], [39], [41], [42] over LA and PA the ASVSpooF 2019 evaluation-set scenarios. It can be seen from the methodological details and results provided in Table 10, that the comparing techniques used a wide range of acoustic characteristics, as well as GMM and deep learning models. For purpose of comparison, our model is significantly simpler and more accurate, with a minimum t-dcf score of 0.1321; and only FFT-LCNN in [41] outperforms our technique in LA attack detection, while our method outperforms in PA attack detection. Similarly, DKU [14] beats our approach in PA attack detection, but their findings for LA attack detection are missing.

Despite obtaining the minimal value of the t-dcf measure is the desired aim, the total cost of the system should not grow to the point that the spoofing detection system's inclusion in real-time applications becomes problematic. In the case of FFT-LCNN [41], the model may suffer from sluggish training, which may take anywhere from hours to days, according to deep learning studies. Due to the linear time operation, our suggested feature extraction methodology is highly efficient, since the feature extraction time of our method is $\Theta(N)$.

To compare our technique to that of top challenge participants, we chose the top 10 teams from the top 50 teams in the LA and PA [71] (Table 11). Then, in terms of min t-dcf score, we compared their performance to our suggested technique and came up with a rating for the proposed system. In both situations, such as the LA and PA cases, our approach was placed ninth. However, in the LA situation, most of the systems that were

scored higher than our technique were ranked lower in the PA scenario, and vice versa. Furthermore, for the PA scenario, we assigned a ranking score of 51 to systems that were among the top 10 in the LA scenario but not among the top 50 in the PA scenario; similarly, for the LA scenario, we assigned a score of 51 to systems that were among the top 10 in the PA scenario but not among the top 50 in the LA scenario. By adding the LA and PA ranking values and dividing by two, the average ranking score of the comparison systems was determined. The average ranking score shows the overall performance of similar systems in both scenarios. For both the LA and PA situations, our approach was placed fourth in terms of cumulative performance based on the sorted ranking score. The ranking score clearly indicates the efficacy of the proposed technique, as well as its additional advantages, such as its small weight.

CHAPTER 6 CONCLUSION AND FUTURE WORK

6.1 Conclusion

ASV (automated speaker verification) is an important feature of speech biometric applications. These apps verify speakers based on their distinct voice features and safeguard user accounts against identity theft. However, security breaches occur as a result of synthetic audio creation algorithms and counterfeited audios created via digital manipulation, causing ASV systems to fail and making voice biometric applications unreliable. In similar fashion, smart speakers such as Google Home, Amazon Alexa, Siri, and other voice-activated devices that rely on the ASV system's robustness are vulnerable to audio spoofing attacks. A reliable countermeasure embedded in any ASV system ensures that these systems will remain protected against the unauthorized access.

The state-of-the-art countermeasure development research lacks in following aspects: a) ignores capturing of higher-order non linearity caused by the microphones against a pre-recorded voice to detect replay attacks. b) Overlooks voice cloning algorithm artifacts to detect voice cloning attack. c) Provides no clue about a possible counterfeiter, d) ignores different scenarios in which a spoofing attack can be launched that may fail countermeasures. e) Less sensitive against noise that make them incompatible for the real-world scenarios. f) Usually suffers class-imbalance problems if not specifically emphasized. In this dissertation, through a comprehensive secure automatic speaker verification (SASV) system all these problems are addressed.

The SASV system we propose identifies registered ASV users and defends against voice cloning/synthesis, voice replays, and cloned voice replay assaults. The audio synthesis

detection module distinguishes between authentic voices and algorithmically created synthetic/cloned audios, as well as providing information on the algorithm used to generate cloned audios. Voice replays and cloned-voice replay assaults are thwarted by the replay detection module.

The proposed system is based on unique sm-ALTP features and asymmetric bagging ensemble learning. By solving the class imbalance problem and recognizing various speaker and spoofing classes, our classifier ensemble technique takes a succession of poor classifiers and creates a stable classifier. According to our findings, the artifacts that arise as a result of microphone characteristics (in the case of replay) or synthetic audio creation techniques may be represented by using neighborhood statistics. However, in this case, the audio representation technique must also capture the speaker’s distinct voice qualities, which are unique to each speaker.

