A Light-Weight Replay Detection Framework For Voice Controlled IoT Devices

Khalid Mahmood Malik, *Senior Member, IEEE*, Ali Javed, *Member, IEEE*, Hafiz Malik, *Senior Member, IEEE*, and Aun Irtaza

***Abstract*—The growing number of voice-controlled devices (VCDs), i.e. Google Home, Amazon Alexa, etc., has resulted in automation of home appliances, smart gadgets, and next generation vehicles, etc. However, VCDs and voice-activated services i.e. chatbots are vulnerable to audio replay attacks. Our vulnerability analysis of VCDs shows that these replays could be exploited in multi-hop scenarios to maliciously access the devices/nodes attached to the Internet of Things. To protect these VCDs and voice-activated services, there is an urgent need to develop reliable and computationally efficient solutions to detect the replay attacks. This paper models replay attacks as a nonlinear process that introduces higher-order harmonic distortions. To detect these harmonic distortions, we propose the acoustic ternary patterns-gammatone cepstral coefficient (ATP-GTCC) features that are capable of capturing distortions due to replay attacks. Error correcting output codes model is used to train a multi-class SVM classifier using the proposed ATP-GTCC feature space and tested for voice replay attack detection. Performance of the proposed framework is evaluated on ASVspoof 2019 dataset, and our own created voice spoofing detection corpus (VSDC) consisting of bona-fide, first-order replay (replayed once), and second-order replay (replayed twice) audio recordings. Experimental results signify that the proposed audio replay detection framework reliably detects both first and second-order replay attacks and can be used in resource constrained devices.**

***Index Terms*—Acoustic ternary patterns, audio replay detection, audio spoofing dataset, gammatone cepstral coefficients, voice-controlled devices.**

I. Introduction

VOICE assistant, a software component of voice-controlled devices (VCD) such as Google home, Amazon Echo, etc., is becoming an essential component of Internet of Things (IoT). This has resulted in realization of novel applications in the commercial domain i.e. voice-based control of appliances in smart homes [24], remote patient checkup in autonomous vehicles, intelligent multimedia surveillance systems [26], and voice-based retrieval of sensitive contents from information systems [27], etc. Although VCDs have revolutionized the IoT domain, they have also introduced various new threats. For instance, spoofing attacks on VCDs may help intruders retrieve sensitive data from healthcare or financial applications or acquire remote access to smart homes [1].

Audio spoofing attacks refer to the impersonation of intruders to the devices through voice replays [3], voice-synthesis [4], and voice conversion (VC) [5], [6]. Among these attacks, voice replay poses the biggest threat due to the pervasiveness of high-quality recording devices and smartphones, and the non-precondition of any advanced technical knowledge [2]. In voice replay attacks, the recorded voice of the genuine target speaker is played back to deceive the VCDs to maliciously control the devices in IoT (Fig. 1). In our previous work [1] on vulnerability analysis of VCDs such as Google home and Amazon echo, we have demonstrated that replay attacks are not just limited to first-order but can be replayed on remote devices located on another subnet if these VCDs are connected.

Fig. 1 shows a practical example to illustrate 1st and 2nd order replay attacks. There are two homes, each having devices in a separate subnet. Home-1 has a baby monitor that is remotely accessible via a mobile application, and its heating system can be managed through Alexa (VCD-1). VCD-1 in home-1 is also connected to VCD-2 located in home-2 via Alexa's drop-in feature. The garage of home-2 is controllable via VCD-3 (Google Home). Now imagine two spoofing scenarios: a) An intruder accesses the baby monitor through his phone, for example, by hacking wireless LAN using tools such as Aircrack [45], and sends a command "Alexa, turn off the heat" to turn off the heat of home-1 as shown in Fig. 1(a). b) In the next attempt shown in Fig. 1(b), the attacker, on the same network, sends the command, "Alexa, hey Google open the Garage door" to the baby monitor in home-1 in order to open the garage door of home-2, which is not part of the compromised WLAN. In this scenario the voice command will propagate through multiple Alexa enabled smart speakers that will eventually reach at Google Home. Eventually due to the inability to detect a replayed voice on multiple hops, the attacker will be able to open the garage door of home-2. It is important to mention that if two of the same devices, e.g. Amazon Alexa, are connected using the drop-in feature, then this will be considered a signal transmission rather than a second-order replay attack. However, if the two devices are different, e.g. Amazon Alexa paired with Google Home, it is considered a second-order replay attack due to the different acoustic properties between first- and second-order replay

Manuscript received November 1, 2019; revised March 25, 2020 and May 21, 2020; accepted May 21, 2020. Date of publication June 3, 2020; date of current version August 24, 2020. This work was supported by grant of National Science Foundation (NSF) of USA via under Grants 1815724 and 1816019. The guest editor coordinating the review of this manuscript and approving it for publication was Dr. Nasir Memon. *(Corresponding author: Khalid Mahmood Malik.)*

Khalid Mahmood Malik and Ali Javed are with the Department of Computer Science and Engineering, Oakland University, Rochester, MI 48309 USA (e-mail: mahmood@oakland.edu; ali.javed@uettaxila.edu.pk).

Hafiz Malik and Aun Irtaza are with the Electrical and Computer Engineering Department, University of Michigan-Dearborn, Dearborn, MI 48128 USA (e-mail: hafiz@umich.edu; airtaza@umich.edu).

Digital Object Identifier 10.1109/JSTSP.2020.2999828

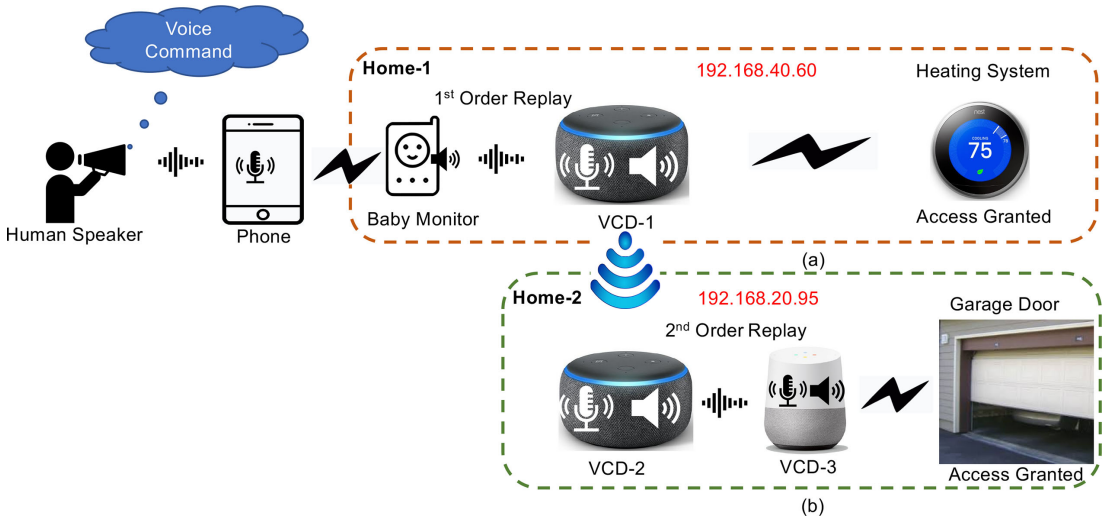

Fig. 1. Representative scenarios of replay attack in IoT using VCDs. (a) First-order replay scenario. (b) Second-order replay scenario.

attacks. We have also demonstrated that when a recorded voice of a verified speaker of Google Home (VCD-3) is played on the phone, this 3rd order chained replay (not shown in the figure) also bypasses the Google Home speaker verification feature. In other words, current VCDs are unable to differentiate between original and replay voices of either authenticated or non-verified users. This calls for development of light-weight anti-spoofing systems for VCDs deployed in IoT environment.

Existing research in IoT anti-spoofing domain is limited to IP and transport layers, however, voice based anti-spoofing in IoT is largely unexplored. In recent years there has been some research efforts to detect audio-based replay attacks [23] in conventional automatic speaker verification (ASV) systems to authenticate users in the financial sector [46]. Without focusing on lightweight solutions, these methods measure the quantifiable non-linearity that occurs due to the missing, changed, or newly added voice attributes i.e. frequencies, amplitude, and phase etc. by the microphone. Additionally, these methods consider this presentation attack detection as a binary classification problem and classify the audio as bonafide or spoof. However, none of the existing work has focused on replay attack detection where multiple microphones and smart speakers are chained together (Fig. 1). Mostly VCDs manufactured by different vendors are present in home/office setup. There exists a possibility where a certain VCD is robust against replay attacks, however, during the chaining process, data is coming from other VCDs (manufactured by different vendor) that are either compromised or prone to replay attacks due to weak or absence of replay detection mechanism. Therefore, the audio received will be considered as a genuine audio, and the countermeasure will eventually fail for all the chained devices. This paper lays the groundwork for multi-order replay attacks detection to overcome the associated threats of VCDs in IoT-based environment. Accordingly, we considered this problem as a tri-class problem where we classify the signal as bonafide, first-order, or second-order replay attack. Additionally, an audio representation mechanism should be less sensitive to noise for the replay detection task as bonafide and replay samples are recorded and replayed under different environmental conditions. In order to achieve these objectives, we propose a light-weight voice anti-spoofing system that can reliably detect the first- and second-order replay attacks through proposed features consisting of acoustic ternary patterns (ATP) and gammatone cepstral coefficients (GTCC).

The main contributions of this paper are as under:

1) We highlight that multi-order replay spoofing attacks are possible and VCDs are unable to detect them.
2) We proposed a noise resistant ATP features descriptor and merge with GTCC for audio signal representation. Additionally, we present the groundwork for multi-order replay attack prevention through the proposed ATP-GTCC light-weight features in VCDs.
3) We developed an open source voice spoofing detection corpus (VSDC) [32] with bonafide, first-order, and second-order replay samples to address the multi-order replay attack detection.

The rest of paper is organized as follows. Section II covers state-of-the-art in voice replay spoofing detection. Section III provides an analysis of multi-order voice replay attacks. Section IV explains the proposed framework. Section V provides the details of datasets and experiments designed for performance evaluation. Finally, section VI concludes our paper.

II. Related Work

This section provides a critical analysis of the existing state-of-the-art anti-spoofing methods.

A. Gaussian Mixture Model-Based Approaches

Anti-spoofing systems developed for non-constrained devices (e.g. banking server responsible for audio-based biometric authentication) have explored Gaussian mixture model (GMM) and its variants using magnitude-oriented [2], [3], [6] and phase-oriented [4], [11], [21] features.

