IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 21 April 2023, accepted 3 May 2023, date of publication 12 May 2023, date of current version 24 May 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3275790

RESEARCH ARTICLE

Voice Spoofing Detection Through Residual Network, Max Feature Map, and Depthwise Separable Convolution

IL-YOUP KWAK[1], (Member, IEEE), SUNGSU KWAG[2], JUNHEE LEE[2], YOUNGBAE JEON[2], JEONGHWAN HWANG[3], HYO-JUNG CHOI[1], JONG-HOON YANG[1], SO-YUL HAN[1], JUN HO HUH[2], CHOONG-HOON LEE[2], AND JI WON YOON[3]

[1]Department of Applied Statistics, Chung-Ang University, Seoul 06974, South Korea
[2]Samsung Research, Seoul 06765, South Korea
[3]School of Cybersecurity, Korea University, Seoul 02841, South Korea

Corresponding author: Jun Ho Huh (junho.huh@samsung.com)

This work was supported in part by the National Research Foundation of Korea (NRF) grant funded by the Ministry of Science and Information Communication Technology (ICT) under Grant RS-2023-00208284, and in part by Chung-Ang University Research Grant in 2022.

ABSTRACT The goal of the "2019 Automatic Speaker Verification Spoofing and Countermeasures Challenge" (ASVspoof) was to make it easier to create systems that could identify voice spoofing attacks with high levels of accuracy. However, model complexity and latency requirements were not emphasized in the competition, despite the fact that they are stringent requirements for implementation in the real world. The majority of the top-performing solutions from the competition used an ensemble technique that merged numerous sophisticated deep learning models to maximize detection accuracy. Those approaches struggle with real-world deployment restrictions for voice assistants which would have restricted resources. We merged skip connection (from ResNet) and max feature map (from Light CNN) to create a compact system, and we tested its performance using the ASVspoof 2019 dataset. Our single model achieved a replay attack detection equal error rate (EER) of 0.30% on the evaluation set using an optimized constant Q transform (CQT) feature, outperforming the top ensemble system in the competition, which scored an EER of 0.39%. We experimented using depthwise separable convolutions (from MobileNet) to reduce model sizes; this resulted in an 84.3 percent reduction in parameter count (from 286K to 45K), while maintaining similar performance (EER of 0.36%). Additionally, we used Grad-CAM to clarify which spectrogram regions significantly contribute to the detection of fake data.

INDEX TERMS Voice assistant security, voice spoofing attack, voice synthesis attack, voice presentation attack detection.

I. INTRODUCTION

As voice assistants continue to rapidly improve to support security- and privacy-sensitive tasks, such as sending and receiving emails, processing payments, taking pictures, and posting on social media, they become highly desirable targets for attackers. The most straightforward and accessible approach for attackers to take advantage of voice assistants is probably through "voice replay attacks." These attacks leverage voice assistant usage recordings to circumvent voice biometric verification features on the victim's target devices by simply playing them back loudspeaker-to-loudspeaker. Advanced attacks known as "voice synthesis (or conversion) attacks" entail gathering voice samples from the target, using machine learning to train the victim's voice biometric models, and then creating new voice attack samples. To train voice models, you can use free resources like Google's Wavenet, and Tacotron [1], [2], [3].

"Automatic speaker verification spoofing and countermeasures challenge" (ASVspoof) has been organized

The associate editor coordinating the review of this manuscript and approving it for publication was Mounim A. El Yacoubi.

since 2015 to encourage researchers to develop voice spoofing attack detection systems and to compete for the improvement of voice detection accuracy. Competitions took place in 2015, 2017, 2019, and 2021 [4], [5], [6], [7].

In this study, we evaluate the performance of our solution to the top-performing ones from the competition using the ASVspoof 2019 dataset. ASVspoof 2019 offers two distinct attack sets: (1) a physical access (PA) attack set produced by playing back recorded voices through loudspeakers and intended to thwart ''voice replay attacks,'' and (2) a logical access (LA) attack set, which is intended to prevent ''voice synthesis attacks,'' and synthetic generation by training victims' voice. The trained voice models directly played to the target voice assistant system. The generation of the second set does not involve recording or replaying. The model that performed the best in comparison to the ASVspoof 2019 PA set had an equal error rate (EER) of 0.39 %.

Many participants in the competition used an ensemble approach involving the use of multiple deep learning models. One ensemble solution that combined multiple light convolutional neural network (LCNN) models achieved an EER of 0.54% on the PA set, and an EER of 1.86% on the LA set [8]. An ensemble solution that used multiple residual network (ResNet) models with the squeeze and excitation (SE) technique achieved an EER of 0.59% for the PA set, and 6.7% for the LA set [9].

The complexity and latency demand that businesses place on models might make such ensemble (multi-model) solutions difficult to use, despite their competitive accuracy outcomes (low EERs). Businesses often need model sizes to be fewer than a few megabytes (including taking into account on-device deployment situations), and the detection (prediction) time to be less than 100 ms due to consumers' expectations of rapid responses and difficulties with ballooning server costs.

To satisfy those business requirements, Kwak et al. propose a deep learning architecture called ''ResMax'' that combines the skip connection concept from ResNet with the max feature map concept from LCNN [10]. By using optimized constant Q transform (CQT) feature, ResMax achieved an EER of 0.37% on the PA set and 2.19% on the LA set. Compared with the top-performing solutions from the ASVspoof 2019 competition that used ensemble approaches (multiple heavy and complex deep learning models), a single ResMax model is capable of achieving the top EER on the PA set and would be ranked third among the LA set solutions.

To extend our prior work, we included three additional experiments in this paper. First, to highlight regions in spectrograms that are significantly contributing to spoof attack detection, we applied gradient-weighted class activation mapping (Grad-CAM [11]) to our models. The three key observations were: 1) Vertically extracted frequency features are activated in PA models, 2) horizontally extracted time-series features are activated in LA models, and 3) the human voice (low) frequency region is activating important features for both PA and LA models but is more significantly impacting the performance of LA models.

Second, we applied depthwise separable convolutions to train a lighter model: We reduced the number of parameters used in the original ResMax model from 286K to 45K while maintaining 0.36% EER on the PA data. The EER increased slightly from 0.30% to 0.36%. Similarly, the EER for the LA data increased slightly from 2.19% to 2.55%.

Third, our revised ''ResMax'' architecture, which combines the ideas of skip connection and max feature map achieved state-of-the art performance (0.30% EER) on the PA set. The best ensemble model from the competition achieved an EER of 0.39%, and the top-performing single model solution that used the LCNN architecture with fast Fourier transform as the feature achieved an EER of 0.5% on the PA set [6]. In the LA set, our ''ResMax'' model achieved 2.19% EER. It was 3rd place after T05 and T45 systems, and 1st place among single models.

II. METHODOLOGY AND MATERIALS

A. FEATURE ENGINEERING

First, we tested with the constant Q transform (CQT) and the short time Fourier transform (STFT), two of the most popular spectral features in the ASVspoof 2017 and 2019 competitions [5], [6], [12]. After a few attempts and monitoring model accuracy, we discovered that the CQT was substantially more accurate and more in line with the suggested model. To increase the model's accuracy, we concentrated on fine-tuning the CQT parameters. By breaking the signal up into smaller frames and evaluating the audio in the frequency domain, the CQT investigates an audio signal $x[n]$ in a time-frequency representation.