The assessment of our technique on the ASVspoof-2019 and VSDC datasets shows that it efficiently captures spoofing patterns even when they are created by unknown algorithms, resulting in a comprehensive security solution for ASV applications.

6.2 Future Work

In the future research work, we aim to perform further analysis of the proposed sm-ALTP features through deep learning models. The fundamental reason is that this study is slightly limited in terms of classification approaches, as there are several classification approaches available including the deep learning approaches but they are not very profoundly explored in this research. However, we want to analyze how various classifiers and particularly deep learning performs with the proposed features as it is one of the

hottest topics in the countermeasure development research.

Another potential area for our future research work is the analysis of spoofing attacks through light-based commands over mems-based microphones. We want to analyze that how our model will react on such commands, and will it be able to reliably counter the spoofing attacks without model optimization or specific optimizations will be required in this regard.

In this research work, we have explored cloned-replay attacks but there are other specific scenarios as well such as 2^{nd} order, and 3^{rd} order replay attacks particularly applicable to smart speakers which are chained together. The higher order represents the level of chaining in smart environments. Due to the deeper chaining, smart speakers trusts on the input audios coming through a neighboring speaker and ignores the possible failure of the countermeasure. Therefore, due to the trust issue, one countermeasure may allow the fake audios to penetrate in the system without even noticing the unauthorized access. However, this is the focus of our future research work.

REFERENCES

- [1] Everyday DSP for programmers: Signal variance. <http://sam-koblenski.blogspot.com/2015/09/everyday-dsp-for-programmers-signal.html>. Accessed: 2019-11-17.
- [2] M. Adiban, H. Sameti, and S. Shehnepoor. Replay spoofing countermeasure using autoencoder and siamese network on ASVspoof 2019 challenge. *arXiv preprint arXiv:1910.13345*, 2019.
- [3] S. M. Adnan, A. Irtaza, S. Aziz, M. ObaidUllah, A. Javed, and M. T. Mahmood. Fall detection through acoustic local ternary patterns. *Applied Acoustics*, 140:296–300, 2018.
- [4] S. Agarwalla and K. K. Sarma. Machine learning based sample extraction for automatic speech recognition using dialectal assamese speech. *Neural Networks*, 78:97–111, 2016.
- [5] M. J. Alam, P. Kenny, G. Bhattacharya, and T. Stafylakis. Development of crim system for the automatic speaker verification spoofing and countermeasures challenge 2015. In *Sixteenth annual conference of the international speech communication association*, 2015.
- [6] S. A. Alim and N. K. A. Rashid. *Some commonly used speech feature extraction algorithms*. IntechOpen, 2018.
- [7] K. R. Alluri and A. K. Vuppala. IIIT-H spoofing countermeasures for automatic speaker verification spoofing and countermeasures challenge 2019. *Proc. Interspeech 2019*, pages 1043–1047, 2019.

- [8] Z. Aslan and A. Mehmet. Performing accurate speaker recognition by use of svm and cepstral features. *The International Journal of Energy and Engineering Sciences*, 3(2):16–25, 2019.
- [9] A. R. Avila, M. J. Alam, D. D. O’Shaughnessy, and T. H. Falk. Blind channel response estimation for replay attack detection. In *INTERSPEECH*, pages 2893–2897, 2019.
- [10] B. Bakar and C. Hanilçi. Replay spoofing attack detection using deep neural networks. In *2018 26th Signal Processing and Communications Applications Conference (SIU)*, pages 1–4. IEEE, 2018.
- [11] R. Baumann, K. M. Malik, A. Javed, A. Ball, B. Kujawa, and H. Malik. Voice spoofing detection corpus for single and multi-order audio replays. *Computer Speech Language*, 65, 101132, 2021.
- [12] U. Bhattacharjee, S. Gogoi, and R. Sharma. A statistical analysis on the impact of noise on MFCC features for speech recognition. In *2016 International Conference on Recent Advances and Innovations in Engineering (ICRAIE)*, pages 1–5. IEEE, 2016.
- [13] L. Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.
- [14] W. Cai, H. Wu, D. Cai, and M. Li. The DKU replay detection system for the ASVspoof 2019 challenge: On data augmentation, feature representation, classification, and fusion. *arXiv preprint arXiv:1907.02663*, 2019.
- [15] J. P. Campbell. Speaker recognition: A tutorial. *Proceedings of the IEEE*, 85(9):1437–1462, 1997.
- [16] S.-Y. Chang, K.-C. Wu, and C.-P. Chen. Transfer-representation learning for detecting spoofing attacks with converted and synthesized speech in automatic speaker verification system. In *INTERSPEECH*, pages 1063–1067, 2019.

