Detecting Replay Attacks Using Single-Channel Audio: The Temporal Autocorrelation of Speech

Shih-Kuang Lee* Yu Tsao† and Hsin-Min Wang*
* Institute of Information Science, Academia Sinica, Taipei, Taiwan
E-mail: sklee@iis.sinica.edu.tw, whm@iis.sinica.edu.tw
† Research Center for Information Technology Innovation, Academia Sinica, Taipei, Taiwan
E-mail: yu.tsao@citi.sinica.edu.tw

***Abstract*—In this paper, we propose to use the temporal autocorrelation of single-channel speech as a new feature for replay detection. Visual comparisons show that the proposed feature distinguishes replay attacks from clean speech and speech with simulated reverberation. Experimental results on the ASVspoof 2019 physical access database show that the proposed feature contains crucial information against replay attacks and that using the proposed feature in a fusion system almost always leads to performance improvements. Furthermore, our best fusion system achieves equal error rate and minimum tandem detection cost function of 0 on the development set for the first time.**

I. Introduction

Replay detection has been greatly improved with the help of deep learning technology over the past few years [1]–[4]. In the Automatic Speaker Verification Spoofing and Countermeasures Challenge (ASVspoof) in 2017 [2] and 2019 [3], [4], it was shown that appropriate neural network architectures can effectively construct countermeasures against replay attacks. Various studies have shown that replay detection systems that use time-frequency representation of speech as input feature perform better than systems that use single frames as input feature. In these state-of-the-art systems, correlations between speech features and replay attacks are found in time-frequency representations, which are not visible in a single frame [5], [6].

Replay attacks are a convenient way to bypass authentication by exploiting vulnerabilities in Automated Speaker Verification (ASV) systems. Since 2017, ASVspoof has introduced replay attacks in the challenge [2]. In 2019, the challenge was further divided into the logical access scenario, including spoofing attacks generated by speech synthesis and voice conversion, and the physical access scenario, including replay attacks using simulated replay speech [3]. In contrast to speech synthesis and voice conversion, which require adjusting the model to mimic the voice of a known target speaker as closely as possible, replay attacks can attack ASV systems by simply replaying the target speaker's voice [4].

The classical way to detect replay attacks is to use the Gaussian mixture models (GMMs). GMM-based replay detection systems can be found in the baseline systems of ASVspoof2017 [2] and ASVspoof 2019 [3]. These systems use GMMs to evaluate whether a single frame from the time-frequency representation of speech contains a replay attack, and then averages the results across all frames to determine whether the speech is a replay attack. However, such computation leads to loss of temporal information because the replay attack detection of these GMM-based systems does not consider the temporal information of speech features.

Replay detection systems built using deep learning models have achieved breakthrough performance growth in recent years [5]. It was observed in ASVspoof 2019 that significant performance gains were obtained with the help of deep learning techniques for replay detection compared to GMM-based systems [3], [4]. These advanced deep learning models for replay detection are based on ResNet model [5]–[7] or light convolutional neural network (LCNN) [8], [9] architectures, which share a common structure of employing two-dimensional convolutional layers to extract and detect replay attack trajectories in time-frequency representations of speech. These models have demonstrated their effectiveness in building countermeasures against replay attacks [5]–[9].

Time-frequency representation refers to the signal representation into which speech is processed before replay detection using GMM-based systems or most deep learning model-based systems. In general, the time-frequency representation of speech can be categorized into magnitude time-frequency representation and phase time-frequency representation. Examples of magnitude time-frequency representation include spectrogram, linear frequency cepstral coefficients (LFCCs) [9]–[11] and constant Q cepstral coefficients (CQCCs) [12], [13], and examples of phase time-frequency representation include modified group delay (MGD) [5], [6] and product spectrum cepstral coefficients (PSCCs) [14]. Researchers have effectively exploited their application in replay detection via deep learning models [5], [6], [9], [14].

Score fusion is a simple and effective method for building valid countermeasures against replay attacks [4]. Among the systems using GMMs or deep learning models combined with different time-frequency representations, each provides different insights for replay detection. To do so, each system calculates a score for the given test utterance; and by exploiting the strength of each countermeasure while complementing the weaknesses, these scores are fused to create a fusion countermeasure. Significant performance improvements are observed in state-of-the-art replay detection systems with the help of score fusion [6], [9].

978-616-590-477-3 ©2022 APSIPA

In this paper, we propose a novel feature for replay detection based on the computation of temporal autocorrelation in weighted prediction error (WPE) dereverberation [15]–[17]. In our experiments, we first implemented a state-of-the-art system for the ASVspoof 2019 physical access scenario [9]. We then examined the proposed feature from various aspects by comparing performance of single systems and fusion systems using different speech features. Experimental results confirm the contribution of the proposed feature in fusion systems. Furthermore, using the proposed feature and through score fusion, our best fusion system is able to reduce the equal error rate (EER) and minimum tandem detection cost function (min-tDCF) to 0 on the development set for the first time.

