Secure Automatic Speaker Verification (SASV) System through sm-ALTP Features and Asymmetric Bagging

Muteb Aljasem, Aun Irtaza, Hafiz Malik, *Senior Member, IEEE*, Noushin Saba, Ali Javed *Member, IEEE*, Khalid Mahmood Malik *Senior Member, IEEE* and Mohammad Meharmohammadi *Senior Member, IEEE*

***Abstract*—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 algorithms and replayed audios. Therefore, to overcome these vulnerabilities, a secure ASV (SASV) system based on the novel sign modified acoustic local ternary pattern (sm-ALTP) features and asymmetric bagging-based classifier-ensemble with enhanced attack vector is presented. 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. 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. Both the voice cloning algorithm detection and cloned-replay attack detection are novel concepts introduced in this paper. The voice cloning algorithm detection module determines the voice cloning algorithm used to generate the fake audios. Whereas, the cloned voice replay attack detection is performed to determine the SASV behavior when audio samples are simultaneously contemplated with cloning and replay artifacts.**

***Index Terms*—ASVspoof 2019, VSDC, logical access (LA) attack, physical access (PA) attack, secure ASV, countermeasures, and classifier ensembles.**

I. Introduction

AUTOMATIC speaker verification (ASV) is an essential component of voice biometric applications. These applications authenticate speakers based on their unique vocal characteristics and protects user accounts against identity theft. However, due to synthetic audio generation algorithms and counterfeited audios through digital manipulation, security breaches occur that fails the ASV systems and, hence, make the voice biometric applications unreliable. Similarly, smart speakers e.g. Google Home, Amazon Alexa, Siri etc., and many voice enabled devices in IoT that rely on the robustness of the ASV system are also prone to audio spoofing attacks as elaborated in [1].

Audio spoofing attacks over ASV systems can be categorized i.e. 1) imitation [2], 2) voice conversion [2], 3) synthesis (voice cloning) [3] 4) and replay [4], attacks. These attacks can be grouped into the Physical Access (PA) and Logical Access (LA) attack categories. In PA attacks, physical channel is accessed to launch the attack, whereas, in case of LA the audio is considered to be transmitted directly to the ASV systems. In replay attacks, which fall under the category of PA attacks, the prerecorded voice of the genuine target speaker is played back to deceive the ASV systems. Replay attack pose a threat as they are easy to launch, and the only precondition to launch this attack is to have a prerecorded speaker voice. Voice cloning technologies, which come under the LA attack category, take the prerecorded voice samples of a speaker and aims to produce speech samples that are perceptually indistinguishable from bonafide speech [5]. The speech samples generated through voice cloning algorithms are also hard to detect and needs the ASV systems to be specifically trained to recognize LA attacks.

In research, many state-of-the-art methods have been proposed to counter voice spoofing attacks. In this regard three community-led challenges of ASVspoof/2015/2017/2019 were launched to promote the development of countermeasures to protect ASV systems from the threat of spoofing [5]. The resulting systems were aimed to combine countermeasures with ASV in a plug-and-play manner either by placing (i) the countermeasure step followed by ASV, (ii) the ASV step followed by countermeasures, or (iii) in parallel [6]. In all these systems the spoofing detection was performed through different feature and classifier combinations by considering the spoofing detection as a binary classification problem [7]. As a first step, these approaches generate the audio representations through various feature combinations. Then, binary classifiers predict an input audio as spoofed or bonafide.

In contrast to the existing approaches, the proposed SASV system (Figure 1) aims to identify the speaker and liveliness of the input audio (i.e., the speech is genuine or spoofed) through a comprehensive framework. Furthermore, an enhanced attack vector is also introduced in the system to make it robust against spoofing attacks. The attack vector of conventional countermeasures are usually comprised of replayed or cloned audios only. In contrast, in the case of an LA attack, our method also detects the voice cloning algorithm, and the replay detection module also detects the cloned replay attacks. Furthermore, existing approaches consider that the generated audio through a voice cloning algorithm directly transfers to the anti-spoofing system, without first going through a physical

Hafiz Malik, and Aun Irtaza are with the Department of Electrical and Computer Engineering, University of Michigan-Dearborn, MI, 30332 USA e-mail: (see https://umdearborn.edu/users/hafiz).

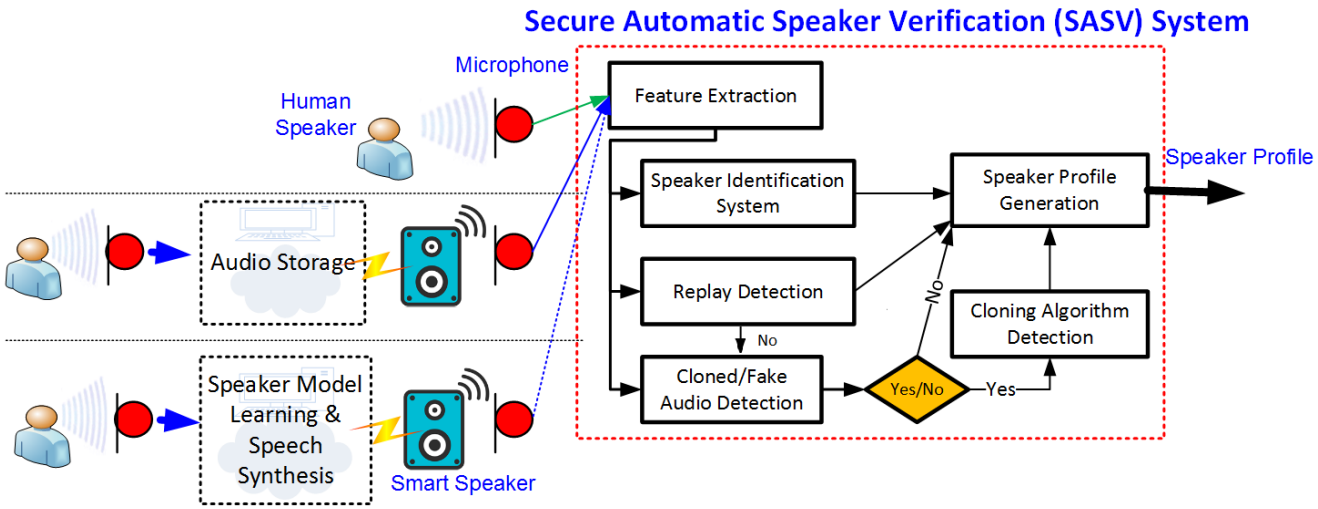


Fig. 1: Block diagram of Secure ASV (SASV) system.

channel. However, we have considered real-world LA attacks over PA attacks, where a physical channel will be used to launch the LA attacks.

For any input audio, after feature extraction, our framework performs the speaker identification step to determine which user is interacting with the system, and passes the speaker ID to the speaker profile generation module. Then, our system determine, if someone has attacked our system through a replayed audio or not, and pass the binary decision to the speaker profile generation module as well. If the input audio is not a replayed audio, then the framework analyzes the input audio for the voice cloning attack that possibly can be launched through a smart speaker or a microphone. For cloned audios, the framework (in the defined settings) also identifies the voice cloning algorithm that was used to generate the cloned voice samples. The voice cloning and algorithm decisions are also passed to the profile generation module. Our speaker profile generation module, grants the system access to only those speakers, where both replay and cloned flags are zero (to represent No). For the bonafide audios, the speaker profile contains the user information e.g., system-user ID, name, account number, account type etc., by accessing the main stream databases as per the application requirements. In the case of spoofing attack, the framework will return, the attack details i.e., which user was attacked, which algorithm or commercial solution was used to generate the fake audios, etc.

For feature extraction, a novel audio representation scheme i.e., sign modified acoustic local ternary pattern (sm-ALTP) features, is proposed. The sm-ALTP features are an extension of the ALTP features that we earlier proposed in [8]. The sm-ALTP captures the features corresponding to the vocal tract of a user and also determines the non-linearity that consequently comes in a signal due to the recording or voice cloning artifacts through the local correlation scores. The liveliness of the voice is determined through the SVM-based classifier ensemble that is generated through the asymmetric bagging and random subspace sampling over the feature repository. The classifier ensemble used in the proposed work takes a series of the weak classifiers and combines the classification output through the weighted normalized voting rule (wNVR) to generate a stable classifier. The generated model then verifies the speakers and detects the voice cloning attack, the cloning algorithm used for the attack (in the defined settings), the voice replays, and the cloned voice replay attacks over the ASVspoof 2019 dataset, and the voice spoofing detection corpus (VSDC). Through voice cloning algorithm detection, we want to further analyze the cases and scenarios that are challenging for our system and can cause failure to any existing countermeasure approach. The intention behind algorithm detection is to counter the commercial solutions that allow even amateurs to generate cloned audios. With our approach, after algorithm/commercial solution detection, the culprits can be identified easily depending on the severity of the case.

