IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 1 August 2024, accepted 18 August 2024, date of publication 21 August 2024, date of current version 2 September 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3447572

RESEARCH ARTICLE

Multi Pattern Features-Based Spoofing Detection Mechanism Using One Class Learning

BESTE USTUBIOGLU[1], GUL TAHAOGLU[1,2], ARDA USTUBIOGLU[3], GUZIN ULUTAS[1], (Senior Member, IEEE), IRENE AMERINI[4], (Member, IEEE), AND MUHAMMED KILIC[1]

[1]Computer Engineering Department, Karadeniz Technical University, 61080 Trabzon, Türkiye
[2]Department of Information Engineering, University of Florence, 50139 Florence, Italy
[3]Department of Management Information Systems, Trabzon University, 61335 Trabzon, Türkiye
[4]Department of Computer, Control and Management Engineering, Sapienza University of Rome, 00185 Rome, Italy

Corresponding author: Gul Tahaoglu (gul.tahaoglu@unifi.it)

This work was supported by the Scientific and Technological Research Council of Türkiye (TUBITAK) under Project 122E013.

ABSTRACT Automatic Speaker Verification systems are prone to various voice spoofing attacks such as replays, voice conversion (VC) and speech synthesis. Malicious users can perform specific tasks such as controlling the bank account of someone, taking control of a smart home, and similar activities, by using advanced audio manipulation techniques. This study presents a Multi-Pattern Features Based Spoofing detection mechanism using the modified ResNet architecture and OC-Softmax layer to detect various LA and PA spoofing attacks. We proposed a novel Pattern features-based audio spoof detection scheme. The scheme contains three branches to evaluate different patterns on a Mel spectrogram of the audio file. This is the first work for the audio spoofing detection task using three different pattern representations of Mel spectrogram with modified ResNet architecture and OC-Softmax layer. Through the proposed network, we can extract pattern images from the Mel spectrogram and gives each of them into modified ResNet architecture. At the last step of each network, we use OC-Softmax to obtain a score for the current pattern image and then the method fuses three scores to label the input audio. Experimental results on the ASVspoof 2019 and ASVspoof 2021 corpuses show that the proposed method achieves better results in the challenges of ASVspoof 2019 than state-of-the-art methods. For example, in the logical access scenario, our model improves the tandem decision cost function and equal error rate scores by 0.06% and 2.14%, respectively, compared with state-of-the-art methods. Additionally, experiments illustrate that the proposed fused decision improved the performance of the system.

INDEX TERMS Deep fake audio, audio forgery, forgery detection.

I. INTRODUCTION

Automatic Speaker verification systems (ASV) have become popular in recent years, especially after the Covid 19 pandemic. Many techniques, such as password-based authentication and fingerprint scanning, which are used to verify the corresponding person, necessitate contact-based interaction. Therefore, utilizing specific methods that do not require any interaction is important when considering health issues that affect people all around the world. Many people use ASV in their daily lives such as by talking with their cellphones, realizing banking operations, and giving some commands to their smart watches. An ASV system gets the voice through the microphone and processes it to decide about it. It accepts or rejects the voice according to the result. The main objective of an ASV system is to decide whether the input audio is genuine or not. Even if ASV-based systems make our lives simpler, they also come with a lot of problems. If a malicious user imitates a legal user's voice to authenticate himself to the system, such a situation can be a problem for ASV systems.

Three different attacks can be applied by attackers to ASV systems: voice conversion, synthetic speech, and replay

The associate editor coordinating the review of this manuscript and approving it for publication was Tianhua Xu.

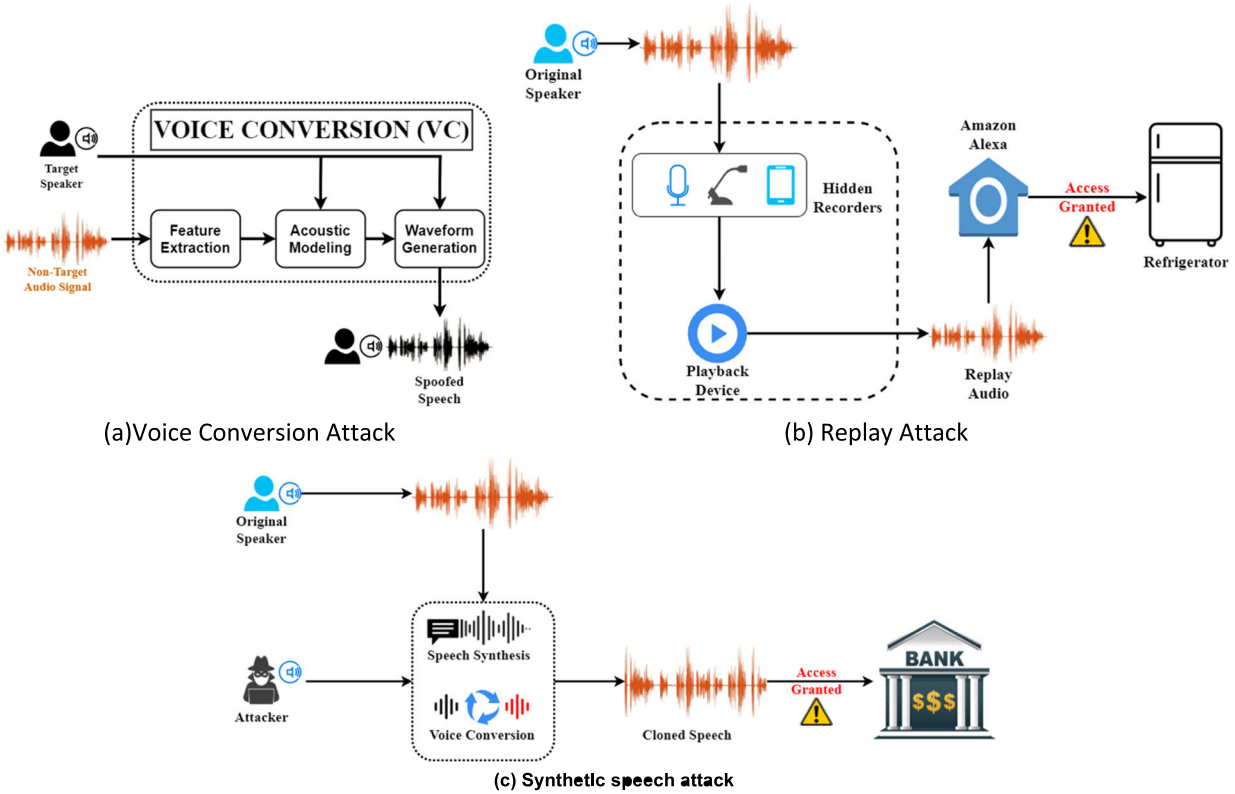


FIGURE 1. Various attack scenarios against ASV system (a) Voice conversion attack (b) Replay Attack (c) Synthetic speech attack.

attack. In the first scenario, shown in Fig. 1(a), an attacker for a voice conversion attack synthetically generates a sound that is very similar to the registered speaker. The voice conversion technique transforms a malicious person's voice into that of an enrolled speaker. An attacker uses feature extraction and acoustic modeling techniques with waveform generation to produce a speech by using the target speaker's voice. Fig. 1(b) shows the general idea of a replay attack. In this scenario, the attacker uses a recorder device to obtain genuine speech, and then he uses this speech to authenticate himself to the system at another time. In the last scenario, the attacker gets the original voice samples from the victim and then uses a speech synthesis system to generate fake speech that imitates the victim's voice as can be seen in Fig. 1(c).

Some problems are encountered by ASV systems that were first visible in a special session in 2013 [1]. Subsequently, the ASVSpoof challenge was organized to provide a public platform for considering anti-spoof methods and to evaluate the performance of the methods according to specific metrics [2]. The most recent challenge, denoted by AsvSpoof 2019, contains three major attack types, as described [3]. ASVSpoof 2019 consists of two scenarios: Logical Access (LA) and Physical Access (PA). While Text to Speech (TTS) and VC attacks were used to create LA dataset, PA consists of replay spoofed voices. PA and LA datasets are divided into three subsets: Training, development, and evaluation subsets. While eight males and twelve females' voices are used to generate the training set, the evaluation set is generated by 21 males and 27 females. The recording environment is equal during the generation of subsets. The same algorithms are used to generate known attacks for both training and development subsets. However, the evaluation subset contains samples that are generated using different synthesizing algorithms. ASVSpoof 2019 dataset came with two different metrics denoted by Equal Error Rate (EER) and Tendum Detection Cost Function (t-DCF) to make a fair comparison between the methods. Therefore, many researchers aim to propose new methods to obtain better EER and t-DCF values on the new challenge.

The methods in the literature have some specific problems: Single spoofing type detectors, limited generalization capability, and high computational cost. The most important problem for audio spoof detection is the generalization capability of detectors. Various low-level spectro-temporal features were analyzed by the researchers to overcome this problem. Even if the works in the literature give lower EER values on the ASVspoof 2019 dataset, they are not able to deal with the generalization problem on ASVspoof 2019 in a well manner especially when we consider the recent improvements on the TTS (Text-to-speech) and VC technologies. Generalization is important for this field because you can deal with unseen attacks if your model is generalized properly. In this work, we aim to capture different characteristics of various PA and LA attacks using pattern information on the spectrogram images.

The proposed approach consists of three stages: Audio visual representation and texture extraction, Training of the Deep Neural Networks and Fusion of the scores. At the first step, the method transforms input audio file into frequency domain using Mel Spectrogram technique and then three different pattern extraction techniques (Local Binary Pattern, Gray Level Cooccurrence Matrix and Local Phase Quantization) are utilized by the method to obtain different characteristics of the Mel Spectrogram image. In the second phase of the proposed method, we modified Resnet18 architecture and used pattern images from three different methods to train three modified Resnet18-based deep neural networks (DNN). Decision obtained from those DNNs are fused together to obtain a final decision of the system.

The major contributions of this work can be listed as follows:

- We design a novel Pattern Features-based audio spoof detection scheme. The scheme contains three branches to evaluate different patterns on a Mel spectrogram of the audio file. As far as we know, this is the first work for the audio spoofing detection task using three different pattern representations of Mel spectrogram with modified ResNet architecture and OC-Softmax layer.
- We combine the decisions for three pattern images for audio spoofing detection. Through the network we designed, we can extract pattern images from the Mel spectrogram and give each of them into modified ResNet architecture. At the last step of each network, we use OC-Softmax to obtain a score for the current pattern image and then the method fuses three scores to label the input audio. Our experiments show that this fusion decision can improve the performance of our model.
- The proposed method achieves better results in the challenges of ASVspoof 2019 and also ASVspoof 2021 datasets than state-of-the-art methods. For example, in the logical access scenario, our model improves the tandem decision cost function and equal error rate scores by 0.06% and 2.14% respectively, compared with state-of-the-art methods.

This paper is organized as follows: While section II gives a brief description of some approaches that are used in the method, the details of the proposed method will be given in section III. Section IV presents the experimental results of the proposed approach and gives a comparison with state-of-the-art methods. The conclusion is also drawn in the last section.

II. RELATED WORKS

The methods in the literature consist of two stages to detect forged audio: Feature extraction and classification. In the feature extraction phase, existing approaches have employed either handcrafted features or deep learning-generated features. In the classification phase, while some methods are utilized from deep neural networks, the others use Gaussian Mixture Model (GMM), Support Vector Machine (SVM), Random Forest Classifier (RFC) at the backend. We can deeply analyze the methods in the literature into two sub-groups: Handcrafted feature-based methods and Deep learning-generated features-based methods.