The remainder of this paper is organized as follows. Section 2 reviews related studies on replay detection. Section 3 describes the proposed feature. Section 4 presents the experiments, results and discussion. Finally, Section 5 provides our concluding remarks.

II. Related Work

Replay detection systems consist of classifiers and speech features. In this section, we review classifiers and speech features for replay detection.

A. Classifiers

1) Gaussian Mixture Model: GMM-based classifiers are a classical approach to build replay detection systems [2], [3], [12]. The classical approach is to determine the replay detection result by the likelihood of the given test utterance containing a replay attack and the likelihood of the given test utterance not containing a replay attack. The score for the given utterance is calculated as the ratio of two likelihoods [12], and then is used to determine whether the utterance contains a replay attack. Since the likelihood of the utterance is calculated by averaging the likelihoods of all frames, the GMM does not exploit the temporal information of speech features. Therefore, we believe that such an architecture would result in limited performance for replay detection.

2) Deep Learning Model: Classifiers based on deep learning models are state-of-the-art for building replay detection systems [7]. Unlike GMM-based replay detection classifiers, deep learning model-based replay detection classifiers generally only need one deep learning model to detect replay attacks in speech, and most deep learning model-based classifiers use the time-frequency representation of speech instead of single frame as input feature [5]–[9]. It has been observed that most deep learning model-based classifiers incorporate two-dimensional convolutional layers in their deep learning model architectures, while taking the time-frequency representation of speech as the feature for the classifier. Comparison with GMM-based classifiers, this architecture can leverage both the temporal and spatial information of the time-frequency representation of speech to identify the association of specific regions in the time-frequency representation with replay attacks [5], [6].

B. Speech Features

1) Magnitude: Magnitude information is a classical feature representation of speech for speech processing applications. Most speech features are created using the Fourier transform, where the speech waveform is transformed from real numbers to complex numbers, i.e., the spectrum, and the magnitude information of speech is the absolute value of the spectrum. Various algorithms and techniques have been developed to exploit the magnitude information of speech for different speech applications, such as spectral subtraction [18], [19] and Wiener filtering [19], [20] for speech enhancement and Mel-frequency cepstral coefficients (MFCCs) for speech recognition [21]. For spoofed speech detection, LFCCs are classification robust to any type of spoofing attack [9]–[11], and the spectrogram computed with the constant Q transform (CQT) [13] exhibits excellent performance in replayed speech detection [6]–[9]. In our previous study, we have shown that the cepstrogram is another powerful feature for countermeasure against replay attacks, which is also derived form the magnitude information of speech [22].

2) Phase: Phase information is another feature representation of speech and is the argument of the spectrum. The phase information of speech was considered unimportant in various speech applications in the past, but has become an emerging field in recent years [23]–[25]. Various techniques have been applied to different speech applications to exploit the phase information of speech for better results, such as group delay [26] and product spectrum [27], [28] for speech recognition and phase spectrum compensation for speech enhancement [24], [29]. For spoofed speech detection, group delay [5], [6] and product spectrum [14] show excellent performance in replay attack detection.

III. The Proposed Feature

In this work, we propose a novel feature for replay detection based on the temporal autocorrelation of speech. In the following, we describe the computation of the temporal autocorrelation of speech and the method for feature construction.

A. Weighted Prediction Error

The general idea of WPE is to estimate the late reflections of reverberant speech, and then subtract them from the reverberant speech to obtain a valid estimate of the early part of the reverberant speech, which includes both direct speech and early reflections of reverberant speech [15]–[17]. The signal model applied in WPE is given as follow:

$$y_{c,t,f} = x^{early}_{c,t,f} + x^{tail}_{c,t,f}, \tag{1}$$

where $y_{c,t,f}$ refers to the multi-channel reverberant speech presented as complex spectrograms, c denotes the channel index, t denotes the frame index, f denotes the frequency bin channel, and $x^{early}_{c,t,f}$ and $x^{tail}_{c,t,f}$ represent the early part and late reverberation of the reverberant speech, respectively.

The procedures for performing WPE on a reverberant speech are given as follows:

$$\lambda_{t,f} = \frac{1}{(\delta+1+\delta)C} \sum_{\tau=t-\delta}^{t+\delta} \sum_{c} |x_{\tau,f,c}^{early}|^2, \tag{2}$$

$$R_f = \sum_{t} \frac{\tilde{y}_{t-\Delta,f} \tilde{y}_{t-\Delta,f}^H}{\lambda_{t,f}}, \tag{3}$$

$$P_f = \sum_{t} \frac{\tilde{y}_{t-\Delta,f} y_{t,f}^H}{\lambda_{t,f}}, \tag{4}$$

$$\mathbf{G}_f = R_f^{-1} P_f, \tag{5}$$

$$\hat{x}_{t,f}^{early} = y_{t,f} - \mathbf{G}_f^H \tilde{y}_{t-\Delta,f}, \tag{6}$$

where $\hat{x}_{t,f}^{early}$ refers to the estimation of clean speech, $\mathbf{G}_f$ represents the prediction filter, and $\lambda_{t,f}$ denotes the time-varying variance.