Our framework also detects cloned replay attacks, which is also a novel concept proposed in this paper. The cloned replays are comprised of the voice samples recorded by playing synthetic voice samples before the microphone. The applications of the cloned replays are possible in the scenarios where an attacker needs to play a recorded voice for impersonation (for instance before the smart speakers i.e., Google Home), but he lacks the prerecorded voice samples of the speaker. Thus, the model evaluation over the enhanced attack vector consequently empowers the proposed ASV system against various possible security breaches. Moreover, due to the lightweight nature of the proposed approach, our system can easily be adopted in resource constrained environments.

The main contributions of the proposed work can be summarized as follows:

- Development of a secure and lightweight ASV framework against multiple audio spoofing attacks.
  - Extension of the attack vector through cloning algorithm detection, and cloned replay attack detection, to further strengthen the ASV systems against the real-world cloning attacks.
- A novel feature extraction approach for audio representa-

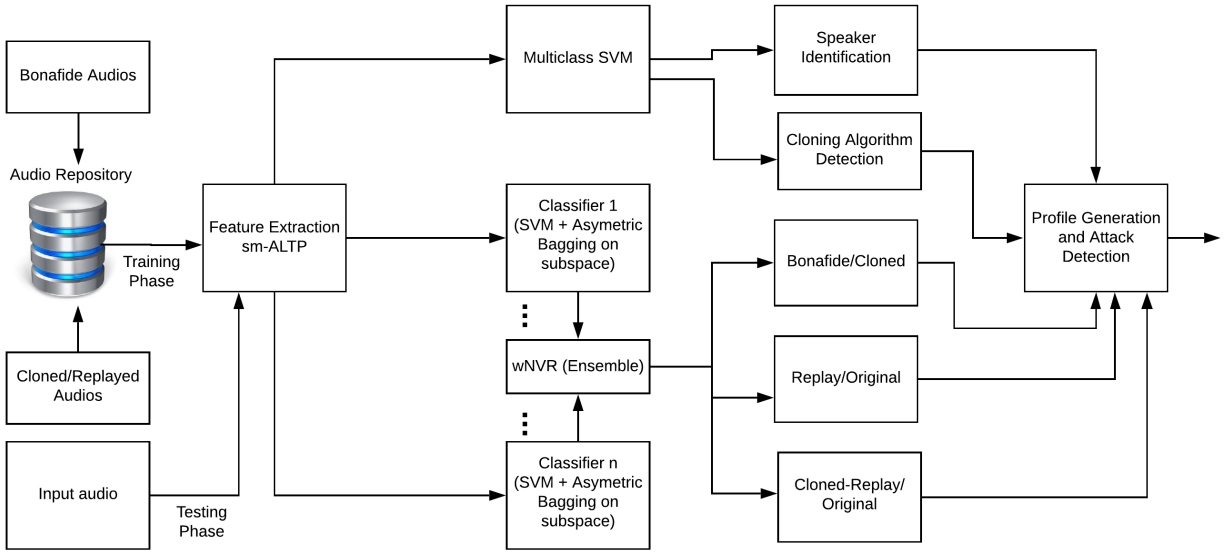


Fig. 2: Detailed architecture of SASV system.

tion capable of capturing speaker as well as attack specific attributes.

The rest of paper is organized as follows: Section II reviews the literature in spoofing attacks detection. Section III details the methodology used in SASV. Section IV provides the dataset and experimental details for performance evaluation. Last, the paper is concluded in section V.

II. Related Work

As we elaborated earlier, the research community has considered audio spoofing attack detection as a binary classification problem and aimed to produce countermeasures through different features and classifier combinations [7]. In feature domain cepstral coefficient features, i.e., constant-Q transform (CQT), Log-CQT, constant-Q cepstral coefficient (CQCC), extended CQCC (eCQCC), inverted CQCC (iCQCC), linear frequency cepstral coefficient (LFCC), Mel-frequency cepstral coefficient (MFCC) have been used widely [5], [9]–[12]. The benefit of CQT-based features stems from a variable spectro-temporal resolution, and captures the tell-tale signs of manipulation artifacts, which indicate spoofing attacks [13]. The CQT-based features provide a greater frequency resolution at low frequencies and a greater time resolution at high frequencies. However, it is difficult to couple them with traditional cepstral analysis approaches, which require post-processing to yield a linear frequency scale. This multi-resolution analysis together with further post-processing may impose a high computational load [14]. The CQCC, which is a derivative of CQT features, provides more spectral detail in the lower-frequency region but neglects the high-frequency region which provides more discriminative information. LFCC performs the time-frequency analysis of the entire input signal through discrete Fourier transform (DFT). However, the spoofing information is mainly found on low and high frequency sub-bands [15]. Therefore, the LFCC feature is unable to provide more spectral detail in the discriminative frequency bands [15]. Other cepstral features i.e., MFCC, are renowned features, however, their performance drops for spoofing detection due to sensitivity towards noise [16].

Phase-based features, e.g., relative phase shift, group delay, modified group delay, phase difference, and cosine normalized phase features, have also been explored in spoofing detection research [17]–[19]. Careful analysis reveals that, phase information is lost/changed during the analysis-synthesis step in some speech-synthesis approaches, which makes bonafide and spoofed speech different from each other. However, in practice, such prior knowledge is not available; thus, these features are not guaranteed to be effective to attacks which have unchanged phase information [20]. Other popular features are deep features, which are deep neural network hidden layer responses, used in [9], [21]–[23]. Although, the deep features provide competitive results, they cannot be used in resource constrained environments due to the need for expensive re-training.

For classification, the Gaussian mixture model (GMM) [5], [9], [24], [25], deep neural networks (DNN) [7], [12], [21], and classifier ensembles [26]–[28] have been widely used. The GMM restates the spoofing detection task as a basic hypothesis test, where whether an utterance belongs to a true speaker or not is determined through the likelihood ratio test. Although the GMM gives promising results, its performance degrades when high dimensional features are used [18], [29]. In contrast to the GMM, DNN classifiers can effectively handle high dimensional features. However, the DNN needs more training data than GMM. On the other hand, classifier ensembles take a series of weak classifiers on the subset of the data and generate a stable classifier by combining the classification outputs [30]. The ensemble approaches hardly overfit, allowing for solutions that are difficult to reach be achieved with a single hypothesis

[31].

A. Details of Specific Approaches

In [32] Todisco et al. trained the GMM classifier using the CQCC features for spoofing attack detection. The features provide a variable-resolution, time-frequency representation of the spectrum to capture the detailed characteristics of the input signal. Then, these characteristics were used to detect the spoofing attack. CQCC features outperformed earlier approaches for spoofing detection with a good margin. However, there was a marked discrepancy between the performance of the known and unknown spoofing attacks.

Nagarsheth et al. [33] used CQCC and high-frequency cepstral coefficients (HFCC) features and applied cepstral mean and variance normalization (CMVN) to generate the tandem features for replay attack detection. The CMVN removes the nuisance channel effects that have primarily been used for automatic speech recognition [34]. The tandem features were fed to a DNN to generate feature embeddings. The features were subsequently passed to a SVM classifier, which determines the replay attack type. The application of CMVN to detect the replay attack may seem counter-intuitive. The speech recording in different acoustic environments using different devices accumulates additional channel effects. CMVN, which aims to attenuate channel effects, uses this information to detect the replay attack. However, this assumption holds only, if bonafide speech was captured across a common, consistent channel [34].

The existing literature on voice replay spoofing detection [4], [35] trained the GMM classifier on various high-frequency features for replay detection. In [4], transmission line cochlea (TLC) features were used in conjunction with the GMM classifier to detect the replay attack. The TLC accurately resembles the auditory system and effectively uses amplitude modulation for replay attack detection. However, in TLC, the input and output signals vary in the same dynamic range. Therefore, for the large energy variation in the input signal, it becomes difficult to capture the discriminative information present in low energy regions. In [35], Witkowaski et al., emphasized that replay spoofing introduces spectral alterations at higher frequencies in the range of 6 to 8 kHz, which can be considered for replay attack detection. Several methods e.g., the inverted-MFCC, linear predictive cepstral coefficients (LPCC), and LPCC residual features in combination of CQCC, MFCC, and Cepstrum features were scrutinised alongside the GMM for replay attack detection. Although the method didn't solve the spoof detection problem completely, it introduced a significant improvement over the baseline CQCC-GMM system in ASVspoof-2017 challenge.