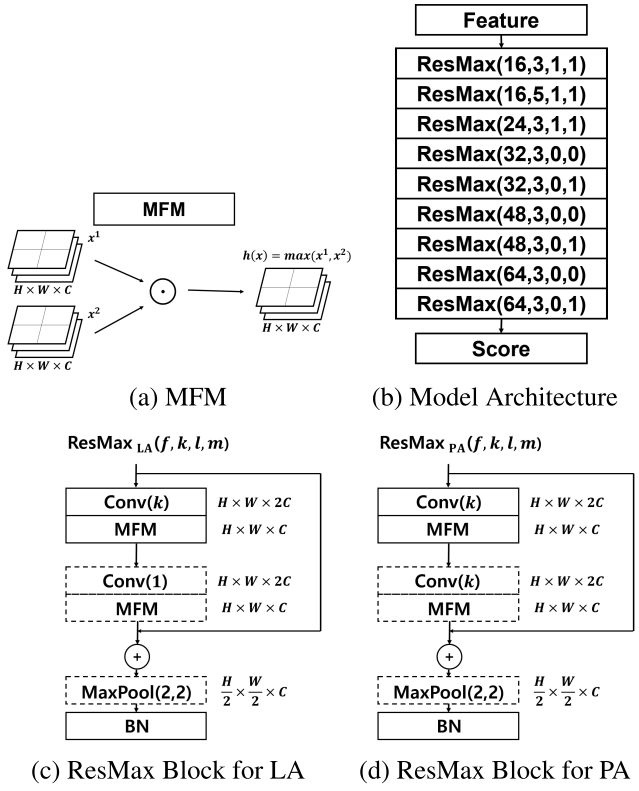
The discrete form of CQT output is as follows:

$$X_l = \sum_{n=0}^{N_l-1} W_{l,n} x_n e^{-j2\pi Qn/N_l}, \tag{1}$$

where l is a frequency bin's index, ranging from 1 to the total number of frequency bins (L); N_l is the size of the bin's frame; W is a windowing function used to taper each frame, and Q is a quality factor that affects the resolution of features.

The CQT uses filters like the center frequency f_l and bandwidth of B_l to the lth frequency domain to convert a given data series into a frequency domain. The centre frequency of the lth filter is $f_l = (2^{1/C})^{l-1} f_{min}$, where f_{min} is the bandwidth of the lowest frequency, and C is the number of octaves in each filter. We calculate Q by using f_l/B_l.

We attempted frequency adjustments in the f_{min} and L parameters for CQT optimization. The highest centre frequency, f_L, was determined using $(2^{1/C})^{L-1} f_{min}$. To examine the consequences of changing low-frequency components, we examined two f_{min} values, 1Hz and 32Hz. In order to study the effects of high-frequency components, we also experimented with different L values to try f_L values at 323Hz and 1024Hz. We fixed the number of bins per octave and the hop size to 12 and 512, respectively, to provide an accurate



(a) MFM (b) Model Architecture

(c) ResMax Block for LA (d) ResMax Block for PA

FIGURE 1. ResMax architecture descriptions: (a) represents an MFM layer; (b) describes the entire model architecture; (c) and (d) represent $\text{ResMax}_{\text{LA}}$ and $\text{ResMax}_{\text{PA}}$, building blocks for the entire model on PA and LA data respectively. The blocks have four parameters: f is the number of ResMax filters, k is the kernel size (k, k) in the convolution layer. l is 1 if a ResMax block has an additional convolution layer followed by an MFM layer (dotted Conv and MFM blocks in (c) and (d)), and otherwise is 0. m is 1 if a ResMax block has an additional max pooling layer (dotted MaxPool block in (c) and (d)), and otherwise is 0.

comparison with constant resolution in the time and frequency domains. The windowing function that we employed was the Hann window.

The duration of each sample was fixed to nine seconds. Any samples longer than nine seconds were truncated after the first nine seconds, while samples shorter than nine seconds were extended by appending audio from the beginning. No normalization techniques were utilized.

B. ResMax: RESIDUAL NETWORK WITH MAX FEATURE MAP

When tested against the ASVspoof 2017 and ASVspoof 2019 evaluation datasets, both LCNN and ResNet-based models demonstrated excellent results with regard to EERs [8], [9], [13], [14].

However, most of the top-performing solutions have combined many deep learning models into an ensemble in order to reduce error rates. Such models do not take into account the criteria for model latency and complexity. Kwak et al. [10] suggested the "**Res**idual network with **Max** feature map" (ResMax) blocks that integrate the max feature map (MFM) ideas from LCNN and the skip connection concept from ResNet.

TABLE 1. Ensemble solutions from ASVspoof 2019 and the models used.

Model	Data	All models used
T10 [15]	PA	LFCC-ResNet, GD gram-ResNet .. Joint gram-ResNet
T44 [16]	PA	logspec-SENet34, CQCC-ResNet .. logspec-SENet50
T45 [8]	LA	LFCC-LCNN, LFCC-CMVN-LCNN .. CQT-LCNN
T50 [17]	PA	CQT-LCNN, LFCC-LCNN, DCT-LCNN
	LA	CQT-CGCNN, CQT-ResNet18 .. CQT-ResNet18IVec
T60 [18]	PA	FFT-CNN, FFT-CRNN, IMFCC-GMM, SVMs-IVec

The top-performing ensemble models [8], [15], [16], [17], [18] that competed in the ASVspoof 2019 are listed in Table 1. The T10 model with ResNet architecture has 36 layers and employs the data augmentation strategy that ranks fourth on the PA data. The T10 model was an ensemble model composed of six ResNet models with distinct characteristics such as STFT, IMFCC, LFCC, GD gram, and Joint gram [15]. Using the CQCC and logspectogram features with ResNet, SENet, and Dilated ResNet architectures, the T44 model came in third place on the PA data. In the T44 model, five models were ensembled, and the best model used the logspectogram feature with SENet34 [16]. Using an ensemble method with features such as FFT, CQT, and LFCC, the T45 model came in second for both the PA and LA data. Among all their models, FFT-LCNN and CQT-LCNN models performed the best on the LA data and PA data, respectively [8]. The T50 model came in fifth on the LA data and attempted data augmentation on the CQT feature using variational autoencoder [17]. The T60 model, which came in third place on the LA data, combined the FFT feature with CNN, CRNN, 1D-CNN, and Wave-U-Net, as well as shallow models of ivector-SVM and IMFCC-GMM [18].

According to the common characteristics of the models that performed well in the competition, there are numerous ResNet models that use skip connection and LCNN models that use MFM activation.

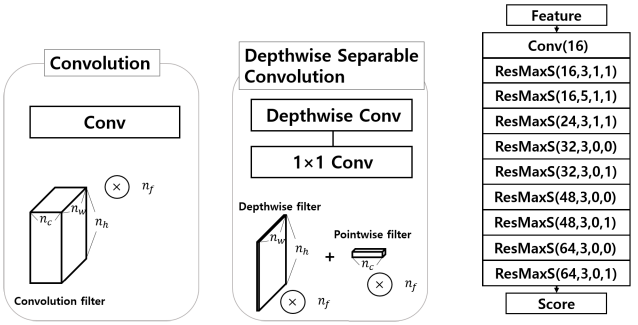
Wu et al [19] suggested using MFM activation as shown in Fig. 1(a) instead of relu activation after the convolution layer, and proposed LCNN architecture. As shown in Fig. 1(a), the MFM layer initializes two tensors with the same dimensions and selects one larger item from the same position in the two tensors.

Figure 1 (b) illustrates the complete model structure consisting of nine ResMax blocks. After the final ResMax block, there is a global average pooling layer followed by a dense layer with softmax activation. To avoid increasing the number of parameters, we skipped using fully connected layers (a series of dense layers) and directly attached the output layer after the convolution layer.

The Fig. 1 (c) and (d) represent $\text{ResMax}_{\text{LA}}$ and $\text{ResMax}_{\text{PA}}$, building blocks for the entire model on PA and LA data respectively. The second convolution layer inside of the skip connection network has 1×1 for LA data, and $k \times k$ filter for PA data.

