Combining the Optimal Seed Number Determining and SCE-based Data Augmentation for Logical Access Attack Detection

Jichen Yan[1*], Jiawei Wu[1], Wen He[1], Zili Gao[2]

[1] School of Cyber, Guangdong Polytechnic Normal University, 510000, China

[2] Zhuhai Fudan Innovation Institute, 519000, China

Corresponding Email:nisonyoung@163.com

https://doi.org/10.70695/skknxc45

Abstract

Recent advancements in voice conversion and text-to-speech technologies enable the creation of natural-sounding speech, posing challenges to automatic speaker verification systems. In response, research into spoofing countermeasures has intensified to safeguard ASV systems from these threats. While advanced spoofing countermeasures can detect known types of spoofing attacks, their effectiveness diminishes against unknown attacks which have not appeared in the training set. In this work, to address the challenge of determining the optimal baseline from the best seed number, and to ensure that both we and others can replicate and potentially enhance the results with ease, we propose a method for determining the optimal seed number. Building on the optimal baseline, we proposed a novel data augmentation technique termed SCE, which is rooted in signal companding and expanding. Specifically, signal companding employs a-law and mu-law algorithms, whereas signal expanding leverages bit-24 and bit-32 variants of the original training set. We believed that our proposed method enhances the robustness of the detection system through data augmentation with SCE. Our investigations utilize the ASVspoof 2019 logical access corpus and employ a ResNet-based system. The results reveal that the SCE technique surpasses the performance of many leading single systems, demonstrating its prowess in tackling the unpredictable nature of attacks. Notably, its t-DCF and EER metrics achieve scores of 0.050 and 1.60% respectively, which can rank top several systems to date.

Keywords Determining the optimal seed number; Data augmentation; Logical access attack detec

1 Introduction

Speaker verification research has travelled a long way and several robust frameworks have been developed in the due course, for example, joint factor analysis [1], i-vector [2,3] and x-vector [4,5]. Presently, the combination of x-vector with probabilistic linear discriminant analysis (PLDA) [6] represents the prevailing approach in speaker verification.

In recent years, automatic speaker verification (ASV) systems have found applications in a broad spectrum of services [7-9]. However, alongside their proliferation, concerns have arisen about spoofing attacks, given the systems' vulnerability to them [10,11]. Typically, ASV systems are trained using genuine speech without accounting for spoofed variants. Consequently, these systems can be susceptible to spoofing attacks when presented with spoofed speech. Currently, the primary forms of spoofing attacks are classified into four main categories: impersonation, replay, voice conversion (VC), and text-to-speech (TTS) attacks [12]. Recent advancements in VC and TTS technologies not only generate high-quality, natural-sounding speech [13], but also pose potential threats to ASV systems [14,15].

The community driven ASVspoof challenge series promotes research on spoofing countermeasures using a benchmark corpus across research groups from last couple of editions. There are mainly two typs of attacks: logical access attacks and physical access attacks. In which, logical access attacks employ VC and TTS techniques, while physical access attacks utilize replay samples within a simulated environment. In this work, We concentrate on detecting logical access attacks, as they present an imminent threat due to their unpredictable nature [15-17].

The literature indicates that many recent advancements in spoofing countermeasures predominantly rely on either front-end handcrafted features or specific classifiers. Among these, linear frequency cepstral coefficients (LFCC), and subband spectral flux coefficients and spectral centroid frequency coefficients [18], as well as cochlear filter cepstral coefficient and instantaneous frequency (CFCCIF) [19], stand out as promising front-ends. These were demonstrated to be effective in the inaugural ASVspoof 2015 for detecting logical access attacks. Subsequently, the constant-Q cepstral coefficients (CQCC) [20,21], derived from long-term constant-Q transform (CQT), emerged as an influential front-end, prompting the proposal of various handcrafted features in that direction [22-24].

In recent years, the application of robust deep learning classifiers has become prevalent in the domain of logical access attack detection. For instance, the squeeze excitation residual network has been utilized in studies such as [25,26], while the deep neural network (DNN) is featured in research like [23,27,28]. Specifically, light convolutional neural networks (LCNN) and the residual network (ResNet) stand out as two of the most frequently employed deep learning classifiers in contemporary literature. LCNN has been adopted in works such as [29-33], whereas ResNet is prominent in studies including [30,33-36].

Typically, systems utilizing LCNN, ResNet, and its derivative, Res2Net [37,38], demonstrate commendable performance, as evidenced in studies like [29,32,34-38].

In addition, some end-to-end systems have been proposed for logical access attacks, for example, AASIST [39], Raw-former [40] and RawBmamba [41].

1.1 The Issue of Replicating Good Results with Deep Neural Networks

Reproducing good results in many machine learning experiments can be challenging, particularly when models are initiated with a random seed number, which dictates initial parameter values of deep neural network to start its training process and significantly influences the performance of trained models. The variability introduced by different seed numbers can lead to different results.To ensure a comprehensive understanding, some studies, like [32], have opted to run their code multiple times, reporting all observed results. It's not uncommon for researchers to struggle to replicate results, even when using identical code and data, if the original outcomes were particularly favorable due to a specific seed number. Attempting to replicate these results can be time-consuming, and occasionally, even the original authors might not be able to reproduce their findings. This variability arises from the nature of random seed numbers. When used, these numbers often determine the initial parameters of deep neural networks. As a result, employing the same training data but different seed numbers can produce distinct models.

When the seed number is fixed during training, results become reproducible. However, this consistency doesn't guarantee optimal outcomes; in some instances, the results might be far from ideal. Aiming to achieve the best model based on the most favorable seed number, while also ensuring that these results can be replicated by ourselves or others, is a primary objective of our study. We delve into strategies to address this challenge.

1.2 The Challenge Associated with Traditional Data Augmentation Methods for Logical Access Attacks

In addition to relying on handcrafted front-end features and robust classifiers, many studies have concentrated on data augmentation as a strategy to enhance performance, especially when confronting the challenge of identifying unknown attack types. For instance, in [42], the authors utilized a parametric sound reverberator and phase shifter on genuine speech samples, simulating unseen conditions pertinent to replayed speech. This work was later expanded upon in the ASVspoof 2019 challenge, where speed perturbation was employed on both genuine and replayed speech, further bolstering the detection of replay attacks [43]. Other studies, such as [44,45], incorporated vocal tract length perturbation alongside speed perturbation, which notably enhanced performance in replay attack detection.

While data augmentation has been employed to address replay or physical access attacks, its application for logical access attacks remains largely unexplored. One plausible explanation for this disparity could be that replay attacks are influenced by the surrounding acoustic environment. Simulating various conditions through data augmentation can thus be instrumental in detecting the unpredictable nature of replay attacks. Conversely, logical access attacks, generated using voice conversion (VC) and text-to-speech (TTS) systems, might not benefit as significantly from conventional data augmentation. This is because such augmentations might inadvertently obscure the subtle differences between genuine and synthesized speech. Motivated by this observation, we aim to devise an effective data augmentation strategy specifically tailored for tackling logical access attacks in this study.

Upon examining the distinctions between bonafide speech and the corresponding synthetic speech produced by TTS or VC, we identified several key differences:

Compared to bonafide speech, some information in the corresponding synthetic speech is lost, as certain characteristics of the source speaker have not been successfully transferred during the VC process.

During VC, the content representation of the source speaker may retain some speaker-specific information, and the identity of the target speaker might include some content information. This is due to the imperfect disentanglement of speaker and content information during the conversion. Consequently, the synthetic speech may carry information from both the source and target speakers simultaneously.