Several works [36]–[38] also focused on channel information, recording and playback device characteristics for replay attack detection. Saranya et al. [37] used MFCC, CQCC, and Mel-Filterbank-Slope (MFS) features to train the GMM for replay attack detection. Their work emphasized that the discriminative information used to categorize a signal as genuine or replayed speech is mainly distributed in two sub-bands, i.e., 0-1 kHz and 7-8 kHz. Yang et al. [38] employed the low frequency frame-wise normalization approach for voice replay spoofing detection.

Existing voice replay spoofing detection approaches have also employed various deep learning models. In [26], a fusion of GMM, DNN and ResNet classifiers was trained on MFCC and CQCC features to detect voice replay attacks. However, this method achieved a lower equal error rate (EER) but at the expense of increased computational cost. Bakar et al. [39] used the long-term average spectrum and MFCC features to train a deep neural network for replay attack detection. To overcome the limitations of higher computational cost, a light-weight CNN model originally proposed for face recognition was used in [40] to detect voice replay spoofing. Despite the computational cost, CNN and other deep learning model require large amounts of data to be trained effectively.

In [41]–[43], the GMM classifier was used for voice cloning attack detection. Leon et al. [41] extracted the relative phase shift features from the harmonic phase of the input audio signal and later used these features to train the GMM classifier for voice cloning detection. The model achieved good results, even though there were only 283 test samples. Moreover, the system is sensitive to the vocoder used for synthetic audio generation. To achieve the good performance, the vocoder used by the impostor must be used to train the system.

Wester et al. [42] employed the GMM-Universal background model (GMM-UBM) using the MFCC and cosine-normalized phase features for cloned voice detection. This is the first work that compared the performance of a system against 100 native English listeners. The results indicate that the automatic detectors outperformed the human listeners for all of the cases except one. The results also suggest that human and automatic countermeasures use different cues to discriminate between spoofed and genuine audios [44].

Patel et al. [43] used MFCCs, cochlear filter cepstral coefficients in combination with cochlear filter cepstral coefficients-instantaneous frequency features, to train a GMM to detect spoofing attack. The main findings of the work was that the countermeasures are more dependent on the robust features as compared to the classifiers.

Janicki et al. [45] used long term prediction residual signals to train SVM for voice cloning attack detection. The work considered the prediction coefficients such as the energy of the prediction error, prediction gains and temporal parameters related to the prediction error signals, etc., to differentiate between genuine and spoofed signals. The performance of the method is dependent on tuning the parameters, which negatively impacts the generalization capabilities. Due to this, the method showed better performance on known attacks as compared to unknown attacks.

B. Limitations of the Existing Approaches

As ASV systems are vulnerable to voice replay and voice cloning attacks, therefore, an effective countermeasure should consider the following facts during the audio representation step—(1) The microphone adds a layer of non-linearity due to inter-modulation distortions, which induces detectable patterns [36]; thus, an audio representation mechanism should be able

to characterize these patterns during audio-fingerprinting to discriminate between original and replayed audios. (2) The subsequent recordings of the same recording, which is very common in audio splicing, consequently introduce higher-order non-linearities and make an audio signal more distinguishable. Therefore, pattern analysis of the audio samples should be considered during the audio representation phase. (3) Similarly, voice cloning algorithms also introduce artifacts and need to be captured while selecting any audio representation schemes. As shown in (Figure 3), the spectral analysis of a genuine audio and its cloned version reveals that the finer lines were appearing in the spectral image of the cloned audio that represents the artifacts caused by the voice cloning algorithm. These lines were missing in the spectral image of the genuine audio. The voice cloning algorithm artifacts are unique; therefore, the cloned audios generated by different algorithms can be discriminated from each other and also from the bonafide ones. (4) An audio representation mechanism for ASV systems should be less sensitive to the noise for speaker verification under different environments. (5) For real-time applications, the ASV systems should consider those features and classifier combinations, which can ensure fast retraining of the model to incorporate new users.

III. Proposed Method

The main objective of the proposed work is to present a secure ASV (SASV) system to verify the registered bonafide speakers, and counter the voice cloning, voice replay, and cloned voice replay attacks. Moreover, in case of a voice cloning attack, it also identifies the cloning algorithm used to generate the cloned audios. In the proposed SASV system, the audio repository comprised of the replayed, cloned, and bonafide speaker-voices. The cloned-voices are generated through multiple voice cloning algorithms against each registered speaker. Thus, for m bonafide speakers and p voice cloning algorithms, we have $(m \times p)$ cloned-speaker classes. To counter the cloned audio samples, which are generated through any unseen voice-cloning algorithm, our model may incorrectly predict the cloning-algorithm type, but it will still detect the cloning attack successfully; in that case our model will label the input audio as cloned audio. Similarly, for replay attack detection, input audio samples are labeled as replayed/bonafide. Thus, there are $q = m + (m \times p) + 2 + 2$ number of speaker classes that we want to recognize.

As shown in Figure 2, for the bonafide voice samples of the registered users and the spoofed samples present in the audio repository, feature extraction is performed through the novel sm-ALTP features. Once, the feature extraction is done, we generate the SVM-based classifier ensembles through asymmetric bagging [30] and subspace sampling. The asymmetric bagging and subspace sampling also overcomes the class imbalance problem that naturally occurs, as bonafide samples are far fewer than the spoofed samples. The classifier ensembles integrates the outcome of multiple SVM classifiers by applying the weighted normalized voting rule (wNVR) to counter the voice cloning and replay attacks. The speaker identification module determines which registered user is interacting with the system, whereas, voice cloning algorithm detection module determines the voice cloning algorithm used to generate the fake audios. As the speech characteristics of each speaker and voice cloning algorithm artifacts are unique, the speaker identification and voice cloning algorithm detection is performed through a multi-class SVM classifier using the polynomial kernel. Once our models are trained, we use trained models to verify the input audio. To grant the system access to the identified speaker, the voice cloning and replay detection modules must give negative results. The details of the proposed method are covered in the following subsections.

A. Feature Extraction

1) Overview of ALTP Features: An input audio signal $Y[n]$ with N samples is partitioned into $i = \{1, 2, \ldots, k\}$ non-overlapping frames/windows $F^{(i)}$ with length $l = 9$. In each frame $F^{(i)}$, c represents the central sample in a frame and has $z^{(j)}$ neighbors, where j represents the neighbor index in the frame $F^{(i)}$. To compute the ALTP response, the difference between c and $z^{(j)}$ is computed by applying the parameter t_h around the sample c. The value of the parameter t_h lies between 0 and 1, and is obtained by performing linear search operation. Next, the sample values in $F^{(i)}$ are quantized to zero that lie in the range of width $\pm t_h$ around c, whereas values above and below $c \pm t_h$ are quantized to 1 and -1 respectively. Thus, we obtain a three-valued function as:

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

The function $p(c, z^{(j)}, t_h)$ is then decomposed into two patterns classes, i.e., upper pattern $P^{up}(.)$ and lower pattern $P^{lw}(.)$ as:

$$P^{up}(c, z^{(j)}, t_h) = \begin{Bmatrix} 1 & p(c, z^{(j)}, t_h) = +1 \\ 0 & Otherwise \end{Bmatrix} \quad (2)$$

Similarly

$$P^{lw}(c, z^{(j)}, t_h) = \begin{Bmatrix} 1 & p(c, z^{(j)}, t_h) = -1 \\ 0 & Otherwise \end{Bmatrix} \quad (3)$$

These upper and lower patterns are then used for upper and lower ALTP representation generation. The upper-ALTP features A_U are computed using eq. 4.