A skip connection adds original input to a processed network, $F(x)$, where output = input + $F(input)$. Here, training

the weight parameters in $F(input)$ imply training the residual (output − input). This residual part would often have values close to zero and helps to solve the vanishing gradient problem during back-propagation. Furthermore, the 2 × 2 MaxPooling layer used in a ResMax block consequently reduces the number of parameters and the model complexity.



(a) Filter comparison (b) ResMaxSep

FIGURE 2. ResMaxSep architecture descriptions: (a) compares convolution filters with depthwise separable convolution filters; (b) represents the entire ResMaxSep model architecture. In the ResMaxS block, the convolution layers in the ResMax block are replaced by depthwise separable convolution layers. All other parameters are the same.

C. ResMaxSep: ResMax WITH DEPTHWISE SEPARABLE CONVOLUTION

Depthwise separable convolutions have been proposed as a method for making traditional convolution layers lighter [20], [21], [22]. The main idea behind depthwise separable convolution is the separation of the traditional convolution into the *depthwise* and *pointwise* convolutions – the effect is a significant reduction in the number of parameters used in a model and overall computational overheads. Fig. 2 (a) compares the traditional convolution and the depthwise separable convolution. Traditional convolutions use a $n_w \times n_h \times n_c$ dimension filter as seen in Fig. 2 (a). If we use n_f numbers of filters, the total number of parameters for a convolution layer becomes $n_w \times n_h \times n_c \times n_f$. However, if we use depthwise separable convolution, a depthwise convolution would need $n_w \times n_h \times n_f$ number of parameters, and a pointwise convolution would need $n_c \times n_f$ number of parameters. Consequently, we can dramatically reduce the number of parameters that a model needs as well as the model training and prediction overheads. If we are using a 3 × 3 convolution filter, for example, the computational cost can be reduced by about eight to nine times [20].

From the original ResMax blocks, convolution layers are converted to depthwise separable convolution layers, as shown in Fig. 1 (b); this new block is referred to as the "ResMaxS" block. Fig. 2 (b) illustrates the entire ResMaxSep model architecture using the ResMaxS block. We added one convolution layer with a 3 × 3 filter at the beginning of the architecture; the rest of the architecture and layout shapes are identical to the original shown in Fig. 1 (c).

D. GRAD-CAM FOR SPOOFING DETECTION

Grad-CAM is a popular technique used to visualize specific regions in given images that are significantly contributing to the performance of CNN models [11]. It is a generalized version of the previously released technique called CAM [23].

We state our proposed ResMax model as $f(x; \hat{\theta})$: x is the input CQT feature, and $\hat{\theta}$ is the estimated parameter vector that minimizes the total cost, $C(\theta; X, Y)$, where X is a set of CQT features from the training data, and Y is the corresponding binary classification label in the training set. To apply the Grad-CAM on a specific convolution layer, one separates the ResMax model ($f(\cdot)$) with $h(\cdot)$ and $g(\cdot)$–$f(x; \theta) = h(g(x; \theta_g); \theta_h)$, where $A = g(x; \theta_g)$ is the output of a specific convolution layer, θ is a parameter vector for model $f(\cdot)$, θ_g is a subset of θ that is used for function $g(\cdot)$, and θ_h is a subset of θ that is used for function $h(\cdot)$. Our aim in using Grad-CAM is to visually highlight parts of a given convolution layer output ($A = g(x; \theta_g)$) that are significantly affecting the prediction performance. To achieve this, we differentiate $h(A; \hat{\theta}_h)$ with respect to A (i.e., $dhA = \partial h(A)/\partial A$). Here, dhA is a three dimensional array: parts that are contributing more to the classification performance would have higher values. Next, we calculate channel wise importance, w_c, using channel wise average pooling dhA. By taking w_c as weights for channel information in A, we can calculate Grad-CAM as follows:

$$L_{\text{Grad-CAM}} = ReLU(\sum_c w_c A_c)$$

where A_c is a two dimensional matrix from the cth channel of three dimensional layer output matrix A, and ReLU is a rectified linear unit. The view of $L_{\text{Grad-CAM}}$ over the input image x is known as the Grad-CAM visualization [11].

One of the objectives of Grad-CAM analysis is to identify frequency ranges that are important in detecting voice spoofing attacks. To use the entire duration of a given sample for spoofing attack detection, we aggregate information on the frequency axis by averaging the Grad-CAM results over the time axis – this aggregation allows identification of important frequency ranges over an entire sample.

E. HYPER PARAMETER TUNING

We utilized the cost-sensitive learning technique [24] to address the imbalanced distribution of attack and genuine samples in the ASVspoof 2019 training set. Our primary metric for evaluation was the equal error rate (EER), which assign equal importance to spoofing attack and genuine data. Therefore, we multiplied weights to the minority class (genuine samples) to ensure their equal influence on the classification model as the majority class (spoofing attack samples). This approach did not alter the overall training time as it simply multiplied weights in the loss function.

We used the grid search method to find the best parameter set for minimizing the EER on the development set. The optimal collection of hyperparameters included feature extraction parameters, cost-sensitive learning parameters, classification

algorithm-specific parameters, and regularization parameters for distinguishing genuine samples from spoofed samples. We used binary cross entropy error loss with ADAM optimizer [25]. Initial weights were set using glorot's uniform initializer [26].

III. EXPERIMENTS

A. EXPERIMENTAL SETUP

To illustrate the ResMax model's performance, it must be compared to existing models that performed well in the ASVspoof 2019 challenge. We used the ASVspoof 2019 competition data for development and evaluation to compare the performance of the ResMax and ResMaxSep models with the ensemble models specified in Table 1.

We used LA and PA data from the ASVspoof 2019 competition to evaluate the performance [6]. There was a training set, a development set, and an evaluation set for each LA and PA data. Using the training sets, we trained the ResMax and ResMaxSep models and tested them on the development and evaluation sets. We utilized the EER metric as the primary measure of accuracy for detecting spoofing, in accordance with the regulations of the ASVspoof 2019 competition. FRR represented the loudspeaker's misclassification rate, while FAR represented the rate at which human voices were misclassified as legitimate. Our binary classifiers generated scores indicating the likelihood of a command being spoken through a loudspeaker, and scores for each command in an evaluation set were calculated. There was a trade-off between FAR and FRR since reducing one error rate increased the other. The EER for a model was determined by the point at which FAR equaled FRR for a given command set. Additionally, we employed the minimum normalized tandem detection cost function (t-DCF) [27], which was also used (and reported) in the competition. The t-DCF values were obtained using a fixed automatic speaker verification method provided by the competition.

We trained the proposed models using 100 epochs and 10 training and evaluation sessions because the models converged to slightly different parameters with each run.

B. EXPERIMENTAL RESULTS

Table 2 compares our ResMax models to the top five performers from ASVspoof 2019 (including multi-model and single-model solutions) in terms of EERs derived from the evaluation set. In the last few rows, we additionally give the EERs of the top-performing single-model solutions; these do not have ranks and are in italics. According to the results, CQT-ResMax outperformed all other single models in terms of both EERs and t-DCFs for both the LA and PA evaluation sets. It has a statistically significant EER advantage over the best performing PA and LA single models ($p < 0.0001$, one sample t-test). The development set evaluation also revealed that the CQT-ResMax model performs best for the PA data in terms of EERs among all single models; however, the T45's single model performed better in LA data.