Synthetic speech generated from TTS may lack certain prosodic patterns when compared to genuine speech produced by humans.

We believe that the reasons traditional data augmentation methods using simulated data don't work for logical access attack detection stem from the discrepancies between synthetic and genuine speech. Specifically, the issues of missing information or added new information in synthetic speech compared to the corresponding genuine speech. For a method to effectively detect logical access attacks, it must address the aforementioned challenges.

To solve the above two problems, on the basis of our previous work [35], a method by combing determining the optimal seed number and data augmentation based on signal companding and expanding (SCE) for logical access attack detection is proposed. Specifically,

The proposed determining the optimal seed number addresses the challenge of reproducing good results with deep neural networks. This is accomplished by investigating the correlation between performance and the seed number.

The proposed SCE combines signal companding techniques based on a-law and mu-law with signal expansion approaches using bit-24 and bit-32 representations.

Signal companding can address the challenge of information loss by leveraging the signals derived from a-law and mu-law companding to detect the anomalies characteristic of logical access attacks.

Signal expanding can tackle the introduction of new information by using bit-24 and bit-32 to expand the original 16-bit signal to 24 bits and 32 bits.

Compared to the original training data, the signals obtained through companding represent compressed versions with some information loss, whereas those from signal expanding represent extended versions with added information.

The contributions of the work can be summarized as following:

A method for determining the optimal seed number is proposed, aiming to address the challenges of consistently reproducing good results with deep neural networks. To the best of the authors' knowledge, this is the first study delving into the reproducibility of results in deep neural network contexts.

A novel data augmentation technique, SCE, is proposed for detecting logical access attacks, building upon the optimal baseline determined from the best seed number. Consequently, we employ a-law and mu-law for signal companding and utilize bit-24 and bit-32 methods for signal expansion, thereby enhancing the training data to train more robust models for detecting logical access attacks.

The remainder of the paper is organized as follows. Section 2 introduces the optimal seed number determining . Section 3 discusses the proposed data augmentation based on SCE. Section 4 describes the experiments conducted in the current study. Finally, the paper was concluded.

2 The Optimal Seed Number Determining

The seed number significantly influences the performance of models trained using neural networks. Given identical network architectures, encompassing the same layers, learning rates, loss functions, etc., varying the seed number, which dictates different initial parameter values, can produce distinct network parameters even with the same training set. Consequently, diverse outcomes can emerge on an identical evaluation set.

To achieve the optimal model based on the optimal seed number and ensure reproducibility of results, particularly those of superior performance, we propose a method for determining the optimal seed number. For a system with a designated training set and development set, the method encompasses the following steps:

Switch the seed number from a randomly generated mode to a manually tuned mode within the system

Define a range for the seed numbers, order them from smallest to largest, and assign the minimal seed number value to the system.

Train the system using the designated seed number and the corresponding training set.

Calculate the experimental results on the development set using the trained model.

Select the subsequent seed number value within the range and return to step (3).

Following step (5), we can gather all the experimental results corresponding to each seed number within the specified range.

Choose the seed number that corresponds to the optimal performance as the system's optimal seed number.

For any system with a training set and a development set, using the proposed ``optimal seed number determining" method, the optimal seed number can be obtained and the most suitable baseline for both the training and development sets can be established. In other words, the proposed method can be generalized to other datasets and adapted or extended to work with other DNN models.

3 Data Augmentation Based on SCE

In this study, building on the optimal baseline determined from the best seed number, we proposed a novel approach to data augmentation utilizing SCE techniques. Specifically, signal companding first compresses and then expands the signals, while signal expanding solely enlarges the signals. Moreover, companding is commonly employed for signals with a broad dynamic range, especially when transmitting over facilities with a more limited dynamic range capacity. Companding is prevalent in telephony speech and numerous other audio applications. We explore the a-law and mu-law signal companding methods, which are two renowned standard versions of the G.711 narrowband audio codec from ITU-T. Furthermore, we employ bit-24 and bit-32 as our signal expanding techniques. These methods are detailed in the subsequent subsections.

3.1 Signal Companding

a-law

The a-law companding technique is standardized for European 8-bit PCM digital communications according to ITU-T guidelines. This method compresses the signal's dynamic range, which enhances the coding efficiency. As a result, it achieves a signal-to-distortion ratio superior to what linear encoding provides for an equivalent bit count. For a specific signal x, supposing A=86.5, the a-law encoding proceeds as outlined below:

$$f_a(x) = \operatorname{sgn}(x) A|x| /(1 + In(A)), |x| < 1/A \tag{1}$$

$$f_a(x) = \operatorname{sgn}(x)(1 + In(A|x|)) /(1 + In(A)), 1/A < |x| \le 1 \tag{2}$$

where the compression parameter \$A=\$86.5 on European standards and \${\text {sgn}}(x)\$ is the

mu-law

The mu-law is an alternative standard companding technique, adopted in North America and Japan according to ITU-T guidelines. It offers a marginally broader dynamic range compared to the a-law method. For a designated signal x, mu-law encoding is executed as described below:

$$f_\mu(x) = \operatorname{sgn}(x)(\text{In}(1 + \mu|x|) /(1 + \mu), -1 < x \le 1 \tag{3}$$

3.2 Signal Expanding

bit-24

The 24-bit representation is achieved by extending the original 16-bit signal to 24 bits using the FFMPEG toolkit.

bit-32

The 32-bit representation is derived by expanding the original 16-bit signal to 32 bits using the FFMPEG toolkit.

3.3 The Relationship and Difference between Signal Companding and Signal Expanding

Although both the a-law and mu-law based signal companding processes involve an initial compression step followed by an expansion step, compared to the original 16-bit signal, the output from a-law and mu-law companding is only 8 bits. This implies that some information from the original signal is lost. Hence, we can leverage the signals derived from a-law and mu-law companding to detect the anomalies characteristic of logical access attacks.

In contrast to the two-step process of signal companding, the bit-24 and bit-32 based signal expanding methods involve only one step: expanding the original 16-bit signal to either 24 or 32 bits. This means that these expanded signals contain additional information compared to the original. This added information is precisely why we utilize the bit-24 and bit-32 types of signals to detect the unique characteristics of logical access attacks.

3.4 Combining Signal Companding and Signal Expanding for Data Augmentation

We employ the a-law and mu-law based signal companding, along with the bit-24 and bit-32 based signal expanding techniques, to augment our training dataset. This enhancement aims to develop a robust countermeasure model for detecting the nuanced characteristics of logical access attacks. Notably, SCE doesn't necessitate any supplementary databases for data augmentation, giving it an advantage over traditional augmentation methods that rely on external datasets with noise or room reverberation. Figure1 gives the relationship between the proposed SCE and data augmentation.

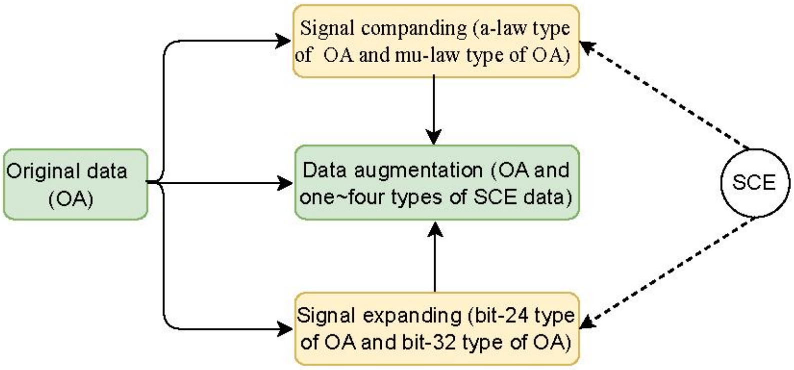
Signal companding (a-law type of OA and mu-law type of OA)
Original data (OA)
Data augmentation (OA and one~four types of SCE data)
SCE
Signal expanding (bit-24 type of OA and bit-32 type of OA)

Fig. 1. The relationship between the proposed SCE and data augmentation.