1) Magnitude-Oriented Features: In [2] constant Q-transform cepstral coefficients (CQCC) were used to train a two-class GMM to classify the audio samples as bonafide

or spoof. In [5], a features-set comprised of Mel-Frequency Cepstral Coefficients (MFCC), Perceptual linear predictive and CQCC was used to train the ensemble classifiers consisting of different variants of GMM. Intuitively, the extensive use of multiple features makes this solution less practical for VCDs. Few works [3], [6] have highlighted the significance of high-frequency bands selectivity to address the replay spoof detection. In this regard, transmission line cochlea (TLC)-amplitude modulation and frequency modulation features were used in [3] to train the GMM. However, amplitude modulation (AM) feature representation has significant computational cost, as it takes more than twice the amplitude frequency to modulate the signal. Similarly, in [6] inverted-MFCC (IMFCC), linear predictive cepstral coefficients (LPCC), and LPCCres features were used to capture the high-frequency features along-with the standard baseline features of CQCC, MFCC, and Cepstrum. GMM was used to classify the original and spoof samples. These approaches [3], [6] perform better than the baseline CQCC-GMM model [2]. However, these methods are unsuitable to locally deploy on resource constrained VCDs due to higher features computational complexity.

Apart from high-frequency bands analysis, some research studies [7], [9], [10] have highlighted that recording and playback device characteristics, reverberation and channel information should be examined for replay attacks detection. In [7], linear prediction residual signals were analyzed to examine the recording and playback device characteristics. More specifically, residual-MFCC (RMFCC) and residual inverse-MFCC (RIMFCC) features were used to train the GMM for replay spoofing detection. In [9], authors examined the channel information and reverberation from the non-voice segments of the audio. MFCC, CQCC, and Mel-Filterbank-Slope (MFS) features were used to train the GMM for replay attack detection. Similarly, in [10] low frequency frame-wise normalization scheme was proposed to capture the artifacts from the playback speech and later used to detect the replay attacks.

2) Phase-Oriented Features: Works such as [4], [11], [21] have used phase-oriented spectral features for replay spoofing detection. In [4], a 36-D feature vector consisting of MFCC, Mel-Frequency Principal Coefficients (MFPC), cos-phase principal coefficients (Cos-phase PC) and Mel-wavelet packet transform (MWPT) was used to train the SVM to classify between the genuine and spoof samples. In [11], teager energy operator (TEO) phase-based features were used to capture the traits of bonafide and spoof samples. It was demonstrated that the TEO alone was unable to provide better classification performance, however accuracy was improved when TEO features were used in combination with magnitude-oriented features. Similarly, magnitude-based features were used in combination of phase-based features to train the GMM for replay spoof detection [21].

B. Deep Learning-Based Approaches

Recently, deep learning approaches have also been investigated for voice anti-spoofing systems. In [12], data augmentation was performed to illustrate that it improves the performance of ASVspoof baseline model [2]. Original spectrogram was employed to deep residual network for features extraction. This work has following limitations; firstly, manual data augmentation is required, which is a laborious activity, and secondly, using only the STFT based spectrogram makes it unable to achieve better results. In [13], MFCC and CQCC were used to train the GMM, ResNet and DNNs for replay attack detection. After analyzing different combinations, it was concluded that the combination of CQCC-GMM, MFCC-ResNet, and CQCC-ResNet achieves the lowest equal error rate (EER). Although this approach is effective however, the fusion of two deep learning models along-with GMM makes it less feasible for VCDs. In [14], a high-pass filter was employed followed by computing the discrete cosine transform (DCT) to obtain the high-frequency cepstral coefficients (HFCC). HFCC was used in combination with CQCC to generate the embeddings through a deep neural network (DNN). These embeddings represent the extracted features that were used to train the SVM to classify the audio samples as bonafide or spoof. Similarly, DNN was also trained using the long-term average spectrum (LTAS) and MFCC features for replay spoofing detection [22].

Instead of extracting certain features (i.e. MFCC, CQCC, etc.) which are then fed to deep learning models for classification, few works [18], [19] have also used machine-learned features. In [18], authors used both the extracted and machine-learned features to train the GMM-Universal Background Model (UBM) for replay attacks detection. More specifically, 11 cepstral features-sets were used to train an auto-encoder to obtain a dense projection of these features. Finally, both the extracted features-set and their learned representations were fed to GMM-UBM for classification. This method [18] achieves better results at the expense of increased features computation cost. The problem of machine-learned features representation through the autoencoder is the ability to learn maximum information rather than the relevant information. Hence, the autoencoder can cause information loss of relevant content.

Few methods have also adopted light-weight deep learning frameworks for replay spoofing detection. In [15], light-weight CNN [16] based on maximum feature-map (MFM) activation was used to detect the replay attacks. MFM was able to reduce the dimension by selecting the most relevant features to perform the classification. This method [15] was extended in [17] to investigate the efficiency of angular margin-based softmax activation function to train the light CNN for cloning and replay spoofing detection. LCNN architecture was also employed in [23] for replay attacks detection.

III. Analysis of Multi-Order Voice Replays

The microphone, an integral component in the processing chain of the replay attack, is a complex electromechanical device. The interactions among its mechanical, electromechanical, and electrical elements transform sound energy into electrical signals. Any nonlinearity in these elements results in a distorted output. The structures of commonly used microphones, e.g., carbon, electric, etc. are known to behave in a nonlinear manner [47]. In general, stiffness of the mechanical suspension

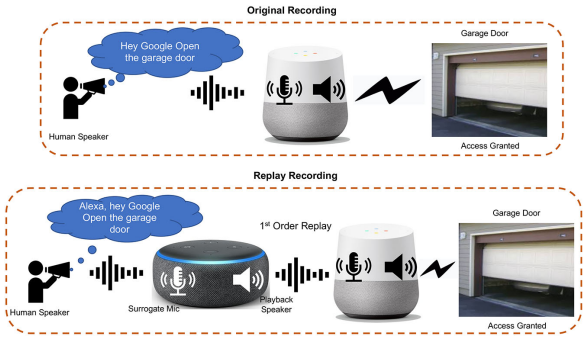

Fig. 2. First-order replay attack scenario.

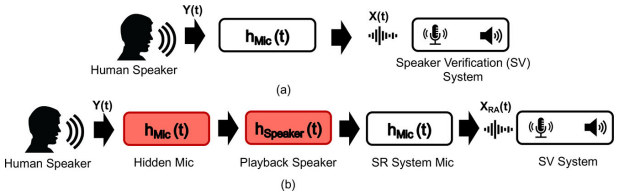

Fig. 3. Process chain comparison between bonafide and replay attack.

and acoustical damping are the dominant causes of nonlinear distortion in most microphones. Microphone distortions can be classified into harmonic, intermodulation, and difference-frequency distortions [48]. Harmonic distortion is the effect of nonlinearity on a pure tone excitation, causing harmonic components in the output. Intermodulation distortion is the effect of nonlinearity produced at the output from an excitation that is sum of a stronger high frequency and a weaker low frequency component. Difference-frequency distortion is the effect of a nonlinearity produced at the output from an excitation of sinusoids of the same amplitude. For good quality microphones, measuring microphone induced distortions at normal sound levels is a difficult task. Intermodulation distortion measurements are considerably more complex than measurements of harmonic distortion. This is one of the factors behind using harmonic distortion as a benchmark for microphone quality characterization. Specifically, second and third harmonics are typically used to qualify microphone quality. The second harmonic is the most dominant distortion component among all harmonic distortion components [49]. It is therefore reasonable to model a microphone as a second-order nonlinear device [50]. The playback speaker also behaves in a nonlinear manner, which can also be modeled as a 2^{nd}-order device.

The 1^{st}-order voice replay attack shown in Fig. 2 (bottom) can be modeled as processing chain of microphone-speaker-microphone (MSM) which is equivalent to a cascade of three 2^{nd}-order nonlinear systems. This is because the microphone as well as speaker are nonlinear devices and it is typically modeled using 2^{nd}-order nonlinear system. The microphone response can be modelled through the following nonlinear function:

$$x[n] = \alpha x[n] + \beta (y[n])^2 \tag{1}$$

Where α is the linear gain and β is the nonlinear coefficient of the microphone. Shown in Fig. 3 is a comparison between a bonafide (3a) and a spoofed recording (3b).

Figs. 2 & 3 show a 1^{st}-order replay attack is equivalent to MSM, which is modeled as 6^{th}-order nonlinear process. The processing chain representing a 1^{st}-order replay attack is therefore expected to introduce higher-order nonlinearity (beyond 7^{th}-order) due to cascade of MSM processing chain. The higher-order nonlinearity introduces higher-order correlations in the frequency domain. In other words, nonlinear systems introduce higher-order correlations which contribute distortions in the form of new frequencies. Thus, the higher-order voice replay attacks are expected to introduce stronger higher-order distortions in the resulting signals. It is important to highlight that microphone/speaker nonlinear modeling assumed here is not optimal. Therefore, the 7^{th}-order polynomial model is not optimal. On the order hand, the direct speech signal (bonafide audio) lacks microphone-speaker processing chain therefore is expected to exhibit relatively low- or higher-order harmonic distortions. The higher-order harmonic distortions therefore can be used to differentiate between a direct and spoofed audio. Spectral features such as higher-order spectral analysis (HOSA), MFCC, GTCC, etc., can also be used to capture the traces of replay attack induced distortions.

In our preliminary work [1], [51], we proposed HOSA-based framework to capture the harmonic distortions due to replay attacks. We have also demonstrated in [1] that replay attacks introduce higher-order distortions, and second-order spectral analysis, e.g., bicoherence can be used to capture traces of harmonic distortions due to replay attacks. Higher computational cost of HOSA makes these features unsuitable for VCDs. To get around this issue, computationally efficient features consisting of 13-dimensional GTCC and 20-dimensional ATP are used to capture the distortions in replayed audios. As an example, Fig. 4 shows plots of frame-level GTCC features for direct (left), 1^{st}-order (center), and, 2^{nd} -order (right) audio recordings. These plots demonstrate that replay attacks introduce distortions (highlighted with doted yellow ellipses) in the resulting replayed recordings; and selected GTCC features are able to capture these distortions. It can also be observed from Fig. 4 that harmonic distortions are more pronounced for the 2^{nd}-order replay audio recording than the 1^{st}-order replay recording. This confirms our claim that higher- order voice replay attacks are expected to introduce stronger higher-order distortions in the resulting signals. The stronger distortions for 2^{nd}-order replay attacks are expected to contribute to better detection performance than 1^{st}-order replay attacks.