B. Replay Detection With Temporal Autocorrelation of Speech

Figs. 1 and 2 present the spectrograms and the prediction filters of clean speech, bona fide trial, and spoofed trial. The bona fide trial is the clean speech with simulated reverberation. The spoofed trial is generated from the bona fide trial with replay attacks. We use these two figures to show how simulated reverberation and replay attacks are revealed in the spectrogram and prediction filters. The spectrogram of clean speech is blurred by simulated reverberation, as shown in Fig. 1 (b), and further smeared by replay, as shown in Fig. 1 (c). WPE is an effective multi-channel dereverberation technique that reduces speech recognition errors for reverberant speech [21]. Estimation of prediction filters in WPE involves computing the temporal autocorrelation of speech and the spatial autocorrelation of audio channels at the same time. Since replay attacks form reverberation in spoofed speech, we are inspired to use the prediction filters, i.e., the temporal autocorrelation of speech, as a feature for replay detection. The temporal autocorrelation of speech in single-channel audio does clearly reveal replay attacks, as shown in Fig. 2 (c).

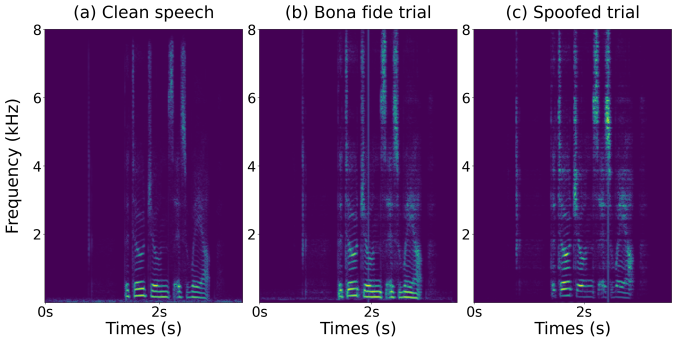

Fig. 1. The spectrograms in the log1p (natural logarithm of (1 + input)) scale [30]: (a) the original clean speech sample p262_227 from VCTK [31], (b) the bona fide trial PA_D_0004063 with simulated reverberation, and (c) the spoofed trial PA_D_0024255.

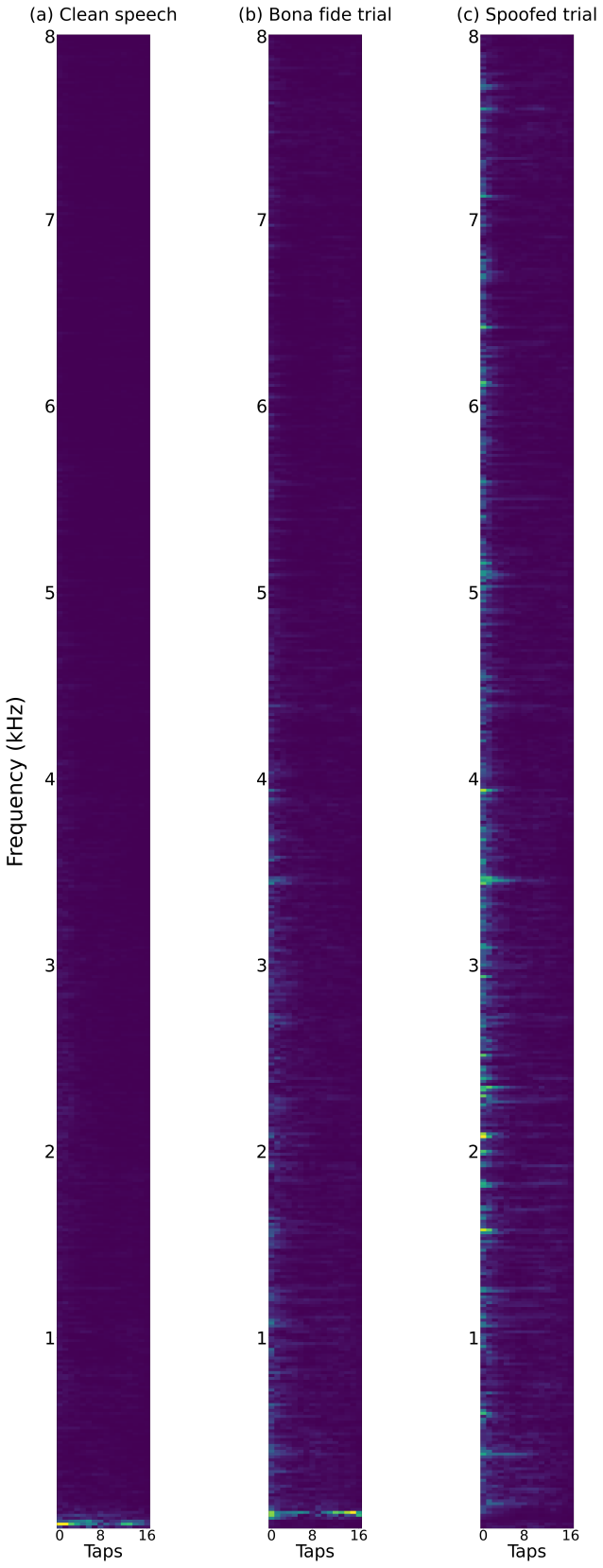

Fig. 2. The prediction filters in the log1p (natural logarithm of (1 + input)) scale [30]: (a) the original clean speech sample p262_227 from VCTK [31], (b) the bona fide trial PA_D_0004063 with simulated reverberation, and (c) the spoofed trial PA_D_0024255.