From Figure 1, it can be seen that the relationship between the proposed SCE and data augmentation, data augmentation consists of original data (OA) and one~four types of SCE data. Next is the augmentation steps:

Step 1: Converting OA into a-law types of OA and mu-law type of OA, respectively.

Step 2: Converting OA into bit-32 type of OA and bit-32 type of OA, respectively.

Step 3: Combing OA and ont~four types of SCE data for data augmentation.

4 Experiment and Evaluation

In this section, the proposed method is evaluated. The corpus details, experimental setup, results, and analysis are discussed in the subsequent subsections.

4.1 Corpus

The ASVspoof 2019 logical access corpus is used for the studies in this work [46]. The database comprises three subsets: training, development, and evaluation sets. The genuine examples in the corpus originate from the VCTK corpus. The corpus features 46 male and 61 female speakers, totaling 107 speakers. There is no speaker overlap among the three subsets. Moreover, the spoofed examples in the evaluation set are crafted using different TTS and VC techniques than those employed for the training and development sets. The ASVspoof 2019 logical access corpus is notably larger and more diverse in terms of attack algorithms than previous synthetic speech or logical access databases [47]. A summary of the ASVspoof 2019 logical access corpus is provided in Table 1.

Table 1. Some detail of ASVspoof 2019 logical access corpus.

Subset	#Male	#Female	#Bonafide	#Spoofed
Training	8	12	2,580	22,800
Development	4	6	2,548	22,296
Evaluation	21	27	7,355	63,882

4.2 Experimental Setup

In this study, following the works [30,33], 84-dim CQT based log power spectrum (LPS) which has 7 octaves and 12 frequency bins in every octave as input to the ResNet-based systems. To obtain a fixed-dimensional LPS, the length is set at 750 frames, achieved either through padding or cropping. This results in an input feature size of 84x750 for each example.

Following the guidelines of the ASVspoof 2019 challenge [46], we use the tandem detection cost function (t-DCF) [48] and equal error rate (EER) as our primary and secondary evaluation metrics respectively. It's worth noting that the ASV-centric t-DCF measure combines our spoofing countermeasure scores with the ASV scores provided with the ASVspoof 2019 corpus. In essence, while the EER evaluates the performance of the logical access attacks detection system on its own, the t-DCF gauges the impact of the logical access attacks detection systems on the reliability of an ASV system [34].

In our experiments, all the ResNet-based classifiers are designed in line with the classic ResNet structure as described in [49] and [50]. Specifically, the architecture begins with a convolutional layer to process the input features and adjust their shape. This is followed by seven residual blocks. Each of these blocks comprises two convolutional layers with a kernel size of 3$\times$7. The design ensures that the input feature of each block is added to its output, addressing potential gradient vanishing issues during the training phase. Notably, apart from the first two residual blocks, the subsequent blocks reduce the size of the feature maps through convolutional strides set at (2,4). The feature maps produced by the final block are transformed into two 128-dimensional features through a global max-pooling layer and a global average-pooling layer. These two feature sets are then concatenated and passed through two fully connected layers. The output from these layers undergoes normalization using the softmax function to yield the final result. It's important to mention that all the residual blocks utilize Leaky ReLu activation functions, with bottleneck layers positioned after every activation function. For training, we've chosen cross-entropy as the loss criterion and employed the Adam optimizer with a momentum of 0.9. The learning rate is set at 0.0001.

There are totally five types of training data used in our experiments, as outlined in Table 2. Specifically, we augment our training data not only using a-law and mu-law based signal companding methods but also employ bit-24 and bit-32 based signal expanding techniques. These augmented data sets aid in training new models for the detection of logical access attacks. In other words, we can increase the amount of training data by 2~5 times with examples derived by selecting one type or several types from a-law, mu-law, bit-24 and bit-32 types of training sets to combine the original training set, furthermore, T_o, T_a, T_{mu}, T_{b24} and T_{b32} stand for the training sets of original type, a-law type, mu-law type, bit-24 type and bit-32 type, respectively. In addition, it should be noted that, in all subsequent experiments, the parameters within the ResNet remain consistent regardless of the number of training set types used.

Table 2. Different types of training data used in our experiments and corresponding method comparison.

Method	Short name	Full name
	T_o	original training set
Signal companding	T_a	a-law type of training set
	T_{mu}	mu-law type of training set
Signal expanding	T_{b24}	bit-24 type of training set
	T_{b32}	bit-32 type of training set

4.3 Development Set Performance with Original Training Set

As highlighted earlier, different seed numbers for the same system can yield varying performances. In this subsection, we'll discuss the process of obtaining the optimal baseline system using the development set. Specifically, we'll outline how to identify the best seed number to achieve this optimal baseline system. To this end, we conducted a series of experiments using the LPS-ResNet system, each with a different seed number, evaluated on the development set.

Following the methodology presented in Section 3, we initially set our seed number range from 100 to 3000, with an interval of 100. Subsequently, we conducted experiments for each seed number within this range. Table 3 displays the experimental results in terms of t-DCF and EER on the development set, utilizing the LPS-ResNet system with different seed numbers and the original training set.

Table 3. Performance with original training set in t-DCF and EER (%) on ASVspoof 2019 logical access development set using LPS-ResNet with different seed num**.**

LPS-ResNet					
#seed num	t-DCF	EER	#seed num	t-DCF	EER
100	0.042	1.26	1600	0.028	0.90
200	0.024	0.78	1700	0.023	0.71
300	0.020	0.63	1800	0.017	0.56
400	0.019	0.67	1900	0.020	0.63
500	0.027	0.82	**2000**	**0.013**	**0.051**
600	0.019	0.59	2100	0.023	0.71
700	0.019	0.58	2200	0.028	0.90
800	0.024	0.75	2300	0.027	0.86
900	0.024	0.75	2400	0.027	0.86
1000	0.024	0.74	2500	0.017	0.56
1100	0.017	0.59	2600	0.028	0.86
1200	0.020	0.67	2700	0.025	0.82
1300	0.019	0.68	2800	0.016	0.62
1400	0.023	0.71	2900	0.027	0.82
1500	0.025	0.95	3000	0.018	0.59

From Table 3, several conclusions can be drawn:

It's observed that different seed numbers yield varying performance. This variability underscores the significance of the seed number in detecting logical access attacks. This further validates the proposed method of leveraging seed numbers on the development set to optimize performance.

The worst result is achieved with a seed number set to 100, where the t-DCF and EER are 0.042 and 1.26%, respectively..

With a seed number set to 2000, we achieve the optimal performance on the development set in terms of both t-DCF and EER, recording their lowest values at 0.013 and 0.051%, respectively. In essence, for the LPS-ResNet, when the ASVspoof 2019 training and development sets are utilized as training and test data respectively, the ideal seed number is determined to be 2000.

4.4 Evaluation Set Performance with Original Training Set

Table 4 presents the results from experiments conducted on the ASVspoof 2019 logical access evaluation set using the LPS-ResNet, with a seed number set to 2000 and employing the original training set.