IV. Proposed Framework

This section provides the description of the proposed replay anti-spoofing framework. The input audio signal is processed to extract the 20-D ATP and 13-D GTCC features that are then fused to create a 33-D ATP-GTCC features-set. For classification, we applied the error correcting output codes (ECOC) model to design a multi-class support vector machine (SVM) classifier. We used the proposed features to train the SVM and classify the audio sample as bonafide, first-order replay or second-order replay. The process flow of the proposed framework is provided in Fig. 5 and the details are as follows.

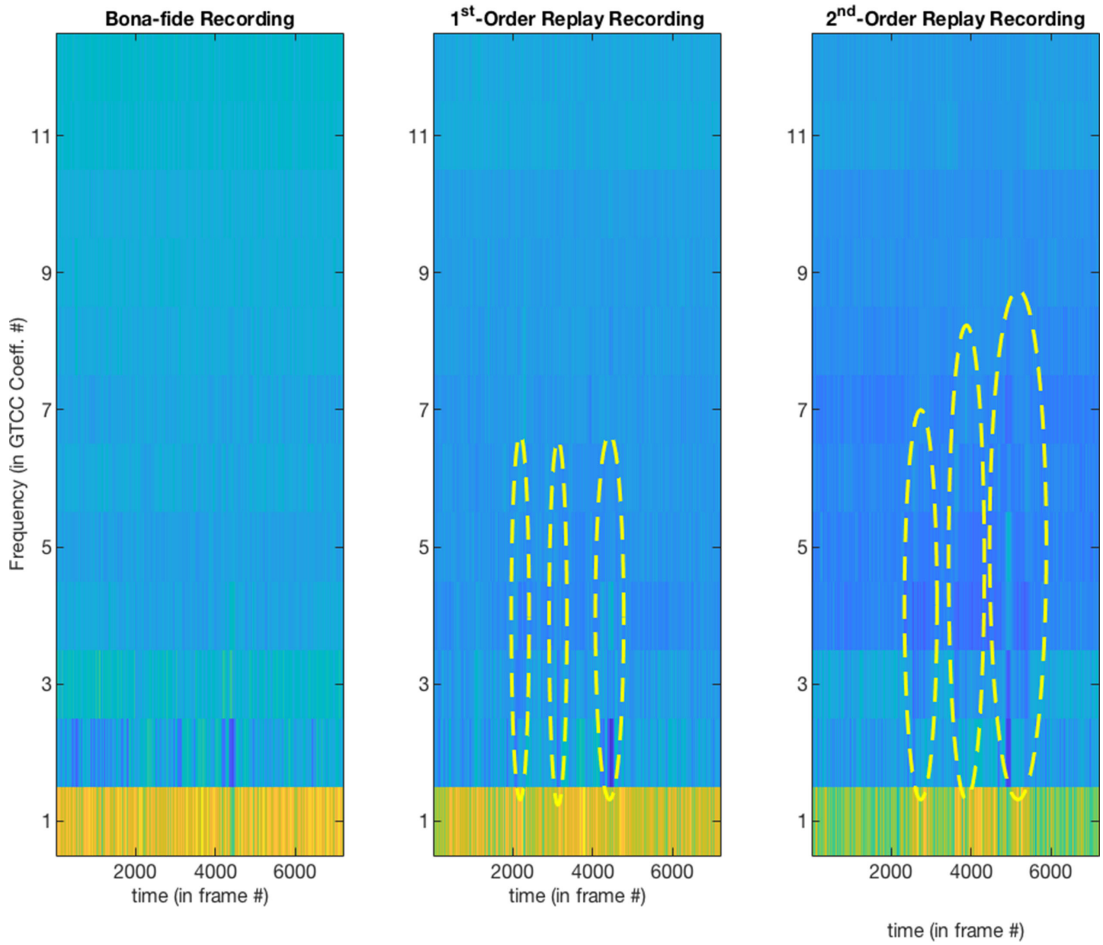

Fig. 4. GTCC features for bonafide, 1st-order replay, and 2nd-order replay.

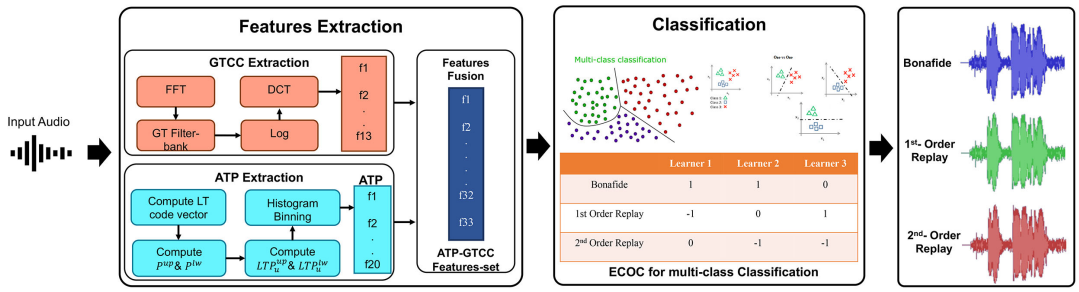

Fig. 5. Architecture of the proposed framework.

A. *Features Extraction*

Effective acoustic features extraction is required to analyze the complex nature of audio signals. The details of proposed ATP and GTCC features extraction is explained below.

1) Acoustic Ternary Patterns (ATP): Inspired by the application of 2D-local ternary patterns in image processing [28], [29], we applied this concept for 1-D audio signals to effectively represent the acoustic signal and detect the replay attacks [52], [53].

For a given input audio signal $Y[n]$ having N samples, we partition the input audio signal into $F^{(i)}$ non-overlapping frames with length l, where $i = \{1, 2, \ldots, m\}$ represents the total number of frames in $Y[n]$ and $l = 9$ in our case. As the ATP features are inspired by image processing research [29] that considers the closest 8 neighbors surrounding a given pixel in a 3×3 window. We employ a similar concept by considering 8 neighbors of a central sample of the audio signal. Thus, one central sample and 8 neighbors become a frame. In each frame $F^{(i)}$, c represents the central sample in a frame with z^j neighbors, where j represents the neighbor index against the sample c (Fig. 6(a)). To compute the local ATP response, we calculate the difference between the magnitude of central sample c and neighboring audio samples z^j by applying the parameter t_h

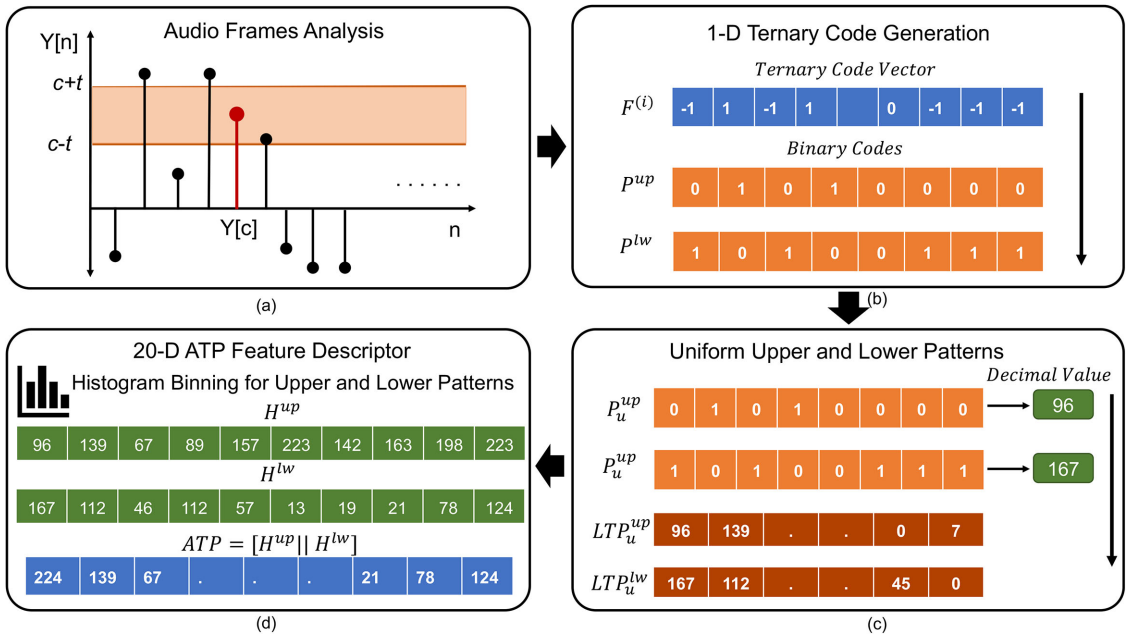

Fig. 6. ATP features computation method.

around the sample c. We initialize the t_h as zero and optimize it by performing a linear search to find the convergence point between 0 and 1. In our case, $t_h = 0.00015$ provide us the most precise results. We quantize the sample values in $F^{(i)}$ to zero that lie in the range of width $\pm t_h$ around the c, whereas values above and below $c \pm t_h$ are quantized to 1 and -1 respectively (Fig. 6(b)). Thus, we obtain a three-valued function as:

$$P\left(z^j, c, t_h\right) = \begin{cases} -1, & z^j - (c - t_h) \leq 0 \\ 0, & (c + t_h) < z^j < (c - t_h) \\ +1, & z^j - (c + t_h) \geq 0 \end{cases} \quad (2)$$

Where $P(z^i, c, t_h)$ denotes the acoustic signal using a three-valued ternary pattern locally. Next, we split the patterns into two classes i.e. upper pattern P^{up} (.) and lower pattern P^{lw} (.). All values quantized to $+1$ are retained in P^{up} (.), while replacing all other values with zeros as follows:

$$P^{up}\left(z^j, c, t_h\right) = \begin{cases} 1, & \text{if } P\left(z^j, c, t_h\right) = +1 \\ 0, & \text{Otherwise} \end{cases} \quad (3)$$

Similarly, we retained all values quantized to -1 in P^{lw} (.) and replaced all the other values with zeros as follows:

$$P^{lw}\left(z^j, c, t_h\right) = \begin{cases} 1, & \text{if } P\left(z^j, c, t_h\right) = -1 \\ 0, & \text{Otherwise} \end{cases} \quad (4)$$