- [17] Z. Chen, Z. Xie, W. Zhang, and X. Xu. Resnet and model fusion for automatic spoofing detection. In *Proc. Interspeech*, pages 102–106, 2017.
- [18] B. Chettri, D. Stoller, V. Morfi, and S. E. a. L. B. M. A. Ramírez, B. Martínez and. Ensemble models for spoofing detection in automatic speaker verification. *arXiv preprint arXiv:1904.04589*, 2019.
- [19] R. K. Das, J. Yang, and H. Li. Long range acoustic and deep features perspective on ASVspoof 2019. In *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 1018–1025, 2019.
- [20] S. Davis and P. Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE transactions on acoustics, speech, and signal processing*, 28(4):357–366, 1980.
- [21] P. L. De Leon, I. Hernaez, I. Saratxaga, M. Pucher, and J. Yamagishi. Detection of synthetic speech for the problem of imposture. In *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4844–4847. IEEE, 2011.
- [22] H. Delgado, M. Todisco, M. Sahidullah, A. K. Sarkar, N. Evans, T. Kinnunen, and Z. Tan. Further optimisations of constant q cepstral processing for integrated utterance and text-dependent speaker verification. In *2016 IEEE Spoken Language Technology Workshop (SLT)*, pages 179–185. IEEE, 2016.
- [23] L. Deng and X. Li. Machine learning paradigms for speech recognition: An overview. *IEEE Transactions on Audio, Speech, and Language Processing*, 21(5):1060–1089, 2013.
- [24] N. W. Evans, T. Kinnunen, and J. Yamagishi. Spoofing and countermeasures for automatic speaker verification. In *Interspeech*, pages 925–929, 2013.

- [25] L. F. Gallardo. *Human and automatic speaker recognition over telecommunication channels*. Springer, 2015.
- [26] A. Gomez-Alanis, A. M. Peinado, J. A. Gonzalez, and A. M. Gomez. A light convolutional GRU-RNN deep feature extractor for asv spoofing detection. *Proc. Interspeech 2019*, pages 1068–1072, 2019.
- [27] T. Gunendradasan, S. Irtza, E. Ambikairajah, and J. Epps. Transmission line cochlear model based AM-FM features for replay attack detection. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6136–6140. IEEE, 2019.
- [28] J. Hao and T. K. Ho. Machine learning made easy: a review of scikit-learn package in python programming language. *Journal of Educational and Behavioral Statistics*, 44(3):348–361, 2019.
- [29] H. Hermansky. Perceptual linear predictive (plp) analysis of speech. *the Journal of the Acoustical Society of America*, 87(4):1738–1752, 1990.
- [30] T. K. Ho. The random subspace method for constructing decision forests. *IEEE transactions on pattern analysis and machine intelligence*, 20(8):832–844, 1998.
- [31] M. A. Hossan, S. Memon, and M. A. Gregory. A novel approach for mfcc feature extraction. In *2010 4th International Conference on Signal Processing and Communication Systems*, pages 1–5. IEEE, 2010.
- [32] A. Irtaza, S. M. Adnan, K. Ahmed, A. Jaffar, A. Khan, A. Javed, and M. T. Mahmood. An ensemble based evolutionary approach to the class imbalance problem with applications in CBIR. *Applied Sciences*, 8(4):495, 2018.