Table 4. Performance with original training data in t-DCF and EER (%) on ASVspoof 2019 logical access evaluation set using the LPS-ResNet when seed num equals 2000.

System	Training set	#seed num	t-DCF	EER
LPS-ResNet	T_o	2000	0.0.66	2.23

From Table 4, it's evident that the LPS-ResNet, with a seed number set to 2000, achieves a t-DCF of 0.066 and an EER of 2.23%. Compared to the performance on the development set, the results on the evaluation set are slightly inferior. This discrepancy can be attributed to the fact that most logical access attacks in the development set are already present in the training set, whereas many attacks in the evaluation set haven't been encountered in the training set. In other words, from the viewpoint of training set, we can define `known attacks' as logical access attacks present both in the development and evaluation sets that have previously appeared in the training set. Conversely, `unknown attacks' can be defined as logical access attacks in the evaluation set that have not been encountered in the training set.

4.5 Data Augmentation Performance with Two Types of Training Data

Building on the best seed number derived from the development set, this subsection explores the performance of data augmentation with two distinct types of training sets. These training sets are categorized based on whether they utilize the original training data, and can be delineated as: with To (original training data) and without To. Table 5 showcases the performance of data augmentation in terms of t-DCF and EER on the ASVspoof 2019 logical access evaluation set using the LPS-ResNet with a seed number of 2000. To clearly differentiate between LPS-ResNet systems utilizing various data augmentation methodologies, we append 'DA' followed by an index to the system name. Here, `DA' represents data augmentation. The associated indices for `DA' are as follows: `o' for original, `a' for a-law type, `mu' for mu-law type, `b24' for bit-24, `b32' for bit-32, and `bb' for combinations of bit-24 and bit-32 training sets.

Table 5. Data augmentation performance in t-DCF and EER (%) with two types of training sets on ASVspoof 2019 logical access evaluation set using LPS-ResNet when seed num equals 2000.

Type	System	Training sets	t-DCF	EER
With T_o	LPS-ResNet-DA_oa	T_o+T_a	0.065	2.17
	LPS-ResNet-DA_omu	T_o+T_{mu}	0.068	2.15
	LPS-ResNet-DA_ob24	T_o+T_{b24}	0.070	2.21
	LPS-ResNet-DA_ob32	T_o+T_{b32}	0.070	2.22
Without T_o	LPS-ResNet-DA_amu	T_a+T_{mu}	0.077	2.49
	LPS-ResNet-DA_ab24	T_a+T_{b24}	0.064	2.01
	LPS-ResNet-DA_ab32	T_a+T_{b32}	0.064	2.01
	LPS-ResNet-DA_mub24	$T_{mu}+T_{b24}$	0.065	2.11
	LPS-ResNet-DA_mub32	$T_{mu}+T_{b32}$	0.065	2.11
	LPS-ResNet-DA_bb	$T_{b24}+T_{b32}$	0.070	2.22

From Table 5, several conclusions can be drawn:

For the LPS-ResNet-DA systems incorporating T_o, the LPS-ResNet-DA_oa system outperforms LPS-ResNet-DA_omu, LPS-ResNet-DA_ob24, and LPS-ResNet-DA_ob32. This suggests that the combination of T_o and T_a offers more complementary benefits than the pairing of T_o with other data types

For LPS-ResNet-DA systems that exclude T_o, both LPS-ResNet-DA_ab24 and LPS-ResNet-DA_ab32 slightly outperform other configurations. This indicates that the pairing of T_a with T_{b24} (or T_{b32}) provides a more complementary benefit compared to combinations like T_a and T_{mu}, T_{mu} and T_{b24}, T_a and T_{b32}, or T_{b24} and T_{b32}.

Comparing the performance of the LPS-ResNet-DA systems incorporating T_o from Table 5 to the LPS-ResNet in Table 4, it's evident that LPS-ResNet-DA_oa, LPS-ResNet-DA_ab24, LPS-ResNet-DA_ab32, LPS-ResNet-DA_mub24, and LPS-ResNet-DA_mub32 slightly outperform the LPS-ResNet in terms of t-DCF. In terms of EER, all the LPS-ResNet-DA configurations exhibit improved performance compared to LPS-ResNet. This suggests that data augmentation enhances the system's ability to detect more logical access attacks in the evaluation set.

For systems incorporating T_o, LPS-ResNet-DA_oa achieves the best performance with a t-DCF of 0.065 and an EER of 2.17%. Conversely, for systems excluding T_o, the top performers are LPS-ResNet-DA_ab24 and LPS-ResNet-DA_ab32, both registering a t-DCF of 0.065 and an EER of 2.01%.

4.6 Data Augmentation Performance with Three Types of Training Data

In this subsection, we aim to examine the performance of data augmentation across three types of training sets. Consistently, we utilize the best seed number, set at 2000, in our system. Table 6 presents the data augmentation performance metrics-t-DCF and EER-across three training data types for the ASVspoof 2019 logical access evaluation set, using the LPS-ResNet with a seed number of 2000. The results are categorized into two types: those incorporating To and those without. To differentiate between the different training data types, the suffix `DA' followed by its respective index is used for LPS-ResNets. It's worth noting that the indices for `DA', such as `o', `a', `mu', `b24', `b32', and `bb', retain the meanings previously described.

Table 6. Data augmentation performance in t-DCF and EER (%) with three types of training sets on ASVspoof 2019 logical access evaluation set using LPS-ResNet when seed num equals 2000.

Type	System	Training sets	t-DCF	EER
With T_o	LPS-ResNet-DA_oamu	$T_o+T_a+T_{mu}$	0.064	2.16
	LPS-ResNet-DA_oab24	$T_o+T_a+T_{b24}$	0.051	1.72
	LPS-ResNet-DA_oab32	$T_o+T_a+T_{b32}$	**0.051**	**1.71**
	LPS-ResNet-DA_omub24	$T_o+T_{mu}+T_{b24}$	0.057	1.89
	LPS-ResNet-DA_omub32	$T_o+T_{mu}+T_{b32}$	0.059	1.89
	LPS-ResNet-DA_obb	$T_o+T_{b24}+T_{b32}$	0.059	1.94
Without T_o	LPS-ResNet-DA_amub24	$T_a+T_{mu}+T_{b24}$	0.069	2.30
	LPS-ResNet-DA_amub32	$T_a+T_{mu}+T_{b32}$	0.069	2.19
	LPS-ResNet-DA_abb	$T_a+T_{b24}+T_{b32}$	0.063	2.05
	LPS-ResNet-DA_mubb	$T_{mu}+T_{b24}+T_{b32}$	0.075	2.41

From Table 6, several conclusions can be obtained:

The LPS-ResNet-DA systems incorporating T_o consistently outperform those without T_o. This observation diverges from the performance trend seen with two types of training set augmentation. The underlying cause may be the significance of T_o in the context of three training set augmentations. This underscores the necessity of utilizing the original training set in data augmentation to achieve optimal performance when dealing with three types of training set augmentations.