IV. Experiments and Results

Our experiments were conducted on the ASVspoof 2019 physical access scenario using models with the same LCNN architecture but different features. We implemented the model of the team T45 [9] in the ASVspoof 2019 challenge [3], as this architecture showed its effectiveness against replay attacks in ASVspoof challenge [4], [8], [9]. The configuration of the dropout layers in the model is not specified in the corresponding paper [9]. We only performed one dropout on the flatten layer to prevent overfitting. For more details on our implementation, please visit our repository[1].

A. Baseline Systems

Table I shows the performance of Team T45's systems (upper four rows) and the performance of our self-implemented systems (lower four rows). In our implementation, the speech features used in LFCC-LCNN and CQT-LCNN are LFCCs and CQT-based logarithmic power magnitude spectrogram (CQTgram), which were obtained with the code from the baseline system provided by the challenge organizer[2], and the default setting is used. This default setting was also used in Team T45's system [9]. LFCCs were extracted using a Hamming window of 20 ms length, 512 FFT bins, and 20 filters. CQTgram was extracted with 96 bins per octave. The speech feature used in DCT-LCNN was obtained with our own code. We followed the configuration of Team T45's DCT-LCNN system; the magnitude spectrogram was extracted by discrete cosine transform (DCT) with a Blackman window of length 863 and frame shift of 128. For each single system (CQT-LCNN, LFCC-LCNN, and DCT-LCNN), the model was separately trained on the training set, selected based on the results on the development set, and evaluated on the evaluation set. The fusion system was achieved by summing the scores of the single systems for each trial [9]. The results were presented in terms of EER and min-tDCF [32], which are the metrics used in the ASVspoof 2019 challenge [3]. From Table I, we can see that the performance of our self-implemented systems is comparable to or better than the performance of the corresponding T45 systems.

TABLE I
Performance comparison of Team T45's systems (reported in [9]) and our self-implemented systems. All systems were implemented using the same LCNN architecture.

	Dev		Eval	
System	**tDCF**	**EER**	**tDCF**	**EER**
CQT-LCNN [9]	0.0197	0.800	0.0295	1.23
LFCC-LCNN [9]	0.0320	1.311	0.1053	4.60
DCT-LCNN [9]	0.0732	3.850	0.560	2.06
Fusion [9]	**0.0001**	**0.0154**	**0.0122**	**0.54**
CQT-LCNN	0.0096	0.374	0.0130	0.514
LFCC-LCNN	0.0145	0.519	0.0299	1.061
DCT-LCNN	0.0385	1.444	0.0774	2.897
Fusion	**0.0014**	**0.057**	**0.0048**	**0.165**

TABLE II
Performance of single systems using different features. All systems were implemented using the same LCNN architecture.

	Dev		Eval	
System	**tDCF**	**EER**	**tDCF**	**EER**
CQT-LCNN	0.0096	0.374	0.0130	0.514
LFCC-LCNN	0.0145	0.519	0.0299	1.061
DCT-LCNN	0.0385	1.444	0.0774	2.897
Spec1724-LCNN	0.0062	0.203	0.0263	0.917
Ceps1724-LCNN	0.0076	0.275	0.0191	0.712
Spec-LCNN	0.0148	0.556	0.0522	1.719
[22] **Ceps-LCNN**	**0.0039**	**0.129**	**0.0105**	**0.370**
TAC-LCNN	0.0863	3.152	0.1560	5.882

B. Performance of Single Systems

Table II shows the performance of our implemented single systems using different features. The first three systems are the same as those in Table I. The speech features used in Spec-LCNN and Spec1724-LCNN were magnitude spectrograms extracted via fast Fourier transform (FFT) with different configurations; Spec-LCNN used a Blackman window of length 1024 and frame shift of 128, and Spec1724-LCNN used a Blackman window of length 1724 and frame shift of 128. The speech features used in Ceps-LCNN and Ceps1724-LCNN were cepstrograms [22] derived by DCT from the magnitude spectrograms used in Spec-LCNN and Spec1724-LCNN, respectively. The temporal autocorrelation of speech used in TAC-LCNN was calculated from a complex spectrogram, which was computed using the same configuration as the spectrogram used in Spec-LCNN and Ceps-LCNN. We used an open source implementation of WPE dereverberation [17] to compute prediction filters as the speech feature. We applied the parameter settings used in CHiME-6 [21] except for the value of `taps`. Following the configuration in [21], the `delay` was set to 2, the `iterations` was set to 3, the `psd_context` was set to 0, the `statistics_mode` was set to `full`, and the value of `taps` was extended from 10 to 16 to fit the LCNN architecture. The results in Table II show that the TAC-LCNN single system using the temporal autocorrelation feature performs worse than single systems using other features. Although this result is disappointing, in subsequent experiments, we will confirm that the temporal autocorrelation feature, combined with other features, can improve the performance of fusion systems.