- [33] A. Irtaza, S. M. Adnan, S. Aziz, M. O. A. Javed, and, and M. T. Mahmood. A framework for fall detection of elderly people by analyzing environmental sounds through acoustic local ternary patterns. In *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 1558–1563. IEEE, 2017.
- [34] A. Janicki. Increasing anti-spoofing protection in speaker verification using linear prediction. *Multimedia Tools and Applications*, 76(6):9017–9032, 2017.
- [35] T. Kinnunen, K. A. Lee, H. Delgado, N. Evans, M. Todisco, M. Sahidullah, J. Yamagishi, and D. Reynolds. t-DCF: a detection cost function for the tandem assessment of spoofing countermeasures and automatic speaker verification. *arXiv preprint arXiv:1804.09618*, 2018.
- [36] T. Kinnunen, M. Sahidullah, H. Delgado, M. Todisco, N. Evans, J. Yamagishi, and K. A. Lee. The asvspoof 2017 challenge: Assessing the limits of replay spoofing attack detection. 2017.
- [37] A. Kumar and J. Mayank. *Ensemble Learning for AI Developers*. Springer, 2020.
- [38] J. Kumar, O. P. Prabhakar, N. K. Sahu, and P. Scholar. Comparative analysis of different feature extraction and classifier techniques for speaker identification systems: A review. *International Journal of Innovative Research in Computer and Communication Engineering*, 2(1):2760–2269, 2014.
- [39] C. Lai, N. Chen, J. Villalba, and N. Dehak. ASSERT: Anti-spoofing with squeeze-excitation and residual networks. *arXiv preprint arXiv:1904.01120*, 2019.
- [40] G. Lavrentyeva, S. Novoselov, E. Malykh, A. Kozlov, O. Kudashev, and V. Shchemelinin. Audio replay attack detection with deep learning frameworks. In *Interspeech*, pages 82–86, 2017.

- [41] G. Lavrentyeva, S. Novoselov, A. Tseren, M. Volkova, A. Gorlanov, and A. Kozlov. STC antispoofing systems for the ASVspoof2019 challenge. *arXiv preprint arXiv:1904.05576*, 2019.
- [42] R. Li, M. Zhao, Z. Li, L. Li, and Q. Hong. Anti-spoofing speaker verification system with multi-feature integration and multi-task learning. In *Proc. Interspeech*, pages 1048–1052, 2019.
- [43] L. Lin, R. Wang, D. Yan, and L. Dong. A robust method for speech replay attack detection. *KSII Transactions on Internet & Information Systems*, 14(1):168–182.
- [44] J. Makhoul. Linear prediction: A tutorial review. *Proceedings of the IEEE*, 63(4):561–580, 1975.
- [45] E. Media. Types of biometrics. <https://www.biometricsinstitute.org/what-is-biometrics/types-of-biometrics/>, 2021.
- [46] J. Mishra, M. Singh, and D. Pati. Processing linear prediction residual signal to counter replay attacks. In *2018 International Conference on Signal Processing and Communications (SPCOM)*, pages 95–99. IEEE, 2018.
- [47] S. Misra, T. Das, P. Saha, U. Baruah, and R. H. Laskar. Comparison of mfcc and lpcc for a fixed phrase speaker verification system, time complexity and failure analysis. In *2015 International Conference on Circuits, Power and Computing Technologies [ICCPCT-2015]*, pages 1–4. IEEE, 2015.
- [48] T. B. Mokgonyane, T. J. Sefara, T. I. Modipa, M. M. Mogale, M. J. Manamela, and P. J. Manamela. Automatic speaker recognition system based on machine learning algorithms. In *2019 Southern African Universities Power Engineering Con-*