A. HANDCRAFTED FEATURE-BASED METHODS

Todisco et al. proposed a new feature that is based on the Constant Q Transform and it is combined with cepstral analysis (CQCC) [4]. Their technique is utilized by GMM to label the audio file as spoof or original. Their results show that the method outperformed all previously reported results on AsvSpoof 2015. In 2017, Alluri et al. employed single-frequency filtering (SFCC) to obtain a spectral and high temporal resolution to detect replay attacks [5]. GMM was used for the classification of handcrafted features denoted by SFCC as the last step. Das et al. combined long-range features with the other known features to improve the detection of spoofing attacks [6]. Their method generated tandem detection cost function to be 0.12 and 0.13 for logical access and physical access datasets respectively. Lavrentyeva et. al. explored various acoustic features as input into their proposed Light CNN architecture [7]. Their experiments showed that the power spectrum contains meaningful information for the detection of spoofed signals. Their technique used log power magnitude spectrum as features and the spectrum was obtained by CQT, FFT, and DCT techniques. Yang et al. developed Log-CQT features combined with multi-layer convolutional neural networks to realize robust performance [8]. Their technique used CNNs with gradient linear units (GLU) for classification purposes. Balamurali et al. examined the effect of different audio features on the effectiveness of the GMM-UBM based detection system [9]. Their work analyzed both audio features (CQCC, LPCC, IMFCC) and learned by autoencoder. Experimental results of the work also designated EER values of the models built with different feature sets. The other study indicated that CQCC based on the magnitude of CQT ignores phase information which can be also changed during the replay process [10]. Their work proposed a CQT-based modified group delay feature (CQTMGD) to capture the phase information of CQT. Multi-branch residual convolutional network was also utilized in their work to classify the input signals. Their results showed that CQTMGD gives better results when compared to the traditional MGD feature. Das et al. found that eCQCC and CQSPIC features are more reliable if you use them as a countermeasure to label the input as authentic or forged [5]. Adiban et al. used CQCC features as input to the autoencoder to obtain more discriminative features [11]. Their method also used various configurations of Siamese networks for classification purposes for the first time. Experimental results indicate that the proposed system improves the baseline works. The other study proposed a new method called subband transform, which represents the signals using subbands. Constant Q equal subband transform, octave subband transform, and mel subband transform are evaluated by their work on the ASVspoof 2015 and ASVspoof2019 logical access datasets [12]. Experimental results indicate that

the proposed subband-based features outperform transforms that are based on full-band transforms. In 2021, Yang et al. proposed a new feature called Modified magnitude-phase spectrum (MMPS) feature to obtain magnitude and phase information from the input signal [13]. Constant-Q-transform is then applied with MMPS to obtain a new handcrafted feature CQT-MMPS. Their results indicated that CQMCC outperformed most commonly used spoofing detection features. Ren et al. used a method to obtain features from the temporal waveform using frequency band calibration via sinc convolution and squeeze-excitation module [14]. Their results showed that the method achieved better generalization capability in the detection of unseen attacks. Aljasem et. al. proposed a method that used sign-modified acoustic local ternary pattern (sm-ALTP) features to determine audio spoofing forgery [15]. Their classifier ensemble approach utilizes a series of weak classifiers and produces a stable classifier. They evaluated their approach to ASVspoof 2019 and VSDC datasets. Dawood et al. suggested a new feature descriptor Center Lop-Sided Local binary patterns (CS-LBP) to represent audio files in the best manner [16]. These features were also fed into the long short-term memory network for the detection of audio forgery. In 2022, Li et al. found that temporal long-term relations and high-frequency information are useful for obtaining artifacts of the spoofed signal [17]. The long-term variable Q transform (L-VQT) is proposed by the authors to catch the clues at the spoofed signal. Modified Densely Connected Convolutional Network (DenseNet) is also used to obtain detection results.

B. DEEP LEARNING GENERATED FEATURES-BASED METHODS

Suggested methods in this field utilize deep learning to detect spoofed audio signals. Kumar et al. used a time-delay shallow neural network (TD-SNN) to detect spoofed audio signals [18]. The method can also handle with variable-length speeches during the testing stage. Their work analyzed GMM on ASVspoof 2019 dataset. Chettri et al. combined the traditional machine learning with deep neural networks to create ensemble models through logistic regression [19]. The authors used large margin cosine loss function (LMCL) and online frequency masking augmentation to increase the learning ability of neural networks [20]. The method is also evaluated on the ASVspoof 2019 logical access (LA) dataset. Zhang et. al. proposed a detection system that used one-class learning to increase its robustness of the method against unseen attacks [21]. They used a specially designed loss function called One Class Softmax (OC-Softmax) to classify the input signals. Modified Resnet-18 architecture is also used by the authors to decide the input, which is the LFCC feature of the audio signal. Li et al. modified Res2Net to enable channel-wise gating mechanism in the connection between feature groups to create a new architecture called Channel-Wise gated Res2Net (CG-Res2Net) [22]. The mechanism used in the method determined channel-wise features according to input. Thus, it could suppress the least related channels. They obtained results on the ASVspoof 2019 dataset.

Xue et. al. utilized physiological-physical feature fusion to detect audio spoofing attacks [23]. A densely connected convolutional network with squeeze and excitation block (SE-DenseNet) and a multi-scale residual neural network with squeeze and excitation block (SE-Res2Net) were used in a combined manner to fuse the features. They also tested their work on the ASVspoof 2019 dataset. Li et al. improved Res2Net to detect audio spoofing attacks in a good manner [24]. Res2Net enables multi-feature scales by modifying ResNet. Res2Net divides the feature maps in each block into multiple channel groups and creates a residual connection between different channel groups. Their results showed that Res2Net outperformed ResNet34 and ResNet50 on physical access (PA) and LA sub-datasets of ASVspoof 2019. In 2022, Ma et al. indicated that filter bank distributions of cepstral features in the frequency domain can affect the system performance [25]. They proposed an improved light convolutional neural network (LCNN) with attention modules (Squeeze and Excitation Block, Convolutional Block Attention Module-CBAM, Dual Attention Network, DANet). They analyzed various attention mechanisms' effects on audio spoof detection. Dua et al. designed three different models and grouped them to analyze the results [26]. MFCC, IMFCC, and CQCC were used as the frontend for all these models.

After we have analyzed the works in the literature in depth, we can now give the details of the proposed method in the next section.

III. PROPOSED METHOD

In this work, we propose a new approach to detecting deep fake audio samples using different characteristics of the audio sample. The input audio signal is converted into the frequency domain using Mel Spectrogram as the first step and then various pattern information from the spectrogram is obtained using specific pattern extraction techniques Local Binary Pattern (LBP), Local Phase Quantization (LPQ), and Gray Level Co-Occurrence Matrix (GLCM). The proposed method trains three deep neural networks (Modified ResNet18 Architecture, the details about which will also be given below) to obtain three different judgments about the input audio signal. Generated scores are then fused as the last step to determine the system's answer. The general architecture of the proposed approach is also given in Fig. 2. As you can see from the figure, the proposed method consists of three parts: Audio visual representation and texture extraction, Training of the Deep Neural Networks, and Fusion of the scores. We will give their details in the following sections.

A. AUDIO VISUAL REPRESENTATION AND PATTERN EXTRACTION

Input audio signal is converted into the frequency domain using the Mel Spectrogram technique as the first step. The method uses Short-Time Fast Fourier Transform (STFT) to generate the corresponding spectrogram of input audio signal

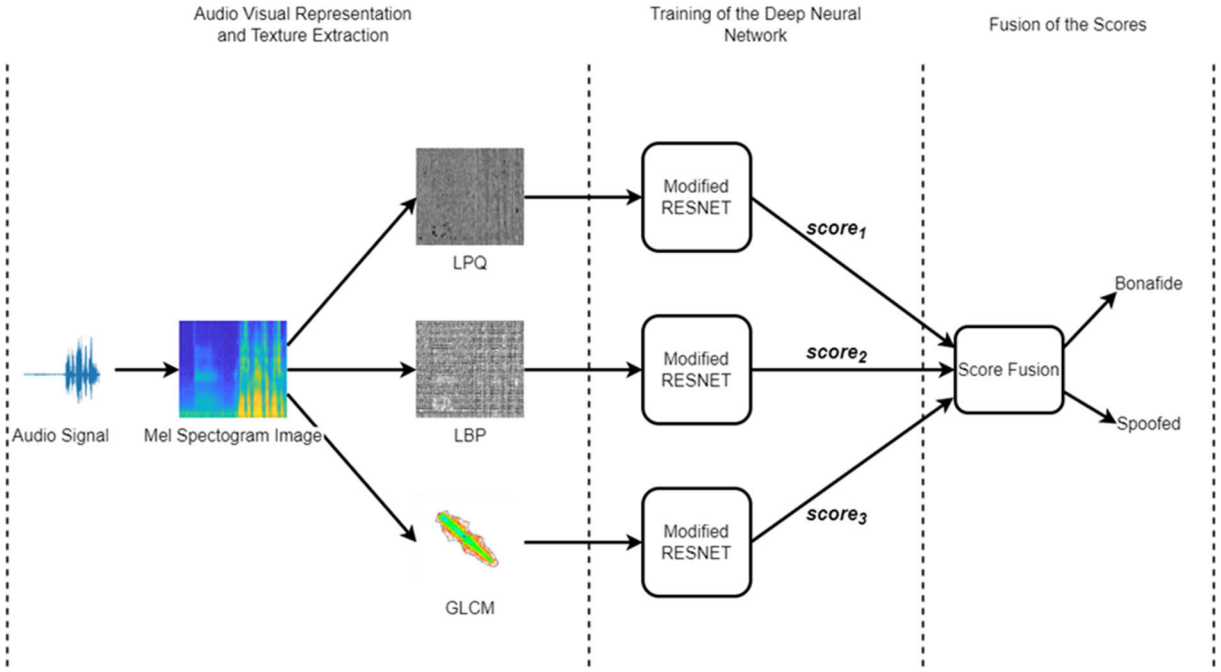


FIGURE 2. General flowchart of the proposed method.

which is divided into frames with 30 ms duration. FFT is realized on all subparts as in (1) after each subframe is multiplied by Hamming window.

$$S(f,t) = \sum_{n=0}^{N-1} w_n x_t(n) \exp(-j2\pi(f/f_s)n)$$
$$f = kf_s/N \tag{1}$$

While Hamming Window function is represented by w_n, x_n corresponds to the original speech. f is the frequency range for $k = 1, 2, \ldots N/2+1$ and N represents the number of samples in each frame. Amplitude, power, phase or log amplitude information of a signal can be used to generate corresponding spectrogram. Chosen frequencies for all different representations are equally apart. However, the human sensory system is more sensitive to low-frequency than high-frequency content. Mel spectrogram uses the properties of the human sensory system to represent the audio in the time-frequency domain. Coefficients of Mel spectrogram $S_{mel}(k,t)$ are calculated using (2)

$$S_{mel}(k,t) = \sum_{l=0}^{L-1} m_k(l) \left|S(l,t)^2\right| \tag{2}$$

L,m_k(l) corresponds to the frequency component number and kth filter of the Mel filter bank respectively. Fig. 3 indicates spectrograms of forged and original speeches.

In the second part of the proposed method, the Mel spectrogram image of the input audio signal will be represented as pattern images. The method utilizes LBP, LPQ, and GLCM techniques to obtain different pattern characteristics of the spectrogram image. After Ojala et al. proposed LBP for texture classification, many researchers have used it in their work to extract pattern information from the image [27]. It is preferred by many researchers because of its simplicity, computational efficiency, coding capability, discriminative power, and robustness against illumination variations. LBP labels each pixel using its neighborhood pixels by thresholding mechanism. It compares the center pixel with the 3×3 neighborhood of it and gives binary values according to the result of the comparison. Eight binary values are used by LBP to code center pixel as a decimal value in $[0 - 255]$ range. LBP code generation for a pixel denoted by p is given in (3).

$$LBP_p = \sum_{x=0}^{X-1} T\left(g_p - g_x\right) \tag{3}$$

where g_p corresponds to the gray level value of the center pixel p, g_x represents the pixels at 3×3 neighborhood of center pixel and T (a) defines the thresholding function. If the value of a is greater than or equal to zero, it returns 1. Otherwise, it returns 0.

Afterward, the proposed method also uses GLCM to obtain different pattern information from the Mel spectrogram image. Relative frequencies of a pair of gray levels at a specified distance d and at a specified angle θ for an image are represented by GLCM matrix. Used distance parameter can be varied from 1 to the size of image and angle parameter can be selected from 0°, 45°, 90° and 135°. GLCM generates different pattern information for varying parameters. Input image must be quantized to decrease the computational complexity of GLCM generation process.

Various GLCMs can also be calculated for varying parameters. Usually, GLCMs are obtained for four different directions, and then the mean of all is calculated to get the final GLCM.