C. Performance of Fusion Systems

Tables III and IV present the performance of our implemented fusion systems. In Table III, the fusion systems combined Spec-LCNN and/or Ceps-LCNN, while in Table IV, the fusion systems combined Spec1724-LCNN and/or Ceps1724-LCNN. We aimed to examine the proposed feature from various aspects. All fusion systems used the same score fusion strategy as the fusion systems in Table I. In Table III, because the same windowing configuration for spectrogram extraction was used in TAC-LCNN, Spec-LCNN, and Ceps-LCNN, the

[1] https://github.com/shihkuanglee/RD-LCNN

[2] https://www.asvspoof.org/asvspoof2019/ASVspoof_2019_baseline_CM_v1.zip

TABLE III
PERFORMANCE OF VARIOUS FUSION SYSTEMS INCORPORATING TAC-LCNN.

System	Dev tDCF	Dev EER	Eval tDCF	Eval EER
CQT	0.0096	0.374	**0.0130**	**0.514**
TAC+CQT	**0.0088**	**0.349**	0.0150	0.613
LFCC	0.0145	0.519	0.0299	1.061
TAC+LFCC	**0.0068**	**0.312**	**0.0157**	**0.547**
DCT	0.0385	1.444	0.0774	2.897
TAC+DCT	**0.0179**	**0.682**	**0.0436**	**1.741**
Spec	0.0148	0.556	0.0522	1.719
TAC+Spec	**0.0099**	**0.353**	**0.0325**	**1.161**
Ceps	0.0039	0.129	0.0105	0.370
TAC+Ceps	**0.0032**	**0.127**	**0.0096**	**0.330**
CQT+LFCC	0.0037	0.166	0.0079	0.283
TAC+CQT+LFCC	**0.0026**	**0.111**	**0.0063**	**0.216**
CQT+DCT	0.0048	0.205	0.0111	0.475
TAC+CQT+DCT	**0.0032**	**0.168**	**0.0105**	**0.398**
CQT+Spec	**0.0021**	0.131	0.0122	0.514
TAC+CQT+Spec	0.0024	**0.109**	**0.0098**	**0.359**
CQT+Ceps	0.0024	0.094	0.0043	**0.149**
TAC+CQT+Ceps	**0.0017**	**0.057**	**0.0043**	0.154
LFCC+DCT	0.0042	0.183	0.0089	0.321
TAC+LFCC+DCT	**0.0022**	**0.109**	**0.0073**	**0.282**
LFCC+Spec	0.0020	0.078	0.0076	0.289
TAC+LFCC+Spec	**0.0017**	**0.076**	**0.0062**	**0.238**
LFCC+Ceps	0.0030	0.109	0.0074	0.254
TAC+LFCC+Ceps	**0.0017**	**0.057**	**0.0052**	**0.205**
DCT+Spec	0.0120	0.499	0.0424	1.488
TAC+DCT+Spec	**0.0079**	**0.275**	**0.0256**	**0.951**
DCT+Ceps	**0.0013**	0.074	0.0066	0.242
TAC+DCT+Ceps	0.0014	**0.057**	**0.0059**	**0.232**
Spec+Ceps	0.0015	0.074	0.0067	0.260
TAC+Spec+Ceps	**0.0010**	**0.041**	**0.0051**	**0.184**
CQT+LFCC+DCT	0.0014	**0.057**	0.0048	0.165
TAC+CQT+LFCC+DCT	**0.0009**	**0.057**	**0.0038**	**0.149**
CQT+LFCC+Spec	**0.0009**	0.039	0.0045	0.177
TAC+CQT+LFCC+Spec	0.0011	**0.037**	**0.0039**	**0.133**
CQT+LFCC+Ceps	0.0022	0.074	0.0042	0.150
TAC+CQT+LFCC+Ceps	**0.0016**	**0.059**	**0.0031**	**0.115**
CQT+DCT+Spec	0.0029	0.168	0.0148	0.591
TAC+CQT+DCT+Spec	**0.0019**	**0.113**	**0.0105**	**0.420**
CQT+DCT+Ceps	0.0015	0.059	0.0034	0.128
TAC+CQT+DCT+Ceps	**0.0008**	**0.039**	**0.0029**	**0.121**
CQT+Spec+Ceps	0.0011	**0.037**	0.0038	0.155
TAC+CQT+Spec+Ceps	**0.0008**	**0.037**	**0.0034**	**0.127**
LFCC+DCT+Spec	0.0021	0.096	0.0073	0.320
TAC+LFCC+DCT+Spec	**0.0009**	**0.057**	**0.0063**	**0.232**
LFCC+DCT+Ceps	0.0012	0.039	0.0040	0.143
TAC+LFCC+DCT+Ceps	**0.0004**	**0.022**	**0.0030**	**0.115**
LFCC+Spec+Ceps	0.0008	0.037	0.0039	0.138
TAC+LFCC+Spec+Ceps	**0.0003**	**0.017**	**0.0027**	**0.109**
DCT+Spec+Ceps	0.0013	0.057	0.0078	0.304
TAC+DCT+Spec+Ceps	**0.0009**	**0.052**	**0.0059**	**0.221**
CQT+LFCC+DCT+Spec	**0.0004**	**0.037**	**0.0041**	0.171
TAC+CQT+LFCC+DCT+Spec	0.0005	**0.037**	0.0041	**0.165**
CQT+LFCC+DCT+Ceps	0.0008	0.052	0.0029	0.104
TAC+CQT+LFCC+DCT+Ceps	**0.0003**	**0.033**	**0.0022**	**0.072**
CQT+LFCC+Spec+Ceps	0.0004	0.022	0.0027	0.094
TAC+CQT+LFCC+Spec+Ceps	**0.0002**	**0.017**	**0.0023**	**0.083**
CQT+DCT+Spec+Ceps	0.0005	0.037	0.0047	0.166
TAC+CQT+DCT+Spec+Ceps	**0.0004**	**0.033**	**0.0035**	**0.149**
LFCC+DCT+Spec+Ceps	0.0003	0.017	0.0031	0.116
TAC+LFCC+DCT+Spec+Ceps	**0.0001**	**0.004**	**0.0028**	**0.098**