- ference/Robotics and Mechatronics/Pattern Recognition Association of South Africa (SAUPEC/RobMech/PRASA)*, pages 141–146. IEEE, 2019.
- [49] P. Nagarsheth, E. Khoury, K. Patil, and M. Garland. Replay attack detection using DNN for channel discrimination. In *Interspeech*, pages 97–101, 2017.
- [50] R. Naika. An overview of automatic speaker verification system. *Intelligent Computing and Information and Communication*, pages 603–610, 2018.
- [51] S. Novoselov, A. Kozlov, G. Lavrentyeva, K. Simonchik, and V. Shchemelinin. Stc anti-spoofing systems for the ASVspoof 2015 challenge. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5475–5479. IEEE, 2016.
- [52] O. Okun, G. Valentini, and M. Re. *Ensembles in machine learning applications*, volume 373. Springer Science & Business Media, 2011.
- [53] T. B. Patel and H. A. Patil. Combining evidences from mel cepstral, cochlear filter cepstral and instantaneous frequency features for detection of natural vs. spoofed speech. pages 2062–2066, 2015.
- [54] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.
- [55] Á. PEDROZA, D. J. J. D. L. ROSA, V. JOSÉ, and A. BECERRA. Limited-data automatic speaker verification algorithm using band-limited phase-only correlation function. *Turkish Journal of Electrical Engineering & Computer Sciences*, 27(4):3150–3164, 2019.

- [56] B. S. M. Rafi, K. S. R. Murty, and S. Nayak. A new approach for robust replay spoof detection in asv systems. In *2017 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pages 51–55. IEEE, 2017.
- [57] S. Raj, P. Prakasam, and S. Gupta. Audio signal quality enhancement using multi-layered convolutional neural network based auto encoder–decoder. *International Journal of Speech Technology*, 24(2):425–437, 2021.
- [58] A. Ravanshad. Ensemble methods.2020. <https://medium.com/@aravanshad/ensemble-methods-95533944783f>, 2020.
- [59] M. Sahidullah, T. Kinnunen, and C. Hanilçi. A comparison of features for synthetic speech detection. 2015.
- [60] J. Sanchez, I. Saratxaga, I. Hernaez, E. Navas, D. Erro, and T. Raitio. Toward a universal synthetic speech spoofing detection using phase information. *IEEE Transactions on Information Forensics and Security*, 10(4):810–820, 2015.
- [61] W. M. Sanjaya, D. Anggraeni, and I. P. Santika. Speech recognition using linear predictive coding (lpc) and adaptive neuro-fuzzy (anfis) to control 5 dof arm robot. In *Journal of Physics: Conference Series*, volume 1090, page 012046. IOP Publishing, 2018.
- [62] S. K. Sarangi and G. Saha. Improved speech-signal based frequency warping scale for cepstral feature in robust speaker verification system. *Journal of Signal Processing Systems*, pages 1–14, 2020.
- [63] M. Saranya, R. Padmanabhan, and H. A. Murthy. Replay attack detection in speaker verification using non-voiced segments and decision level feature switching. In *2018*

- International Conference on Signal Processing and Communications (SPCOM)*, pages 332–336. IEEE, 2018.
- [64] A. K. Sarkar, Z. Tan, H. Tang, S. Shon, and J. Glass. Time-contrastive learning based deep bottleneck features for text-dependent speaker verification. *Ieee/acm Transactions on Audio, Speech, and Language Processing*, 27(8):1267–1279, 2019.
- [65] R. E. Schapire. The strength of weak learnability. *Machine learning*, 5(2):197–227, 1990.
- [66] H. Tang, Z. Lei, Z. Huang, H. Gan, K. Yu, and Y. Yang. The GMM and i-vector systems based on spoofing algorithms for speaker spoofing detection. In *Chinese Conference on Biometric Recognition*, pages 502–510. Springer, 2019.
- [67] L. Tang, P. Zhou, and X. Wei. A speaker verification system based on emd. In *2009 Third International Conference on Genetic and Evolutionary Computing*, pages 553–556. IEEE, 2009.
- [68] D. Tao, X. Tang, X. Li, and X. Wu. Asymmetric bagging and random subspace for support vector machines-based relevance feedback in image retrieval. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (7):1088–1099, 2006.
- [69] M. Todisco, H. Delgado, and N. Evans. A new feature for automatic speaker verification anti-spoofing: Constant q cepstral coefficients. In *Odyssey*, volume 45, pages 283–290, 2016.
- [70] M. Todisco, H. Delgado, and N. Evans. Constant q cepstral coefficients: A spoofing countermeasure for automatic speaker verification. *Computer Speech & Language*, 45:516–535, 2017.