Similar to the concept of uniform patterns in image processing [29], we employed this concept for audio signals as these patterns can effectively capture the maximum traits of the audio signal. Uniform patterns possess significant details of the signal as compared to non-uniform patterns that include redundant and less important content of the input signal. It is also important to mention that there exist more uniform patterns in comparison to non-uniform patterns. We computed the upper uniform $LTP_u^{up}(.)$ and lower uniform $LTP_u^{lw}(.)$ patterns from the P^{up} (.) and P^{lw} (.) as shown in Fig. 6(c), and represented these patterns in decimal values as follows:

$$LTP_u^{up}\left(z^j, c, t_h\right) = \sum_{j=0}^{j=7} P_u^{up}\left(z^j, c, t_h\right) \times 2^j \quad (5)$$

$$LTP_u^{lw}\left(z^j, c, t_h\right) = \sum_{j=0}^{j=7} P_u^{lw}\left(z^j, c, t_h\right) \times 2^j \quad (6)$$

In the next step, we compute the histogram of LTP_u^{up} and LTP_u^{lw}, where we assigned one histogram bin for each uniform pattern and include all non-uniform patterns in a bin while reducing minimum information (Fig. 6(d)). Histograms are calculated as:

$$H^{up}\left(LTP^{up}, b\right) = \sum_{k=1}^{K} \delta\left(LTP_k^{up}, b\right) \quad (7)$$

$$H^{lw}\left(LTP^{lw}, b\right) = \sum_{k=1}^{K} \delta\left(LTP_k^{lw}, b\right) \quad (8)$$

Here b represents the histogram bins corresponding to the uniform ATP codes, $\delta(.)$ is the Kronecker delta function. After conducting extensive experiments during the ATP code generation process, we observed that the first 10 uniform patterns both from the upper and lower patterns were sufficient to capture all distortions available in the samples. Therefore, we used the 10-D ATP code each for the upper and lower uniform patterns and combined the histograms to create a 20-D ATP feature descriptor as:

$$ATP = [H^{up} || H^{lw}] \quad (9)$$

Where $||$ denotes the concatenation operator, and [] is used to represent that two histograms will concatenate to provide the ATP feature vector.

2) Gammatone Cepstral Coefficients (GTCC): Gammatone Cepstral Coefficient (GTCC) features [54] are gaining importance due to the improved characteristics of filter responses

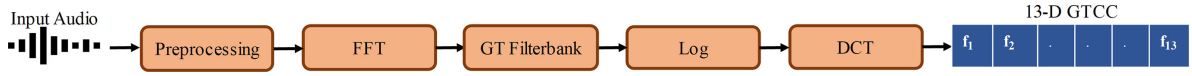

Fig. 7. GTCC features computation method.

that better resembles the human auditory system. To capture the distortions in frequency scale, we can use any spectral features i.e. MFCC, GTCC, etc. The computational cost of GTCC is equivalent to MFCC, however GTCC are more robust to noise [25] and provides superior classification performance over MFCC. Therefore, we selected the GTCC features to better capture the distortions with extremely efficient ATP features for representation of the audio signal. Additionally, they are never employed for audio replay attacks detection, therefore, we employed GTCC to evaluate their effectiveness in terms of reliable audio replay attacks detection.

GTCC features are a biologically inspired modification of the MFCC that uses gammatone filters with equivalent rectangular bandwidth (ERB) bands. As reported in [30], the magnitude response of 4$^{\text{th}}$-order gammatone filter (GT) is similar to a reox function [44] that can be used to represent the human auditory response. GT filter provides more frequency components in the low-frequency range with narrow bandwidth and less frequency components in the high-frequency range with wider bandwidth that reveals the spectral information effectively. Moreover, designing an efficient feature descriptor is better accomplished through GTCC, as the n^{th}-order GT filter can be represented through a set of n 1$^{\text{st}}$-order GT filters arranged in cascade form.

To extract the GTCC features, we applied the fast Fourier transform (FFT) on each audio frame to analyze the spectrum. A gammatone filter bank comprising of various GT filters is employed to the FFT of the audio signal and energy of each sub-band E_n is computed. In the next step, logarithm (Log) of each E_n is computed followed by applying the discrete cosine transform (DCT) on this signal to obtain the GTCC features. The GTCC features are computed as follows:

$$GTCC_k = \sqrt{\frac{2}{Z}\sum_{z=1}^{Z} log\left(E_n\right)\cos\left[\frac{\pi z}{Z}\left(k-\frac{1}{2}\right)\right]}\ 1 \le k \le K. \tag{10}$$

Where E_n, Z, and K represents the signal energy for n^{th} spectral band, number of gammatone filters and number of GTCC respectively. Log and DCT are computed to model the subjective perception of loudness and reduce the auto-correlation in the log-compressed filter outputs for better energy compression. The process of GTCC computation that returns the 13-dimensional GTCC coefficients is shown in Fig. 7. We employed the window length of 30 ms and overlapping factor of 20 ms to extract the GTCC features.

For cepstral features, 0$^{\text{th}}$-order coefficient contains the average power of the input audio signal, whereas, the 1$^{\text{st}}$-order coefficient denotes the distribution of spectrum energy between low- and high-frequencies. Although higher-order coefficients represent increasing levels of spectral details based on the sampling rate and estimation method, however, it is important to mention that 13 to 20 cepstral coefficients are usually considered optimal for audio signal analysis. Since we aim to propose a light-weight anti-spoofing framework, therefore we extracted 13 GTCC features and fused with ATP features for audio signal representation. The implementation of ATP and GTCC features computation are available at [8].

B. Classification

We employ the error correcting output codes (ECOC) model [43] to design a multi-class classifier through combining three binary classifiers to detect the bonafide, first-order, and second-order replay samples. ECOC comprises of encoding and decoding stages. In the encoding stage, ECOC creates a codeword for each class based on different binary problems. Whereas in the decoding stage, ECOC classifies the given test input based on the value of the output code.

There are three classes in our case, so during the encoding stage, three different groups of classes are created and three dichotomizers (binary learners) are trained. Next, we obtain the code-word of length three for each class. Each bit of the code-word indicates the response of the given dichotomizer. More specifically, we adopt the ternary ECOC model that use three codes $\{-1,0,1\}$ in the encoding process. The ternary ECOC model ignores one class and compares the other two during one vs one scheme as shown in Fig. 5. So, our ternary coding matrix CM is $\{-1,0,1\}$. Here 0 is used to ignore one class when the other two classes are used in the particular binary classifier. At the decoding stage, we obtain a code against each audio for three binary classifiers that is then compared against the base-codewords of each class. We employ the hamming distance to measure the distance between the codewords of given test input and the three classes. The given test sample is assigned to the class having the closest code-word.

Our ECOC model uses three SVM learners to classify the bonafide, first-order replay, and second-order replay samples. The training samples consist of V number of features for bonafide, first-order-, and second-order-replay samples created as: $\{x^i, c^i\}$, $i = 1,\ldots,V$, where $c^i\{-1,0,1\}$ represents the bonafide, first-order- and second-order-replay classes. Each SVM classifier is trained using the proposed features for two classes at a time. We solve the following binary classification problem to train the two classes q and r.

$$\min_{w^{q,r},b^{q,r},^{q,r}} \frac{\frac{1}{2}(w^{q,r})^T(w^{q,r}) + P\sum_t \xi_t^{q,r}(w^{q,r})^T}{\left\{\begin{array}{l}(w^{q,r})^T\emptyset(U_t) + b^{q,r} \ge 1-\xi_t^{q,r},\ c_t = q\\ (w^{q,r})^T\emptyset(U_t) + b^{q,r} \le \xi_t^{q,r}-1,\ c_t = r\\ \xi_t^{q,r} \ge 0\end{array}\right\}} \tag{11}$$

TABLE I
DETAILS OF VOICE SPOOFING DETECTION CORPUS

Audio Samples	Sample Rate	Environment	Microphones for Bona fide Recordings		Playback Devices	Human Speakers	1PR-2PR Replay Configuration		
			Make	Model			Source	Target	Connection Method
Bona-fide: 4000	96kbps	Recording-Chamber	Audio-Technica	ST95MKII	Polk R150	Male: 10	Echo dot 2	Echo dot 2	Amazon drop-in
		Kitchen Table	Shure	SM58	Bose 141		Echo dot 2	Echo dot 3	Amazon drop-in
1st-order Replay: 4000		Living Room	Behinger	ECM 8000	Presonus Eric-E5	Female:9	Echo dot 2	Echo dot 2	Amazon drop-in
		Office Desk	Electro-Voice	635 A/B	Bose Soundlink-415859		Echo dot 3	Echo dot 3	Amazon drop-in
2nd-order Replay: 4000		Dining Room	Blue Yeti	Internal microphones for mobile devices	SBT 6050R		Echo dot 3	EchoPlus Gen-2	Amazon drop-in
		Vehicle	MacBook Pro		MacBook Pro Internal Speakers 2018		Echo dot 3	Asus-Tablet	Google Meet
Total: 12000		Ground	Acer		Acer Nitro Spin 5		LG G6	Laptop	Google Meet
			Samsung Galaxy S7		Acer Aspire E5-574G				
			IPhone 5S						
			IPhone 7						
			IPhone 8						
			IPhone rx						

Where U_t represents the training data which is mapped to a higher dimensional space by the function $\emptyset$, and P is the penalty parameter. We want to maximize the margins between the samples of bonafide and spoof classes through minimizing the $\frac{1}{2}(w^{q,r})^T(w^{q,r})$. The penalty term $\sum_t \xi_t^{q,r}(w^{q,r})^T$ is employed to reduce the number of training errors as our data is not linearly separable. We aim to search for a balance between the regularization term $\frac{1}{2}(w^{q,r})^T(w^{q,r})$ and the training errors.

We apply the voting scheme through analyzing the $(w^{q,r})^T\emptyset(U_t) + b^{q,r}$. More explicitly, if sign $((w^{q,r})^T\emptyset(U_t) + b^{q,r})$ indicates that U_t belongs to the q^{th} class, then we increment the voting counter for q^{th} class and vice versa. Finally, we adopt the majority voting scheme to predict the class of U_t where U_t belongs to the class getting the majority of votes.

V. EXPERIMENTAL RESULTS AND DISCUSSION

A. Dataset

Performance of our framework is evaluated on the proposed VSDC [32] and ASVspoof 2019 [23] corpus. The existing spoofing datasets like ASVspoof [23] and ReMASC [20] only contains first-order replay samples against the bonafide audio samples, therefore, the VSDC is specifically designed to evaluate the performance of the proposed framework in multi-order replay attacks scenario (first- and second-order). We ensured that the proposed VSDC is diverse in terms of microphones, playback devices, environment, speaker genre, and number of speakers. For audio recording, we used multiple professional and cell-phones microphones. Additionally, we captured and replayed the samples in different environments to ensure that our recorded and replayed samples encompass noise and interferences. To generate the first- and second-order replay samples we used variety of different playback devices to counter the effect of any playback device characteristics. Ten male and nine female speakers were involved to record the original audio samples in different environments. The details of our dataset are provided in Table I.