On the PA evaluation set, CQT-1_120-ResMax surpassed all ensemble solutions in terms of both EERs and t-DCFS, achieving an EER of 0.30%. Its EER advantage over T28 (the top-performing ensemble solution) was statistically significant ($p < 0.0001$, one sample t-test). With an EER of 0.16 percent, the CQT-1_120-ResMax model scored third among all ensemble models on the development set. All other models had lower EERs on the development set than the ResMax model, which had higher EERs on the evaluation set, implying that the other models might be overfitted to the development set.

As for the LA set, CQT-1_100-ResMax rated third on the evaluation set with an EER of 2.19%, and fourth on the development set with an EER of 0.56% among all ensemble solutions. It did not have a statistically significant advantage over the 4th ensemble solution (T60) in terms of EERs, but it did have a statistically significant advantage over the 5th ensemble solution (T24) in the evaluation set ($p < 0.0001$, one sample t-test).

In the case of ensemble solutions, we added the number of parameters utilized across all models; we present the numbers of parameters only for those available published papers. Given that CQT-ResMax is a single model solution with considerably fewer parameters (indicating model complexity and computational latency) than ensemble solutions, both LA and PA accuracy results are competitively high and represent major achievements.

C. ResMaxSep RESULTS

In the ResMaxSep model architecture, we applied depthwise separable convolutions to lighten the ResMax model. Fig. 3 shows the number of parameters used in different classification models and their EERs on the LA and PA datasets. The original ResMax model used 262K and 286K parameters for the LA and PA datasets, and it was the lightest model among the models presented in the ASVspoof 2019 competition. Impressively, ResMaxSep is approximately six times lighter than the original ResMax model, and it uses just 44K and 45K parameters for LA and PA data, respectively. The ResMaxSep model also performs competitively well: the average EER on the PA set was 0.36%, which is 0.06% greater than the EER of the original ResMax model; however, this difference was not statistically significant ($p > 0.1$, two sample t-test). For the LA dataset, it achieved an average EER of 2.55%, which is 0.12% higher than EER of the original; however, this difference was not statistically significant ($p > 0.2$, two sample t-test). The LA model performance is also slightly better than that of the T60 system (2.64%), which ranked fourth overall (see Table 2); however, this difference was not statistically significant ($p > 0.3$, one sample t-test). In summary, the ResMaxSep model produced similar or slightly poorer performance than ResMax. Significantly, however, the model's weight was reduced by 1/6.

D. TIME RELATED ISSUES

We conducted experiments related to the required time in practical applications of the ResMax and ResMaxSep

TABLE 2. ResMax performance on the ASVspoof 2019 development sets and evaluation sets. The EERs and t-DCF results are compared against the top five models from each dataset; models are sorted based on the EER in a descending order. Systems that use a single model are in *italic*. #Mo describes the number of models used in an ensemble system. Two top performing single model systems are also shown at the end of each table. #Params represents the number of parameters contained in the models. (Results that are not public are denoted as hyphens).

LA

#	*Model*	*t-DCF (Dev)*	*EER (Dev)*	*t-DCF (Eval)*	*EER (Eval)*	*#Mo*	*# Params*
1	T05	-	-	0.0069	0.22	-	-
2	T45	0.0000	0.000	0.0510	1.86	5	1484K
3	***CQT-1_100-ResMax***	0.0179	0.56	0.0600	2.19	1	262K
4	T60	0.0	0.0	0.0755	2.64	4	-
5	T24	-	-	0.0953	3.45	-	-
6	T50	0.027	0.90	0.1118	3.56	-	-
	T45 (FFT-LCNN)	0.0009	0.040	0.1028	4.53	1	371K
	T45 (LFCC-LCNN)	0.0043	0.157	0.1000	5.06	1	371K

PA

#	*Model*	*t-DCF (Dev)*	*EER (Dev)*	*t-DCF (Eval)*	*EER (Eval)*	*#Mo*	*# Params*
1	***CQT-1_120-ResMax***	0.0042	0.16	0.0086	0.30	1	286K
2	T28	-	-	0.0096	0.39	-	-
3	T45	0.0001	0.0154	0.0122	0.54	3	1113K
4	T44	0.003	0.129	0.0161	0.59	5	5811K
5	T10	0.0064	0.24	0.0168	0.66	6	1330K
6	T24	-	-	0.0215	0.77	-	-
	T28	-	-	-	0.50	1	-
	T45 (CQT-LCNN)	0.0197	0.800	0.0295	1.23	1	371K
	T44 (logspec-SENet)	0.015	0.575	0.0360	1.29	1	1344K

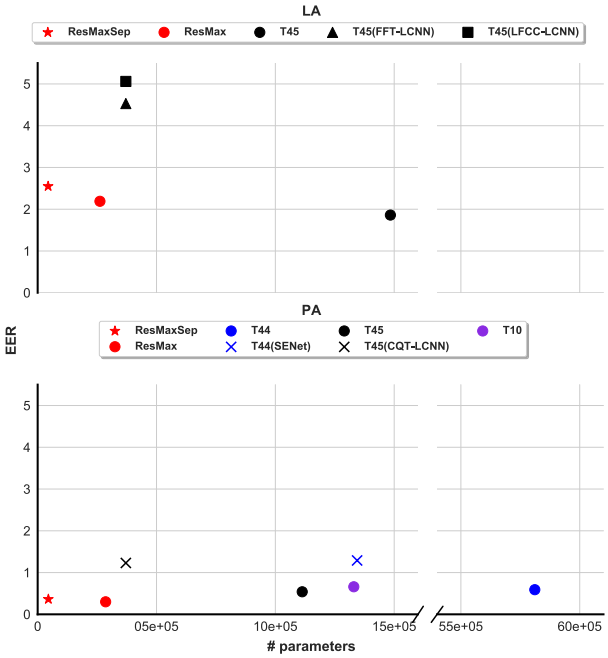


FIGURE 3. The ResMaxSep model has a smallest number of parameters and their performance is not a big difference in LA and PA data.

models. The models are trained on Intel(R) Xeon(R) Gold 6230R CPU clocked at 2.10GHz, and NVIDIA RTX 8000 GPU. To confirm that the model evaluation can be performed at a reasonable speed even on low-spec devices, we also evaluated the model using only the Intel i5-9400F CPU.

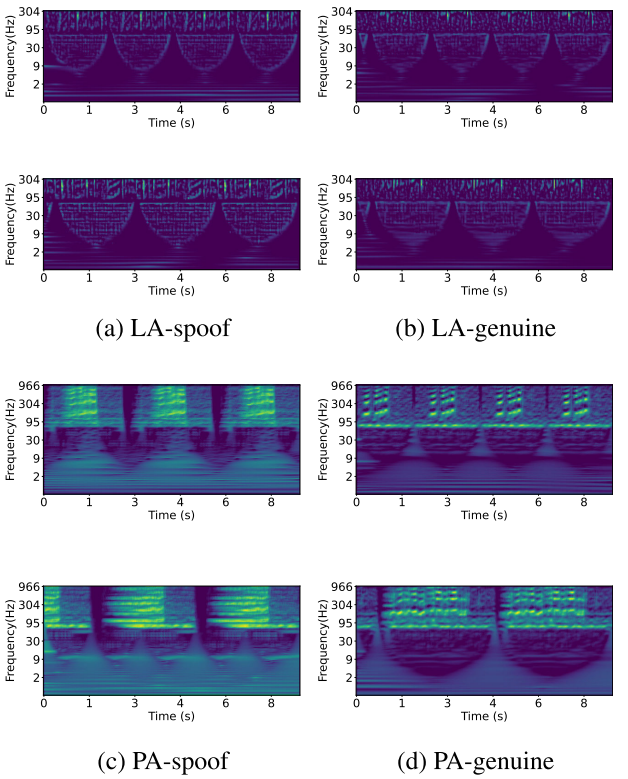
Table 3 displays the amount of time taken for training and evaluation of the ResMax and ResMaxSep models on the LA and PA datasets. The LA set and PA set had 25,380 and 54,000 training data, and 71,237 and 134,730 evaluation data, respectively. On the LA and PA datasets, the ResMaxSep model required an additional training time of about two and five hours, respectively, compared to the ResMax model. However, the time required for training is relatively less important compared to the inference time, as it is only necessary once at the beginning of model training. The inference time (evaluation time) using the GPU was between 48 to 53 seconds, which translates to approximately 0.67 to 0.74 milliseconds per sample. Assuming the situation where the GPU cannot be used, an evaluation was performed using the CPU, and the ResMax and ResMaxSep models took 302 and 472 seconds, respectively, for the LA dataset. When calculated as processing speed per sample, they were 4.24 and 6.49 milliseconds, respectively. For the PA evaluation dataset, ResMax and ResMaxSep models took 870 and 1352 seconds, respectively, with processing speeds per sample of 6.45 and 10.03 milliseconds, respectively. The processing speeds achieved by the ResMax and ResMaxSep models using only the CPU, as described above, are sufficiently low such that devices with low processing capabilities can effectively handle them without encountering any difficulties.