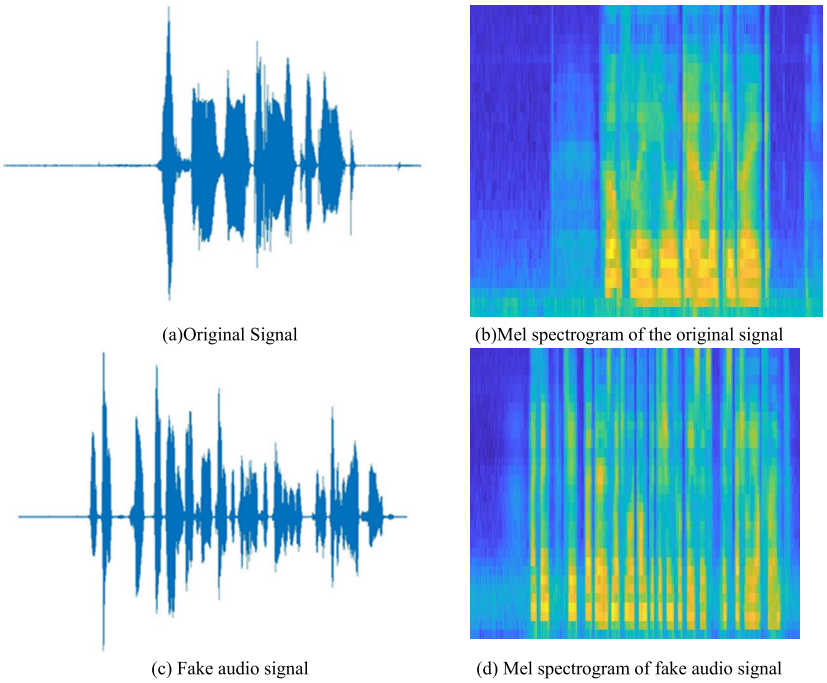


FIGURE 3. Mel spectrogram images of corresponding original and fake audio signals from ASVSpoof 2019.

The last phase of the second part extracts LPQ from the Mel Spectrogram image. Heikkila et al. proposed LPQ to create a blur-independent representation of texture images [28]. Assume that original image and point spread function of blurring operation are denoted by $i(x)$ and $b(x)$ respectively. Frequency domain representation of the convolution operation for blurred image can be expressed as $G(U) = I(u) \cdot B(u)$. $I(U)$ and $B(U)$ represent Discrete Fourier Transform (DFT) of original image and point spread function of blurring operation. The phase angle of $B(u)$ is real-valued all time because its point spread function is symmetric. So, phase angle of $B(u)$ becomes either 0 or π. The method computes the Short-Term Fourier Transform (STFT) in the local neighborhood of N_x for each pixel position. Local spectra are computed using STFT as in (4).

$$\sum_y i(y)\, v(y-x)\, e^{-j2\pi u^T y} \tag{4}$$

Where $v(x)$ is a window function which defines the neighborhood of N_x. If the size of the rectangular region is $N_R \times N_R$, the function returns one for $|x|$ and $|y|$ less then $N_R/2$. Otherwise $v(x)$ returns zero. For each pixel position, LPQ computes four frequency points for small scalar value a as $s_1 = [a, 0]^T$, $s_2 = [0, a]^T$, $s_3 = [a, a]^T$ and $s_4 = [a, -a]^T$. The value of a must satisfy $B(U)$ to be greater than zero. The method calculates a vector with four elements as in (5) for each pixel.

$$F(x) = [F(s_1, x), F(s_2, x), F(s_3, x), F(s_4, x)] \tag{5}$$

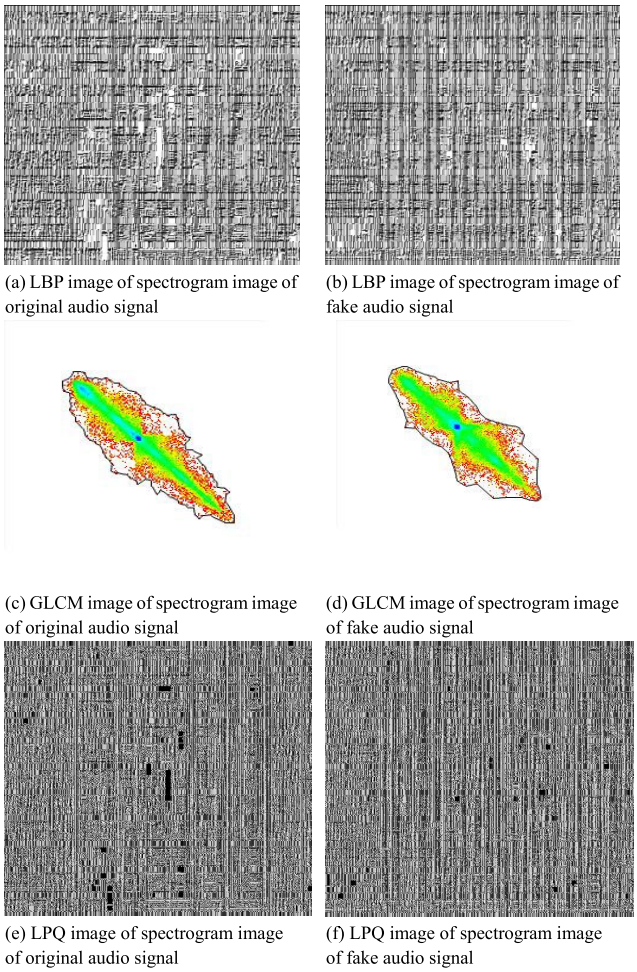
Signs of the real and imaginary parts of each component in $F(x)$ are used to determine the phase information. Each element of $F(x)$ which also have real and imaginary parts, represented by two binary values. For example, if current imaginary part is bigger than or equal to zero, it will be represented by 1. At last, $F(x)$ vector with four elements is represented by eight binary values and it also corresponds to a value in range $[0 - 255]$.

We used original and fake audios from the Asvspoof 2019 LA dataset to show each pattern information, as shown in Figure 4. The created textural images contain additional information, which will contribute positively to the model training. It can be seen from the figure that the texture images extracted from the fake and original audio are different from the texture images extracted from the original audio. The LBP and LPQ textural images obtained from the original audio have more detail than fake ones. The GLCM image of the original audio is longer and narrower than the fake ones. These differences would contribute positively to detecting fake audio from the original audio.

In this part of the proposed method, the input audio is converted into the frequency domain using the Mel spectrogram and then three different pattern extraction techniques are used to extract different patterns of the frequency representation. Those representations will be used to train three DNNs in the next section.

B. TRAINING OF THE DEEP NEURAL NETWORKS

In the second part of the proposed approach, three modified Resnet-18 architectures are learned distinctly using three different pattern-based features, which are calculated in the former step. We modified the network architecture proposed in [30]. The modified architecture uses the deep residual network ResNet-18 but adds a self-attentive temporal pooling



(a) LBP image of spectrogram image of original audio signal

(b) LBP image of spectrogram image of fake audio signal

(c) GLCM image of spectrogram image of original audio signal

(d) GLCM image of spectrogram image of fake audio signal

(e) LPQ image of spectrogram image of original audio signal

(f) LPQ image of spectrogram image of fake audio signal

FIGURE 4. Three types of textural images for original and fake audios.

layer instead of a global average pooling layer. Resnet has recently become a high-performance deep network model used especially in computer vision, speech, and emotion recognition [24], [31], [32].

The more hidden layers in a deep neural network, the lower the performance will be. Resnet neural network model aims to solve this problem. Although deep networks generally learn directly from underlying mapping, the Resnet model performs learning by residual function [32]. Thus, residual mapping instead of underlying mapping is more convenient for optimization, as pushing a residual to zero is easier than fitting it into an underlying mapping [32]. ResNet consists of several residual units. These residual units contain two convolutional layers with 3 × 3 filter sizes. Batch normalization (BN) is performed after each convolution [33]. After the first convolution, the shortcut connection addition operation, and ReLU activation functions are performed.

We extend it by adding two pairs of convolutional layers with a kernel size of 3 × 3, having every 512 kernels. With these additional convolutional layers and the fully connected layer that is featured afterward, the total amount of layers is brought up to 22. The proposed architecture takes as input the texture images generated from the Mel spectrogram image and produces a classification score for each texture image as output. The description of layers with filters, max-pooling function, dropout, and activation function is discussed in the following.

TABLE 1. The details of used architecture.

Layer name	Output size	Building blocks strides, neurons
conv1	113 x 390	9 x 3, 16, stride 3 x 1
conv2_x	113 x 390	$\begin{bmatrix} 3\,x\,3 & 64 \\ 3\,x\,3 & 64 \end{bmatrix}$ x 2, stride 1 x 1
conv3_x	57 x 195	$\begin{bmatrix} 3\,x\,3 & 128 \\ 3\,x\,3 & 128 \end{bmatrix}$ x 2, stride 2 x 2
conv4_x	29 x 98	$\begin{bmatrix} 3\,x\,3 & 256 \\ 3\,x\,3 & 256 \end{bmatrix}$ x 2, stride 2 x 2
conv5_x	8 x 25	$\begin{bmatrix} 3\,x\,3 & 512 \\ 3\,x\,3 & 512 \end{bmatrix}$ x 2, stride 4 x 4
conv6_x	3 x 9	$\begin{bmatrix} 3\,x\,3 & 512 \\ 3\,x\,3 & 512 \end{bmatrix}$ x 2, stride 3 x 3
conv7_x	1 x 9	$\begin{bmatrix} 3\,x\,3 & 512 \\ 3\,x\,3 & 512 \end{bmatrix}$ x 2, stride 1 x1
Self-Attention	512	256
Fully_Connected_1	256	256
Fully_Connected_2	2	2
OC-SoftMax	1	-

The network model contains 20 convolution layers, 2 down sampling layers, two fully connected layers(fc), and a self-attention layer. Self- attention layer was used to process inputs of variable length and give higher coefficients to parts of the input [30]. In this way, all local vectors from the input were combined into a single global vector. The input texture image size of our model is 113 × 390, in addition to the first convolution layer, the convolution kernel size is 9 × 3, and the other layers are 3 × 3. After the attention of the last convolution layer, an eigenvector is created by full connection, then the classification probability is created by OC-Softmax [21]. We used OC-Softmax proposed in [21]. as a loss function. The aim of this loss function is composed of two different margins to compress bonafide speech and separate the spoofing attacks. One-class Softmax (OCSoftmax) is indicated in (6).

$$L_{OCS} = \frac{1}{N}\sum_{i=1}^{N} \log(1 + e^{\alpha(m_{y_i} - w_0 x_i)(-1)^{y_i}}) \qquad (6)$$

where α is a scale factor, w_0 is weight vector, N is the number of samples in a mini batch, x_i and y_i embedding vectors. Two margins ($m_0, m_1 \in [-1; 1], m_0 > m_1$) shows bonafide speech and spoofing attacks, respectively.

Two convolution layers of identical color, as shown in Fig 8, constitute a residual block. Shortcut connections are those skipping two layers (curved arrows in Fig. 8). The shortcut connections build an identity mapping, and the inputs are inserted into the output of the multiple layers. Our model creates an eigenvector containing two probabilities, which are

utilized to show that the input texture image will be included in the class with the highest probability.

C. FUSION OF THE SCORES

Three different score in $[0-1]$ range are obtained from three DNNs at the former step of the algorithm. Weights will be determined using individual scores denoted by $s_{LBP}, s_{GLCM}, s_{LPQ}$. To determine corresponding weights in a correct manner, we used score results from the development set of the LA dataset. The proposed technique started each weight from 0 and increased it by 0.01 for each iteration. At each step, we obtain the weighted sum of scores with new weights and decide according to the weighted sum. The method obtains decisions for all audios in the development dataset using the current weight configuration and records the EER value for the current iteration. At the next step of the algorithm, weights will be updated, and then all samples from the development dataset will be processed again to obtain the current EER. At the end of the algorithm, the proposed approach chooses a specific weight configuration that gives min EER during tests. The corresponding weight determination algorithm is given below.