The ASVspoof 2019 dataset for replay spoofing consists of training, development and evaluation sets. The training set contains 54 000 samples, the development set contains 33 534 samples and the evaluation set contains 1 53 522 bonafide and replay samples. Unlike our dataset, the ASVspoof dataset includes samples of different lengths, even some of the bonafide and corresponding spoof samples vary in duration.

TABLE II
RESULTS OF THE PROPOSED METHOD ON DIFFERENT KERNELS

Dataset	SVM Kernel	EER %	Precision %	Recall %	F1-Score %	Accuracy %
VSDC	Linear	18	82	82	82	82.2
	Quadratic	1.16	98.3	98.3	98.3	98.3
	Cubic	0.66	99.3	99.3	99.3	99.3
	RBF	**0.6**	**99.3**	**99.3**	**99.3**	**99.4**
ASVspoof	Linear	2	93.47	93	93.23	93.1
	Quadratic	1.5	98.5	98.5	98.5	98.8
	Cubic	1.1	99.25	99.25	99.25	99.2
	RBF	**1**	**99**	**99**	**99**	**99.1**

B. Performance Evaluation of Proposed Framework

Performance of the proposed framework is evaluated using the EER, precision, recall, f1-score, and accuracy. For experimentation, we used the proposed features to train the SVM using ECOC model to classify among the bonafide, first-order-, and second-order-replay samples. For VSDC, we used 70% of the samples for training and rest 30% for testing purposes, whereas, for ASVspoof dataset we used the training set to train the model and evaluation set for testing. We employed the ten-fold cross validation scheme to train the model with different SVM kernels i.e. linear, quadratic, cubic, and radial basis function (RBF). We selected the penalty parameter or box constraint to 1 and kernel scale or gamma to 1.4 as we obtained best results on these parameter settings.

1) Results of the Proposed ATP-GTCC Features: We performed the experiments through proposed features-set and different SVM kernels on both datasets, and results are reported in Table II. We achieved an EER of 18%, 1.16%, 0.66% and 0.6% using the linear, quadratic, cubic and RBF kernels respectively on VSDC. Whereas for the ASVspoof 2019 dataset, we achieved an EER of 2%, 1.5%, 1.1% and 1% using the linear, quadratic, cubic and RBF kernels respectively. From the results

TABLE III
DETAILS OF FEATURE VECTORS

Feature Vectors	Features
Spectral 40-D	GTCC [log energy, 1-13], MFCC [log energy, 1-13], Spectral (Kurtosis, Skewness, Slope, Centroid, Flatness, Entropy, Decrease, Rolloff point, Flux, Crest, Spread), Energy
ATP-Spectral 31-D	ATP [20-D], Spectral (Kurtosis, Skewness, Slope, Centroid, Flatness, Entropy, Decrease, Rolloff point, Flux, Crest, Spread)
ATP-MFCC 33-D	ATP [20-D], MFCC [13-D]
ATP-GTCC 33-D	**ATP [20-D], GTCC [13-D]**

presented in Table II we can observe that the SVM using the RBF kernel provides best results as compared to other kernels. More specifically, we obtained an EER of 0.6%, precision, recall, and f1-score of 99.3%, and accuracy of 99.4% on VSDC. And, obtained an EER of 1%, precision, recall, and f1-score of 99%, and accuracy of 99.1% on the ASVspoof dataset.

We observed from the results (Table II) that the polynomial and RBF SVM kernels provide superior classification performance due to the non-linearities present in first- and second-order replay samples. Whereas, SVM tuned with the linear kernel achieves the highest EER. We also observed that the third order polynomial (cubic) kernel achieves a lower EER as compared to second order polynomial (quadratic) for multi-order replay spoofing. This observation shows the effectiveness of higher-order polynomial function to better distinguish the distortions available in higher-order replays. More specifically, SVM using the RBF kernel achieves the lowest EER of 0.6% and 1% on our proposed ATP-GTCC features-set on VSDC and ASVspoof datasets. Our findings on different SVM kernels conclude that RBF outperforms all other kernels for replay spoofing classification. From these observations, we argue that SVM using the RBF kernel can better discriminate the characteristics present in the bonafide and spoof samples.

2) Performance Comparison of Proposed Features With Different Features-Combinations: To justify the effectiveness of the proposed features in detecting the distortions present in the spoof samples, we generated different ATP and spectral feature-combinations (Table III) and evaluated their performance on both datasets through SVM using the RBF kernel. The results obtained are reported in Table IV.

From Table IV we can clearly observe that our proposed ATP-GTCC features-set outperforms other features by achieving the lowest EER. More specifically, we achieved an EER of 2.5% and 1.5% on ATP-spectral, 2.33% and 6.75% on MFCC-GTCC-spectral, 1.33% and 0.75% on ATP-MFCC and 0.6% and 1% on our proposed ATP-GTCC features for VSDC and ASVspoof datasets respectively. Similarly, we achieved the highest precision, recall, f1-score, and accuracy for our proposed ATP-GTCC features-set as compared to other features as shown in Table IV.

From the experiments, we observed that the fusion of ATP with spectral features improves the classification performance and achieves a lower EER. Based on the results, we conclude that the proposed ATP-GTCC features-set can reliably be used to classify among the bonafide, first-order replay and second-order replay samples. Additionally, our proposed features are computationally efficient, making them a reliable features descriptor for VCDs. Our proposed ATP-GTCC features-set lays the groundwork for detecting replay attacks in resource constraint VCDs connected in IoT environment.

TABLE IV
COMPARATIVE ANALYSIS OF PROPOSED AND OTHER SPECTRAL FEATURES

Dataset	Features	EER %	Precision %	Recall %	F1-Score %	Accuracy %
VSDC	MFCC-GTCC-Spectral	2.33	97.7	97.7	97.7	97.6
	ATP-Spectral	2.5	97.7	97.3	97.49	97.4
	ATP-MFCC	1.33	98.7	98.7	98.7	98.8
	ATP-GTCC	**0.6**	**99.3**	**99.3**	**99.3**	**99.4**
ASVspoof	MFCC-GTCC-Spectral	6.75	93.47	93	93.23	93.1
	ATP-Spectral	1.5%	98.5	98.5	98.5	98.8
	ATP-MFCC	0.75	99.25	99.25	99.25	99.2
	ATP-GTCC	**1%**	**99%**	**99%**	**99%**	**99.1%**

C. Performance Comparison of First- and Second-Order Replay Attacks

We hypothesize that the second-order replays contain more distortions compared to the first-order replays. To evaluate our claim, we measured the performance of the proposed method for first- and second-order replay attacks detection separately on our VSDC. For the purpose of this experiment, we partitioned our dataset into two collections; one collection containing 4000 bonafide and 4000 first-order replay samples, and the other containing 4000 bonafide and 4000 second-order replay samples.

First, we evaluated the performance of the proposed method on the bonafide and first-order replay samples of our dataset. We used 70% of the samples from each of the two classes, extracted the ATP-GTCC features, and trained the SVM using the RBF kernel. We used the remaining 30% samples of both classes for testing to classify the bonafide and first-order replay samples, and obtained an EER of 0.7%, accuracy of 99.3%, and precision, recall, and F1-score of 99.2%. Next, we evaluated the performance of our method on the bonafide and second-order replay samples. We used 70% of the samples from each of the two classes (bonafide and second-order replays), extracted the ATP-GTCC features and trained the SVM using RBF kernel. Again, we used the remaining 30% samples of both classes for testing to classify the bonafide and second-order replay samples, and achieved an EER of 0.5%, accuracy of 99.5%, and precision, recall, and F1-score of 99.4%. The results of this experiment prove our hypothesis that the first-order replay attacks are more challenging to detect than the second-order replay attacks.

D. Performance Comparison using Different Classifiers

This experiment is designed to compare the performance of the proposed method against other machine learning classifiers

TABLE V
DETECTION RESULTS OF DECISION TREES

Dataset	Features Set	Tree-Depth	EER%	Precision%	Recall%	F1-Score%	Accuracy%
VSDC	GTCC-MFCC-Spectral	Fine	11	89	89	89	89
		Medium	21.83	78.3	78	78.13	77.8
		Coarse	40.83	59.5	57.7	58.55	57.8
	ATP-Spectral	Fine	18.66	81.3	81.3	81.3	81.3
		Medium	26.66	74	72	73.97	72
		Coarse	37.83	62.6	60.3	61.46	60.5
	ATP-MFCC	Fine	16.83	83.3	83	83.14	83.3
		Medium	29	72	71	71	71.2
		Coarse	40.16	60	59	59.5	60
	ATP-GTCC	Fine	16.33	84.1	83	83.56	82.7
		Medium	29.5	71	69.3	70.15	69.1
		Coarse	44.33	55.9	54	54.91	53.7
ASVspoof	GTCC-MFCC-Spectral	Fine	15.25	84.92	84.5	84.71	84.2
		Medium	19.25	81.22	80	80.61	80.1
		Coarse	24.5	77.13	72.5	74.74	72.6
	ATP-Spectral	Fine	18.5	81.5	81.5	81.5	81.7
		Medium	23.5	76.5	76.5	76.5	76.3
		Coarse	29.75	70.98	68.5	69.72	68.7
	ATP-MFCC	Fine	17.25	82.91	82.5	82.7	82.3
		Medium	20.75	79.7	78.5	79.1	78.3
		Coarse	24	77.66	73	75.26	73.1
	ATP- GTCC	Fine	15	85.35	84.5	84.92	84.6
		Medium	18.5	82.47	80	81.22	80.3
		Coarse	21	80.21	77	78.57	77.1

to indicate the significance of the proposed method for accurate classification of bonafide and spoof samples. For this purpose, we trained various classifiers (decision trees [33], k-nearest neighbor (KNN) [34], naïve bayes [35], ensemble classifiers [36], and BiLSTM deep learning framework [37]) on the extracted features. We performed different experiments using different feature-sets to train these classifiers as done for SVM. Again, for VSDC, we used 70% of the samples for training and the remaining 30% for testing purposes, whereas for the ASVspoof dataset, we used the training set to train the model and evaluation set for testing.