For the LPS-ResNet-DA systems incorporating T_o , all systems with three types of training sets outperform their counterparts with two types of training sets, as seen in Table 5, when evaluated based on t-DCF. In terms of EER, with the exception of LPS-ResNet-DA_oamu, the remaining five systems also surpass the two-type training set systems from Table 5. This suggests that augmenting with three types of training sets provides more complementary information than using just two types, especially for detecting logical access attacks. Moreover, it's evident that the performance of LPS-ResNet-DA_oab24 and LPS-ResNet-DA_oab32 significantly exceeds that of LPS-ResNet-DA_oamu, LPS-ResNet-DA_omub24, LPS-ResNet-DA_omub32, and LPS-ResNet-DA_obb. This can be attributed to the complementary information provided by the combination of T_o , T_a, and either T_{b24} or T_{b32}, making them more effective at detecting logical access attacks compared to other three-type training set augmentations.

In Table 5, it was observed that the combination of T_o and T_a yielded superior performance for systems with T_o . However, in Table 6, augmenting with both T_o and T_a , along with T_{mu} , delivered inferior results compared to augmenting with T_o, T_a, and either T_{b24} or T_{b32}. This discrepancy might be attributed to the unique complementary information present in the combination of T_o, T_a, and either T_{b24} or T_{b32}. In contrast, the T_o, T_a, and T_{mu} combination does not appear to generate as much of this valuable complementary information

The optimal performance is achieved by LPS-ResNet-DA_oab32, registering a t-DCF of 0.051 and an EER of 1.71%. Following closely, LPS-ResNet-DA_oab24 delivers the second-best performance with a t-DCF of 0.051 and an EER of 1.72%.

4.7 Data Augmentation Performance with Four Types of Training Data

In this section, we aim to understand the performance of data augmentation using four types of training sets. Table 7 presents the data augmentation results in terms of t-DCF and EER for four training set configurations on the ASVspoof 2019 logical access evaluation set, utilizing the LPS-ResNet with a seed number set to 2000. As before, the experimental results are categorized based on the inclusion or exclusion of T_o. To differentiate between the LPS-ResNets with various training set configurations, we use the `DA' suffix followed by an index. The indices, such as `o', `a', `mu', `b24', `b32', and `bb', retain the meanings previously discussed.

Table 7. Data augmentation performance in t-DCF and EER (%) with four types of training sets on ASVspoof 2019 logical access evaluation set using LPS-ResNet when seed num equals 2000.

Type	System	Training sets	t-DCF	EER
With T_o	LPS-ResNet-DA_oamub24	$T_o+T_a+T_{mu}+T_{b24}$	0.058	2.03
	LPS-ResNet-DA_oamub32	$T_o+T_a+T_{mu}+T_{b32}$	0.057	1.89
	LPS-ResNet-DA_oabb	$T_o+T_a+T_{b24}+T_{b32}$	**0.050**	**1.60**
	LPS-ResNet-DA_omubb	$T_o+T_{mu}+T_{b24}+T_{b32}$	0.057	1.94
Without T_o	LPS-ResNet-DA_amubb	$T_a+T_{mu}+T_{b24}+T_{b32}$	0.062	2.07

From Table 7, we can draw the following conclusions:

The LPS-ResNet-DA systems utilizing T_o consistently outperform those without T_o, a conclusion previously derived from Table 6.

For the LPS-ResNet-DA systems with T_o, in terms of t-DCF, LPS-ResNet-DA_oabb delivers the best performance. The results for LPS-ResNet-DA_oamub24, LPS-ResNet-DA_oamub32, and LPS-ResNet-DA_omubb are closely matched. Regarding EER, LPS-ResNet-DA_oabb remains the top performer. LPS-ResNet-DA_oamub32 and LPS-ResNet-DA_omubb secure the second and third spots, respectively. However, LPS-ResNet-DA_oamub24 lags behind, registering the least favorable performance.

From Table 6, we observed that the augmentation combination of T_o, T_a, and either T_{b24} or T_{b32} resulted in enhanced performance. However, in Table 7, the inclusion of T_{mu} to the combination of T_o, T_a, and either T_{b24} or T_{b32} led to a significant decline in performance. This suggests that there might be conflicting information introduced with the inclusion of T_{mu}. In contrast, the combination of T_o, T_a, T_{b24}, and T_{b32} seems to provide more synergistic and complementary information.

The optimal performance is achieved by LPS-ResNet-DA_oabb, with a t-DCF of 0.050 and an EER of 1.60%.

4.8 Data Augmentation Performance with Five Types of Training Data

In this section, we aim to evaluate the data augmentation performance utilizing all five types of training sets. Table 8 presents the data augmentation results in terms of t-DCF and EER, using all five training sets on the ASVspoof 2019 logical access evaluation set with the LPS-ResNet when the seed number is set to 2000. Specifically, LPS-ResNet-DA_oamubb indicates that the training leverages To,Ta, Tmu, Tb24, and Tb32.

Table 8. Data augmentation performance in t-DCF and EER (%) with five types of training sets on ASVspoof 2019 logical access evaluation set using LPS-ResNet when seed num equals 2000.

System	Training sets	t-DCF	EER
LPS-ResNet-DA_oamubb	$T_o+T_a+T_{mu}+T_{b24}+T_{b32}$	0.051	1.67

From Table 8, LPS-ResNet-DA_oamubb registers a t-DCF and EER of 0.051 and 1.67%, respectively. When juxtaposed with the results of LPS-ResNet-DA_oabb from Table 7, we note an absolute increase in t-DCF and EER by 0.01 and 0.07%, respectively. This suggests that the inclusion of Tmu with To, Ta, Tb24, and Tb32 introduces some non-beneficial information. Furthermore, when comparing with the results of LPS-ResNet-DA_oab24 and LPS-ResNet-DA_oab32 from Table 6, we observe a slight reduction in EER, while the t-DCF remains unchanged. This implies that the combined use of To, Ta, Tmu, Tb24, and Tb32 offers a more robust capability to detect logical access attacks than simply combining To, Ta, and either Tb24 or Tb32.

4.9 Comparison with Traditional Data Augmentation Methods

In this subsection, we would like to compare the proposed SCE method with traditional data augmentation techniques, including simulated noisy data and speed perturbation. For the simulated noisy data method, we integrate three types of noise to enhance the training data, thereby developing new anti-spoofing models. We utilize the NoiseX-92 database [51] for our comparative noise data augmentation analysis. Specifically, we incorporate three categories of noise: street, volvo, and white noise, each with a signal-to-noise ratio (SNR) of 20 dB. As for speed perturbation, we adopt the approach delineated in [43] and [52], applying a standard 3-way speed perturbation with factors of 0.9, 1.0, and 1.1 during model training.

Table 9 presents a performance comparison between traditional data augmentation techniques and SCE, as measured by t-DCF and EER, on the ASVspoof 2019 logical access evaluation set using LPS-ResNet with a seed number set to 2000. In this table, the abbreviations SND and SP denote simulated noisy data and speed perturbation, respectively. The system names employ the suffix DA and its associated index to differentiate among the LPS-ResNets. The term DA refers to data augmentation, and its indices, i.e., o, s, v, w, sp1.1, and sp0.9, correspond to the original, street noise, volvo noise, white noise, speed perturbation with a factor of 1.1, and speed perturbation with a factor of 0.9 training sets, respectively. Likewise, T_o, T_s, T_v, T_w, $T_{sp1.1}$, and $T_{sp0.9}$ represent the training sets associated with each of these respective categories.

Table 9. Performance comparison between traditional data augmentation methods and SCE in t-DCF and EER (%) on ASVspoof 2019 logical access evaluation set using LPS-ResNet when seed num equals 2000.