Algorithm 1 Determining best weight values

Input: $s_{LBP}, s_{GLCM}, s_{LPQ}$
Output: w_1, w_2, w_3

```
minEER = 999
for w_LBP = 0: 0.01: 1
  for w_GLCM = 0: 0.01: 1
    w_LPQ = 1 − (w_LBP + w_GLCM);
    s = w_LBP · s_LBP + w_GLCM · s_GLCM + w_LPQ · s_LPQ;
    EER = DevTest (DevDataset, s);
    if EER < minEER
      minEER = EER;
      w_1 = w_LBP; w_2 = w_GLCM; w_3 = w_LPQ;
    end if
  end for
end for
return w_1, w_2, w_3
```

Where *DevTest* is a function which returns the EER value for the current score values on the development dataset. $s_{LBP}, s_{GLCM}, s_{LPQ}$ represents the score values of LBP based, GLCM based and LPQ based DNNs for development dataset and each of them contains same number of elements as development dataset. Vector s accommodates weighted score values for all samples in the development dataset. Function DevTest calculates EER using the current score values. The fusion algorithm given above will return the best weight values for three architectures. Thus, the proposed system will decide the originality of the input audio using the predetermined weight values with current score values, as in (7).

$$s = w_1 \cdot s_{LBP} + w_2 \cdot s_{GLCM} + w_3 \cdot s_{LPQ} \tag{7}$$

After the last phase of the proposed method, the system will decide the originality of the input audio. In the next section, we will also provide a detailed experimental analysis of the work.

IV. EXPERIMENTAL RESULTS

In this section, we give the details of the experiments to show the performance of the proposed method and to make a fair comparison between the method and similar works in the literature.

A. IMPLEMENTATION DETAILS

The proposed method utilized three different pattern extraction techniques, as indicated in the previous section. LBP-based, GLCM-based, and LPQ-based subparts of the method will be evaluated individually, and then fusion results will be given to show the general performance of the proposed method. We have realized the proposed approach using the following parameters. Our network model was performed with PyTorch2. The model utilizes Adam optimizer with the $\beta 1$ and $\beta 2$ parameters to update the weights in ResNet. $\beta 1$, $\beta 2$ was set to 0.9 and 0.999 respectively. We selected $\alpha = 20$, m0 = 0:9 and m1 = 0:2 for the OC-Softmax as a hyper-parameter in the loss functions. We determined parameters in the loss function using the Stochastic Gradient Descent (SGD). The batch size is set to 16. We initially determined the learning rate as 0.0003 and then reduced it 50% for every 10 epochs [21], [41]. The model is trained on the network for 100 epochs on a Tesla P100-PCIE-16GB. Then, we choose the model with the lowest validation EER for evaluation.

B. DATASETS

ASVSpoof 2019 dataset was used to train and test of the proposed system [34]. And it is also tested using ASVSpoof2021 dataset [46].

ASVSpoof2019 consists of two sub-datasets: Logical Access (LA) and Physical Access datasets. The details of these subsets can be given as follows and the number of audios is given in Table 2. The Logical Access dataset consists of two attack types, voice conversion and speech synthesis attacks, and three different subsets: the Training, Development, and Evaluation datasets. While the training dataset accommodates 25380 audio samples (2580 are bonafide samples and 22800 are spoofed samples), 2540 bonafide samples and 22296 spoofed samples were used to create the development dataset. The evaluation dataset also consists of 7355 bonafide samples and 63882 spoofed samples as indicated in Table 1. Attacks in the training and development section were created using a set of 6 different algorithms (A01-A06), while attacks in the evaluation partition using a set of 13 algorithms (A07-A19). While A01-A04 was created with TTS, A05 and A06 created with two VC approaches in the training and development partitions. For these spoofing attacks the waveform conversion and generation techniques are respectively as follows:

- A01; AR RNN- WaveNet
- A02; AR RNN -WORLD
- A03; FF-WORLD
- A04; CAR-Waveform concat
- A05; VAE- WORLD
- A06; GMM+UBM Spectral filtering -OLA

Seven TTS algorithms (A07-A12 and A16), three VC algorithms (A17-A19), and three TTS+VC algorithms (A13-A15) were used to generate the evaluation partition.

- A07; RNN-WORLD
- A08; AR RNN- Neural source-filter
- A09; RNN- Vocaine
- A10; AR RNN+CNN- WaveRNN
- A11; AR RNN+ CNN- Griffin-Lim
- A12; RNN-WaveNet
- A13; Momentum match- Waveform filtering
- A14; RNN-STRAIGHT
- A15; RNN- WaveNet
- A16; CART-Waveform concat
- A17; VAE-Waveform filtering
- A18; Linear- MFCC vocoder
- A19; GMM-UBM Spectral filtering- OLA

The physical Access (PA) dataset, which contains replay attacks and spoofing scenarios, was created using a fixed microphone in an environment. Sounds propagate, and various obstacles, such as walls and floors, reflect the sound in this scenario. Varying source/receiver positions are considered to simulate room acoustics in a good manner, as indicated in [34].

Physical Access (PA) dataset, which contains replay attacks and spoofing scenario, was created using a fixed microphone which is in an environment. Sounds propagate and various obstacles such as walls and floors reflect the sound in this scenario. Varying source/receiver positions are considered to simulate room acoustics in a well manner as indicated in [34]. Room size S, reverberation time $T60$ and the talker to ASV distance D_s define the noise-free acoustic environment which also accommodates the ASV system. Each parameter gets different values and using these values 27 different configurations are denoted by environment identifiers (EIDs) (aaa, aab, …, ccc). While the training dataset in the PA consists of 54000 samples, the development dataset accommodates 29700 samples. The evaluation dataset also contains 134730 samples in the PA scenario.

TABLE 2. General structure of ASV spoof 2019 LA and PA sets.

	Logical Access			Physical Access	
	Attacks	Bonafide	Spoof	Bonafide	Spoof
Train	A01-A06	2580	22,800	5400	48,600
Dev	A01-A06	2548	22,296	5400	24,300
Eval	A07-A19	7,355	63,882	18090	119,367

The ASVSpoof 2021 dataset was published with three different scenarios. It is stated to use ASVSpoof 2019 as a training and development set. In this study, the method was tested on the ASVSpoof 2021 LA set. The dataset contains audios that database was transmitted across a telephony network such as a PSTN or VoIP, using various codecs, sampling rates and bitrates which are represented by C1-C7 given in Table 3.

TABLE 3. Summary of ASVSpoof 2021 LA evaluation conditions.

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

C. METRİCS

Two metrics were utilized in spoof speech detection. The first of these metrics is EER. In ASV systems, EER is described as the point where the False Acceptance Rate (FAR) equals the False Rejection Rate (FRR). FAR shows the probability of the system admitting a forged record as original, while FRR shows the probability of refusing an original record as forged. The other of these metrics is the minimum normalized tandem detection cost function (min t-DCF). It shows the overall protection rate for ASV systems and ASV performance. Its formula is given in equation (8).

$$min_{t-DCF} = min_{\mathrm{T}_{cm}} \left\{ \frac{C_0 + C_1 P^{cm}_{miss}(\mathrm{T}_{cm}) + C_2 P^{cm}_{fa}(\mathrm{T}_{cm})}{t - DCF_{default}} \right\} \tag{8}$$

where $P^{cm}_{miss}(\mathrm{T}_{cm})$ and $P^{cm}_{fa}(\mathrm{T}_{cm})$ are the miss and false alarm rate of the output of the anti-spoofing system is called a countermeasure (CM) for threshold T_{cm}. C_0, C_1 and C_2 are the coefficients. $t - DCF_{default} = C_0 + \min(C_1, C_2)$.

These metrics show the similarity of the suspicious record with bonafide speech. A low EER and min t-DCF values in ASV systems indicate high system accuracy.

D. ASVSPOOF 2019 LA DATASET RESULTS

In this section, we give the sub-results for development and evaluation dataset results of LA in two subsections. The last part of this section also shows fused results and comparisons with similar works.

1) DEVELOPMENT DATASET RESULTS

In this section, we used the LA development set to show the effectiveness of each branch of the method. All samples from this dataset are converted into the frequency domain using Mel Spectrogram, and then their LBP, GLCM, and LPQ textures are obtained separately. After that, the trained system tested these textured images. Table 4 gives the EER (%) results of each attack for each branch. As we can see from the table, while the LBP branch gives the best EER value for the

A03 attack, the worst EER value is obtained for the samples of the A06 attack. When we look at the results, the best three EER values for this experiment are obtained for the A03, A02, and A05 attack types. WORLD waveform generator is used for speech synthesis to generate A02, A03, and A05 samples. The worst EER for this experiment is obtained for the A06 attack, and samples from this attack type are generated using spectral filtering plus an OLA waveform generator. Thus, we can conclude that the LBP branch gives good results for the samples that are generated using a vocoder; it generates the worst result for the A06 attack type, which uses spectral filtering plus OLA to generate spoofed samples.

When it comes to GLCM branch results, the best case is obtained for the A03 attack; the worst EER value is obtained for the samples of the A06 attack. The best three EER values for this experiment are obtained for the A03, A01, and A02 attack types. WORLD waveform generator and WaveNet are used for speech synthesis purposes to generate A01, A02, and A03 samples. The worst EER for this experiment is obtained for the A06 attack, and samples from this attack type are generated using spectral filtering plus an OLA waveform generator. Thus, the GLCM branch gives good results for the samples that are generated using TTS algorithms; it generates the worst result for the A06 attack type, which uses spectral filtering plus OLA to generate spoofed samples.

LPQ branch gives the best EER value for A03 attack with achieved EER of 0.47% and gives the worst EER value for A06 attack as indicated in Table 4. Results for A01, A02 and A03 attack scenarios are the best three EER values for this experiment. The worst EER for this experiment is obtained for A06 attack and samples from this attack type are generated using spectral filtering plus OLA waveform generator. Thus, we can conclude that LPQ branch gives good results for the samples which are generated using TTS and some specific VC algorithms, it generates the worst result for A06 attack type which uses spectral filtering plus OLA to generate spoofed samples. LPQ branch also gives good results for VC attacks which uses WORLD waveform generator.

TABLE 4. EER (%) results the LBP branch of the method on the development dataset of LA.

Attacks	LBP	GLCM	LPQ
A01	0.8292	2.71299	0.47767
A02	0.15922	3.14413	0.54383
A03	**0.12615**	**1.37303**	**0.47767**
A04	0.94189	4.03949	0.8292
A05	0.47767	7.18362	0.8292
A06	3.72104	15.74028	8.95472

For the second experiment, we tested the performance of the branches for TTS-based and VC-based spoof samples, as given in Table 5. For each branch, Modified Resnet18 architecture was trained using TTS-based spoof samples and bonafide samples from the training dataset of LA and it was tested on the TTS-based spoof samples (A01 to A04) of the development dataset. While the lowest EER was obtained with LPQ for TTS attacks, the lowest EER was obtained with LBP for VC attacks.

TABLE 5. EER (%) results of the branches for Speech synthesis and voice conversion attacks in development set.

Attacks	LBP	GLCM	LPQ
TTS	0.58027	3.06116	**0.51075**
VC	**2.78588**	12.40385	6.93806

We also give the score distribution of each branch for the best and worst scenarios. As you can see from Table 4, the LBP branch achieved the best EER for A03 and the worst EER for A06. Therefore, we plot the score distribution of the samples from these two attack types to support the obtained LBP results in Table 5. While Fig. 5(a) shows the score values of the samples from A03, score values from A06 are also given in Fig. 5(b). Fig. 5 shows that while scores from A03 can be separated into two regions using a threshold value, score values from A06 cannot be divided into two parts in a simple manner. While some samples from bona fide classes are labeled as fake by the system, some samples from spoofed samples are labeled as original, as shown in Fig. 5(b).

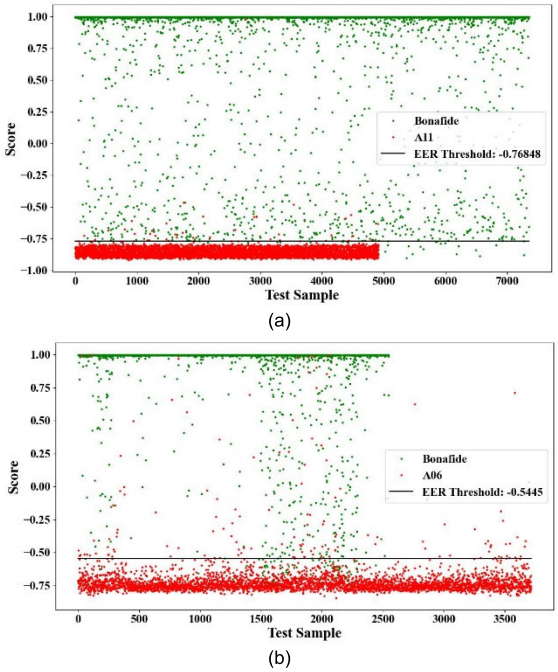


FIGURE 5. Score distribution of samples from A03 and A06 attack types using LBP branch.