1) Classification Using Decision Trees: For experimental purposes, we created the decision trees at different levels: coarse-level, using only few decision nodes (maximum number of splits is 4); medium-level, with more decision nodes (maximum number of splits is 20); and fine-level using a large number of decision nodes (maximum number of splits is 100). It is to be noted that fine trees have more depth in the structure and coarse has the least. We trained the decision trees on both datasets for all of these levels. We repeated the experiments for different features-sets as adopted for SVM. The detailed results of these experiments performed on different features are reported in Table V.

For GTCC-MFCC-spectral features-set, we achieved the lowest EER of 11% and 15.25% on the VSDC and ASVspoof 2019 datasets respectively for decision trees trained on fine level. Similarly, for ATP-spectral features-set, we repeated all the experiments and achieved a minimum EER of 18.66% on VSDC and 18.55% on the ASVspoof 2019 dataset. For ATP-MFCC combination, we achieved the lowest EER of 16.8% and 17.3% on the VSDC and ASVspoof 2019 datasets. Finally, for our proposed ATP-GTCC features, we also attained the best results on the decision trees trained at fine-level depth. More specifically, we obtained an EER of 16.3% and 15% on VSDC and ASVspoof datasets respectively.

Results show that decision trees at coarse-level train faster as compared to medium- and fine-level trees since coarse-level trees have the fewest nodes. However coarse-level decision trees are less accurate in terms of classification as compared to medium- and fine-level trees. It is important to mention that fine-level trees have the most depth of all trees and performs best for all features-sets and on both datasets. We conclude that fine-level decision trees are most effective and coarse-level trees are most efficient among all the three levels used for training.

2) Classification using Naïve Bayes: We performed an experiment to train different features-sets on Naïve Bayes with gaussian and kernel distributions separately and results are shown in Table VI.

For GTCC-MFCC-spectral features, we achieved an EER of 30.66% on VSDC, and 17% on the ASVspoof dataset. For ATP-spectral features, we obtained an EER of 31% and 26.5% on VSDC and ASVspoof datasets. Similarly, for ATP-MFCC features we achieved an EER of 31.3% and 20.75% on VSDC and ASVspoof datasets. Finally, for our proposed ATP-GTCC features-set we achieved the lowest EER of 27% on our dataset and 19.75% on the ASVspoof 2019 dataset.

Table VI shows that Naïve Bayes performs better with the kernel distribution as compared to gaussian distribution. However, this superior performance comes at the expense of increased computational cost and memory. This is because modeling each feature with the kernel distribution requires calculating a separate kernel density estimate for each class according to the training data of that class.

3) Classification Using K-Nearest Neighbor (KNN): We also tested the classification performance of KNN on both datasets. Our experiments demonstrate that KNN achieves very low EER

TABLE VI
DETECTION RESULTS OF NAÏVE BAYES

Dataset	Features Set	Distribution	EER%	Precision%	Recall%	F1-Score%	Accuracy%
VSDC	GTCC-MFCC-Spectral	Kernel	30.66	69.6	68.7	69.13	68.5
		Gaussian	40.33	59.9	58.3	59.12	58.3
	ATP-Spectral	Kernel	31	69	69	69	68.8
		Gaussian	48.5	51.5	50.3	50.93	50.4
	ATP-MFCC	Kernel	31.33	68.7	68.7	68.7	68.6
		Gaussian	43.16	57.4	53.3	55.27	57.3
	ATP-GTCC	Kernel	27	73	73	73	73.1
		Gaussian	38.33	62.1	60	61.02	60
ASVspoof	GTCC-MFCC-Spectral	Kernel	17	83	83	83	82.7
		Gaussian	29	73.6	65.5	69.31	65.8
	ATP-Spectral	Kernel	26.5	73.74	73	73.37	72.8
		Gaussian	27.25	73.1	72	72.55	71.9
	ATP-MFCC	Kernel	20.75	80	78	78.99	77.8
		Gaussian	23.25	77.44	75.5	76.46	75.2
	ATP-GTCC	Kernel	19.75	81.03	79	80	79
		Gaussian	22.25	78.76	76	77.36	76.4

TABLE VII
DETECTION RESULTS OF KNN

Dataset	Features Set	EER %	Precision %	Recall %	F1-Score %	Accuracy %
VSDC	GTCC-MFCC-Spectral	4.16	95.8	95.7	96	95.84
	ATP-Spectral	3	97	97	97	96.9
	ATP-MFCC	2	98	98	98	97.7
	ATP-GTCC	1.83	98.3	98	98.16	98.2
ASVspoof	GTCC-MFCC-Spectral	10.5	89.5	89.5	89.5	89.7
	ATP-Spectral	9.5	90.5	90.5	90.5	90.5
	ATP-MFCC	7.75	92.5	92.5	92.5	91.9
	ATP-GTCC	7.5	92.46	92	92.23	92.1

as shown in Table VII. However, KNN is computationally expensive and requires more memory as it stores all the training data. In addition, KNN is sensitive to irrelevant features and scale of data.

For our experiments using the KNN, we tuned three parameters: the number of neighbors (k_n), the distance metric to compute the nearest neighbors (NN), and distance weights. More specifically, we set the distance metric to Euclidean and distance weight to equal for the first three experiments and used different values of neighbors; first with $k_n = 1$ (fine KNN), then $k_n = 10$ (medium KNN), and finally $k_n = 100$ (coarse KNN). In the next two experiments, we fixed the number of neighbors $(k_n = 10)$ and distance weight to equal while changing the distance metric to cosine and cubic. Finally, in the last experiment of KNN, we modified the distance weight from equal to squared inverse, fixed $k_n = 10$ and the distance metric to Euclidean (Table VII). We assigned different weights to the neighbors based on distance, because KNN assumes that closer samples are possibly similar, so it makes sense to distinguish among the nearest neighbors during classification of new samples. Therefore, we assigned higher weights to the closer neighbors to ensure their maximum contribution towards deciding the class of the new instance. More specifically, each NN is assigned a weight based on the squared inverse mechanism, in which closer neighbors have higher weights and vice versa.

For GTCC-MFCC-spectral features-set, weighted and cosine KNN performs best among all variations of KNN on VSDC. It is to be noted that both weighted and cosine KNN use 10 nearest neighbors. More specifically, both weighted and cosine KNN achieves an EER of 2.66% on VSDC.

For the ASVspoof 2019 dataset, fine KNN achieves the lowest EER of 8.75%. For ATP-spectral features set, fine KNN performs best on VSDC with an EER of 3.33%, whereas weighted KNN performs best on the ASVspoof dataset, obtaining an EER of 10.75%. Similarly, for ATP-MFCC features set we achieved an EER of 1.58% on VSDC and 7.75% on the ASVspoof dataset using fine KNN. Finally, for our ATP-GTCC features set we also achieved the best results with fine KNN on both datasets. More precisely, we achieved an EER of 0.75%, precision of 99.3%, recall of 99.2%, f1-score of 99.25%, and accuracy of 99.3% on VSDC, and an EER of 7%, precision, recall, accuracy, and f1-score of 93% on the ASVspoof dataset.

From the results (Table VII), we conclude that KNN with minimum value of k_n provides the best results for all sets containing ATP features. Whereas, conventional spectral features perform best when k_n is set to around 10. In addition, KNN using the Euclidean distance performs best in all features-sets for both datasets. We also observed that assigning different weights to neighbors based on distance in KNN results in better performance for conventional spectral features sets.

4) Classification Using Ensemble Classifiers: Ensemble methods are created through integrating multiple classifiers to build a predictive model with the aim of achieving better accuracy.

Some ensemble classifiers combination cause data overfitting, however, ensemble methods like bagging and boosting decrease the variance and bias, respectively. Though ensemble classifiers can achieve better accuracy, this is at the expense of increased computational cost. In light of the above facts, we evaluated the performance of different ensemble classifiers on the selected

TABLE VIII
DETECTION RESULTS OF ENSEMBLE BAGGED TREES

Dataset	Features Set	EER %	Precision %	Recall %	F1-Score %	Accuracy %
VSDC	GTCC-MFCC-Spectral	2.66	97.3	97.3	97.3	97.2
	ATP-Spectral	3.33	96.7	96.7	96.7	96.7
	ATP-MFCC	1.58	98.5	98.3	98.41	98.6
	ATP-GTCC	0.75	99.3	99.3	99.3	99.3
ASVspoof	GTCC-MFCC-Spectral	8.75	91.46	91	91.23	91
	ATP-Spectral	10.75	89.45	89	89.22	89.4
	ATP-MFCC	7.75	92.46	92	92.23	91.9
	ATP-GTCC	7	93	93	93	93

TABLE IX
DETECTION RESULTS OF BILSTM DEEP LEARNING

Dataset	Features-Set	EER %
VSDC	GTCC-MFCC-Spectral	15.3
	ATP-Spectral	14.5
	ATP-MFCC	13.9
	ATP-GTCC	13.1
ASVspoof	GTCC-MFCC-Spectral	15.7
	ATP-Spectral	14.3
	ATP-MFCC	13.1
	ATP-GTCC	12.7

features sets. More specifically, we employed five different ensemble methods: boosted trees [38], bagged trees [39], subspace discriminant [40], subspace KNN [41], and RUSBoosted trees [42].

As in earlier experiments, we evaluated the performance of ensemble methods on different features-sets. Ensemble bagged trees performed best for all features-sets as compared to other ensemble methods. The results of ensemble bagged trees on all features-sets and both datasets are provided in Table VIII. We achieved the lowest EER of 4.16% and 10.5% on GTCC-MFCC-spectral features-set, 3% and 9.5% on ATP-spectral, 2% and 7.75% on ATP-MFCC, and 1.83% and 7.5% on proposed ATP-GTCC features-set for VSDC and ASVspoof datasets respectively. From the results, we conclude that our ATP-GTCC features-set provides better classification performance as compared to other features on ensemble bagged trees classifier.

5) Classification Using Deep Learning: The significance of recurrent neural networks (RNN) in analysis of sequential and time series data motivated us to apply the BiLSTM deep learning model (a type of RNN) for audio replay attack detection. For this experiment, we evaluated the performance of BiLSTM framework on different features-sets, and results are reported in Table IX.