E. ResMax VISUALIZATION WITH GRAD-CAM

To identify specific frequency regions that are significantly affecting the model performance, we applied Grad-CAM to the output of the second convolution layer in the first ResMax block. Fig. 4 shows eight randomly selected Grad-CAM images from the LA and PA datasets for both genuine and spoofed samples (two images for each category). The lighter yellow and green colors indicate more significant

TABLE 3. The time required for training and evaluation. The "evaluation" refers to the time taken to process each sample using a GPU or a CPU.

Models	Data	Training	Evaluation (gpu)	Evaluation (cpu)
ResMax	LA	46m 56s	0.67ms	4.24ms
	PA	1h 41m 23s	0.73ms	6.45ms
ResMaxSep	LA	3h 4m 2s	0.74ms	6.49ms
	PA	6h 31m 3s	0.71ms	10.03ms



(a) LA-spoof (b) LA-genuine

(c) PA-spoof (d) PA-genuine

FIGURE 4. Grad-CAM visualization of eight randomly picked samples in (a) spoof from LA data, (b) genuine from LA data, (c) spoof from PA data, and (d) genuine from PA data. Two-samples are picked from each category.

feature importance and activation. As for the PA samples, the extracted features tend to use a wide frequency range between 1 Hz to 966 Hz – i.e., both low and high frequency ranges are exploited. We surmise that this is because, in general, the fundamental frequency of human voice cannot fall below 80 Hz whereas samples replayed through loudspeakers may also utilize frequencies below 80 Hz; such differences are examined for detection. Highlighted regions tend to flow horizontally (along the time-axis) for some fixed frequency ranges. In the case of LA samples, the human voice frequency range (i.e., excluding frequencies lower than 80 Hz) tend to have more significant impact on the model performance. In contrast to the PA samples, highlighted regions tend to drop vertically to cover a range of frequencies for a given time information.

To conduct deeper analyses on the importance of different frequency ranges, we performed average pooling on the time axis to show how the activation amount (importance) changes across different frequencies. We randomly selected 500 genuine and 500 spoofed samples from LA and PA datasets

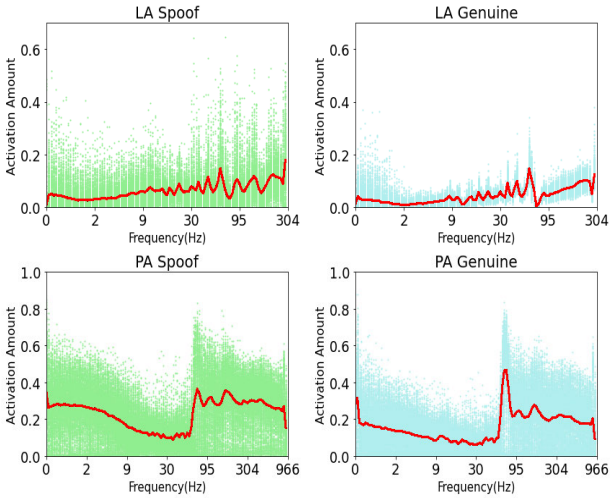


FIGURE 5. 500 frequency-wise information of spoof (left) and genuine (right) samples for LA (upper row) and PA (lower row) datasets. The 500 individual information are expressed softly, and the averaged frequency-wise information is expressed in bold (red colored).

each, and applied average pooling. The graphs in Fig. 5 show activation amount of the selected 500 samples across frequencies range from 0 to 966Hz for the PA dataset, and frequencies range from 0 to 304Hz for the LA dataset. Along with individual sample plots, we also show the average values with the bold red lines. For LA samples, the average lines show more peaks as there are a smaller number of subjects (voice types) available in the LA dataset compared with PA samples. We further surmise that LA samples show higher variances across all frequencies due to various types of voice conversion (VC) and text to speech (TTS) techniques being applied while generating the attack set – the LA model probably exploits a diverse frequency region to detect attack samples generated through multiple techniques. For both datasets, the human voice frequency region (80Hz to 14KHz) shows stronger activation for both genuine and spoofed samples. In particular, we can observe the sudden pick at the human voice frequency region for the PA data. In the PA scenario, we are directly recording the genuine human voice of 80Hz to 14KHz, and replaying sounds from electronic speakers. Thus, we can observe a sudden pick from the 80Hz region in the PA graph, as shown in Fig. 4 (c) and (d). The PA model also shows strong activation at lower frequencies to detect attacks that are replayed through loudspeakers.

F. INCORRECTLY CLASSIFIED SAMPLES AND VISUALIZING ACTIVATION AMOUNT

We observed the performance of the development set dependent on environmental variables to investigate misclassified samples utilizing CQT-1_120-ResMax on PA data. Table 4 shows how EER performance varies based on the development set's verification and recording conditions. We trained ten CQT-1_120-ResMax models and measured the average EER values in each setting. The most important factor was

the replay device's quality. When it was A (perfect), the performance was the worst, with an average EER of 0.0067, and it increased as the playback device's quality decreased. The more high-quality speakers attackers utilize in replay assaults, the more difficult it is to detect the attacks. The reverberation time (T60) and room size, which normally induce the channel effect, had no effect on performance. Furthermore, the EER was substantially greater at 0.0017 when the T60 reverberation period was short, in the ranges of 50-200 ms (A). Another intriguing finding is that when the talker-to-ASV or attacker-to-talker distance is sufficient, the model has a somewhat higher risk.

TABLE 4. Detection performance on the ASVspoof2019 Physical Access evaluation sets in various environments. The A, B, C represent the classes of each factor which is well described in [6]. All numerical values represent the average of EER.

	Factors	A	B	C
Verification Env.	Room size (S)	0.0047	0.0044	0.0041
	T60 (R)	0.0055	0.0029	0.0038
	Talker-to-ASV distance	0.0059	0.0036	0.0042
Recording Env.	Attacker-to-talker distance (D_a)	0.0051	0.0036	0.0041
	Replay Device Quality (Q)	0.0067	0.0036	0.0009

We randomly sampled 500 voice samples under each environmental condition, and calculated the frequency-wise activation amount from their Grad-CAM results. The highest differential frequency-wise activation amount was detected from the replay device quality. The Fig. 6 shows the averaged frequency-wise activation amount from each of the PA device qualities. The range of the human voice extends from 80Hz to 14kHz, the speaker generated sounds possibly have less activation amount from 80Hz. From the Fig. 6, we can detect gradual degradation of activation amount from high quality (A) to low quality (C) in the frequency range from 80Hz to 966Hz.