In another experiment, we show the score distribution of the GLCM branch for the best and worst scenarios. GLCM branch achieved the best and worst EERs for A03 and A06, respectively. Score distributions of the samples from these two attack types are given in Fig. 6. While Fig. 6(a) shows the score values of the samples from A03, score values from

A06 are also given in Fig 6(b). As you can see from Fig. 6(a), scores from A03 can be separated into two regions using a threshold value, and score values from A06 cannot be simply divided into two parts. While some samples from bona fide classes are labeled as fake by the system, some samples from spoofed samples are labeled as original, as shown in Fig. 6(b).

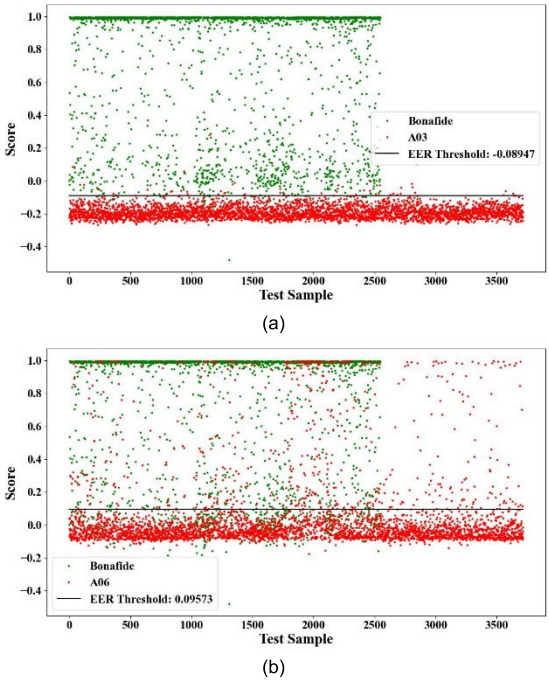


FIGURE 6. Score distribution of samples from A03 and A06 attack types using GLCM branch.

We also show the score distribution of the LPQ branch for two scenarios: The best and worst scenario. While the LPQ branch generates the best EER value for A01 and A03, it gives the worst EER for A06. Score distributions for these scenarios are given in Fig. 7. While Fig. 7(a) shows the score values of the samples from A03, score values from A06 are also given in Fig. 7(b). As you can see from Fig. 7(a), scores from A03 can be separated into two regions using a threshold value; score values from A06 cannot be divided into two parts in a simple manner. While some samples from the bonafide class are labeled as fake from the system, some samples from spoofed samples are labeled as original, as can be seen in Fig. 7(b).

2) EVALUATION DATASET RESULTS

The evaluation dataset results of each branch are discussed in this section. There are 13 attack types for the evaluation dataset of LA. While seven of them denoted by (A07 to A12 and A16) represent TTS-based attack, three of them denoted by (A17 to A19) accommodate VC-based samples. The remaining three systems (A13 to A15) contain spoof samples that were created using the VC-TTS-based technique. We used bonafide and spoof samples of the training dataset of LA to train the modified Resnet18 architecture at each branch and then we tested the training system on the

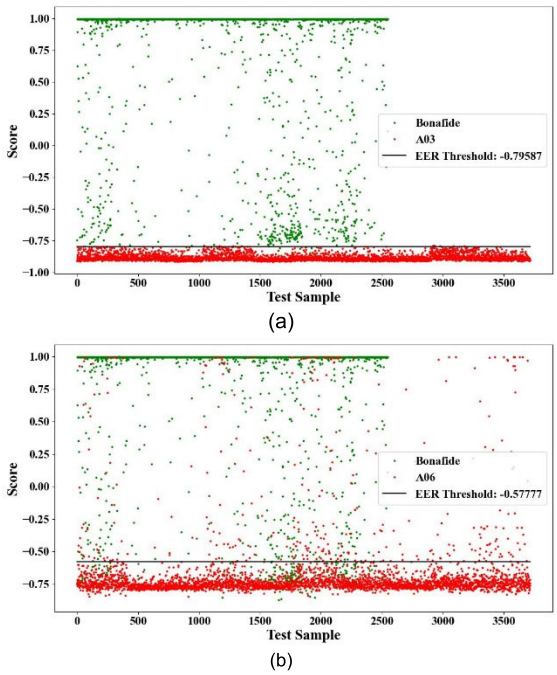


FIGURE 7. Score distribution of samples from A03 and A06 attack types using LPQ branch.

evaluation dataset. Obtained results are also given in Table 6. As you can see from the table LBP branch achieved the best three EER for A09, A13, and A15 attack types. It means that the LBP branch worked well on TTS-VC-based and TTS-based spoof samples when compared to VC-based spoof samples.

GLCM branch achieved the best three EER for A11, A09, and A13 attack types. It means that the GLCM branch worked well on TTS-VC-based and TTS-based spoof samples when compared to VC-based spoof samples.

LPQ branch gives the best EER value for A03 attack with achieved EER of 0.47% and gives the worst EER value for A06 attack.

In the next experiment, we tested the performance of the branches for TTS-VC, TTS and VC based spoof samples for each branch. The EER results are given in Table 6. The lowest results have been given with bold. The table indicates that the LBP branch generates the worst three EER for VC-based spoof samples. When we consider the performance of the LBP branch per attack type group, the LBP branch achieved the best EER of 0.31% for TTS-VC-based spoof samples. The same experiment also shows that the LBP branch does not work well on the VC-based spoof samples. Overall, the LBP branch achieved an EER of 2.27% and a min t-DCF of 0.06 on the evaluation dataset. The evaluation dataset contains unseen attacks during the training stage. However, the LBP branch of the proposed method gives an EER of 2.27% even if it is encountered with unseen attacks.

Table 7 also indicates that the GLCM branch generates the worst EER for VC-based spoof samples by achieving an EER of 26.6%. When we consider the performance of the GLCM

TABLE 6. General performance of the LBP branch of the method on the evaluation dataset of LA.

Attacks	LBP	GLCM	LPQ
A07	0.89637	5.14396	0.62814
A08	0.34293	2.29865	0.5059
A09	0.06452	0.71642	**0.47196**
A10	1.05935	6.83825	0.93712
A11	0.48893	0.66888	0.75717
A12	0.44819	5.94187	0.75717
A13	0.30218	1.85047	0.91335
A14	0.40744	7.5309	0.58739
A15	0.20372	14.04319	0.75717
A16	0.40744	7.36792	1.16461
A17	5.10321	18.90194	10.64444
A18	2.56009	36.69361	10.9636
A19	4.76029	20.13107	6.30857

branch for each attack type, the GLCM branch achieved the best EER of 4.67% for TTS-based spoof samples. GLCM branch does not work well on the VC-based spoof samples with an achieved EER of 26.5%.

The LPQ branch achieved the best EER of 0.75% for TTS-based spoof samples. When we look at the obtained results for VC-based attack types, we can conclude that the LPQ branch does not work well on the VC-based spoof samples. Overall, the LPQ branch achieved an EER of 4.44%.

TABLE 7. EER (%) results of the branches for speech synthesis, voice conversion TTS-VC based attacks.

Attacks	LBP	GLCM	LPQ
TTS-VC	**0.31576**	8.45444	0.76056
TTS	0.62232	**4.67589**	**0.75037**
VC	4.34945	26.59582	9.76165

We also want to show the score distribution of the LBP branch for the best and worst scenarios in the evaluation dataset. Table 8 indicates that the LBP branch achieved the best EER for A09 and it also achieved the worst EER for A17. Therefore, we plot the score distribution of the samples from these two attack types to support the obtained results in Table 8. While Fig. 8(a) shows the score values of the samples from A09, score values from A17 are also given in Fig 8(b). Fig 8 shows that while scores from A09 can be separated into two regions using a threshold value, score values from A17 cannot be simply divided into two parts. While some samples from bona fide classes are labeled as fake from the system, some samples from spoofed samples are labeled as original as can be seen in Fig. 8(b).

When it comes to score distribution of GLCM branch, the score values of the samples from A11, score values from A18 are given in Fig. 9(a) and (b). It is seen that while scores from A11 can be separated into two regions using a threshold value, score values from A18 cannot be simply divided into two parts. While some samples from the bona fide class are labeled as fake from the system, some samples from spoofed samples are labeled as original as can be seen in Fig. 9(b).

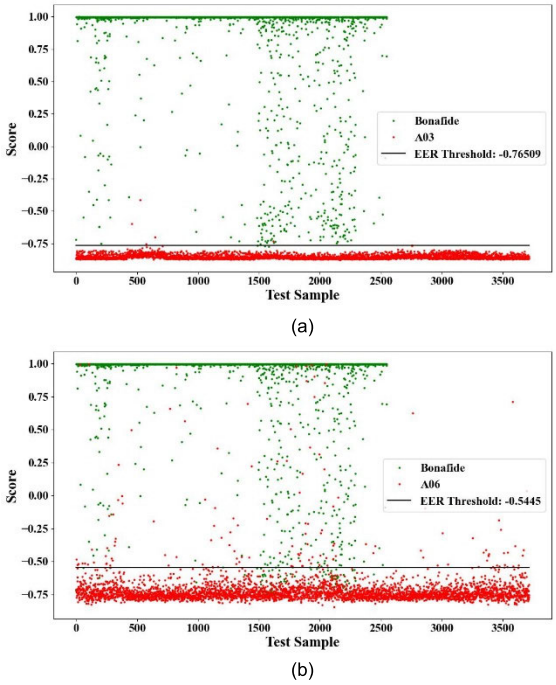


FIGURE 8. Score distribution of samples from A09 and A17 attack types using LBP branch.

Score distribution for the LPQ branch for the best and worst scenario in the evaluation dataset is also given in Fig. 10. The LPQ branch achieved the best and worst EER for A09 and A18 respectively. Obtained results were also supported using score distributions of the samples from these two attack types. While Fig. 10(a) shows the score values of the samples from A09, score values from A18 are also given in Fig 10(b). Fig 10 shows that while scores from A11 can be separated into two regions using a threshold value, score values from A18 cannot be simply divided into two parts. While some samples from bona fide classes are labeled as fake from the system, some samples from spoofed samples are labeled as original as can be seen in Fig. 10(b).

3) ABLATION STUDIES

In this section, performance evaluations of the use of different Resnet models and the fusion of branches are given. Firstly, it analyzes the performance of the ResNet18, ResNet34, and ResNet50 models to decide which is more suitable for the proposed detection system. It is known that the number of layers and parameters increase, respectively, in these three models [32]. Although more parameters provide greater flexibility and accuracy of the model, they increase the risk of overfitting and causing complexity. For the proposed system to be suitable for adaptation to real-time systems, a method that can provide rapid test results with fewer parameters is more appropriate than large models. The number of learnable parameters of the three modified Resnet models that provide information about model capacity and complexity are as in Table 8. As can be seen in the table below, ResNet18 has lower parameters.

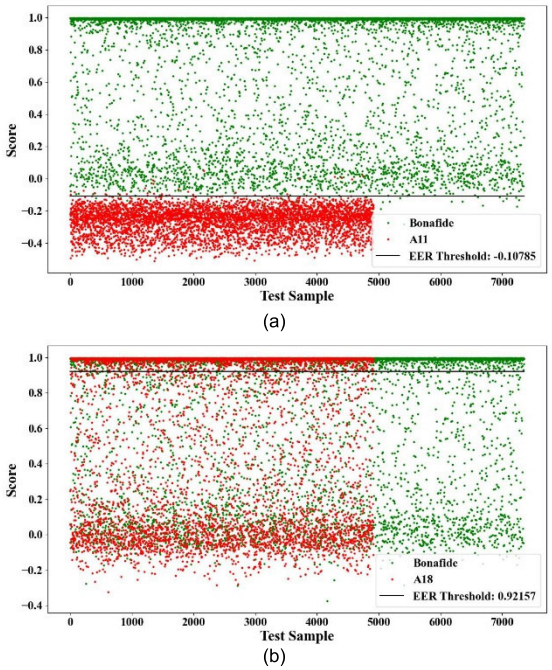


FIGURE 9. Score distribution of samples from A11 and A18 attack types using GLCM branch.