- [71] M. Todisco, X. Wang, V. Vestman, M. Sahidullah, H. Delgado, A. Nautsch, J. Yamagishi, N. Evans, T. Kinnunen, and K. A. Lee. ASVspoof 2019: Future horizons in spoofed and fake audio detection. *arXiv preprint arXiv:1904.05441*, 2019.
- [72] J. Unpingco. *Python for probability, statistics, and machine learning*, volume 1. Springer, 2016.
- [73] X. Wang and X. Tang. Random sampling for subspace face recognition. *International Journal of Computer Vision*, 70(1):91–104, 2006.
- [74] X. Wang, J. Yamagishi, M. Todisco, H. Delgado, A. Nautsch, N. Evans, M. Sahidullah, V. Vestman, T. Kinnunen, K. A. Lee, et al. The ASVspoof 2019 database. *arXiv preprint arXiv:1911.01601*, 2019.
- [75] L. Wen and M. Hughes. Coastal wetland mapping using ensemble learning algorithms: A comparative study of bagging, boosting and stacking techniques. *Remote Sensing*, 12(10):1683, 2020.
- [76] M. Wester, Z. Wu, and J. Yamagishi. Human vs machine spoofing detection on wide-band and narrowband data. pages 2047–2051, 2015.
- [77] M. Witkowski, S. Kacprzak, P. Zelasko, K. Kowalczyk, and J. Galka. Audio replay attack detection using high-frequency features. In *INTERSPEECH*, pages 27–31, 2017.
- [78] Z. Wu, E. S. Chng, and H. Li. Detecting converted speech and natural speech for anti-spoofing attack in speaker recognition. In *Thirteenth Annual Conference of the International Speech Communication Association*, 2012.
- [79] Z. Wu, T. Kinnunen, N. Evans, J. Yamagishi, C. Hanilçi, M. Sahidullah, and A. Sizov. Asvspoof 2015: the first automatic speaker verification spoofing and countermea-

- sures challenge. In *Sixteenth annual conference of the international speech communication association*, 2015.
- [80] Z. Wu, L. P. L. De, C. Demiroglu, A. Khodabakhsh, S. King, Z. Ling, D. Saito, B. Stewart, M. W. W. Toda, et al. Anti-spoofing for text-independent speaker verification: An initial database, comparison of countermeasures, and human performance. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, 24(4):768–783, 2016.
- [81] Z. Wu, J. Yamagishi, T. Kinnunen, C. Hanilçi, M. Sahidullah, A. Sizov, N. Evans, M. Todisco, and H. Delgado. Asvspoof: the automatic speaker verification spoofing and countermeasures challenge. *IEEE Journal of Selected Topics in Signal Processing*, 11(4):588–604, 2017.
- [82] X. Xiao, X. Tian, S. Du, H. Xu, E. S. Chng, and H. Li. Spoofing speech detection using high dimensional magnitude and phase features: The NTU approach for asvspoof 2015 challenge. pages 2052–2056, 2015.
- [83] J. Yang and R. K. Das. Low frequency frame-wise normalization over constant-q transform for playback speech detection. *Digital Signal Processing*, 89:30–39, 2019.
- [84] Y. Yang, H. Wang, H. Dinkel, Z. Chen, S. Wang, Y. Qian, and K. Yu. The SJTU robust anti-spoofing system for the ASVspoof 2019 challenge. *Proc. Interspeech 2019*, pages 1038–1042, 2019.
- [85] H. Yu, Z. Tan, Z. Ma, R. Martin, and J. Guo. Spoofing detection in automatic speaker verification systems using DNN classifiers and dynamic acoustic features. *IEEE transactions on neural networks and learning systems*, 29(10):4633–4644, 2017.