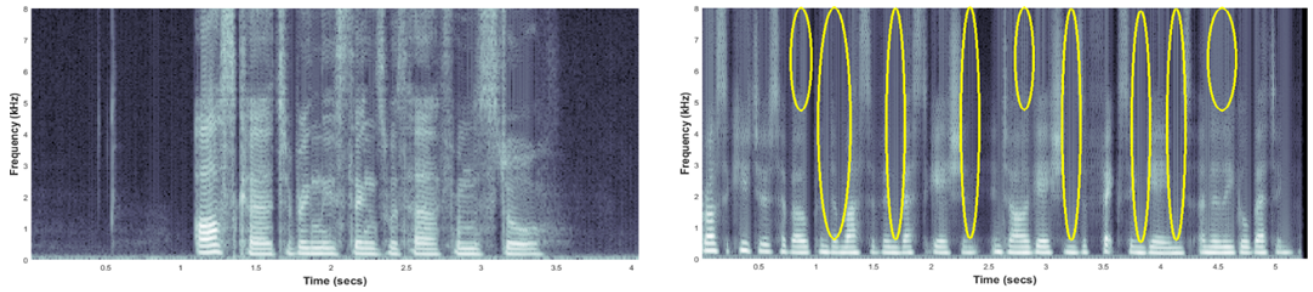
$$A_U = \sum_{j=0}^{j=l} P^{up}(c, z^{(j)}, t_h) * 2^j \quad (4)$$

whereas, lower-ALTP features A_L are computed through eq. 5.

$$A_L = \sum_{j=0}^{j=l} P^{lw}(c, z^{(j)}, t_h) * 2^j \quad (5)$$

Then, the histograms of A_U and A_L are computed by applying the Kronecker delta function $\delta(.)$ as described in eq. 6 and eq. 7

$$H^u(b) = \sum_{a=1}^{a=k} \delta(A_U^a, b) \quad (6)$$



(a) Original (b) Cloned

Fig. 3: Spectral Analysis of (a) original and (b) cloned utterances. The vertical lines, which appear only in the cloned audios, can serve as a potential clue for cloning attack. These lines can be captured by applying neighborhood statistics.

$$H^l(b) = \sum_{a=1}^{a=k} \delta(A_L^a, b) \tag{7}$$

Where b represents the bin and a represents the frame index. After computing the $H^u(b)$ and $H^l(b)$, the ALTP features are obtained by concatenating ($||$) both histograms as:

$$H_A = [H^u(b) \,||\, H^l(b)] \tag{8}$$

2) Limitations of ALTP Features: The ALTP features were originally proposed for indoor applications, i.e., fall detection [8], [46]; and due to the tolerance against noise showed very good performance as a feature descriptor against state-of-the-art feature extraction methods. However, there are some vulnerabilities of ALTP that needs to be overcome for application in ASV systems. These vulnerabilities are— (a) non-static pattern detection—as shown in Figure 3 the spectral analysis of the cloned audio reveals that the artifacts have a non-static repetition pattern, which can be more effectively captured through a dynamic threshold mechanism. However, the ALTP has only the static threshold, i.e., $\pm th$; thus, room for improvement exists in ALTP for ASV applications. (b) Signal volatility—To effectively capture the artifacts in cloned and replayed audios, it is important to know how quickly the signal is changing in terms of artifacts [47]. However, the ALTP features lack this attribute. Hence, the performance drops against the spoofed audios. (c) Brute-force Optimization—in ALTP a brute-force approach for threshold optimization was required; consequently, error reduction was not guaranteed in time critical applications. (d) Noise uniformity—ALTP was robust against the uniform noise that remains consistent in the audio scenes e.g. indoor audios. In contrast, in outdoor environments as the noise is non-uniform, therefore, the static threshold-based feature extraction becomes inconsistent and, hence, demands a different approach for noise suppression.

3) Motivation for the sm-ALTP Features: In order to overcome the limitations of ALTP features and to detect the liveliness of the voice in an effective way, sm-ALTP features are proposed. The sm-ALTP features use the dynamic optimizable threshold that effectively captures signal artifacts and generates different representations for bonafide and spoofed voices. Thus, the difference of representation for bonafide and spoofed vices results in the form of a strong CM approach. Furthermore, the exploitation of the vocal tract information, which was missing in the ALTP features, can also boost speaker identification/ recognition capabilities.

4) sm-ALTP Features: sm-ALTP features overcome the vulnerabilities of ALTP features by defining a dynamic optimizable threshold and capturing the vocal tract of the speaker. In sm-ALTP we compute the three valued function as:

$$p(c, z^{(j)}, \sigma\alpha) = \begin{Bmatrix} -1 & z^{(j)} - (c - \sigma\alpha) \leq 0 \\ 0 & (c + \sigma\alpha) < z^{(j)} < (c - \sigma\alpha) \\ +1 & z^{(j)} - (c + \sigma\alpha) \geq 0 \end{Bmatrix} \tag{9}$$

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

$$\sigma = \sqrt{\frac{\sum \left(z^{(j)}\right)^2 - \left(\frac{\sum z^{(j)}}{l}\right)^2}{l - 1}} \tag{10}$$

By replacing t_h with $(\sigma \times \alpha)$ we overcome the limitations (a),(c), and (d) of the ALTP features (section III-A2), which demands the incorporation of the signal variance in terms of neighborhood statistics. Another limitation of the ALTP feature was that the t_h needed the brute-force optimization through linear search. However, by defining the following convex function we can optimize the new threshold value, i.e., $\sigma\alpha$.

$$J(\sigma) = min \frac{\alpha}{2M} \sum_{q=1}^{q=M} \left(g\left(\theta^T \sigma\left(x^{(q)}\right)\right) - y^{(q)} \right)^2 \tag{11}$$

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

$$p(y^{(q)}|x^{(q)}; \sigma) = \frac{1}{\sqrt{2\pi}\sigma} exp\left(- \frac{y^{(q)} - x^{(q)}}{2\sigma^2} \right) \tag{12}$$

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

$$\sigma_{new} = \sigma - \alpha * \frac{\partial \sigma}{\partial z^{(j)}} \left(\sqrt{\frac{\sum \left(z^{(j)}\right)^2 - \left(\frac{\sum z^{(j)}}{l}\right)^2}{l - 1}} \right) \tag{13}$$

Where

$$\frac{\partial \sigma}{\partial z^{(j)}} = \left[\frac{\partial \sigma}{\partial z^{(1)}} \quad \frac{\partial \sigma}{\partial z^{(2)}} \quad \cdots \quad \frac{\partial \sigma}{\partial z^{(l)}} \right] \tag{14}$$

Thus

$$\frac{\partial \sigma}{\partial z^{(j)}} = \frac{1}{\sqrt{l-1}} * \frac{\partial}{\partial z^{(j)}} \left[\hat{A} + \hat{B} \right]^{1/2} \tag{15}$$

Where

$$\hat{A} = \left(z^{(1)}\right)^2 - \left(\frac{\sum z^{(j)}}{l} \right)^2 + \cdots + \left(z^{(c-1)}\right)^2 - \left(\frac{\sum z^{(j)}}{l} \right)^2 \tag{16}$$

And

$$\hat{B} = \left(z^{(c+l)}\right)^2 - \left(\frac{\sum z^{(j)}}{l} \right)^2 + \cdots + \left(z^{(l)}\right)^2 - \left(\frac{\sum z^{(j)}}{l} \right)^2 \tag{17}$$

or in compact form we can write it as:

$$\frac{\partial \sigma}{\partial z^{(j)}} = \frac{1}{\sqrt{l-1}} * \frac{\partial}{\partial z^{(j)}} \left(\sum \left(z^{(j)}\right)^2 - \left(\frac{\sum z^{(j)}}{l} \right)^2 \right)^{1/2} \tag{18}$$

thus, the partial derivative will return:

$$\frac{\partial \sigma}{\partial z^{(j)}} = \frac{1}{2\sqrt{l-1}} * \left(\sum \left(z^{(j)}\right)^2 - \left(\frac{\sum z^{(j)}}{l} \right)^2 \right)^{-1/2} * \left(2z^{(j)} - \frac{2 \sum z^{(j)}}{l^2} \right) \tag{19}$$

or

$$\frac{\partial \sigma}{\partial z^{(j)}} = \frac{1}{2\sqrt{l-1}} * \frac{1}{\sqrt{\left(\sum \left(z^{(j)}\right)^2 - \left(\frac{\sum z^{(j)}}{l} \right)^2 \right)}} * \left(2z^{(j)} - \frac{2 \sum z^{(j)}}{l^2} \right) \tag{20}$$

By replacing the eq. 2-5 with $(\sigma \times \alpha)$ we get the $H^u(b)$ and $H^l(b)$ using eq. 6 and 7 and generate feature representation as:

$$H = [H^u(b) \,||\, H^l(b)] \tag{21}$$

The feature representation H captures the patterns present in the input signal, but this representation lacks the vocal tract information that can be captured through the cepstral coefficients at Mel-scale [48]. For instance, at 1000 Hz the cepstral coefficients of a particular speaker always appear negative due to the phoneme representation attributed to the vocal structure of that particular speaker, and this frequency occurs very frequently; in case of sm-ALTP a large positive histogram-spike will appear, but it will not provide any information regarding the vocal behavior at this particular frequency. Therefore, we have further processed the sm-ALTP representation using eq. 22.

$$H_s = H \times sgn(\mu_t(C_\gamma(t))) \times \beta \tag{22}$$

Where $C_\gamma(t)$ is the t^{th} order MFCC of the γ^{th} frame (more details in [49]), μ_t is the frame-wise mean of $C_\gamma(t)$, and $t = \{1, 2, \ldots, 20\}$. $C_\gamma(t)$ is applied by computing the frame energy $E(f)$ with index f as represented in eq. 23.