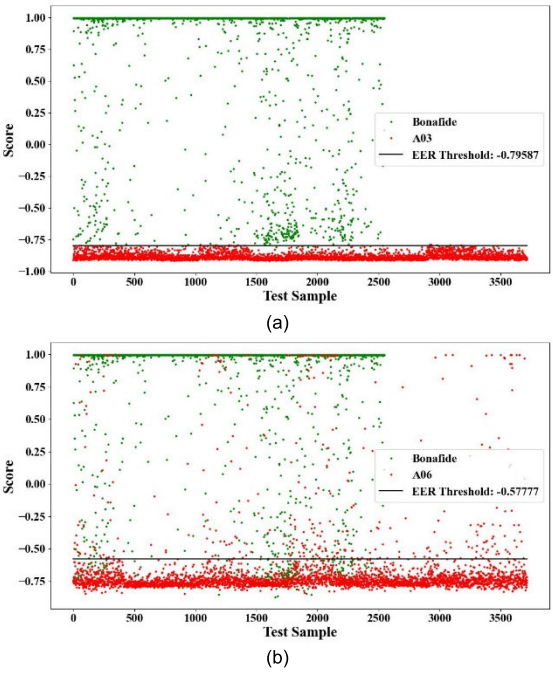


FIGURE 10. Score distribution of samples from A11 and A18 attack types using LPQ branch.

We tested LBP branch results on each attack type in the LA Eval set. The EER (%) and $\boldsymbol{min_{tdcf}}$ results are given in Table 9. For scenarios A08-09, A11-12, and A15-16, mostly unseen and produced by TTS attacks, the results of ResNet18 are better than those given in bold. For the other scenarios, which are generally VC attacks, the results of Resnet18 are the second best. For these reasons, we proposed to use the modified Resnet18 model in the detection system.

TABLE 8. EER (%) results the LBP branch of the method on the development dataset of LA.

Models	Number of learnable parameters
ResNet18	12450290
ResNet34	22558450
ResNet50	28323058

TABLE 9. EER (%) and $\boldsymbol{min_{tdcf}}$ results for each attack types of LA evalset of each models.

	ResNet-18		ResNet-34		ResNet-50	
	EER %	min_{tdcf}	EER %	min_{tdcf}	EER %	min_{tdcf}
A07	6.48	0.17	5.02	0.13	12.04	0.32
A08	**0.61**	**0.01**	1.32	0.13	1.56	0.04
A09	**0.09**	**0.01**	1.01	0.06	0.13	0.01
A10	7.18	0.20	2.85	0.08	13.73	0.37
A11	**1.44**	**0.04**	3.76	0.10	3.00	0.07
A12	**1.44**	**0.03**	4.71	0.11	3.94	0.10
A13	4.71	0.13	2.28	0.05	9.84	0.26
A14	5.18	0.13	1.93	0.06	8.60	0.23
A15	**1.44**	**0.03**	3.35	0.09	3.80	0.09
A16	**2.68**	**0.06**	3.72	0.08	5.39	0.12
A17	12.41	0.69	6.93	0.19	14.32	0.69
A18	13.59	0.68	5.73	0.38	15.97	0.75
A19	12.82	0.45	4.92	0.13	22.73	0.740

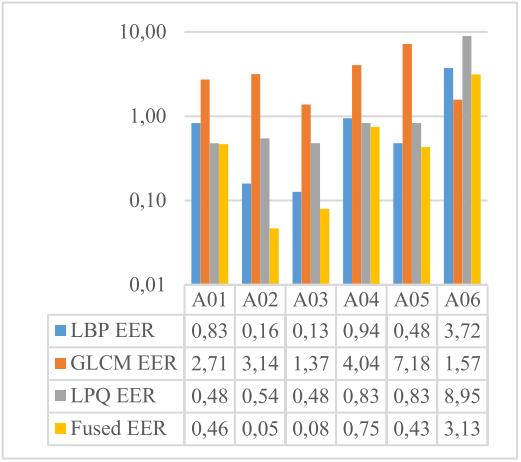
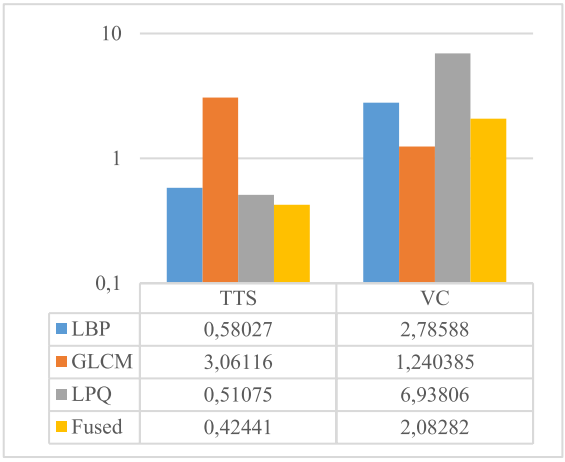


FIGURE 11. General performance of the fused system on the development dataset of LA.

Secondly, the fused system was tested using the samples from the development set of LA for each attack type. Scores from three different branches are fused to decide about the input audio. The EER results of each branch and fused system are given in Figure 11. For each attack type, the EER results of the fused system are better than all branch results. The fused system provides the best EER value for the A02 attack, with an achieved EER of 0.05%, and gives the worst EER value for the A06 attack, with 3.1%. Thus, the fused branch offers good results for the samples that are generated using TTS; it generates the worst result for the A06 attack type, which uses spectral filtering plus OLA to generate spoofed samples.

The fused system was also trained using TTS and VC-based spoof samples of LA separately and it was tested on the TTS and VC-based spoof samples of the development



	TTS	VC
LBP	0,58027	2,78588
GLCM	3,06116	1,240385
LPQ	0,51075	6,93806
Fused	0,42441	2,08282

FIGURE 12. General performance of the fused system for Speech synthesis and Voice conversion attacks.

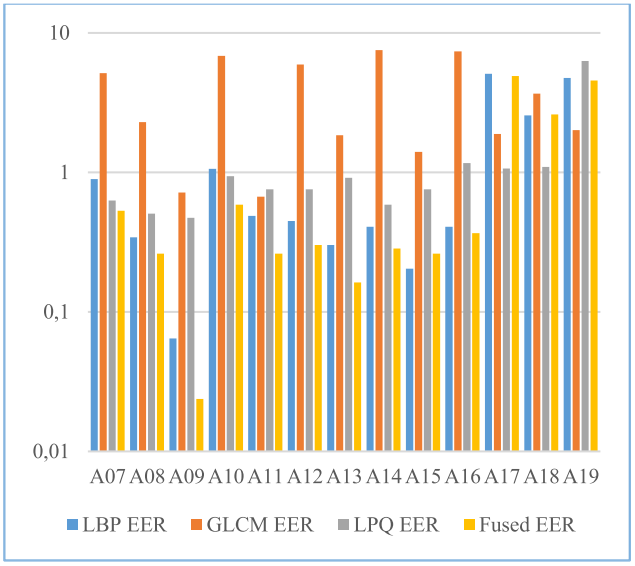
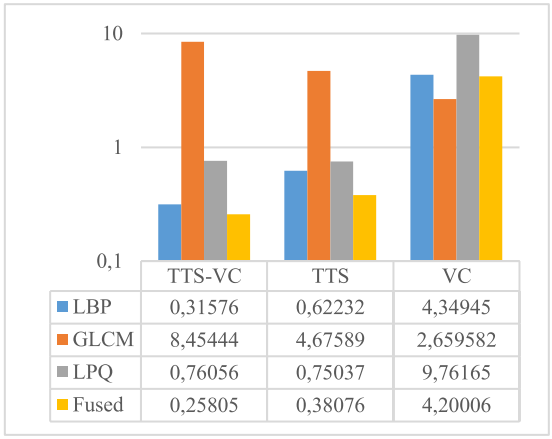


FIGURE 13. General performance of the fused system on the evaluation dataset of LA.

dataset. Original samples from the training class were also used to train the system. Fig. 12 indicates that the fused system achieved an EER of 0.4% for TTS-based samples. As you can see from the figure, it achieved an EER of 2.08% for VC-based samples. The fused system gives better results during the detection of TTS-based samples as you can see from the table. Overall, the system achieved an EER of 1.17% and a min t-DCF of 0.03 on the development dataset.

The evaluation dataset was also used to test the fused system. Scores from three different branches are fused to obtain the decision about the input audio. From Fig 13, it is seen that the best two EERs were achieved at the A09 and A13 attack types.

The fused system worked well on TTS-VC-based and TTS-based spoof samples when compared to VC-based spoof samples as shown in Figure 14. It is shown that fused system gives the worst EER for VC-based spoof samples by achieving an EER of 4.2 %. It achieved the best EER of 0.25% for



	TTS-VC	TTS	VC
LBP	0,31576	0,62232	4,34945
GLCM	8,45444	4,67589	2,659582
LPQ	0,76056	0,75037	9,76165
Fused	0,25805	0,38076	4,20006

FIGURE 14. General performance of the fused system for Speech synthesis, Voice conversion TTS-VC based attacks.

TABLE 10. EER and min_{t_dcf} values for all branches and fused system on the evaluation and development datasets.

Development Dataset	LBP Branch	GLCM Branch	LPQ Branch	Fused System
EER %	1.8	7.38	4.15	**1.17**
min_{t_dcf}	0.05	0.21	0.11	**0.03**

Evaluation Dataset	LBP Branch	GLCM Branch	LPQ Branch	Fused System
EER %	2.27	12.25	4.44	**2.14**
min_{t_dcf}	0.06	0.29	0.1	**0.06**

TABLE 11. The name of attacks such that our system achieved best and worst EERs.

	LBP	GLCM	LPQ	Fused System
Attack Type which gives Best EER (Development Dataset)	A03	A03	A03	A02
Attack Type which gives Best EER (Evaluation Dataset)	A09	A11	A09	A09
Attack Type which gives Worst EER (Development Dataset)	A06	A06	A06	A06
Attack Type which gives Worst EER (Evaluation Dataset)	A17	A18	A18	A19

TTS-VC-based spoof samples. And all fused results are better than each individual branch. When we look at the obtained results for VC-based attack types, we can conclude that system does not work well on the VC-based spoof samples. The evaluation dataset of AsvSpoof 2019 contains unseen attacks which are not encountered during the training stage. In this experiment, we test the system using unseen attacks and we can conclude that it works well even if it is encountered with unseen attacks.

We also give obtained EER and min_{t_dcf} values for all branches for the development and evaluation dataset as the last experiment for this section. Table 10 shows all EER

TABLE 12. Evaluation results of the studies in the literature on the LA while P indicates primary systems, S denotes the single systems.

System	min_{tdcf}	EER %	System
Proposed GLCM Branch	0.29	12.25	S
Proposed LPQ Branch	0.10	4.44	S
Proposed LBP Branch	0.06	2.27	S
Proposed Fused Approach (GLCM+LPQ)	0.10	4.24	P
Proposed Fused Approach (GLCM+LBP)	0.06	2.21	P
Proposed Fused Approach (LPQ+LBP)	0.06	2.16	P
Proposed Fused Approach (GLCM+LPQ+LBP)	**0.06**	**2.14**	P

TABLE 13. Evaluation results of the studies in the literature on the LA while P indicates primary systems, S denotes the single systems.