TABLE IV
RESULTS OF THE FUSION SYSTEMS WITH SPEC1724-LCNN & CEPS1724-LCNN.

System	Dev tDCF	Dev EER	Eval tDCF	Eval EER
CQT+LFCC+DCT+Spec1724	**0.0002**	**0.017**	0.0030	0.109
TAC+CQT+LFCC+DCT+Spec1724	0.0003	**0.017**	**0.0028**	**0.105**
CQT+LFCC+Spec1724+Ceps	**0.0000**	**0.002**	0.0019	0.077
TAC+CQT+LFCC+Spec1724+Ceps	**0.0000**	**0.002**	**0.0016**	**0.055**
CQT+DCT+Spec1724+Ceps	0.0002	0.017	0.0026	0.099
TAC+CQT+DCT+Spec1724+Ceps	**0.0001**	**0.015**	**0.0025**	**0.095**
LFCC+DCT+Spec1724+Ceps	0.0000	0.002	0.0022	0.093
TAC+LFCC+DCT+Spec1724+Ceps	**0**	**0**	**0.0020**	**0.072**
CQT+LFCC+DCT+Cpes1724	0.0017	0.074	**0.0036**	0.145
TAC+CQT+LFCC+DCT+Cpes1724	**0.0011**	**0.057**	0.0036	**0.133**
CQT+LFCC+Spec+Cpes1724	0.0007	0.039	0.0033	0.137
TAC+CQT+LFCC+Spec+Cpes1724	**0.0006**	**0.035**	**0.0033**	**0.116**
CQT+DCT+Spec+Cpes1724	0.0015	0.072	0.0070	0.293
TAC+CQT+DCT+Spec+Cpes1724	**0.0010**	**0.070**	**0.0060**	**0.226**
LFCC+DCT+Spec+Cpes1724	0.0007	**0.037**	0.0045	0.188
TAC+LFCC+DCT+Spec+Cpes1724	**0.0006**	0.041	**0.0042**	**0.160**
CQT+LFCC+Spec1724+Cpes1724	0.0006	**0.022**	0.0027	0.111
TAC+CQT+LFCC+Spec1724+Cpes1724	**0.0003**	0.033	**0.0025**	**0.088**
CQT+DCT+Spec1724+Cpes1724	0.0012	0.041	0.0052	0.203
TAC+CQT+DCT+Spec1724+Cpes1724	**0.0008**	**0.037**	**0.0038**	**0.188**
LFCC+DCT+Spec1724+Cpes1724	0.0007	0.033	0.0036	0.137
TAC+LFCC+DCT+Spec1724+Cpes1724	**0.0005**	**0.022**	**0.0032**	**0.127**

comparison of fusion systems with and without TAC-LCNN is clear. From the table, we can see that almost every fusion system with TAC-LCNN outperforms the corresponding system without TAC-LCNN. Next, we examined whether the combination of TAC-LCNN with systems with features based on spectrograms extracted using different windowing configurations is effective. From Table IV, we can see that the combination of TAC-LCNN with Spec1724-LCNN and/or outperforms its corresponding system without TAC-LCNN in most cases. This result again confirms the contribution of TAC-LCNN in fusion systems. In addition, it is worth mentioning that this is the first time a fusion system (see TAC+LFCC+DCT+Spec1724+Ceps) has achieved EER and min-tDCF of 0 on the development set.