$$C_\gamma(t) = \sum_{f=0}^{g-1} log\big[E(f)\big] cos\big[t\big(f - \frac{1}{2}\big)\frac{\pi}{q}\big] \tag{23}$$

The parameter $\beta = 0.1$ in eq. 22 is used for the feature normalization in H_s. Our final representation of sm-ALTP features then can be represented as:

$$H_{sm} = \big[\mu_t(C_\gamma(t)) || H_s\big] \tag{24}$$

B. Classifier Comity Learning for Ensembles

No matter how powerful a feature extraction method is, the characteristics of data in terms of data-quality, data-collection mechanism, and dataset size affects the classification performance in ASV systems. For instance, if a training-set is comprised of fewer bonafide representations, and far more spoofed representations, it may cause a classifier to tend towards the spoofed class. In this particular case, higher classification accuracy may be an outcome of the bias towards the spoofed class; in reality, the classifier is giving far lower performance for the bonafide samples, which is a primary goal of any ASV system. Thus, even the higher classification accuracy will become insignificant. Meanwhile, it is fundamentally important to identify the reasons why classifiers generate the wrong output. In order to achieve this objective, for cloning attack detection we also identify the cloning-algorithm used for spoofed audio generation. By capturing the correlation between spoofed samples and the cloning-algorithm, classification models can be further improved. Furthermore, we have ensured that the complexity of the testing process may not increase in a way that makes the classification model inappropriate for a real-time application.

1) Training-Phase—Asymmetric Bagging and Subspace Sampling: In order to generate multiple classifiers, asymmetric bagging and subspace sampling are used [30]. In asymmetric bagging, bootstrapping is executed over the spoofed class samples as there are far more spoofed samples as to bonafide samples. This way each classifier is trained over a balanced set using the complete bonafide-set and a subset of spoofed samples, thus improving unstable SVM classification performance. The stable SVM classifiers then become able to discriminate well even the unseen bonafide and spoofed samples. However, if instead of using the asymmetric bagging, other data balancing methods are used, i.e., up-sampling, or down-sampling, the classifier either becomes over-fit or under-fit. After the asymmetric bagging, the aggregation of multiple classifiers is performed through the weighted normalized voting rule (wNVR) over the development-set.

2) Weighted Normalized Voting Rule (wNVR): After training multiple classifiers, wNVR is applied to aggregate the outcomes of all of these classifiers. The reason to choose wNVR over majority voting rule (MVR) is that MVR is unable to take advantage of the accurate classifiers and give equal weight to all of the classifiers [50].

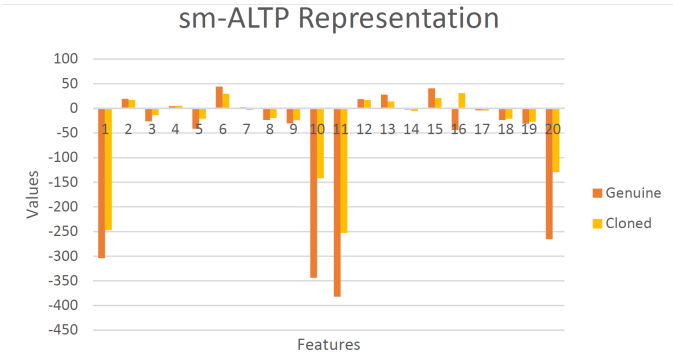


Fig. 4: sm-ALTP representation (eq. (21)) for genuine and cloned audios.

Let $w = \{1, 2, \ldots, Q\}$ classifiers are used to generate the ensemble classifier by applying weighted cross-entropy function as described in eq (25):

$$C(x) = \sum_{w=1}^{Q} \lambda_w \sum_{b=1}^{M} \sum_{k=1}^{K} [y_b = k] \log \frac{e^{\theta_k^T x_b}}{\sum_{v=1}^{K} e^{\theta_v^T x_b}} \tag{25}$$

Where λ is the weight, to take the advantage of more accurate classifier for $k = \{1, 2, \ldots, K\}$ number of classes to be classified, $b = \{1, 2, \ldots, M\}$ are the number of instances x_b in the development-set. The final class-label $C^*(x)$ is then generated through the eq. (26):

$$C^*(x) = sgn\left[C(x) - \frac{K-1}{2 \times s}\right] \tag{26}$$

The parameter s is the normalization factor to control the bias/variance effect.

3) Testing Phase: After the training and model optimization, the trained model can be used for the evaluation purposes. The evaluation-set is comprised of the examples having seen and unseen bonafide speakers, and in case of a voice-cloning attack having samples generated through seen and unseen algorithms. After model evaluation, any query audio sample can be passed to the final model, and it can perform the ASV tasks in the real-time scenarios.

C. Overcoming the Limitations of Existing Approaches

As described in section II-B, the existing approaches ignore some important signal characteristics during feature extraction, which consequently lowers their performance. For instance, the first three limitations emphasize that during replay and voice cloning, the inter modulation and algorithm artifact appear, which exhibits distinguishable patterns. The proposed approach performs the pattern analysis of the input signal, thus effectively capturing these artifacts to distinguish the spoofed signals from the bonafide. For instance, as shown in Figure 4, the bonafide and cloned signals exhibit the peak at the same feature points, but due to the difference of peaks, these signals are still easily distinguishable. Moreover, at some feature points e.g., feature 16 in Figure 4, the bonafide and spoofed signals exhibit the peaks at opposite directions. The difference of the feature values in Figure 4 shows that the cloned audio appears similar to the genuine one, but the essential signal components i.e., pitch, loudness, etc., are still not perfectly replicated. However, the lower level analysis of the input signal through the proposed approach easily reveals this difference.

TABLE I: Number of non-overlapping target speakers and number of utterances in training and development sets of the ASVspoof 2019 database.

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

Another limitation of the audio representation approaches was that their robustness against noise was not easily quantifiable. However, the proposed approach is robust against noise, and we can easily verify this claim. For instance, consider the audio frame shown in Figure 5. We can observe that the additive noise, which can either increase or decrease the value of central sample c in a frame $F^{(i)}$ and become a cause to generate the wrong code against c, will become ineffective. The reason is that, the value of the sample c now lies in a range of upper and lower threshold values; hence, becomes more tolerant against additive values by noise. Moreover, due to the less complex features, fast model retraining is possible; thus, it makes our approach effective for the applications that have continuous user enrollment requirements.

IV. Experiments and Results

A. Dataset

Performance of the proposed method is evaluated on ASVspoof 2019 [51] dataset, and voice spoofing detection corpus (VSDC) [52].

ASVspoof 2019 dataset (Table I) is further comprised of two datasets, i.e., logical access (LA) dataset for voice-cloning attacks detection, and physical-access (PA) dataset for replay attack detection. The LA-dataset has 25,380 samples for training-, 24,844 samples for development-, and 71,933 samples for evaluation- purposes. The training- and development-set contains the voice samples of 20 speakers (different speakers in both sets) that serves as the bonafide classes whereas, the spoofed-set has cloned samples of the same speaker utterances generated through 2 voice-conversion and 4 speech synthesis

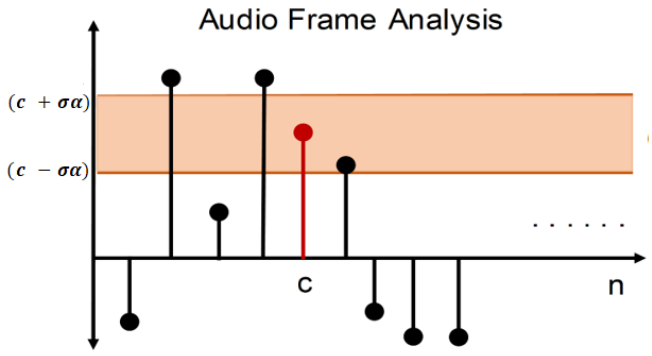


Fig. 5: Effect of the dynamic threshold over the audio frames.

TABLE II: Details of Voice Spoofing Detection Corpus (VSDC).