There are 17 attack types in the LA data, six of which are used in the training and development sets and were deemed the known attack set. The evaluation set includes two known attacks (A16, A19) and 11 unknown assaults. The averaged EER values for the 13 assault types in the evaluation set are shown in Table 5 and Fig. 7. The assault types A08, A17, A18, and A19 have high EERs. A17, A18, and A19 are assaults that exclusively use voice conversion. According to [6], the variation in the mean EER is bigger than the variation in the other sets. Despite the fact that the audio samples from A19 have previously been trained in the training set, it has a considerably higher EER than other attacks.

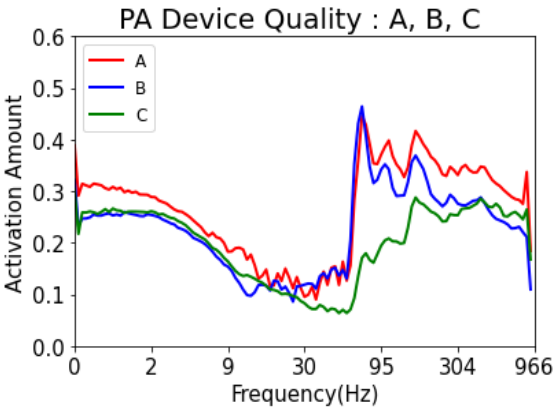


FIGURE 6. The averaged activation amount of 500 random sample voices from each of PA device qualities.

TABLE 5. Detection performance on the ASVspoof2019 logical access evaluation sets in various attacks. Detailed description of each attack is in [6]. All numerical values represent the average of EER.

ID	Type	Description	EER
A07	TTS	vocoder + GAN, LSTM-RNN	0.0022
A08	TTS	neural waveform, AR LSTM-RNN	0.0388
A09	TTS	vocoder, LSTM-RNN	0.0003
A10	TTS	neural waveform, Attention seq2seq	0.0045
A11	TTS	griffin lim, Attention seq2seq	0.0039
A12	TTS	neural waveform	0.0002
A13	TTS,VC	waveform concatenation & filtering	0.0051
A14	TTS,VC	vocoder, LSTM-RNN	0.0012
A15	TTS,VC	neural waveform, LSTM-RNN	0.0030
A16	TTS	waveform concatenation	0.0039
A17	VC	waveform filtering, VAE	0.0561
A18	VC	vocoder, i-vector/PLDA	0.0225
A19	VC	spectral filtering, GMM-UBM	0.0317

We randomly sampled 500 voice samples of each attack system, and calculated the frequency-wise averaged activation amount from their Grad-CAM results. The Fig. 8 shows the frequency-wise averaged activation amount of 500 voice samples from the selected attack systems; A08 (TTS), A12 (TTS), A13 (TTS, VC) and A17 (VC). As these are synthetic voice samples generated from diverse VC or TTS systems, the difference is somewhat unpredictable unless we study each system in detail.

Furthermore, the averaged voice samples confounded different patterns of each attack system. Thus, we visualized each frequency wise activation amount using UMAP in Fig. 9 [28], [29]. We can clearly detect the differences between each attack system from the UMAP visualization.

There were also differences in the EER performance with respect to gender. In the LA data, the EER for male and female was 0.0389% and 0.0299%, respectively. In the PA data, the EERs for males and females were 0.0040% and 0.0013%, respectively. The EER performances for the female

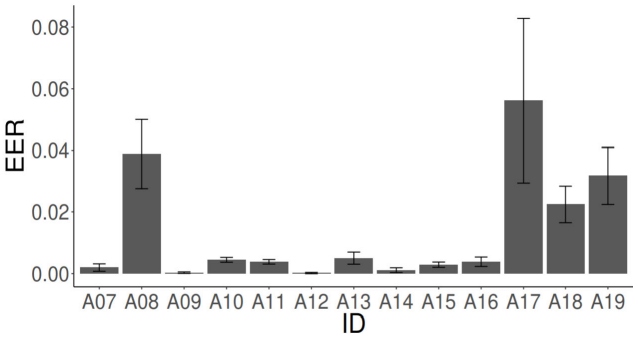


FIGURE 7. The averaged EER for 13 attack types in the evaluation set. The barplot indicates averaged EER with one standard deviation error bar.

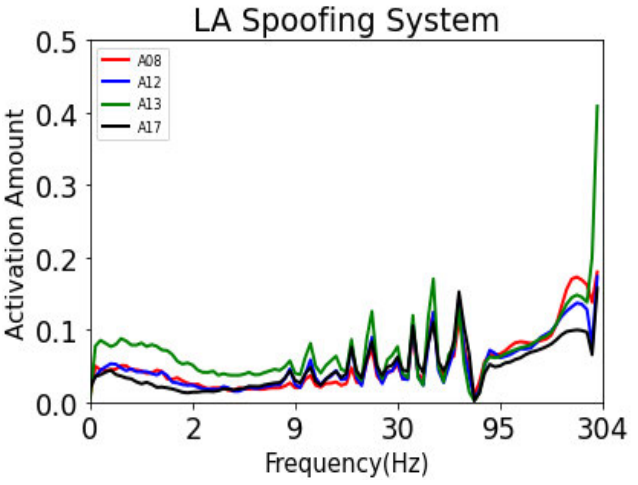


FIGURE 8. The averaged activation amount for A08, A12, A13, A17 attack types in evaluation set.

voice were approximately 1.3 and 3.1 times better than those for the male voice for LA and PA evaluation sets, respectively. One possible reason is that we have more females in the training set (8 males and 12 females for both PA and LA data sets).

The Fig. 10 represents the frequency-wise averaged activation amount of 500 voice samples from each gender for LA (top-side) and PA (bottom-side) data sets. Interestingly, the differences among males and females were only detected for the frequency range 80–300Hz for both of LA and PA datasets. The fundamental frequency of the male voice varies from 80 Hz to 400 Hz, whereas that of the female voice ranges from 120 Hz to 800 Hz [30].

IV. RELATED WORK

In modern society, various types of attacks are threatening security [31]. In the field of voice, detecting voice spoofing attacks is an important issue. Several strategies for detecting speech replay or synthesis assaults employing loudspeakers have been developed. Liu et al. [32] offered wearable technologies to detect voice liveness, such as spectacles, headphones, or necklaces. They reached about 97% accuracy in identifying liveness when utilizing headphones. To test voice

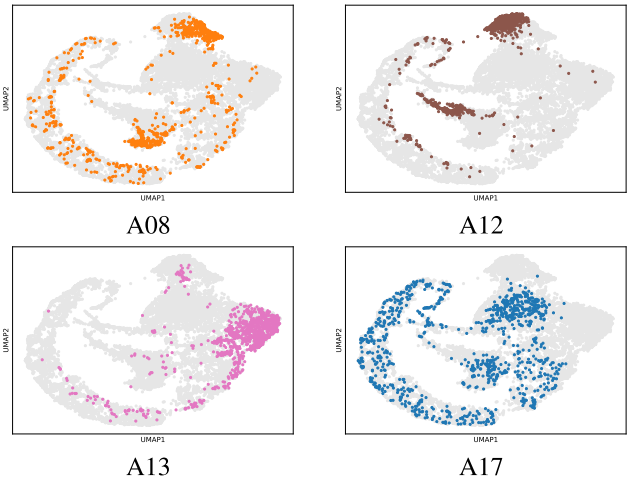


FIGURE 9. UMAP visualization of four attack systems (A08, A12, A13 and A17) in the evaluation set.