Method	System	Training sets	t-DCF	EER
SND	LPS-ResNet-DA_osv	$T_o+T_s+T_v$	0.076	2.39
	LPS-ResNet-DA_osw	$T_o+T_s+T_w$	0.059	2.01
	LPS-ResNet-DA_ovw	$T_o+T_v+T_w$	0.073	2.39
	LPS-ResNet-DA_osvw	$T_o+T_s+T_v+T_w$	0.065	2.19
SP	LPS-ResNet-DA_osp1.1sp0.9	$T_o+T_{sp1.1}+T_{sp0.9}$	0.065	2.22
SCE	LPS-ResNet-DA_oab32	$T_o+T_a+T_{b32}$	0.051	1.71
	LPS-ResNet-DA_oabb	$T_o+T_a+T_{b24}+T_{b32}$	0.050	1.60
	LPS-ResNet-DA_oamubb	$T_o+T_a+T_{mu}+T_{b24}+T_{b32}$	0.051	1.67

From Table 9, several conclusions can be drawn as following:

For the simulated noisy data method, when compared to the performance of LPS-ResNet in Table [4], all systems except LPS-ResNet-DA_ovw, LPS-ResNet-DA_osv, and LPS-ResNet-DA_osvw exhibit improved performance. Additionally, each system among LPS-ResNet-DA_oab32, LPS-ResNet-DA_oabb, and LPS-ResNet-DA_oamubb outperforms LPS-ResNet-DA_ovw in either t-DCF or EER.

Using the speed perturbation method, LPS-ResNet-DA_osp1.1sp0.9 exhibits performance nearly identical to that of LPS-ResNet.

The proposed SCE method significantly outperforms both the simulated noisy data method and the speed perturbation technique. This indicates that traditional data augmentation approaches might not be effective for enhancing performance in detecting logical access attacks. In comparison, the SCE system exhibits superior performance over these conventional augmentation methods.

4.10 Comparison with the State-of-the-art Single Systems

In this subsection, we would like to contrast the proposed SCE system with state-of-the-art single systems evaluated on the ASVspoof 2019 logical access dataset. We've taken into consideration some of the top-performing systems from the ASVspoof 2019 challenge, as well as more recent contributions published after the challenge. These systems employ a variety of front-end features and classifiers.

We first consider the novel front-end systems that utilize single frequency filtering cepstral coefficients (SFFCC), zero time windowing cepstral coefficients (ZTWCC), and instantaneous frequency cepstral coefficients (IFCC), as reported in [53]. In addition, we examine several deep learning systems such as LCNN, ResNet, and deep neural network (DNN). These systems use diverse inputs like constant-Q statistics-plus principal information coefficients (CQSPIC) [54], mel frequency cepstral coefficient (MFCC), LFCC, CQCC, feature genuinization (FG) [31], and LPS of discrete Fourier transform (DFT) and fast Fourier transform (FFT) [28,29,55]. Additionally, we also consider the spoofing identity vector (SIV) [56], extracted from gated recurrent convolutional neural networks (GRCNNs) using log magnitude spectrogram (LMS) and modified group delay (MGD), as well as their respective signal-to-noise mask (SNM) features as inputs.

Of the systems previously mentioned, several are based on ResNet. However, we also consider a modified version of ResNet that supports multiple feature scales, known as Res2Net [37,38]. Specifically, we examine systems such as SE-ResNet50 [38], MCG-Res2Net50 [37], and MLCG-ResNet50 [37]. Here, SE-ResNet50 integrates the squeeze-and-excitation (SE) block with the blocks in Res2Net50. Meanwhile, MCG-Res2Net50 and MLCG-ResNet50 incorporate Res2Net50 with single-group channel-wise gate (SCG) and multi-group latent-space channel-wise gate (MLCG), respectively.

Lastly, we also consider four recent end-to-end systems that use raw audio as input: Res-TSSDNet, RawGAT-ST, AASIST and AASIST-L. Their details are as following:

The Res-TSSDNet is a time-domain synthetic speech detection Net (TSSDNet) that integrates ResNet structures [57].

RawGAT-ST represents a spectro-temporal graph attention network. This network initially learns the relationships between cues across different sub-bands and temporal intervals, followed by a model-level graph fusion of both spectral (S) and temporal (T) dimensions [58].

Audio anti-spoofing using integrated spectro-temporal graph attention networks (AASIST) is modified RawGAT-ST by using heterogeneous stacking graph attention layer and max graph operation.

AASIST-L is the lightweight variant of AASIST.

RawGAT-ST, AASIST and AASIST-L are based on spectro-temporal graph attention network.

Table 10. Performance comparison of the proposed spoofing countermeasure using data augmentation with the state-of-the-art single systems on ASVspoof 2019 logical access evaluation set in terms of t-DCF and EER (%).

Type	Systems	t-DCF	EER
GMM-based	CQCC-GMM [46]	0.237	9.57
	LFCC-GMM [46]	0.212	8.09
	SFFCC-GMM [53]	0.323	13.97
	ZTWCC-GMM [53]	0.141	6.13
	IFCC-GMM [53]	0.357	15.59
	CQSPIC-GMM [28]	0.164	7.74
DNN-based	LFCC-DNN [28]	0.234	9.65
	CQCC-DNN [28]	0.308	12.79
	CQSPIC-DNN [28]	0.183	7.81
ResNet-based	MFCC-ResNet [55]	0.204	9.33
	CQCC-ResNet [55]	0.217	7.69
	(LPS-DFT)-ResNet [55]	0.274	9.68
	LFCC-ResNet [34]	0.059	2.19
LCNN-based	LFCC-LCNN [29]	0.100	5.06
	(LFCC-CMVN)-LCNN [29]	0.183	7.86
	(LPS-FFT)-LCNN [29]	0.103	4.53
	FG-LCNN [31]	0.102	4.07
	LFCC-LCNN-LSTM-sum [32]	0.052	1.92
PLDA-based	SIV-PLDA [56]	0.095	3.85
LCNN-Datt-based	LFCC-LCNN-Datt [59]	0.078	2.76
Capsule Net-based	LFCC-Capsule Net [60]	0.054	1.97
Res2Net-based	CQT-SE-Res2Net50 [38]	0.074	2.50
	CQT-MLCG-Res2Net50 [37]	0.069	2.15
	CQT-MCG-Res2Net50 [37]	0.052	1.78
End-to-end	Raw-audio-Res-TSSDNet [57]	0.048	1.64
	Raw-audio-RawGAT-ST [58]	0.034	1.06
	Raw-audio-AASIST-L [39]	0.031	0.99
	Raw-audio-AASIST [39]	0.028	0.83
The proposed	**LPS-ResNet-DA_oabb**	**0.050**	**1.60**

Table 10 shows the performance comparison of our proposed system with SCE to other known single system results on ASVspoof 2019 evaluation set. In which,

One-class softmax (OC-Softmax) is used as loss function in LFCC-ResNet in [34].

Mean-square error for probability-to-similarity gradient (MSE-for-PSGrad) is used as loss function for LFCC-LCNN-LSTM-sum [32].

PLDA represents probabilistic linear discriminant analysis in SIV-PLDA [56].

Datt stands for dual attention in LFCC-LCNN-Datt [59].

Net stands for network in LFCC-Capsule Net [60].

CQT stands for CQT-based spectrum in the systems based on Res2Net [37,38].

From Table 10, it's evident that our proposed SCE system surpasses the majority of the state-of-the-art single systems in both performance metrics, t-DCF and EER, with the exceptions being the four end-to-end systems. Although our results slightly edge out LFCC-LCNN-LSTM-sum [30] from and CQT-MCG-Res2Net50~[35] in terms of t-DCF, our system demonstrates a significantly enhanced performance when measured by EER. We believe this improvement can be attributed to the use of data augmentation in our approach. This positions our proposed system as a robust anti-spoofing solution capable of addressing the unknown aspects of logical access attacks generated using TTS and VC.