- [86] H. Zeinali, T. Stafylakis, G. Athanasopoulou, J. Rohdin, I. Gkinis, L. Burget, J. Černocký, et al. Detecting spoofing attacks using VGG and sincnet: but-omilia submission to asvspoof 2019 challenge. *arXiv preprint arXiv:1907.12908*, 2019.
- [87] C. Zhang and Y. Ma. *Ensemble machine learning: methods and applications*. Springer, 2012.
- [88] C. Zhang, C. Yu, and J. Hansen. An investigation of deep-learning frameworks for speaker verification antispooofing. *IEEE Journal of Selected Topics in Signal Processing*, 11(4):684–694, 2017.
- [89] Y. Zhao, A. K. Shrivastava, and K. L. Tsui. Regularized gaussian mixture model for high-dimensional clustering. *IEEE transactions on cybernetics*, 49(10):3677–3688, 2018.

ABSTRACT**SECURE AUTOMATIC SPEAKER VERIFICATION SYSTEM**

by

MUTEB ALJASEM**December 2021****Advisor:** Dr. Mohammad Mehrmohammadi**Major:** Electrical Engineering**Degree:** Doctor of Philosophy

The growing number of voice-enabled devices and applications consider automatic speaker verification (ASV) a fundamental component. However, maximum outreach for ASV in critical domains e.g., financial services and health care, is not possible unless we overcome security breaches caused by voice cloning, and replayed audios collectively known as the spoofing attacks. The audio spoofing attacks over ASV systems on one hand strictly limit the usability of voice-enabled applications; and on the other hand the counterfeiter also remains untraceable. Therefore, to overcome these vulnerabilities, a secure ASV (SASV) system is presented in this dissertation.

The proposed SASV system is based on the concept of novel sign modified acoustic local ternary pattern (sm-ALTP) features and asymmetric bagging-based classifier-ensemble. The proposed audio representation approach clusters the high and low frequency components in audio frames by normally distributing frequency components against a convex function. Then, the neighborhood statistics are applied to capture the user specific vocal tract information. This information is then utilized by the classifier ensemble that is based on the concept of weighted normalized voting rule to detect various spoofing attacks.

Contrary to the existing ASV systems, the proposed SASV system not only detects the conventional spoofing attacks (i.e. voice cloning, and replays), but also the new attacks that are still unexplored by the research community and a requirement of a the future. In this regard, a concept of cloned-replays is presented in this dissertation, where, replayed audios contains the microphone characteristics as well as the voice cloning artifacts. This depicts the scenario when voice cloning is applied in real-time. The voice cloning artifacts suppresses the microphone characteristics thus fails replay detection modules and similarly with amalgamation of microphone characteristics the voice cloning detection gets deceived. Furthermore, the proposed scheme can be utilized to obtain a possible clue against the counterfeiter through voice cloning algorithm detection module that is also a novel concept proposed in this dissertation. The voice cloning algorithm detection module determines the voice cloning algorithm used to generate the fake audios.

Overall the proposed SASV system simultaneously verifies the bonafide speakers and detects the voice cloning attack, cloning algorithm used to synthesize cloned audio (in the defined settings), and voice-replay attacks over the ASVspoof 2019 dataset. In addition, the proposed method detects the voice replay and cloned voice replay attacks over the VSDC dataset. Rigorous experimentation against state-of-the-art approaches also confirms the robustness of the proposed research.

AUTOBIOGRAPHICAL STATEMENT



Muteb Aljasem received the B.S. degree in electrical engineering from West Virginia University, Morgantown, WV, USA, and the M.S. degree in electrical engineering from the University of Michigan, Dearborn, MI, USA. He is currently pursuing the Ph.D. degree in electrical engineering with Wayne State University, Detroit, MI, USA. He has been a teacher's assistant for a period of two years. He has one U.S. patent. His research interests include data science, machine learning, and cybersecurity.