System	t-DCF	EER %	System
T05 [3]	0.0069	0.22	P
T45 [7]	0.0510	1.84	P
T60 [18]	0.0755	2.64	P
T24 [3]	0.0953	3.45	P
T50 [30]	0.1671	3.56	P
FFT SENet dual band fusion [39]	0.0498	1.56	P
FFT-VAD SENet dual band fusion [39]	0.10	3.5	P
CQT-1__100-ResMax [38]	0.06	2.19	P
GMM Fusion [36]	0.074	2.92	P
SVM Fusion (Polynomial Kernel) [36]	0.074	2.92	P
Multinomial logistic regression fusion [36]	0.11	4.50	P
BestGMM + LCNN + LogSpec + RLogSpec [40]	0.07	2.57	P
LCNN + LogSpec + RLogSpec [40]	0.18	9.67	P
BaseLine1 CQCC + GMM [4]	0.23	9.57	S
BaseLine2 LFCC + GMM [4]	0.21	8.09	S
sm-ALTP-Asymmetric Bagging [15]	0.13	5.22	S
CQCC-DNN, [6]	0.3	12.79	S
LFCC-DNN, [6]	0.23	9.65	S
CQSPIC – DNN, [6]	0.18	7.81	S
CQSPIC – GMM, [6]	0.16	7.74	S
CQ-EST-DNN, [12]	0.23	11.79	S
DF-MST-DNN, [12]	0.27	10.91	S
CQ-OST-DNN, [12]	0.18	8.03	S
Spec + ResNet + CE [35]	0.27	9.68	S
MFCC + ResNet + CE [35]	0.2	9.33	S
CQCC + ResNEt + CE [35]	0.21	7.69	S
Spec + LCGRNN + GKDE-Softmax [36]	0.08	3.77	S
Spec + LCGRNN + GKDE-Triplet [36]	0.07	3.03	S
LFCC & Face + SE-DenseNet + log-softmax (Xue, Hao, Wang, et. al. 2021)	0.07	2.82	S
CQT – MMPS – ResNet [13]	0.11	3.72	S
CQT – MMPS – LCNN [13]	0.17	5.99	S
Proposed Fused Approach (GLCM+LPQ+LBP)	**0.06**	**2.14**	**P**

values with corresponding min_{t_dcf} values for each branch and fused system on the development and evaluation datasets. As you can see from the table, the fused system achieved the best EER when compared to other branches. Table also shows all EER values with corresponding min_{t_dcf} values for each branch and fused system on the evaluation dataset. The fused system achieved the best EER when compared to other branches as you can see from the table. Overall, the fused system achieved 1.17% and 2.14% EERs for development and evaluation datasets respectively. We also give the name of attacks such that our system achieved the best and worst EERs as in Table 11. As you can see from the table, all branches and fused systems achieved the worst EER for the A06 attack type on the development dataset. All branches and fused system also give the worst EER for the VC-TTS-based spoof samples on the evaluation dataset (A17, A18, A19). While only the GLCM branch achieved the best EER for A11 attack types on the evaluation dataset, other branches and fused systems give the best EER for the A09 attack type. While the three branches achieved the best EER for A03, the fused system gives the best performance for A02 as you can see in Table 11.

TABLE 14. EER and min_{t_dcf} values for all branches and fused system on the evaluation and development datasets.

Development Dataset	EER %	min_{t_dcf}
LBP Branch	18.39	0.52
GLCM Branch	29.06	0.79
LPQ Branch	24.40	0.48
Fusion (LBP + GLCM)	16.55	0.46
Fusion (LBP + LPQ)	17.85	0.48
Fusion (LPQ + GLCM)	23.28	0.49
Fusion (LBP + GLCM + LPQ)	**16.07**	**0.44**
Evaluation Dataset	**EER %**	min_{t_dcf}
LBP Branch	20.31	0.61
GLCM Branch	32.03	0.84
LPQ Branch	24.93	**0.51**
Fusion (LBP + GLCM)	19.72	0.59
Fusion (LBP + LPQ)	19.47	0.56
Fusion (LPQ + GLCM)	25.40	0.53
Fusion (LBP + GLCM + LPQ)	**18.82**	0.55

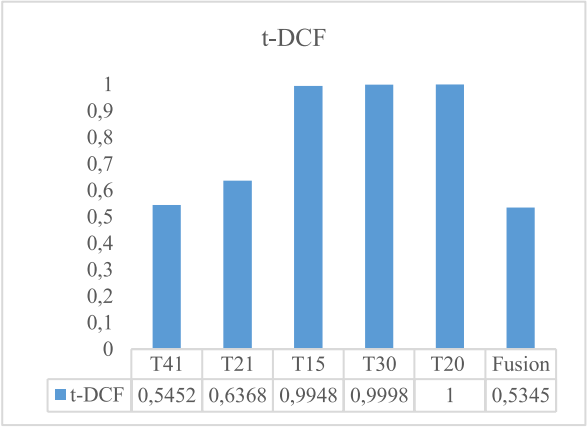


FIGURE 15. Evaluation results of the studies in the literature on the PA-Eval set.

TABLE 15. Comparison of EER (%) performance pooled over attacks on ASVSpoof 2021 LA.

	C1	C2	C3	C4	C5	C6	C7	Pooled
CQCC-GMM [4]	10.57	14.76	20.58	11.61	13.58	14.01	11.21	15.62
LFCC-GMM [42]	12.72	21.21	35.55	15.28	18.76	18.46	12.73	19.30
LFCC-LCNN [43]	6.71	8.89	12.02	6.34	9.25	11.00	6.66	9.26
RawNet2[44]	5.84	6.59	16.72	6.41	6.33	10.66	7.95	9.50
UR-AIR [45]	4.73	4.47	6.20	5.06	4.71	5.18	5.82	5.46
LBP	4.0	6.43	4.36	4.04	4.59	4.51	2.85	4.85
LPQ	6.54	8.91	7.07	6.41	5.9	6.43	4.62	7.69
GLCM	12.93	12.98	12.76	12.99	8.87	8.62	9.02	12.95
Fusion	**3.41**	**5.94**	**3.7**	**3.67**	**3.94**	**4.0**	**2.39**	**4.22**

TABLE 16. Comparison of min_{t_dcf} performance pooled over attacks on ASVSpoof 2021 LA.

	C1	C2	C3	C4	C5	C6	C7	Pooled
CQCC-GMM [4]	0.2858	0.5116	0.6183	0.3715	0.4703	0.4791	0.2935	0.4974
LFCC-GMM [42]	0.3615	0.6334	0.8307	0.4409	0.5819	0.6061	0.3661	0.5758
LFCC-LCNN [43]	0.2186	0.3354	0.4286	0.2218	0.3505	0.4217	0.2166	0.3445
RawNet2[44]	0.2073	0.3584	0.6367	0.2387	0.3489	0.4959	0.2908	0.4257
UR-AIR [45]	0.1902	**0.2911**	0.3720	0.2104	0.2996	0.3618	0.2186	0.3094
LBP	0.1909	0.3611	0.3820	0.1938	0.2585	0.3045	0.1305	0.3252
LPQ	0.2148	0.3420	0.3905	0.1988	**0.2124**	**0.2785**	0.1206	0.3216
GLCM	0.3528	0.4592	0.5010	0.3747	0.3062	0.3546	0.2342	0.4497
Fusion	**0.1737**	0.3346	**0.3645**	**0.1713**	0.2322	0.2817	**0.1179**	**0.3016**

For the final comparative overview, the performance of each branch and each fusion of two branches (GLCM+LPQ, GLCM+LBP, LPQ+LBP) and fusion of three branches (LPQ+LBP+GLCM) are analyzed. The average EER and min_{tdcf} results of the approaches are given in Table 12. Using the proposed approach LBP-based approach also gives the better EER value than LPQ and GLCM branch results. When we look at the fusion results of two systems (GLCM+LPQ, GLCM+LBP, LPQ+LBP), the EER result of LPQ+LBP is the best result of them. Although GLCM branch results were the worst, it was observed that even in the LBP+LPQ fusion case, some spurious sounds could not be detected, while these sounds were detected with GLCM. Therefore, GLCM+LPB+LPQ triple fusion result is better with 2.14 EER. We can say that fusion of the scores improves the EER value of the proposed approach.

4) COMPARISON WITH SIMILAR WORKS IN THE LITERATURE

In this section, we give the evaluation dataset results of LA and for the proposed approach and similar works in the literature to further evaluate the effectiveness of the method. While primary systems (P) use score fusion approaches, single systems (S) utilize one score value to evaluate the originality of input audio. The results of the works in literature and our final fused system are given in Table 13. We also selected the top five teams amongst the fifty best teams of the LA scenario denoted by T05, T24, T45, T50, and T60. Min t-DCF scores and EER values are used for comparison. While the lowest performing studies were [6] and [10], with the proposed fused approach, superior and satisfactory results have been obtained in many studies.

E. ASVSPOOF 2019 PA DATASET RESULTS

We also performed the same experiment for PA scenario the EER and $\min_{t\_dcf}$ results are given in Table 14. Figure 17 shows the comparison with T41, T21, T15, T30, T20 primary methods results given in [3]. Min t-DCF scores are used for comparison. The proposed approach gives the smallest EER with a 0.5345 value as shown in Fig. 15 between the primary systems. When the proposed method is compared to both primary and single systems, it is observed that it gives superior performance in both LA and PA eval sets.

F. ASVSPOOF 2021 LA DATASET RESULTS

In this section, the comparative test results on ASVSpoof 2021 LA dataset for each codec condition (C1-C7).

The comparison with the state-of-the-arts with the four baseline systems CQCC-GMM [4], LFCC-GMM [42], LFCC-LCNN [43], RawNet2 [44] and UR-AIR [45]. The fusion system has better performance than these baseline systems and UR-AIR [45] with lowest EER on all conditions as seen in Table 15. We also present min_{t_dcf} performance given in Table 16. Only for C2, C5 and C6 scenarios, although the metric result of our method is slightly higher than the best results, it can be said that the best results for other cases are obtained with the proposed method.

V. CONCLUSION

We proposed a novel multi–Pattern Features based audio spoof detection scheme using the modified ResNet architecture and OC-Softmax layer to detect various LA and PA spoofing attacks. Through the proposed network, we extracted LBP, GLCM, and LPQ images from the Mel

spectrogram and gave each of them into modified ResNet architecture. At the last step of each network, we used OC-Softmax to obtain a score for the current pattern image and then the method fuses three scores to label the input audio. Experimental results on the ASVspoof 2019 and ASVspoof 2021 corpus show that the proposed method achieves better results in the challenges of ASVspoof 2019 than state-of-the-art methods. The proposed model improves the tandem decision cost function and equal error rate scores by 0.06% and 2.14%, respectively, in the logical access scenario and the tandem decision cost function scores by 0.5345% in the physical access scenario. These results were obtained from the evaluation set of ASVspoof corpus containing the data of unseen speakers. Despite the data of unseen speakers, we also obtained superior results from the evaluation set of the ASVspoof 2019 and ASVspoof 2021. We will improve the performance of our method on different voice spoofing datasets in our future work.

REFERENCES

[1] N. Evans, T. Kinnunen, and J. Yamagishi, "Spoofing and countermeasures for automatic speaker verification," presented at the 14th Annu. Conf. Int. Speech Commun. Assoc. (INTERSPEECH), Lyon, France, Aug. 2013.

[2] Z. Wu, T. Kinnunen, N. Evans, J. Yamagishi, C. Hanilçi, M. Sahidullah, and A. Sizov, "ASVspoof 2015: The first automatic speaker verification spoofing and countermeasures challenge," presented at the 15th Annu. Conf. Int. Speech Commun. Assoc. (INTERSPEECH), Dresden, Germany, Sep. 2015.

[3] M. Todisco, X. Wang, V. Vestman, M. Sahidullah, H. Delgado, A. Nautsch, J. Yamagishi, N. Evans, T. H. Kinnunen, and K. A. Lee, "ASVspoof 2019: Future horizons in spoofed and fake audio detection," presented at the 19th Annu. Conf. Int. Speech Commun. Assoc. (INTERSPEECH), Graz, Austria, Sep. 2019.

[4] M. Todisco, H. Delgado, and N. Evans, "A new feature for automatic speaker verification anti-spoofing: Constant Q cepstral coefficients," in *Proc. Speaker Lang. Recognit. Workshop (Odyssey)*, Jun. 2016, pp. 283–290.

[5] K. N. R. K. R. Alluri, S. Achanta, S. R. Kadiri, S. V. Gangashetty, and A. K. Vuppala, "Detection of replay attacks using single frequency filtering cepstral coefficients," in *Proc. 17th Annu. Conf. Int. Speech Commun. Assoc. (Interspeech)*, Stockholm, Sweden, Aug. 2017, pp. 2596–2600.

[6] R. K. Das, J. Yang, and H. Li, "Long range acoustic features for spoofed speech detection," in *Proc. 19th Annu. Conf. Int. Speech Commun. Assoc. (INTERSPEECH)*, Graz, Austria, Sep. 2019, pp. 1058–1062.

[7] G. Lavrentyeva, S. Novoselov, A. Tseren, M. Volkova, A. Gorlanov, and A. Kozlov, "STC antispoofing systems for the ASVspoof2019 challenge," in *Proc. 19th Annu. Conf. Int. Speech Commun. Assoc. (INTERSPEECH)*, Graz, Austria, Sep. 2019, pp. 61–67.