Audio Samples		Sample Rate	Environment	Microphone		Recording Device	Recording Source	Recording Device	
Bonafide	4000	96K	Recording Chamber Kitchen Table Living Room Office Desk Dining Room Vehicle Ground	Make	Model	Zoom R16 Olympus LS-12	Male Speakers 10 Female Speakers 10	1st Order Replay	2nd Order Replay
Replay	4000			Audio-Technica shure Behinger Electro-Voice	ST95 MKII SM58 ECM 8000 635 A/B			Zoom R16 Laptop Audio Asus GL 504 GM-DS74 USB Audio Card Ugreen 30521	Echo plus Gen-2 Echo plus Gen-3
Cloned Replay	4000								
Total	12000								

algorithms comprised of 120 (20×6) cloned speaker-plus-algorithm classes. The voice-conversion algorithms are based on (i) neural-network-based, and (ii) transfer-function-based methods. In contrast, the speech synthesis algorithms are an implementation of (i) waveform concatenation, (ii) neural-network-based parametric speech synthesis using source-filter vocoders, and (iii) neural-network-based parametric speech synthesis using Wavenet [51]. The evaluation-set includes unseen bonafide and spoofed speech samples collected from 67 speakers, and the spoofed-set includes samples generated through 19 algorithms including the GAN-based, and deep neural network-based methods. The PA-dataset comprises of 54,000, 33,534, and 1,53,522 training, development, and evaluation samples, respectively (Table I). The details of ASVspoof 2019 corpus can be found at [51].

VSDC was designed for replay and cloned replay attack detection. Cloned-replay represents the recording of cloned voice samples; for this the ASVspoof cloning samples were used to generate the replay samples in a manner similar to what was done for the bonafide voice recordings. The samples in the dataset are diverse in terms of environment, configurations, speaker-genre, recording, playback-devices, and number of speakers (Table II). More specifically, the samples contain noise and interference as well. To generate the replays, different playback devices were used to combat the effect of a specific playback device. VSDC includes the voice samples of ten male and nine female speakers who volunteered their services for data collection.

B. Experiment I—Performance Evaluation for Speaker Verification

In this experiment, the performance of the proposed method is evaluated for bonafide speaker verification. Bonafide speaker verification is the primary task performed by any ASV system. For this experiment, all the 2580 audio samples corresponding to the 20 bonafide speakers were selected from the ASVspoof 2019 dataset. Amongst these samples 70% of the data (i.e., 1806 records) was used for training of the model and 30% data (i.e., 774 records) was used for the testing purposes. As shown in Table III, the proposed method achieved on average 99% precision, recall, f1-score, and accuracy values. For most of the classes the evaluation rates were 100%, whereas there was no class that had more than 1 misclassified sample; and amongst 774 testing samples only 7 samples were misclassified, Moreover, even if we changed the training and testing ratios as 30-70 (i.e., 774 records for training and 1806 records for testing), our method still gave 98% average precision, recall, f1-score, and accuracy values, which clearly signifies that our method effectively captures the unique vocal tract information of the registered speakers; thus, our method is reliable for the in-domain ASV tasks.

TABLE III: Performance of the proposed method for bonafide Speaker Verification over LA-training dataset.

Precision	Recall	f1-Score	Accuracy
0.99	0.99	0.99	0.99

C. Experiment II—Voice Cloning Algorithm Detection

In this experiment, we evaluated the performance of the proposed method for synthetic audio generation algorithms detection using ASVspoof 2019 LA-training dataset. The synthetic audio generation algorithms is comprised of both voice conversion, and speech synthesis algorithms as described in section IV-A. For this experiment, amongst 22,800 samples, 70% of the data (i.e., 15,874 samples) was used for model training to recognize 6 algorithm classes, and 30% of the data (i.e., 6,803 samples) was used for model testing. From the results presented in Table IV, it can be observed that our method gave approximately 100% performance in terms of all the performance evaluation measures. Even if we increased the testing samples from 6,803 to 15,874 and decreased the training samples from 15,874 to 6,803, the algorithm detection performance of the proposed method still remained constant. Hence, the results confirm that each algorithm induces its specific properties/artifacts in the generated cloned audios that usually differ from the other audio generation algorithms, and a good audio representation with an effective classification mechanism can exploit these artifacts to perform the algorithm level detection; consequently, the attack detection profile becomes more reliable. This feature can also benefit the audio forensics applications by inciting more credibility particularly in court cases.

D. Experiment III—Performance Evaluation for Compromised Speaker Identification

The objective of this experiment is to identify which registered user voices have been compromised to attack the application. Through compromised user identification additional security measures could be taken to further protect the target user accounts. Thus, in this experiment, we combined the algorithm and speaker information and used this information to generate the true labels for model evaluation. The algorithms are represented with the label 'A01' to 'A06' as described in Table IV, and users IDs are represented as 'LA_00xx' and the term 'spoof' is included to show that the audios are synthetic. Thus, using 6 voice cloning algorithms, against 20 registered

TABLE IV: Performance evaluation of the proposed method for synthetic algorithm recognition over LA-training dataset.

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			

TABLE V: Speaker identification whose voices were used to attack the system with a certain voice cloning algorithm over LA-training dataset.

Algo + Speaker ID	Precision	Recall	F1-score
A01_LA_0079_spoof	0.99	1.00	0.99
A02_LA_0086_spoof	0.98	1.00	0.99
A03_LA_0091_spoof	0.98	1.00	0.99
A04_LA_0095_spoof	1.00	1.00	1.00
A05_LA_0081_spoof	1.00	1.00	1.00
A06_LA_0095_spoof	0.96	0.91	0.94
Accuracy for 120 classes	0.97		

speakers, present in ASVspoof 2019 LA-training dataset, we generated 120 audio classes. In Table V we present the results of the 6 randomly selected classes, from the results, we can observe that our method gives 97% accuracy, and the average value of all the performance evaluation measures is also 97%. The difference between the accuracy values of Table IV and Table V is 2.6%, which is due to the probability of a sample's partial association with a particular output label; for instance, a miss-classified sample in terms of the real target speaker can still be associated with the correct voice cloning algorithm. Moreover, as in case of algorithm detection (table IV) as there were only 6 classes, the margin of error was lower. However, our approach still gives high performance even by applying the drill down operation. Thus, on the basis of the results we can say that our method reliably provides us the information regarding the compromised speakers, which is also a unique attribute of our method.

E. Cross-Dataset Evaluation

For this experiment, 76,236 unseen examples were selected for evaluation purposes. Amongst these examples 9,902 examples are bonafide, and 66,334 examples are cloned. These 76,236 examples are comprised of 5000 examples from the ASVspoof 2019 development-set, and 71,236 examples from the evaluation-set, which are never used for the training purposes. All of these examples have unseen speakers (20 speakers from development-set and 67 speakers from evaluation-set), and 19 different voice-cloning, and voice-conversion algorithms (including 6 algorithms mentioned in Table IV and the remaining 13 in Table VII) are used for cloned audio generation of these 87 speakers. As algorithms used for the voice cloning are never used for training of our method, our method cannot predict algorithm labels. Therefore, for this experiment we trained our model using the training-set with two labels, i.e., bonafide and cloned. Thus, the aim of this experiment was to evaluate if our method is able to discriminate between any bonafide/cloned audios, no matter who the speaker is or how the cloning is performed.

From the results presented in Table VI, we can observe that our method has 88% overall accuracy. By further applying the drill-down operation on this accuracy value, we found that the accuracy of the bonafide class is 86%, whereas, for the cloned class the average accuracy is 90%; hence, the overall accuracy becomes 88%. Amongst these 87 speakers, for 72 speakers the average accuracy remains above 90%, which is fairly high considering that only 20 speakers are used for training purposes, and those 20 speakers are not considered for evaluation purposes in this experiment. Similarly, as shown in Table VII, if we analyze the 13 algorithms that were not used for training, it can be observed that for 8 algorithms accuracy is nearly 100%; whereas, for 2 algorithms accuracy is above 90%. The most problematic algorithms are A17-A19, where accuracy significantly drops. However, it can be observed from Table VII that the number of samples in all these algorithm classes have fewest samples. A17, which has lowest accuracy is just approximately 27% (in terms of sample size) of A09 which has highest accuracy of 100% and also contains the most samples. Therefore, based on this we can conclude that model optimization has positive correlation with sample size, and although external algorithm labels are not used but still our model identifies the correlation between the specific types of artifacts that any synthetic algorithm introduce, and it returns the correct output for most of the samples.