D. Comparison with State-of-the-Art Models

Table V compares the performance of several state-of-the-art systems and our fusion systems on the ASVspoof 2019 physical access database. It is clear that all our fusion systems with TAC-LCNN outperform the three state-of-the-art fusion systems compared in this experiment. The best performance on the evaluation set is achieved by the "TAC+CQT+LFCC+Spec1724+Ceps" system, and the min-DCF and EER are 0.0016 and 0.055, respectively. For the development set, the best performance is achieved by the "TAC+LFCC+DCT+Spec1724+Ceps" system, and the min-DCF and EER are reduced to 0. Our fusion systems combining the temporal autocorrelation feature achieve new state-of-the-art performance on the ASVspoof 2019 physical access database.

TABLE V
PERFORMANCE COMPARISON WITH STATE-OF-THE-ART SYSTEMS.

System	Dev		Eval	
	tDCF	EER	tDCF	EER
T45-Fusion [9]	0.0001	0.0154	0.0122	0.54
T28-Fusion [6]	0.0049	0.20	0.0096	0.39
Res2Net-CQT+LFCC+Spec [7]	0.0028	0.096	0.0075	0.287
CQT+LFCC+DCT	0.0014	0.057	0.0048	0.165
TAC+CQT+LFCC+DCT	0.0009	0.057	0.0038	0.149
[22] CQT+LFCC+Spec+Ceps	0.0004	0.022	0.0027	0.094
TAC+CQT+LFCC+Spec+Ceps	0.0002	0.017	0.0023	0.083
CQT+LFCC+Spec1724+Cpes1724	0.0006	0.022	0.0027	0.111
TAC+CQT+LFCC+Spec1724+Cpes1724	0.0003	0.033	0.0025	0.088
CQT+LFCC+Spec1724	0.0005	0.017	0.0031	0.111
TAC+CQT+LFCC+Spec1724	0.0005	0.017	0.0027	0.099
CQT+LFCC+Spec1724+Ceps	0.0000	0.002	0.0019	0.077
TAC+CQT+LFCC+Spec1724+Ceps	0.0000	0.002	**0.0016**	**0.055**
CQT+LFCC+DCT+Spec1724+Ceps	0.0000	0.002	0.0018	0.066
TAC+CQT+LFCC+DCT+Spec1724+Ceps	0.0000	0.002	0.0016	0.061
LFCC+DCT+Spec1724+Ceps	0.0000	0.002	0.0022	0.093
TAC+LFCC+DCT+Spec1724+Ceps	**0**	**0**	0.0020	0.072

V. CONCLUSION

In this paper, a novel feature for replay detection using temporal autocorrelation of single-channel speech is proposed. The computation of WPE dereverberation inspired us to use the prediction filters as the feature to detect replay attacks. Visual comparisons show that the proposed feature distinguishes replay attacks from bona fide speech. Experimental results show that all fusion systems incorporating the proposed feature achieve performance improvements compared to the corresponding systems without the proposed feature. One of our fusion systems achieves EER and min-tDCF of 0 on the development set of the ASVspoof 2019 physical access database. To the best of our knowledge, this is the first time a fusion model has achieved such a result. Our best fusion model also achieves new state-of-the-art performance on the evaluation set.

ACKNOWLEDGMENT

This work was supported by the NSTC-Taiwan Grant 111-2221-E-001-002.

REFERENCES

[1] Y. Gong, J. Yang, and C. Poellabauer, "Detecting Replay Attacks Using Multi-Channel Audio: A Neural Network-Based Method," *IEEE Signal Processing Letters*, vol. 27, pp. 920–924, 2020.

[2] T. Kinnunen, M. Sahidullah, H. Delgado, *et al.*, "The ASVspoof 2017 Challenge: Assessing the Limits of Replay Spoofing Attack Detection," in *Proc. Interspeech 2017*, 2017, pp. 2–6.

[3] M. Todisco, X. Wang, V. Vestman, *et al.*, "ASVspoof 2019: Future Horizons in Spoofed and Fake Audio Detection," in *Proc. Interspeech 2019*, 2019, pp. 1008–1012.

[4] A. Nautsch, X. Wang, N. Evans, *et al.*, "ASVspoof 2019: Spoofing Countermeasures for the Detection of Synthesized, Converted and Replayed Speech," *IEEE Transactions on Biometrics, Behavior, and Identity Science*, vol. 3, no. 2, pp. 252–265, 2021.

[5] F. Tom, M. Jain, and P. Dey, "End-To-End Audio Replay Attack Detection Using Deep Convolutional Networks with Attention," in *Proc. Interspeech 2018*, 2018, pp. 681–685.

[6] X. Cheng, M. Xu, and T. F. Zheng, "Replay detection using CQT-based modified group delay feature and ResNeWt network in ASVspoof 2019," in *2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, 2019, pp. 540–545.

[7] X. Li, N. Li, C. Weng, *et al.*, "Replay and Synthetic Speech Detection with Res2Net Architecture," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 6354–6358.

[8] G. Lavrentyeva, S. Novoselov, E. Malykh, A. Kozlov, O. Kudashev, and V. Shchemelinin, "Audio Replay Attack Detection with Deep Learning Frameworks," in *Proc. Interspeech 2017*, 2017, pp. 82–86.