TABLE X
DETECTION PERFORMANCE OF DIFFERENT CLASSIFIERS WITH PROPOSED FEATURES

Dataset	Classifiers	EER%
VSDC	Decision Trees	4.16
	Naïve Bayes	27
	KNN	0.75
	Ensemble Models	1.83
	BiLSTM	13.1
	SVM	**0.6**
ASVspoof	Decision Trees	15
	Naïve Bayes	19.75
	KNN	7
	Ensemble Models	7.5
	BiLSTM	12.7
	SVM	**1**

We used different numbers of hidden layers and tuned different parameters during network training. More specifically, we tuned the following parameters: number of hidden units, state activation function, gate activation function, batch size, and maximum epochs. We performed the experiments using 100, 200, and 300 number of hidden units, and obtained best results for each features-set on 200 hidden units. For the state activation function, we tuned the system on tanh and soft-sign and found that tanh outperforms the soft-sign state activation function in almost all experiments. Similarly, for the gate activation function, we used sigmoid and hard-sigmoid and obtained best results on sigmoid function. For network training, we tuned the maximum number of epochs in different ranges and received best results on 200 epochs for each experiment. Mini-batch size was also set to different values of 16, 24, 32, 64, and 128 for each features-set. For all cases we found best results on mini-batch size set to 64. The results obtained on each features-set using the BiLSTM framework are shown in Table IX. Again, our proposed ATP-GTCC features-set provides best performance as compared to other features-sets. It is also important to mention that the BiLSTM framework achieves much higher EER (Table IX) as compared to SVM. Therefore, we conclude from this experiment that BiLSTM model is less effective for audio replay spoofing detection.

6) Analysis of Features Performance Comparison: Performance comparison of different classifiers, using the proposed features-set, shows that SVM performs best and Naïve Bayes is the worst in terms of EER. More specifically, SVM achieves the lowest EER of 0.6%, whereas Naïve Bayes achieves the highest EER of 27% on our proposed ATP-GTCC features (Table X). Therefore, we argue that SVM can reliably be used to classify the bonafide and replay spoof samples.

E. Performance Comparison With Existing Methods

This experiment is designed to compare the performance of the proposed method against state-of-the-art replay attack detection methods. For this purpose, a comparative analysis of the proposed method is performed with these techniques [3, 5, 6, 10, 12–15, 18–19, 22]. The EER values of the proposed and comparative methods are provided in Fig. 8.

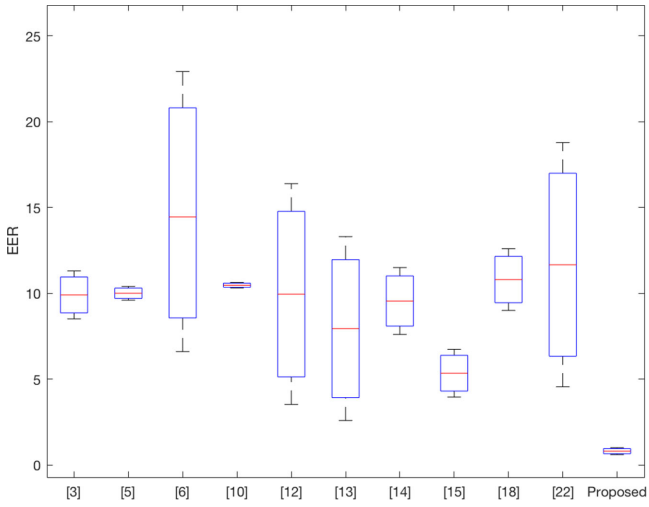

Fig. 8. Performance comparison with existing state-of-the-arts.

Gunendradasan *et al.* [3] evaluated the performance of their method on ASVspoof corpus. TLC-AM features achieved an EER of 8.51% and 8.68% and TLC-FM features obtained an EER of 10.11% and 11.3% on V1 and V2 corpus respectively. However, the fusion of TLC-AM and TLC-FM features improve the detection performance and obtained the EER of 7.32% on V1 corpus and 7.59% on V2 corpus. Ji *et al.* [5] achieved an average EER of 10.2% with a limitation of data overfitting due to the small size of the ASVspoof development set. Witkowski *et al.* [6] achieved an average EER of 6.6% and 22.93% on ASVspoof development and evaluation datasets respectively. Yang *et al.* [10] achieved an ERR of 10.63% and 10.31% on CQNSC and CQNCC features on ASVspoof V2 corpus. Cai *et al.* [12] achieved an EER of 16.39% on the evaluation set and 3.52% on the development set of ASVspoof. Chen *et al.* [13] achieved an EER of 2.58% on development and 13.3% on evaluation dataset of ASVspoof. Nagarsheth *et al.* [14] achieved an EER of 7.6% and 11.5% on the development and evaluation datasets of ASVspoof. Lavrentyeva *et al.* [15] obtained an EER of 3.95% on development and 6.73% on evaluation dataset of ASVspoof. Balamurali *et al.* [18] achieved an EER of 10.8%, whereas, Bakar *et al.* [22] obtained an EER of 18.78% and 4.55% on development set, and 24.81% and 18.1% on the evaluation set of ASVspoof. Finally, the proposed method outperforms the existing state-of-the-art replay spoofing detection methods and achieved an EER of 0.6% and 1% on VSDC and ASVspoof datasets 2019 respectively.

F. Performance Evaluation on Mixed Dataset

The purpose of this experiment is to evaluate the performance of the proposed method under more diverse conditions where the dataset samples are heterogeneous in nature i.e. speakers, environments, microphones and playback devices, sampling rate, etc. For this purpose, we have created the training and testing sets comprising of bonafide and spoof samples from both the ASVspoof and VSDC datasets. For this experiment, we have taken 8000 bonafide and first-order replay samples from the ASVspoof dataset, and same from the VSDC (16 000 audio samples in total). Afterwards, we used 70% (11 200 audio samples) of the data for training and the remaining 30% (4 800 audio samples) for testing purposes using the RBF kernel based SVM classifier. We obtained an EER of 9.9% and accuracy of 90% that clearly shows the effectiveness of the proposed method in terms of replay attack detection even when the audio samples were highly diverse.

G. Comparative Analysis of Features Computation Cost

The proposed framework selects efficient features to counter the replay spoofing attacks for VCDs in IoT environment. In this sub-section we provided a comparative analysis of the computational complexity of proposed features against different features-sets.

The computational cost of proposed ATP-GTCC features-set is $O(n) + O(n\log(n))$. The computational cost of ATP-MFCC features is similar to our proposed features, however our proposed ATP-GTCC features provides superior classification performance over ATP-MFCC on both VSDC and ASVspoof datasets. Additionally, ATP-Spectral (Table III) has a computational cost of $O(n) + O(n\log(n))$, whereas GTCC-MFCC-Spectral combination is the most complex having computational cost of $O(n\log(n)) + O(n\log(n)) + O(n\log(n))$.

From the above-mentioned statistics, we can clearly observe that the proposed ATP-GTCC features has the lowest complexity as compared to other features. Hence, ATP-GTCC features can reliably be used in resource constrained environments.

VI. Conclusion

The proposed anti-spoofing framework is the first attempt to address the issue of detecting the first- and second-order replay attacks. The voice replay attack is modeled as a nonlinear process that introduces harmonic distortions. Stronger harmonic distortions are used to quantify replay attacks which is used for developing an anti-spoofing framework. We proposed a light-weight ATP-GTCC features-set to better capture the non-linear characteristics, due to distortions, of first- and second-order replay samples. The proposed light-weight solution is best fit for resource constraint VCDs connected in IoT environment. The EER of 0.6% on VSDC and 1% on the ASVspoof datasets signifies the effectiveness of the proposed framework in terms of replay attack detection.

We plan to extend the proposed framework to develop a hybrid anti-spoofing system that can effectively be used to combat both replay and cloning attacks. Additionally, we are also planning to extend our VSDC to include the cloning samples as well, since the existing audio cloning datasets like ASVspoof lack high-quality cloning audios.

Reference

[1] K. M. Malik, H. Malik, and R. Baumann, "Towards vulnerability analysis of voice-driven interfaces and countermeasures for replay attacks," in *Proc. IEEE Conf. Multimedia Inf. Process. Retrieval*, 2019, pp. 523–528.

[2] M. Todisco, D. Héctor, and E. Nicholas, "Constant Q cepstral coefficients: A spoofing countermeasure for automatic speaker verification," *Comput. Speech Lang.*, vol. 45, pp. 516–535, 2017.

[3] T. Gunendradasan, S. Irtza, E. Ambikairajah, and J. Epps, "Transmission line cochlear model based AM-FM features for replay attack detection," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2019, pp. 6136–6140.

[4] S. Novoselov, A. Kozlov, G. Lavrentyeva, K. Simonchik, and V. Shchemelinin, "STC anti-spoofing systems for the ASVspoof 2015 challenge," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2016, pp. 5475–5479.

[5] Z. Ji *et al.*, "Ensemble learning for countermeasure of audio replay spoofing attack in ASVspoof2017," in *Proc. Inter Speech*, 2017, pp. 87–91.

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

[7] J. Mishra, M. Singh, and D. Pati, "Processing linear prediction residual signal to counter replay attacks," in *Proc. Int. Conf. Signal Process. Commun.*, 2018, pp. 95–99.

[8] M. S. Saranya, R. Padmanabhan, and H. A. Murthy, "Replay attack detection in speaker verification using non-voiced segments and decision level feature switching," in *Proc. Int. Conf. Signal Process. Commun.*, 2018, pp. 332–336.

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

[10] A. P. Tapkir and H. A. Patil, "Significance of teager energy operator phase for replay spoof detection," in *Proc. Asia-Pacific Signal Inf. Process. Assoc. Annu. Summit Conf.*, 2018, pp. 1951–1956.

[11] W. Cai, D. Cai, W. Liu, G. Li, and M. Li, "Countermeasures for automatic speaker verification replay spoofing attack: On data augmentation, feature representation, classification and fusion," in *Proc. Inter Speech*, 2017, pp. 17–21.

[12] Z. Chen, Z. Xie, W. Zhang, and X. Xu. "ResNet and model fusion for automatic spoofing detection," in *Proc. Inter Speech*, 2017, pp. 102–106.

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

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

[15] X. Wu, R. He, Z. Sun, and T. Tan, "A light CNN for deep face representation with noisy labels," *IEEE Trans. Inf. Forensics Secur.*, vol. 13, no. 11, pp. 2884–2896, Nov. 2018.

[16] G. Lavrentyeva, S. Novoselov, A. Tseren, M. Volkova, A. Gorlanov, and A. Kozlov, "STC antispoofing systems for the ASVspoof 2019 challenge," 2019, arXiv:190405576.

[17] B. T. Balamurali, K. W. E. Lin, S. Lui, J. Chen, and D. Herremans, "Towards robust audio spoofing detection: a detailed comparison of traditional and learned features," 2019, arXiv:1905 12439.