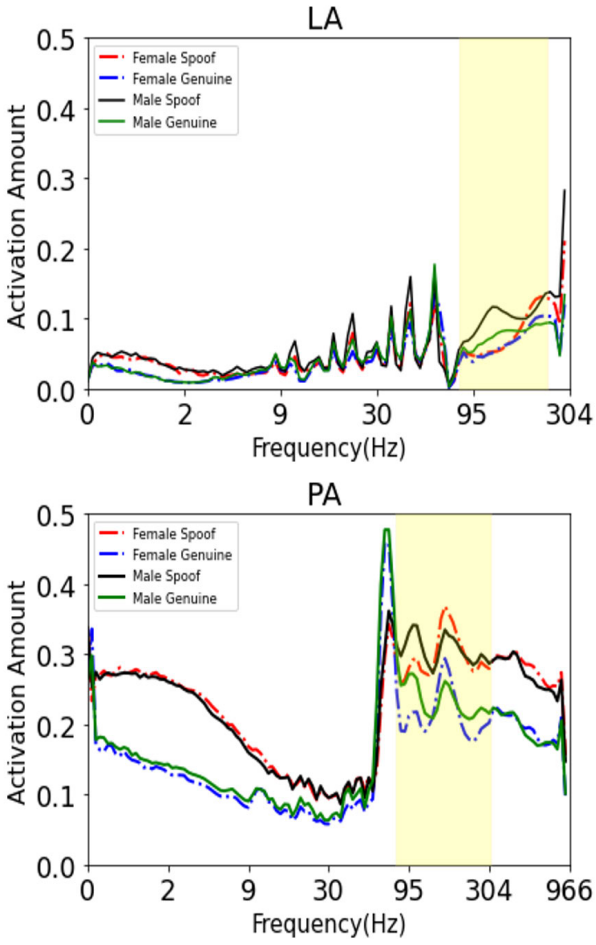


FIGURE 10. The frequency-wise averaged activation amount of 500 voice samples from each gender for LA (top-side) and PA (bottom-side) data.

liveness, Zhang et al. [33] tracked unique articulatory motions using sound wave reflection techniques. They reached a 99.9% accuracy by having users physically hold their gadgets near their ears. Blue et al. [34] employ sub-bass over

excitation and low frequency signal characteristics to detect electronic speakers, obtaining 100% TAR and 1.72 FRR in calm environments. These techniques, like any other speech biometric-based technology, will all suffer considerable accuracy losses when subjected to background noise variations and environmental changes. Furthermore, merging numerous sophisticated models and characteristics necessitates extensive computational resources that are unsuitable for practical usage.

Several researchers offered machine learning-based liveness detection algorithms as part of the ASVspoof 2015, 2017 and 2019 competitions. The ASVspoof competition have been held by every two years from 2015 and changed train, validation and evaluation dataset in each competition. As representative solution for ASVspoof challenge in 2015, the Gausian Mixture Model (GMM) classifier with voice, music related features are mostly proposed and used for the baseline of later competition (2017, 2019). According ASVspoof 2019 competition, EERs ranged from 0.22 to 92.36 percent for LA attacks and 0.39 to 92.64 percent for PA attacks. For both the LA and PA datasets, the top five systems [8], [15], [16], [17], [18] all used an ensemble method that combined numerous models.

This is because the competition focused solely on the accuracy aspects of developing a voice spoofing attack detection system without taking into account real-world deployment scenarios where lowering model complexity and detection latency is critical. For example, In order for the voice spoofing detection model to be implemented on device in Samsung mobile phones, it is necessary to satisfy the requirements such as model size and execution speed, etc. As a result, the majority of the top-performing solutions investigated to date would fail to meet these criteria. In comparison, even with a single model and only 262K parameters, the proposed ResMax models can perform comparably well. ResMaxSep, only has 45K parameters, and is capable of achieving similar level of performance compared to the original.

Many methods have used features that decompose raw audio into frequency and time information, such as STFT, Mel-spectrogram, and CQT. Some researchers have employed an approach that utilizes raw audio directly, such as RawNet, RawNet2, and AASIST [35], [36], [37]. These methods are also interesting approaches, and in particular, AASIST has achieved very high performance with an error rate of around 1% in LA data.

V. DISCUSSIONS

A. IMPORTANT FREQUENCY REGIONS

The performance of deep learning models are difficult to explain, and are often called ''black boxes'' because of the model complexity and large number of parameters trained [38]. To compensate for these shortcomings, studies have been conducted to improve the explanatory power of deep learning models [38], [39]. Attempts have been made to explain the results and classification features from the voice spoofing attack detection solutions [14], [40]. However, such results identified and explained the highly activated parts for only a few examples. Conversely, our Grad-CAM analyses were performed on much larger datasets to identify more generally applicable explanations for detecting spoofing attacks. To the best of our knowledge, this is the first attempt to visualize and compare activated frequency regions between genuine and spoofed samples. We observed that for both the PA and LA samples, human voice frequency range is an important region significantly contributing to the performance of our classifiers. As for the PA model, the energy differences in the lower frequency region also contributed to the model performance – we surmise that samples replayed through loudspeakers inherently produced sounds at lower frequencies.

B. SPOOFING DETECTION IN NOISY ENVIRONMENT

There was the Audio Deep Synthesis Detection Challenge (ADD 2022) [41] hosted as one of the grand challenges from the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) 2022 conference. The track 1 of ADD 2022 is to identify spoofing attacks in noisy situations like real-world noises and background music effects. The 3rd place winning team considered the ResMax model [42], [43] as one of their final ensemble models. Especially, ResMax had the best single model performance among the five single models considered in their ensemble system. There were 43 participating teams in the ADD 2022 challenge [41]. The winning team in the competition applied wav2vec2 [44] to extract features and fine-tuned their model, achieving 21.7% EER [45]. The 3rd-place winning team that utilized the ResMax model in their ensemble model achieved 23.8% EER and the single ResMax model alone achieved 24.7% EER [43]. These results show that the ResMax model is one of the competitive models in voice spoofing detection models, and also shows good performance in noisy environment settings.

C. SPOOFING DETECTION AGAINST ADVANCED ADVERSARIAL ATTACK

While voice spoofing detection solutions are widely studied under signal processing and computer security communities, new advanced technologies are also suggested that can bypass them by manipulating voice spoofing samples [46], [47]. In the study of Zhang et al. [46], they carried voice command samples to ultrasound frequency band which is inaudible for humans and recovered them by leveraging non-linearity of the microphone circuit. Wang et al. [47] also eliminated suspicious features of frequency domain using software-based inverse filter. To build more secure voice spoofing detection solutions under the demands of real-world applications like voice-assistants, we should consider not only typical replay and synthesized spoofing voice samples, but also such advanced adversarial spoofing samples. In further research, we will be able to analyze such crafted samples using Grad-CAM, understand how our approaches effectively detect such

advanced attacks, and finally improve our models and parameter settings to defend against both traditional and advanced attacks towards robust and efficient voice spoofing deep learning model.

VI. CONCLUSION

Existing voice spoofing attack detection systems were created without taking into account real-world model complexity and detection latency requirements, and they frequently consist of numerous large and sophisticated deep learning models. Given the model size and latency constraints, such solutions would be deemed unsuitable. In comparison, our CQT-1 120-ResMax model outperformed the top-performing PA solution on the evaluation set using only a single deep learning model with much less parameters, achieving an EER of 0.30% compared to the current best competition EER of 0.39% by an ensemble solution.