For a good algorithm, a higher accuracy value is one of the many requirements including algorithm performance in terms of precision, recall, and f1-score in class dependent scenarios. The reason for the class dependent analysis is that in case of imbalanced data, if a classifier even ignores the minor class, it will still give higher overall accuracy and other performance evaluation measures. However, such higher evaluation values are unacceptable, as usually the minor class is the class of interest that must be considered. By observing the results presented in Table VI, we can see that our method has a 67% precision rate for the bonafide class and 97% for cloned class. As the precision measure also takes into account the false positive rate, for the highly imbalance data (as in our case where 13:87 ratio exist in both classes) the precision rate drops for the bonafide class; however, the false positives in the cloned class are less, thus, they did not impose a very high negative impact on the precision rate of the cloned class. However, in case of recall we only considered the correctly classified examples in a class against all the relevant examples for that specific class; therefore, in case of the bonafide class, the recall rates are 91%, which are approximately 24% higher than those of the precision rate. Similarly, the recall rates drop by 6% for the cloned class and becomes 91%. Thus, our method performs well in terms of recall rate for the bonafide class as well as for the precision rate of the cloned class. By combining the precision and recall rates through the f1-score, we get 81% and 94% for bonafide and cloned classes, respectively. The difference in the f1-score indicates that our model needs an enhanced training-set to better classify the unseen bonafide examples. However, in real-world scenarios, as we need our proposed SASV system to only correctly

TABLE VI: Performance evaluation for cloning detection for unseen speakers and seen/unseen algorithms by training over the LA-training set and testing through LA-development, and LA-evaluation sets.

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				

TABLE VII: 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

classify the registered bonafide speakers over which the model is trained as bonafide (as shown in Table III and discussed in section IV-B), miss-classifying the unregistered users although they are bonafide is a good thing from the security perspective. The overall EER of the system is 5.22%, which is significantly lower considering the difference in the training and evaluation set sizes.

F. Replay Attack Detection

In a replay attack, the pre-recorded voice of any bonafide speaker is played back before the ASV systems. As voice samples belong to the genuine speakers, the artifacts that appear during the voice cloning are missing in the replay samples; thus, the audio fingerprints match the bonafide speakers, and impersonation occurs. However, deeper analysis of the replay samples reveals that a recorded voice also contains non-linear components that can be used as a clue for replay attack detection. In order to detect replay attacks, we first elaborate what a replay sample is comprised of:

1) Replay and Cloned Replay Patterns: A first-order voice replay attack can be modeled as a processing chain of microphone-speaker-microphone (MSM) which is equivalent to a cascade of three 2nd–order systems considering that the speakers also behave in a non-linear manner. The processing chain representing a first order replay attack is therefore expected to introduce higher order non-linearity due to the cascading of the MSM processing chain. The higher-order harmonic distortions therefore can be used to differentiate between a bonafide and spoofed audio. However, in case of cloned replays (introduced in the VSDC), the voice cloning artifacts further contain the non-linear components and have a behavior similar to that of the deeper chaining of the MSM. Moreover, by simultaneously capturing the non-linear components and cloning artifacts through an effective audio representation mechanism, cloned replays can be detected.

TABLE VIII: Performance evaluation for replay- and cloned replay attack detection using PA-evaluation set of ASVspoof 2019, and VSDC dataset.

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	–
ASVspoof 2019	Bonafide	98	98	98	1.1 / 0.0335
	Replay	98	98	98	

TABLE IX: 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 2019	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**	-

2) Replay and Cloned Replay Attack Detection: In this experiment, we evaluated the performance of the proposed method for the replay and cloned replay attack detection on VSDC and PA-evaluation set of ASVspoof 2019. From the results presented in Table VIII, we can observe that our method achieves remarkable performance on both datasets for audio replay attack detection. More specifically, we obtained an average precision of 98.3% and 99%, recall of 98.5% and 99%, and F1-score of 98.4% and 99%, EER of 1.33 and 1.1 and min t-dcf score of 0.089 and 0.0335 on VSDC and ASVspoof datasets, respectively. We can observe from the results that the proposed method performs slightly better on ASVspoof dataset over VSDC due to the fact that samples of VSDC are generated in more challenging and diverse conditions as compared to ASVspoof dataset. In VSDC, our method achieves better performance for the cloned replay attack detection as compared to the first order replay attack, confirming our findings that cloned signals become more distorted after replay as compared to normal samples; thus, they become more distinguishable as well.

G. Comparison Against Other Feature Extraction Approaches

To further elaborate the effectiveness of the proposed sm-ALTP features, we compared our features to several acoustic features for spoofing attack detection. The selected features were comprised of various combinations of MFCC, GTCC, ALTP, and spectral features. The performance of various feature combinations was then evaluated on both VSDC, and ASVspoof 2019 LA and PA training datasets. From the results presented in Table IX, it can be observed that the proposed features outperformed all the comparative features for all types of spoofing attacks in terms of EER and min t-dcf scores. Hence, the comparison results confirm again the robustness of the proposed sm-ALTP features.

H. Comparison Against State-of-the-art Methods

To further evaluate the effectiveness of the proposed method for spoofing attack detection, we compared our method to single-model approaches i.e., [21], [51], [53], [54], [55] over LA and PA the evaluation-set scenarios of ASVspoof 2019. From the methodological details and results presented in Table X, it can be observed that the comparative methods deployed large variety of acoustic features, with GMM, and deep learning models. In comparison, our model is much simpler and more accurate with min t-dcf score of 0.1321; and amongst all the different methods used by the comparative studies, only FFT-LCNN in [54] performs better than our method in LA attack detection, but our method supersedes in terms of PA attack detection. Similarly, DKU [10] outperforms our method in PA attack detection, however their LA attack detection results are unavailable. Although achieving the minimum value of t-dcf measure is the desired goal, by doing so the overall cost of the system should not increase in a way that the integration of the spoofing detection system may become difficult in real-time applications. If we consider the case of FFT-LCNN [54], the model may suffer from slow training which may span from hours to days as established in deep learning research. However, as the feature extraction time of our method is $\Theta(N)$, due to the linear time operation, our proposed feature extraction approach is very efficient.

In order to compare our method to top challenge competitors, we selected the top 10 teams amongst the 50 best performing teams of LA and PA scenarios [5] (Table XI). Next, we compared their performance to our proposed method in terms of min t-dcf score and obtained a ranking of the proposed system. Our method in both cases i.e., LA and PA scenarios, was ranked in the 9th position. However, most of the systems that were ranked higher than our method in the LA scenario were lower than our method in the PA scenario and vice versa. Furthermore, regarding the systems which were amongst the top 10 in the LA scenario but were not amongst the top 50 of the PA scenario, we assigned them the ranking score of 51 for the PA scenario; similarly, the systems which were listed amongst the top 10 of the PA scenario but were not amongst the top 50 of the LA scenario were assigned the score 51 for the LA scenario. Then, we obtained the average ranking score of the comparative systems by adding the LA and PA ranking values and dividing by 2. The average ranking score illustrates the cumulative performance of the comparative systems in both scenarios. Based on the sorted ranking score, our method was ranked 4th in terms of cumulative performance for both the LA and PA scenarios. The ranking score clearly demonstrates the effectiveness of the proposed approach with additional benefits i.e., lightweight nature.

TABLE X: 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 [51]	LFCC-GMM	11.96	0.212	13.54	0.3017
	CQCC-GMM	9.87	0.236	11.04	0.2454
ASSERT [53]	logSpec-SENet	11.75	0.216	1.29	0.036
	logspec-CQCC-SENet34-Mean-std-ResNet-SENet50-Dialated ResNet	6.70	0.155	**0.59**	**0.016**
STC [54]	LFCC-CMVN-LCNN	7.86	0.183	4.6	0.105
	FFT-LCNN	**4.53**	**0.103**	2.06	0.56
BUT-Omilia [21]	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 [55]	MFCC-CQCC-FBank-multi task learning	7.63	0.213	0.96	0.0266
DKU [10]	GD gram-ResNet	-	-	**1.08**	**0.0282**
Proposed	**sm-ALTP-Asymmetric Bagging**	**5.22**	**0.132**	**1.1**	**0.0335**

V. Conclusion

This paper presents a secure automatic speaker verification (SASV) system that can recognize registered ASV users, and also counter voice cloning, voice replays, and cloned voice replay attacks. Voice cloning detection module discriminates the original voices against the algorithmically generated synthetic/cloned audios and also provides information about the algorithm that was used for cloned audio generation. The replay detection module counters the voice replays and cloned-voice replay attacks. The proposed framework is based on novel sm-ALTP features and ensemble learning through asymmetric bagging. Our classifier ensemble approach takes a series of weak classifiers and generates a stable classifier by overcoming the class imbalance problem to recognize multiple speakers and spoofing classes. Our findings suggest that the artifacts that consequently appear due to microphone characteristics (in case of replay) or synthetic audio generation algorithms can be represented by applying the neighborhood statistics. However, the audio representation approach in this regard must also capture a speaker's specific vocal characteristics that are unique for all the speakers. The evaluation of the ASVspoof 2019 and VSDC datasets reveals that our approach effectively captures the spoofing patterns even when they are generated through unseen algorithms, thus providing a comprehensive security solution for ASV applications.