[18] A. K. Sarkar, Z. Tan, H. Tang, and J. Glass, "Time-contrastive learning based deep bottleneck features for text-dependent speaker verification," in *Proc. IEEE/ACM Trans. Audio, Speech, Lang. Process.*, 2019.

[19] Y. Gong, J. Yang, J. Huber, M. MacKnight, and C. Poellabauer, "ReMASC: Realistic replay attack corpus for voice controlled systems," 2019, arXiv:1904.03365.

[20] B. Bakar and C. Hanilçi, "Replay spoofing attack detection using deep neural networks," in *Proc. 26th Signal Process. Commun. Appl. Conf.*, 2018, pp. 1–4.

[21] ASVspoof Challenge. [Online]. Available: "https://www.asvspoof.org," Accessed on Jul. 25, 2019.

[22] V. Tiwari, M. F. Hashmi, A. Keskar, and N. C. Shivaprakash, "Virtual home assistant for voice-based controlling and scheduling with short speech speaker identification," *Multimedia Tools Appl.*, pp. 1–26, 2018.

[23] D. Cooper, "Speech detection using gammatone features and one-class support vector machine," M.S. Thesis, Dept. Elec. Eng. & Comp. Sci., Univ. of Central Florida, Orlando, FL, USA, 2013.

[24] A. V. Memos, E. K. Psannis, I. Yutaka, B. Kim, and B. B. Gupta, "An efficient algorithm for media-based surveillance system (EAMSuS) in IoT smart city framework," *Future Gener. Comput. Syst.*, vol. 83, pp. 619–628, 2018

[25] S. Badaskar, "Voice-based media searching," U.S. Patent 9,547,647, issued Jan. 17, 2017.

[26] X. Tan and W. Triggs, "Enhanced local texture feature sets for face recognition under difficult lighting conditions," *IEEE Trans. Image Process.*, vol. 19, no. 6, pp. 1635–1650, Jun. 2010.

[27] T. Ojala, M. Pietikäinen, and D. Harwood, "A comparative study of texture measures with classification based on featured distributions," *Pattern Recognit.*, vol. 29, no. 1, pp. 51–59, 1996.

[28] R. Patterson, "Spiral vos final report, Part A: The auditory filterbank," Cambridge Electronic Design, Contract Rep., 1988.

[29] B. R. Glasberg and B. C. J. Moore. "Derivation of auditory filter shapes from notched-noise data," *Hearing Res.*, vol. 47, no. 1-2, pp. 103–138, 1990.

[30] Voice Spoofing Detection Corpus [Online]. Available: http://www.secs.oakland.edu/∼mahmood/datasets/audiospoof.html

[31] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*. Boca Raton, FL: Chapman & Hall, 1984.

[32] S. Zhang, X. Li, M. Zong, X. Zhu, and R. Wang, "Efficient KNN classification with different numbers of nearest neighbors," *IEEE Trans. Neural Netw. Learning Syst.*, vol. 29, no. 5, pp. 1774–1785, May 2018.

[33] Z. Fu, G. Lu, K. M. Ting, and D. Zhang,"Learning naive bayes classifiers for music classification and retrieval," in *Proc. 20th Int. Conf. Pattern Recognit.*, 2010, pp. 4589–4592.

[34] T. M. Dietterich, "Ensemble methods in machine learning," in *Proc. Int. Workshop Multiple Classifier Syst.*, 2000, pp. 1–15.

[35] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional LSTM and other neural network architectures," *Neural Netw.*, vol. 18, no. 5-6, pp. 602–610, 2005.

[36] O. Hubacek, G. Sourek, and F. Zelezny, "Learning to predict soccer results from relational data with gradient boosted trees," *Mach. Learn.*, vol. 108, no. 1, pp. 29–47, 2019.

[37] J. Sun, J. Lang, H. Fujita, and H. Li, "Imbalanced enterprise credit evaluation with DTE-SBD: Decision tree ensemble based on SMOTE and bagging with differentiated sampling rates," *Inf. Sci.*, vol. 425, pp. 76–91, 2018.

[38] R. Hang, Q. Liu, H. Song, and Y. Sun, "Matrix-based discriminant subspace ensemble for hyperspectral image spatial–spectral feature fusion," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 2, pp. 783–794, Feb. 2015.

[39] Y. Zhang, G. Cao, B. Wang, and X. Li. "A novel ensemble method for k-nearest neighbor," *Pattern Recognit.*, vol. 85, pp. 13–25, 2019.

[40] J. Moeyersons, C. Varon, D. Testelmans, B. Buyse, and S. V. Huffel, "ECG artefact detection using ensemble decision trees," in *Proc. Comput. Cardiology (CinC)*, 2017, pp. 1–4.

[41] S. Escalera, O. Pujol, and P. Radeva, "Separability of ternary codes for sparse designs of error-correcting output codes," *Pattern Recognit. Lett.*, vol. 30, no. 3, pp. 285–297, 2009.

[42] W. Abdullah, "Auditory based feature vectors for speech recognition systems," *Adv. Commun. Softw. Technol.*, pp. 231–236, 2002.

[43] Aircrack-Ng, [Online]. Available: https://www.aircrack-ng.org, Accessed on Aug. 8, 2019.

[44] J. Kollewe, "HSBC rolls out voice and touch ID security for bank customers | Business," The Guardian. Accessed on Aug. 8, 2019.

[45] R. Brockbank and C. Wass, "Non-linear distortion in transmission systems," *J. Inst. Electr. Eng.- Part III: Radio Commun. Eng.*, vol. 92, no. 17, pp. 45–56, Mar. 1945.

[46] V. D. Svetislav, "Distortion in microphones," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. '76*, Apr. 12–14, 1976.

[47] M. T. Abuelma. "Harmonic and intermodulation distortion in carbon microphones," *Appl. Acoust.*, vol. 31, no. 4, pp. 233–243, 1990.

[48] H. Malik and J. Miller, "Microphone identification using higher-order statistics," in *Proc. 46th AES Conf. Audio Forensics*, Jun. 14–16, 2012.

[49] H. Malik, "Securing speaker verification system against replay attack," in *Proc. 46th AES Conf. Audio Forensics*, Jun. 14–16, 2012.

[50] S. M. Adnan, A. Irtaza, S. Aziz, M. O. Ullah, A. Javed, and M. T. Mahmood, "Fall detection through acoustic local ternary patterns," *Appl. Acoust.*, vol. 140, pp. 296–300, 2018.

[51] A. Irtaza, S.M. Adnan, S. Aziz, A. Javed, M.O. Ullah, and M.T. Mahmood, "A framework for fall detection of elderly people by analyzing environmental sounds through acoustic local ternary patterns," in *Proc. IEEE Int. Conf. Syst. Man Cybern.*, Oct. 2017, pp. 1558–1563.

[52] X. Valero and F. Alias, "Gammatone cepstral coefficients: Biologically inspired features for non-speech audio classification," in *IEEE Trans. Multimedia*, vol. 14, no. 6, pp. 1684–1689, Dec. 2012.

[53] [Online]. Available: https://github.com/alijaved21/Voice-Replay-Anti-spoofing

Khalid Mahmood Malik (Senior Member, IEEE) received the Ph.D. degree from the Tokyo Institute of Technology Tokyo, Japan, in 2010. He is currently working as Assistant Professor with the School of Engineering and Computer Science, Oakland University, Rochester, MI, USA. Prior to that he has worked as Visiting Researcher with Sanyo Electric, Japan, and then Project Manager R&D with DTS Inc. Japan, from 2010 to 2014. His current research interests include multimedia forensics, development of intelligent decision support systems using analysis of medical imaging and clinical text, secure multicast protocols for intelligent transportation systems, and automated ontology and knowledge graph generation. His research is supported by the National Science Foundation (NSF), Brain Aneurysm Foundation, and Oakland University. He is founding member of the Center of Cybersecurity, Oakland University, MI, USA.

Ali Javed (Member, IEEE) received the B.Sc. (honors and third position) degree in software engineering from UET Taxila, Pakistan in 2007. He received his MS and Ph.D. degrees in computer engineering from UET Taxila, Pakistan in 2010 and 2016. He received Chancellor's Gold Medal in MS Computer Engineering degree. He is serving as an Assistant Professor in Software Engineering Department at UET Taxila, Pakistan. He has served as a Postdoctoral Scholar in SMILES lab at Oakland University, MI, USA in 2019 and as a visiting PhD scholar in ISSF Lab at University of Michigan, MI, USA in 2015. His areas of research interest are image processing, computer vision, multimedia forensics, video content analysis, medical image processing, and multimedia signal processing. He has published more than 50 papers in leading journals and conferences. Dr. Javed is a recipient of various research grants from HEC Pakistan, National ICT R&D Fund, NESCOM, and UET Taxila Pakistan. He has also served as an HOD in Software Engineering Department at UET Taxila in 2014. He was selected as an Ambassador of Asian Council of Science Editors from Pakistan in 2016. He is also a Member of Pakistan Engineering Council since 2007.

Hafiz Malik (Senior Member, IEEE) received the B.Sc. degree in electronics and computer engineering. He is Associate Professor in the Electrical and Computer Engineering (ECE) Department at University of Michigan – Dearborn. His current research in the areas of automotive cybersecurity, IoT security, sensor security, multimedia forensics, steganography/steganalysis, information hiding, pattern recognition, and information fusion is funded by the National Science Foundation, National Academies, Ford Motor Company, and other agencies. He has published more than 100 papers in leading journals, conferences, and workshops. He is a founding member of the Cybersecurity Center for Research, Education, and Outreach at UM-Dearborn and member leadership circle for the Dearborn Artificial Intelligence Research Center at UM-Dearborn. He is also a Member of the Scientific and Industrial Advisory Board (SIAB) of the National Center of Cyber Security Pakistan. He is a Member of MCity Working Group on Cybersecurity, since 2015.

Aun Irtaza has completed his Ph.D. in 2016 from FAST-nu, Islamabad Pakistan. During his Ph.D. he remained working as a Research Scientist in the Gwangju Institute of Science and Technology (GIST), South Korea. He became an Associate Professor in 2017 and Department of Computer Science Chair in 2018 in the University of Engineering and Technology (UET) Taxila, Pakistan. He is currently working as visiting Associate Professor in the University of Michigan-Dearborn. His current research areas include computer vision, multimedia forensics, audio-signal processing, medical image processing, and Big data analytics. He has more than 40 publications in IEEE, Springer, and Elsevier Journals.