Comparing with Raw-audio-Res-TSSDNet, it can be found that our system is a little worse in terms of t-DCF while a little better in terms of EER, which means that our system can be competitive with Raw-audio-Res-TSSDNet. However, our system performs worse than Raw-audio-RawGAT-ST, Raw-audio-AASIST and Raw-audio-AASIST-L by a large margin, there are several reasons for this:

Raw-audio-RawGAT-ST, Raw-audio-AASIST and Raw-audio-AASIST-L employs both spectral and temporal information extracted directly from raw audio. In contrast, our system solely relies on CQT-based spectral information for its input.

Raw-audio-RawGAT-ST, Raw-audio-AASIST and Raw-audio-AASIST-L can learn the relationship between cues at different sub-bands and temporal intervals by using a novel spectro-temporal graph attention network.

A new graph pooling strategy is used in Raw-audio-RawGAT-ST, which has the ability to reduce the graph dimension and to improve discrimination.

Additionally, the three end-to-end systems benefit from the implementation of a model-level fusion.

In summary, when compared with the current state-of-the-art single systems, our proposed method secures a position among the top several systems on the ASVspoof 2019 evaluation set in terms of t-DCF and EER. This underscores the significance of our approach in determining the optimal seed number and leveraging SCE-based data augmentation.

5 Conclusion

In this study, we initially propose a method to seek the best seed number, addressing the challenge of deriving the optimal model from this number. This ensures the reproducibility of results, particularly superior performances. Building on the optimal baseline, we present SCE for detecting logical access attacks. Furthermore, the proposed SCE is an innovative data augmentation technique that employs a-law and mu-law-based signal companding as well as bit-24 and bit-32-based signal expansion . Evaluations on the ASVspoof 2019 logical access corpus indicate that the proposed data augmentation is highly effective in identifying previously unseen attacks on the evaluation set, especially when compared to non-augmented approaches.

Moreover, when compared with conventional techniques, such as simulated noisy data and speed perturbation-based methods, our approach demonstrates superior effectiveness. In the context of the ASVspoof 2019 logical access corpus, our SCE-augmented system surpasses the majority of the prevailing state-of-the-art single systems. Specifically, in terms of t-DCF and EER, our system ranks among the top three and top two, respectively. In the future, our focus is to adapt and expand the SCE technique to other speech and audio processing applications.

Conflicts of Interest

The authors declare no conflicts of interest.

Reference

1. Kenny, P., Boulianne, G., Dumouchel, P. & Dumouchel, P. Front-end factor analysis versus eigenchannels in speaker verification. IEEE Transactions on Audio, Speech, Lang. Process. 15, 1435−1447 (2007).
2. Dehak, N., Kenny, P. J., Dehak, R., Dumouchel, P. & Quellet, P. Front-end factor analysis for speaker verification. IEEE Transactions on Audio, Speech, Lang. Process. 19, 788−798 (2010).
3. Lei, Y., Scheffer, N., Ferrer, L. & Mclaren, M. A novel scheme for speaker recognition using a phonetically-aware deep neural network. In IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 1695−1699,(Florence, Italy, 2014).

4. Snyder, D., Garcia-Romero, D., Povey, D. & Khudanpur, S. Deep neural networks embeddings for text-independent speaker verification. In 18th Annual Conference of the International Speech Communication Association(INTERSPEECH), 999−1003 (Stockholm, Sweden, 2017).
5. Snyder, D., Garcia-Romero, D., Sell, G., Povey, D. & Khudanpur, S. X-vectors: robust DNN embeddings for speaker recognition. In IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP), 5329−5333 (Calgary,Canada, 2018).
6. Ioffe, S. Probabilistic linear discriminant analysis. In European Conference on Computer Vision (ECCV), 531−542 (Graz,Austria, 2006).
7. Lee, K. A., Ma, B. & Li, H. Speaker verification makes its debut in smartphone. In SLTC Newsletter (February 2013).
8. Das, R. K., J., S. & S. R. M, P. Development of multi-level speech based person authentication system. J. Signal Process. Syst. 88, 259−271 (2017).
9. Jelil, S., Shrivastava, A., Das, R. K., Prasanna, S. R. M. & Sinha, R. SpeechMarker: A voice based multi-level attendance application. In 20th Annual Conference of the International Speech Communication Association, 3665−3666 (2019).
10. Wu, Z. & Li, H. On the study of replay and voice conversion attacks to text-dependent speaker verification. Multimedia Tools Applications. 75, 5311−5327 (2016).
11. Das, R. K., Tian, X., Kinnunen, T. & Li, H. The attacker's perspective on automatic speaker verification: An overview. In 21st Annual Conference of the International Speech Communication Association, 4213−4217 (2020).
12. Wu, Z. et al. Spoofing and countermeasures for speaker verification: A survey. Speech Commun. 66, 130 − 153 (2015).
13. Zhao, Y. et al. Voice conversion challenge 2020—intra-lingual semiparallel and cross-lingual voice conversion . In ISCA Joint Workshop for the Blizzard Challenge and Voice Conversion Challenge 2020, 80−98 (2020).
14. Lorenzo-Trueba, J. et al. Can we steal your vocal identity from the internet?: Initial investigation of cloning Obama's voice using GAN, WaveNet and low-quality found data. In Odyssey, 240−247 (2018).
15. Das, R. K. et al. Predictions of subjective ratings and spoofing assessments of voice conversion challenge 2020 submissions. In ISCA Joint Workshop for the Blizzard Challenge and Voice Conversion Challenge 2020, 99−120 (2020).
16. Todisco, M. et al. ASVspoof 2019: Future horizons in spoofed and fake audio detection. In 20th Annual Conference of the International Speech Communication Association, 1008−1012 (2019).
17. Das, R. K., Yang, J. & Li, H. Assessing the scope of generalized countermeasures for anti-spoofing. In IEEE International Conference on Acoustic, Speech and Signal Processing, 6589−6593 (2020).
18. Sahidullah, M., Kinnunen, T. & Hanilci, C. A comparison of features for synthetic speech detection. In 16th Annual Conference of the International Speech Communication Association, 2087−2091 (2015).
19. Patel, T. B. & Patil, H. A. Combining evidences from mel cepstral, cochlear filter cepstral and instantaneous frequency features for detection of natural vs. spoofed speech. In 16th Annual Conference of the International Speech Communication Association, 2062−2066 (2015).
20. Todisco, M., Delgado, H. & Evans, N. A new feature for automatic speaker verification anti-spoofing: constant Q cepstral coefficients. In the speaker and language recognition workshop, 283−290 (2016).
21. Todisco, M., Delgado, H. & Evans, N. Constant Q cepstral coefficients: A spoofing countermeasure for automatic Speaker verification. Comput. Speech & Lang. 45, 516−535 (2017).
22. Yang, J., Das, R. K. & Zhou, N. Extraction of octave spectra information for spoofing attack detection. IEEE/ACM Transactions on Audio, Speech Lang. Process. 27, 2373−2384 (2019).
23. Yang, J., Das, R. K. & Li, H. Significance of subband features for synthetic speech decetion. IEEE Transactions on Information Forensics Security 15, 2160−2170 (2020).
24. Yang, J. & Das, R. K. Long-term high frequency features for synthetic speech detection. Digit. Signal Process. 97 (Feb,17.2020).
25. Lai, C.-I., Chen, N., Villaba, J. & Dehak, N. ASSERT:Anti-spoofing with squeeze-excitation and residual networks. In Annual Conference of the International Speech Communication Association, 1013−1017 (Graz, Austria, 2019).
26. Monteiro, J. & Alam, J. Development of voice spoofing detection systems for 2019 edition of automatic speaker verification and countermeasures challenge. In IEEE Automatic Speech Recognition and Understanding Workshop 2019, 1003−1010 (2019).