[9] G. Lavrentyeva, S. Novoselov, A. Tseren, M. Volkova, A. Gorlanov, and A. Kozlov, "STC Antispoofing Systems for the ASVspoof2019 Challenge," in *Proc. Interspeech 2019*, 2019, pp. 1033–1037.

[10] M. Sahidullah, T. Kinnunen, and C. Hanilçi, "A comparison of features for synthetic speech detection," in *Proc. Interspeech 2015*, 2015, pp. 2087–2091.

[11] H. Tak, J. Patino, A. Nautsch, N. Evans, and M. Todisco, "Spoofing Attack Detection Using the Non-Linear Fusion of Sub-Band Classifiers," in *Proc. Interspeech 2020*, 2020, pp. 1106–1110.

[12] M. Todisco, H. Delgado, and N. Evans, "A New Feature for Automatic Speaker Verification Anti-Spoofing: Constant Q Cepstral Coefficients," in *Proc. The Speaker and Language Recognition Workshop (Odyssey 2016)*, 2016, pp. 283–290.

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

[14] J. Monteiro and J. Alam, "Development of Voice Spoofing Detection Systems for 2019 Edition of Automatic Speaker Verification and Countermeasures Challenge," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2019, pp. 1003–1010.

[15] T. Nakatani, T. Yoshioka, K. Kinoshita, M. Miyoshi, and B.-H. Juang, "Speech Dereverberation Based on Variance-Normalized Delayed Linear Prediction," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 7, pp. 1717–1731, 2010.

[16] T. Yoshioka and T. Nakatani, "Generalization of multi-channel linear prediction methods for blind mimo impulse response shortening," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 10, pp. 2707–2720, 2012.

[17] L. Drude, J. Heymann, C. Boeddeker, and R. Haeb-Umbach, "NARA-WPE: A Python package for weighted prediction error dereverberation in Numpy and Tensorflow for online and offline processing," in *Speech Communication; 13th ITG-Symposium*, 2018, pp. 1–5.

[18] S. Boll, "Suppression of acoustic noise in speech using spectral subtraction," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 27, no. 2, pp. 113–120, 1979.

[19] P. Loizou, *Speech Enhancement: Theory and Practice (2nd ed.)* CRC Press, 2013.

[20] N. Wiener, *Extrapolation, Interpolation, and Smoothing of Stationary Time Series: With Engineering Applications*. The MIT Press, Aug. 1949.

[21] S. Watanabe, M. Mandel, J. Barker, *et al.*, "CHiME-6 Challenge: Tackling Multispeaker Speech Recognition for Unsegmented Recordings," in *Proc. 6th International Workshop on Speech Processing in Everyday Environments (CHiME 2020)*, 2020, pp. 1–7.

[22] S.-K. Lee, Y. Tsao, and H.-M. Wang, "A Study of Using Cepstrogram for Countermeasure Against Replay Attacks," *arXiv preprint arXiv:2204.04333*, 2022.

[23] D. Wang and J. Lim, "The unimportance of phase in speech enhancement," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 30, no. 4, pp. 679–681, 1982.

[24] K. Paliwal, K. Wójcicki, and B. Shannon, "The importance of phase in speech enhancement," *Speech Communication*, vol. 53, no. 4, pp. 465–494, 2011.

[25] P. Mowlaee, R. Saeidi, and Y. Stylianou, "Advances in phase-aware signal processing in speech communication," *Speech Communication*, vol. 81, pp. 1–29, 2016, Phase-Aware Signal Processing in Speech Communication.

[26] H. Kawahara, M. Morise, R. Nisimura, and T. Irino, "An interference-free representation of group delay for periodic signals," in *Proceedings of The 2012 Asia Pacific Signal and Information Processing Association Annual Summit and Conference*, 2012, pp. 1–4.

[27] D. Zhu and K. Paliwal, "Product of power spectrum and group delay function for speech recognition," in *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, 2004, pp. I–125.

[28] R. M. Hegde, H. A. Murthy, and V. R. R. Gadde, "Significance of the Modified Group Delay Feature in Speech Recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 1, pp. 190–202, 2007.

[29] A. P. Stark, K. K. Wojcicki, J. G. Lyons, and K. K. Paliwal, "Noise Driven Short-Time Phase Spectrum Compensation Procedure for Speech Enhancement," in *Proc. Interspeech 2008*, 2008, pp. 549–552.

[30] S.-Y. Chuang, H.-M. Wang, and Y. Tsao, "Improved lite audio-visual speech enhancement," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 1345–1359, 2022.

[31] C. Veaux, J. Yamagishi, and K. MacDonald, "CSTR VCTK Corpus: English Multi-speaker Corpus for CSTR Voice Cloning Toolkit," University of Edinburgh, The Centre for Speech Technology Research (CSTR), 2017.

[32] T. Kinnunen, K. A. Lee, H. Delgado, *et al.*, "t-DCF: a Detection Cost Function for the Tandem Assessment of Spoofing Countermeasures and Automatic Speaker Verification," in *Proc. The Speaker and Language Recognition Workshop (Odyssey 2018)*, 2018, pp. 312–319.