ACKNOWLEDGMENT

This work was supported by a grant from the National Science Foundation (NSF) of USA via Awards No. (1815724) and (1816019).

References

[1] K. M. Malik, H. Malik, and R. Baumann, "Towards vulnerability analysis of voice-driven interfaces and countermeasures for replay attacks," in *2019 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*. IEEE, 2019, pp. 523–528.

[2] W. Mireia C. Farrús, M. Wagner, E. Daniel E. Erro, and F. J. Hernando Pericás, "Automatic speaker recognition as a measurement of voice imitation and conversion," *The Intenational Journal of Speech. Language and the Law*, vol. 1, no. 17, pp. 119–142, 2010.

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

[4] 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)*. IEEE, 2019, pp. 6136–6140.

TABLE XI: Comparison of the proposed method to the top 10 teams of LA and PA scenarios of ASVspoof 2019.

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] M. Todisco, X. Wang, V. Vestman, Md. 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.

[6] T. Kinnunen, K. A. Lee, H. Delgado, N. Evans, M. Todisco, Md. 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.

[7] 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*, vol. 29, no. 10, pp. 4633–4644, 2017.

[8] S. M. Adnan, A. Irtaza, S. Aziz, M. ObaidUllah, A. Javed, and M. T. Mahmood, "Fall detection through acoustic local ternary patterns," *Applied Acoustics*, vol. 140, pp. 296–300, 2018.

[9] 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)*, 2019, pp. 1018–1025.

[10] 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.

[11] 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*, pp. 1038–1042, 2019.

[12] 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.

[13] M. Todisco, H. Delgado, and N. Evans, "Constant q cepstral coefficients: A spoofing countermeasure for automatic speaker verification," *Computer Speech & Language*, vol. 45, pp. 516–535, 2017.

[14] H. Delgado, M. Todisco, Md. 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)*. IEEE, 2016, pp. 179–185.

[15] L. Lin, R. Wang, D. Yan, and L. Dong, "A robust method for speech replay attack detection.," *KSII Transactions on Internet & Information Systems*, vol. 14, no. 1, pp. 168–182.

[16] 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)*. IEEE, 2016, pp. 1–5.

[17] 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*, vol. 10, no. 4, pp. 810–820, 2015.

[18] 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," 2015, pp. 2052–2056.

[19] Z. Wu, L. Phillip L. De, C. Demiroglu, A. Khodabakhsh, S. King, Z. Ling, D. Saito, B. Stewart, M. Wester 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)*, vol. 24, no. 4, pp. 768–783, 2016.

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

[21] H. Zeinali, T. Stafylakis, G. Athanasopoulou, J. Rohdin, I. Gkinis, L. Burget, J. Černockỳ, et al., "Detecting spoofing attacks using VGG and sincnet: but-omilia submission to asvspoof 2019 challenge," *arXiv preprint arXiv:1907.12908*, 2019.

[22] 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*, pp. 1068–1072, 2019.

[23] 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*, vol. 27, no. 8, pp. 1267–1279, 2019.

[24] KNRK. R. Alluri and A. K. Vuppala, "IIIT-H spoofing countermeasures for automatic speaker verification spoofing and countermeasures challenge 2019," *Proc. Interspeech 2019*, pp. 1043–1047, 2019.

[25] 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*. Springer, 2019, pp. 502–510.

[26] Z. Chen, Z. Xie, W. Zhang, and X. Xu, "Resnet and model fusion for automatic spoofing detection.," in *Proc. Interspeech*, 2017, pp. 102–106.

[27] B. Chettri, D. Stoller, V. Morfi, and S. Emmanouil and L. Bob M. A. Ramírez, B. Martínez and, "Ensemble models for spoofing detection in automatic speaker verification," *arXiv preprint arXiv:1904.04589*, 2019.

[28] Z. Ji, Z. Y. Li, P. Li, M. An, S. Gao, D. Wu, and F. Zhao, "Ensemble learning for countermeasure of audio replay spoofing attack in asvspoof2017.," in *INTERSPEECH*, 2017, pp. 87–91.

[29] Y. Zhao, A. K. Shrivastava, and K. L. Tsui, "Regularized gaussian mixture model for high-dimensional clustering," *IEEE transactions on cybernetics*, vol. 49, no. 10, pp. 3677–3688, 2018.

[30] 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*, vol. 8, no. 4, pp. 495, 2018.

[31] "Ensemble methods," https://medium.com/@aravanshad/ensemble-methods-95533944783f, Accessed: 2020-05-09.

[32] M. Todisco, H. Delgado, and N. Evans, "A new feature for automatic speaker verification anti-spoofing: Constant q cepstral coefficients.," in *Odyssey*, 2016, vol. 45, pp. 283–290.

[33] P. Nagarsheth, E. Khoury, K. Patil, and M. Garland, "Replay attack detection using DNN for channel discrimination.," in *Interspeech*, 2017, pp. 97–101.

[34] H. Delgado, M. Todisco, Md Sahidullah, N. Evans, T. Kinnunen, K. Lee, and J. Yamagishi, "Asvspoof 2017 version 2.0: meta-data analysis and baseline enhancements," in *Odyssey 2018 - The Speaker and Language Recognition Workshop*, 2018, pp. 1–9.

[35] M. Witkowski, S. Kacprzak, P. Zelasko, K. Kowalczyk, and J. Galka, "Audio replay attack detection using high-frequency features.," in *INTERSPEECH*, 2017, pp. 27–31.

[36] 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)*. IEEE, 2018, pp. 95–99.

[37] MS. 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)*. IEEE, 2018, pp. 332–336.

[38] J. Yang and R. K. Das, "Low frequency frame-wise normalization over constant-q transform for playback speech detection," *Digital Signal Processing*, vol. 89, pp. 30–39, 2019.

[39] B. Bakar and C. Hanilçi, "Replay spoofing attack detection using deep neural networks," in *2018 26th Signal Processing and Communications Applications Conference (SIU)*. IEEE, 2018, pp. 1–4.

[40] G. Lavrentyeva, S. Novoselov, E. Malykh, A. Kozlov, O. Kudashev, and V. Shchemelinin, "Audio replay attack detection with deep learning frameworks.," in *Interspeech*, 2017, pp. 82–86.

[41] L. Phillip L. De, M. Pucher, J. Yamagishi, I. Hernaez, and I. Saratxaga, "Evaluation of speaker verification security and detection of hmm-based synthetic speech," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 8, pp. 2280–2290, 2012.

[42] M. Wester, Z. Wu, and J. Yamagishi, "Human vs machine spoofing detection on wideband and narrowband data," 2015, pp. 2047–2051.

[43] 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," 2015, pp. 2062–2066.

[44] Z. Wu, J. Yamagishi, T. Kinnunen, C. Hanilçi, Md. Sahidullah, A. Sizov, N. Evans, M. Todisco, and H. Delgado, "Asvspoof: the automatic speaker verification spoofing and countermeasures challenge," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 4, pp. 588–604, 2017.

[45] A. Janicki, "Increasing anti-spoofing protection in speaker verification using linear prediction," *Multimedia Tools and Applications*, vol. 76, no. 6, pp. 9017–9032, 2017.

[46] A. Irtaza, S. M. Adnan, S. Aziz, M. ObaidUllah 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)*. IEEE, 2017, pp. 1558–1563.

[47] "Everyday DSP for programmers: Signal variance," http://sam-koblenski.blogspot.com/2015/09/everyday-dsp-for-programmers-signal.html, Accessed: 2019-11-17.

[48] "Mel frequency cepstral coefficient (MFCC) tutorial," http://practicalcryptography.com/miscellaneous/machine-learning/guide-mel-frequency-cepstral-coefficients-mfccs/, Accessed: 2019-11-20.

[49] Á. PEDROZA, D. J. JOSÉ 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*, vol. 27, no. 4, pp. 3150–3164, 2019.

[50] 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*, , no. 7, pp. 1088–1099, 2006.

[51] X. Wang, J. Yamagishi, M. Todisco, H. Delgado, A. Nautsch, N. Evans, Md. Sahidullah, V. Vestman, T. Kinnunen, K. A. Lee, et al., "The ASVspoof 2019 database," *arXiv preprint arXiv:1911.01601*, 2019.

[52] 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.

[53] C. Lai, N. Chen, J. Villalba, and N. Dehak, "ASSERT: Anti-spoofing with squeeze-excitation and residual networks," *arXiv preprint arXiv:1904.01120*, 2019.

[54] 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.

[55] 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*, 2019, pp. 1048–1052.