27. Das, R. K., Yang, J. & Li, H. Long range acoustic features for spoofed speech detection. In 20th Annual Conference of the International Speech Communication Association, 1058−1062 (2019).

28. Das, R. K., Yang, J. & Li, H. Long range acoustic and deep features perspective on ASVspoof 2019. In Automatic Speech Recognition and Understanding Workshop, 1018−1025 (2019).

29. Lavrentyva, G. et al. STC antispoofing systems for the ASVspoof2019 challenge. In 20th Annual Conference of the International Speech Communication Association, 1033−1037 (2019).

30. Yang, Y. et al. The SJTU robust anti-spoofing system for the ASVspoof 2019 challenge. In 20th Annual Conference of the International Speech Communication Association, 1038−1042 (2019).

31. Wu, Z., Das, R. K., Yang, J. & Li, H. Light convolutional neural network with feature genuinization for detection of synthetic speech attacks. In 21st Annual Conference of the International Speech Communication Association, 1101−1105 (2020).

32. Wang, X. & Yamagishi, J. A comparative study on recent neural spoofing countermeasures for synthetic speech detection. In 22nd Annual Conference of the International Speech Communication Association (2021).

33. Yang, J., Wang, H., Das, R. K. & Qian, Y. Modified magnitude-phase spectrum information for spoofing detection.IEEE/ACM Transactions on Audio, Speech Lang. Process. 29, 1065−1078 (2021).

34. Zhang, Y., Jiang, F. & Duan, Z. One-class learning towards generalized voice spoofing detection. IEEE Singal Process Letters, 28, 937−941 (2021).

35. Das, R. K., Yang, J. & Li, H. Data augmentation with signal companding for detection of logical access attacks. In IEEE International Conference on Acoustic, Speech and Signal Processing, 6349−6353 (2021).

36. Hua, G., Teoh, A. B. J. & Zhang, H. Towards end-to-end synthetic speech detection. IEEE Signal Process. Lett. 28,1265−1269 (2021).

37. Li, X., Wu, X., Lu, H., Liu, X. & Meng, H. Channel-wise gated Res2Net: towards robust detection of synthetic speech attacks. In 22nd Annual Conference of the International Speech Communication Association (2021).

38. Li, X. et al. Replay and synthetic speech detection with Res2Net aritecture. In IEEE International Conference on Acoustic, Speech and Signal Processing, 6354−6358 (2021).

39. w. Jung, J. et al. AASIST: audio anti-spoofing using integrated spectro-temporal graph attention networks. In IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 6367−6371 (Singapore, Singapore,2022).

40. X. Liu, L. W., M. Liu & etc. Leveraging positional-related local-global dependency for synthetic speech detection. In 2023 IEEE International Conference on Acoustics, Speech and Signal Processing, 1 − 5 (IEEE, 2023).

41. Y. Chen, J. X., J. Yi & etc. Rawbmamba: End-to-end bidirectional state space model for audio deepfake detection. arXiv preprint arXiv:2406.06086v2 (2024).

42. Cai, W., Cai, D., Liu, W., Li, G. & Li, M. Countermeasures for automatic speaker verification replay spoofing attack : On data augmentation, feature representation, classification and fusion. In 18th Annual Conference of the International Speech Communication Association, 17−21 (2017).

43. Cai, W., Wu, H., Cai, D. & Li, M. The DKU replay detection system for the ASVspoof 2019 challenge: On data augmentation, feature representation, classification, and fusion. In 20th Annual Conference of the International Speech Communication Association, 1023−1027 (2019).

44. Zhao, Y., Togneri, R. & Sreeram, V. Data augmentation and post selection for improved replay attack detection. In Asia-Pacific Signal and Information Processing Association Annual Summit and Conference 2019, 818−821 (2019).

45. Zhao, Y., Togneri, R. & Sreeram, V. Replay anti-spoofing countermeasure based on data augmentation with post selection. Computer, Speech & Language. 64, 101115 (2020).

46. Todisco, M. et al. ASVspoof 2019: Future horizons in spoofed and fake audio detection. In 20th Annual Conference of the International Speech Communication Association, 1008−1012 (2019).

47. Nautsch, A. et al. ASVspoof 2019: Spoofing countermeasures for the detection of synthesized, converted and replayed speech. IEEE Transactions on Biom. Behav. Idemtity 3, 252−265 (2021).

48. Kinnunen, T. et al. t-DCF: a detection cost function for tandem assessment of spoofing countermeasures and automatic speaker verification. In The speaker and language recognition workshop, 312−319 (2018).

49. He, K., Zhang, X., Ren, S. & Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, 770−778 (2016).

50. Wu, Q., Xiong, S. & Zhu, Z. Replay speech answer-sheet on intelligent language learning system based on power spectrum decomposition. IEEE Access 9, 104197−104204 (2021).

51. Varga, A. & Steeneken, H. J. Assessment for automatic speech recognition: II. NOISEX-92: A database and an experiment to study the effect of additive noise on speech recognition systems, Speech Communication, 12, 247 − 251 (1993).

52. Ko, T., Peddinti, V., Povey, D. & Khudanpur, S. Audio augmentation for speech recognition. In 16th Annual Conference of the International Speech Communication Association, 3586–3589 (2015).

53. Alluri, K. N. R. K. R. & Vupala, A. K. IIIT-H spoofing countermeasures for automatic speaker verification spoofing and countermesures challenge 2019. In 20th Annual Conference of the International Speech Communication Association, 1043−1047 (2019).

54. Yang, J., You, C. & He, Q. Feature with complementarity of statistics and principal information for spoofing detection. In 19th Annual Conference of the International Speech Communication Association, 651−655 (2018).

55. Alzanto, M., Wang, Z. & Srivastava, M. B. Deep residual neural networks for audio spoofing detection. In 20th Annual Conference of the International Speech Communication Association, 1078−1082 (2019).

56. Gomez-Alanis, A., Peinado, A. M., Gonzalez, J. A. & Gomez, A. M. A gated recurrent convolutional neural network for robust spoofing detection. IEEE/ACM Transactions on Audio, Speech Lang. Process. 27, 1985− 1999 (2019).

57. Hua, G., Teoh, A. B. J. & Zhang, H. Towards end-to-end synthetic speech detection. IEEE Singal Process. Lett. 28,1265−1269 (2021).

58. Tak, H. et al. End-to-end spectro-temporal graph attention networks for speaker verification anti-spoofing and speech deepfake detection. In Proc. ASVspoof workshop (2021).

59. Ma, K., Liang, T., Zhang, S., Huang, S. & He, L. Improved lightCNN with attention modules for ASV spoofing detection.In IEEE International Conference on Multimedia and Expo, 1−6 (2021).

60. Luo, A., Li, E., Liu, Y., Kang, X. & Wang, Z. J. A capsule network based approach for detection of audio spoofing attacks.In IEEE International Conference on Acoustic, Speech and Signal Processing, 6359 − 6363 (2021).