[8] 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," in *Proc. 19th Annu. Conf. Int. Speech Commun. Assoc. (INTERSPEECH)*, Graz, Austria, Sep. 2019, pp. 1038–1042.

[9] B. T. Balamurali, K. E. Lin, S. Lui, J.-M. Chen, and D. Herremans, "Toward robust audio spoofing detection: A detailed comparison of traditional and learned features," *IEEE Access*, vol. 7, pp. 84229–84241, 2019.

[10] X. Cheng, M. Xu, and T. F. Zheng, "A multi-branch ResNet with discriminative features for detection of replay speech signals," *APSIPA Trans. Signal Inf. Process.*, vol. 9, no. 1, p. e28, 2020.

[11] M. Adiban, H. Sameti, and S. Shehnepoor, "Replay spoofing countermeasure using autoencoder and Siamese networks on ASVspoof 2019 challenge," *Comput. Speech Lang.*, vol. 64, Nov. 2020, Art. no. 101105.

[12] R. K. Das, J. Yang, and H. Li, "Assessing the scope of generalized countermeasures for anti-spoofing," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2020, pp. 6589–6593.

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

[14] Y. Ren, W. Liu, D. Liu, and L. Wang, "Recalibrated bandpass filtering on temporal waveform for audio spoof detection," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Anchorage, AK, USA, Sep. 2021, pp. 3907–3911.

[15] M. Aljasem, A. Irtaza, H. Malik, N. Saba, A. Javed, K. M. Malik, and M. Meharmohammadi, "Secure automatic speaker verification (SASV) system through sm-ALTP features and asymmetric bagging," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 3524–3537, 2021.

[16] H. Dawood, S. Saleem, F. Hassan, and A. Javed, "A robust voice spoofing detection system using novel CLS-LBP features and LSTM," *J. King Saud Univ. Comput. Inf. Sci.*, vol. 34, no. 9, pp. 7300–7312, Oct. 2022.

[17] J. Li, H. Wang, P. He, S. M. Abdullahi, and B. Li, "Long-term variable Q transform: A novel time-frequency transform algorithm for synthetic speech detection," *Digit. Signal Process.*, vol. 120, Jan. 2022, Art. no. 103256.

[18] M. G. Kumar, S. R. Kumar, M. S. Saranya, B. Bharathi, and H. A. Murthy, "Spoof detection using time-delay shallow neural network and feature switching," in *Proc. IEEE Autom. Speech Recognit. Understand. Workshop (ASRU)*, Dec. 2019, pp. 1011–1017.

[19] B. Chettri, D. Stoller, V. Morfi, M. A. M. Ramírez, E. Benetos, and B. L. Sturm, "Ensemble models for spoofing detection in automatic speaker verification," presented at the 19th Annu. Conf. Int. Speech Commun. Assoc. (INTERSPEECH), Graz, Austria, Sep. 2019.

[20] T. Chen, A. Kumar, P. Nagarsheth, G. Sivaraman, and E. Khoury, "Generalization of audio Deepfake detection," in *Proc. Speaker Lang. Recognit. Workshop (Odyssey)*, Nov. 2020, pp. 132–137.

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

[22] X. Li, X. Wu, H. Lu, X. Liu, and H. Meng, "Channel-wise gated Res2Net: Towards robust detection of synthetic speech attacks," presented at the 21st Annu. Conf. Int. Speech Commun. Assoc. (INTERSPEECH), Brno, Czech Republic, Aug. 2021.

[23] J. Xue and H. Zhou, "Physiological-physical feature fusion for automatic voice spoofing detection," *Frontiers Comput. Sci.*, vol. 17, no. 2, pp. 1–10, Apr. 2023.

[24] B. Li and D. Lima, "Facial expression recognition via ResNet-50," *Int. J. Cognit. Comput. Eng.*, vol. 2, pp. 57–64, Jun. 2021.

[25] X. Ma, T. Liang, S. Zhang, S. Huang, and L. He, "Improved lightCNN with attention modules for ASV spoofing detection," in *Proc. IEEE Int. Conf. Multimedia Expo. (ICME)*, Jul. 2021, pp. 1–6.

[26] M. Dua, C. Jain, and S. Kumar, "LSTM and CNN based ensemble approach for spoof detection task in automatic speaker verification systems," *J. Ambient Intell. Humanized Comput.*, vol. 13, no. 4, pp. 1985–2000, Apr. 2022.

[27] T. Ojala, M. Pietikainen, and D. Harwood, "Performance evaluation of texture measures with classification based on Kullback discrimination of distributions," in *Proc. 12th Int. Conf. Pattern Recognit.*, vol. 1, Oct. 1994, pp. 582–585.

[28] J. Heikkila and V. Ojansivu, "Methods for local phase quantization in blur-insensitive image analysis," in *Proc. Int. Workshop Local Non-Local Approximation Image Process.*, Aug. 2009, pp. 104–111.

[29] M. Hall-Beyer, "GLCM texture: A tutorial," *Nat. Council Geographic Inf. Anal. Remote Sens. Core Curriculum*, vol. 3, no. 1, p. 75, Mar. 2020.

[30] J. Yang, R. K. Das, and H. Li, "Significance of subband features for synthetic speech detection," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 2160–2170, 2020.

[31] J. Yamagishi, T. Kinnunen, N. Evans, and P. L. Leon, "Introduction to the issues on spoofing and countermeasures for automatic speaker verification," *IEEE J. Sel. Topics Signal Process.*, vol. 11, no. 4, pp. 585–587, Jun. 2017.

[32] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[33] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn. (PMLR)*, 2015, pp. 448–456.

[34] A. Janicki, F. Alegre, and N. Evans, "An assessment of automatic speaker verification vulnerabilities to replay spoofing attacks," *Secur. Commun. Netw.*, vol. 9, no. 15, pp. 3030–3044, Jul. 2016.

[35] M. Alzantot, Z. Wang, and M. B. Srivastava, "Deep residual neural networks for audio spoofing detection," in *Proc. 19th Annu. Conf. Int. Speech Commun. Assoc. (INTERSPEECH)*, Graz, Austria, Sep. 2019, pp. 1078–1082.

[36] H. Tak, J. Patino, A. Nautsch, N. Evans, and M. Todisco, "Spoofing attack detection using the non-linear fusion of sub-band classifiers," presented at the 20th Annu. Conf. Int. Speech Commun. Assoc. (INTERSPEECH), Shanghai, China, Oct. 2020.
[37] A. Gomez-Alanis, J. A. Gonzalez-Lopez, and A. M. Peinado, "A kernel density estimation based loss function and its application to ASV-spoofing detection," *IEEE Access*, vol. 8, pp. 108530–108543, 2020.
[38] I.-Y. Kwak, S. Kwag, J. Lee, J. H. Huh, C.-H. Lee, Y. Jeon, J. Hwang, and J. W. Yoon, "ResMax: Detecting voice spoofing attacks with residual network and max feature map," in *Proc. 25th Int. Conf. Pattern Recognit. (ICPR)*, Jan. 2021, pp. 4837–4844.
[39] Y. Zhang, W. Wang, and P. Zhang, "The effect of silence and dual-band fusion in anti-spoofing system," presented at the 20th Annu. Conf. Int. Speech Commun. Assoc. (INTERSPEECH), Brno, Czech Republic, Aug. 2021.
[40] L. Wei, Y. Long, H. Wei, and Y. Li, "New acoustic features for synthetic and replay spoofing attack detection," *Symmetry*, vol. 14, no. 2, p. 274, Jan. 2022.
[41] H. Ling, L. Huang, J. Huang, B. Zhang, and P. Li, "Attention-based convolutional neural network for ASV spoofing detection," in *Proc. 20th Annu. Conf. Int. Speech Commun. Assoc. (INTERSPEECH)*, Brno, Czech Republic, Aug. 2021, pp. 4289–4293.
[42] M. Sahidullah, T. Kinnunen, and C. Hanilçi, "A comparison of features for synthetic speech detection," presented at the 15th Annu. Conf. Int. Speech Commun. Assoc. (INTERSPEECH), Dresden, Germany, Sep. 2015.
[43] X. Wang and J. Yamagishi, "A comparative study on recent neural spoofing countermeasures for synthetic speech detection," presented at the 20th Annu. Conf. Int. Speech Commun. Assoc. (INTERSPEECH), Brno, Czech Republic, Aug. 2021.
[44] H. Tak, J. Patino, M. Todisco, A. Nautsch, N. Evans, and A. Larcher, "End-to-end anti-spoofing with RawNet2," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2021, pp. 6369–6373.
[45] X. Chen, Y. Zhang, G. Zhu, and Z. Duan, "UR channel-robust synthetic speech detection system for ASVspoof 2021," presented at the 20th Annu. Conf. Int. Speech Commun. Assoc. (INTERSPEECH), Brno, Czech Republic, Sep. 2021.
[46] J. Yamagishi, X. Wang, M. Todisco, M. Sahidullah, J. Patino, A. Nau, A. Lee, T. Kinnunen, N. Evans, and H. Delgado, "ASVspoof2021: Accelerating spoofed and deep fake speech detection," presented at 20th Annu. Conf. Int. Speech Commun. Assoc. (INTERSPEECH), Brno, Czech Republic, Aug. 2021.

BESTE USTUBIOGLU received the B.Sc.E. and M.Sc.E. degrees in computer engineering and the Ph.D. degree from Karadeniz Technical University (KTU), Türkiye, in 2010, 2013, and 2018, respectively. She is currently an Assistant Professor with the Computer Engineering Department, KTU. She works on cybersecurity and machine learning.

GUL TAHAOGLU received the bachelor's degree in computer engineering from Erciyes University, in 2012, and the master's and Ph.D. degrees in computer engineering from Karadeniz Technical University, Türkiye, in 2015 and 2021, respectively. She has been an Assistant Professor with Karadeniz Technical University, since 2022. She has been a Visiting Research Fellow with the Signal Processing and Communications Laboratory, University of Florence, Italy. Her main research interest includes multimedia forensics.

ARDA USTUBIOGLU received the B.Sc.E. and M.Sc.E. degrees in computer engineering and the Ph.D. degree in computer engineering from Karadeniz Technical University (KTU), Türkiye, in 2005, 2009, and 2018, respectively. He is currently an Assistant Professor in management information systems with Trabzon University. He works on data security, digital image watermarking, and deep learning.

GUZIN ULUTAS (Senior Member, IEEE) received the B.S., M.S., and Ph.D. degrees from the Computer Engineering Department, Karadeniz Technical University, in 2002, 2004, and 2012, respectively. Her previous work experience includes Karadeniz Technical University as a Research Assistant, from 2002 to 2004. She was with Ondokuz Mayis University as a Research Assistant, from 2005 to 2009. She joined Karadeniz Technical University, in 2009, as a Lecturer. She was with the Computer Engineering Department as an Assistant Professor, from 2012 to 2018, and an Associate Professor, from 2018 to 2024. She is currently a Professor. Her main research interests include multimedia security, forensic science, and network security. She has been a member of Advisory Board of Tubitak which is a Scientific Council of Türkiye, since 2022.

IRENE AMERINI (Member, IEEE) received the Laurea degree in computer engineering and the Ph.D. degree in computer engineering, multimedia, and telecommunication from the University of Florence, Italy, in 2006 and 2010, respectively. She is currently a Postdoctoral Researcher with the Media Integration and Communication Center, University of Florence, and also an Associate Professor at the Department of Computer, Control, and Management Engineering. She was a Visiting Scholar with Binghamton University, NY, USA, in 2010. She has been a Visiting Research Fellow of the School of Computing and Mathematics, Charles Sturt University, Australia, since 2018, offered by Australian Government Department of Education and Training through the Endeavour Scholarship & Fellowship Program. Her main research interests include digital image processing, multimedia content security technologies, secure media, and multimedia forensics. She is a member of the IEEE Information Forensics and Security Technical Committee and EURASIP SAT Biometrics, Data Forensics, and Security. She received Italian Habilitation for Associate Professor in Telecommunications and Computer Science. She is an Associate Editor of IEEE Access and a guest editor of several international journals.

MUHAMMED KILIC received the B.S. degree in computer engineering from Karadeniz Technical University, in 2020. He is currently pursuing the master's degree in computer engineering. He is also a Research Assistant. His research interest includes audio forgery detection.

Open Access funding provided by 'Università degli Studi di Firenze' within the CRUI CARE Agreement