In the LA set, we place third with an EER of 2.19%, barely behind the second best ensemble solution in the evaluation set, which earned an EER of 1.86%. Although CQT-1 120-ResMax used the fewest parameters among the single model systems, it displayed substantial superiority in detection accuracy. Furthermore, we tried using ResMaxSep, lowering the number of parameters by six times (286K vs. 45K for PA data) while maintaining a comparable level of performance: ResMaxSep produced an EER of 0.36% on the PA dataset, which was slightly worse than the original (EER of 0.30%); on the LA dataset, it achieved 2.55% EER, which was roughly 0.12% worse than the original. However, ResMaxSep only uses 45 K parameters and would be more useful for real-world on-device deployment. In our future works, we aim to investigate inverted residuals and linear bottlenecks from MobileNextV2 [48] to decrease model complexity, examine ResMax and ResMaxSep using advanced adversarial attacks, and assess their performance in noisy environmental conditions using ADD 2022 and ASVspoof 2021 datasets [7], [41].

REFERENCES

[1] Y. Wang, R. J. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio, Q. Le, Y. Agiomyrgiannakis, R. Clark, and R. A. Saurous, "Tacotron: Towards end-to-end speech synthesis," in *Proc. INTERSPEECH*, Stockholm, Sweden: ISCA, Aug. 2017, pp. 4006–4010.

[2] W. Ping, K. Peng, A. Gibiansky, S. O. Arik, A. Kannan, S. Narang, J. Raiman, and J. Miller, "Deep voice 3: Scaling text-to-speech with convolutional sequence learning," 2018, *arXiv: 1710.07654*.

[3] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan, R. A. Saurous, Y. Agiomvrgiannakis, and Y. Wu, "Natural TTS synthesis by conditioning wavenet on MEL spectrogram predictions," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2018, pp. 4779–4783.

[4] Z. Wu, T. Kinnunen, N. Evans, J. Yamagishi, C. Hanilçi, M. Sahidullah, and A. Sizov, "ASVspoof 2015: The first automatic speaker verification spoofing and countermeasures challenge," in *Proc. INTERSPEECH*, Dresden, Germany: ISCA, Sep. 2015, pp. 2037–2041.

[5] T. Kinnunen, M. Sahidullah, H. Delgado, M. Todisco, N. Evans, J. Yamagishi, and K. A. Lee, "The ASVspoof 2017 challenge: Assessing the limits of replay spoofing attack detection," in *Proc. INTERSPEECH*, Stockholm, Sweden: ISCA, Aug. 2017, pp. 2–6.

[6] M. Todisco, X. Wang, V. Vestman, M. Sahidullah, H. Delgado, A. Nautsch, J. Yamagishi, N. Evans, T. H. Kinnunen, and K. A. Lee, "ASVspoof 2019: Future horizons in spoofed and fake audio detection," in *Proc. INTERSPEECH*, Sep. 2019, pp. 1008–1012.

[7] A. Tomilov, A. Svishchev, M. Volkova, A. Chirkovskiy, A. Kondratev, and G. Lavrentyeva, "STC antispoofing systems for the ASVspoof2021 challenge," in *Proc. Ed. Autom. Speaker Verification Spoofing Countermeasures Challenge*. Brno, Czech Republic: ISCA, Sep. 2021, pp. 61–67.

[8] G. Lavrentyeva, S. Novoselov, A. Tseren, M. Volkova, A. Gorlanov, and A. Kozlov, "STC antispoofing systems for the ASVspoof2019 challenge," in *Proc. INTERSPEECH*, Sep. 2019, pp. 1033–1037.

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

[10] I.-Y. Kwak, S. Kwag, J. Lee, J. H. Huh, C.-H. Lee, Y. Jeon, J. Hwang, and J. W. Yoon, "ResMax: Detecting voice spoofing attacks with residual network and max feature map," in *Proc. 25th Int. Conf. Pattern Recognit. (ICPR)*, Jan. 2021, pp. 4837–4844.

[11] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 618–626.

[12] M. Todisco, H. Delgado, and N. Evans, "Constant Q cepstral coefficients: A spoofing countermeasure for automatic Speaker verification," *Comput. Speech Lang.*, vol. 45, pp. 516–535, Feb. 2017.

[13] F. Tom, M. Jain, and P. Dey, "End-to-end audio replay attack detection using deep convolutional networks with attention," in *Proc. INTERSPEECH*. Hyderabad, India: ISCA, Sep. 2018, pp. 681–685.

[14] C. Lai, A. Abad, K. Richmond, J. Yamagishi, N. Dehak, and S. King, "Attentive filtering networks for audio replay attack detection," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 6316–6320.

[15] W. Cai, H. Wu, D. Cai, and M. Li, "The DKU replay detection system for the ASVspoof 2019 challenge: On data augmentation, feature representation, classification, and fusion," in *Proc. INTERSPEECH*, 2019, pp. 1023–1027.

[16] C.-I. Lai, N. Chen, J. Villalba, and N. Dehak, "ASSERT: Anti-spoofing with squeeze-excitation and residual networks," in *Proc. INTERSPEECH*, Sep. 2019, pp. 1013–1017.

[17] Y. Yang, H. Wang, H. Dinkel, Z. Chen, S. Wang, Y. Qian, and K. Yu, "The SJTU robust anti-spoofing system for the ASVspoof 2019 challenge," in *Proc. INTERSPEECH*, Sep. 2019, pp. 1038–1042.

[18] B. Chettri, D. Stoller, V. Morfi, M. A. M. Ramírez, E. Benetos, and B. L. Sturm, "Ensemble models for spoofing detection in automatic speaker verification," in *Proc. INTERSPEECH*, Sep. 2019, pp. 1018–1022.

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

[20] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*.

[21] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1800–1807.

[22] X. Zhang, X. Zhou, M. Lin, and J. Sun, "ShuffleNet: An extremely efficient convolutional neural network for mobile devices," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6848–6856.

[23] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2921–2929.

[24] C. Elkan, "The foundations of cost-sensitive learning," in *Proc. 17th Int. Joint Conf. Artif. Intell. (IJCAI)*. Washington, DC, USA: IJCAI, 2001, pp. 973–978.

[25] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

[26] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. 13th Int. Conf. Artif. Intell. Statist.*, vol. 9, May 2010, pp. 249–256.

[27] T. Kinnunen, K. A. Lee, H. Delgado, N. Evans, M. Todisco, M. Sahidullah, J. Yamagishi, and D. A. Reynolds, "T-DCF: A detection cost function for the tandem assessment of spoofing countermeasures and automatic speaker verification," in *Proc. Speaker Lang. Recognit. Workshop (Odyssey)*, Jun. 2018, pp. 312–319.

[28] L. McInnes, J. Healy, and J. Melville, "UMAP: Uniform manifold approximation and projection for dimension reduction," 2018, *arXiv:1802.03426*.

[29] L. McInnes, J. Healy, N. Saul, and L. Großberger, "UMAP: Uniform manifold approximation and projection," *J. Open Source Softw.*, vol. 3, no. 29, p. 861, Sep. 2018.

[30] J. Benesty, M. M. Sondhi, and Y. Huang, *Springer Handbook of Speech Processing*. Berlin, Germany: Springer, 2007.

[31] X. Cai, K. Shi, K. She, S. Zhong, and Y. Tang, "Quantized sampled-data control tactic for T–S fuzzy NCS under stochastic cyber-attacks and its application to truck-trailer system," *IEEE Trans. Veh. Technol.*, vol. 71, no. 7, pp. 7023–7032, Jul. 2022.

[32] R. Liu, C. Cornelius, R. Rawassizadeh, R. Peterson, and D. Kotz, "Vocal resonance: Using internal body voice for wearable authentication," *Proc. ACM Interact., Mobile, Wearable Ubiquitous Technol.*, vol. 2, no. 1, pp. 1–23, Mar. 2018.

[33] L. Zhang, S. Tan, and J. Yang, "Hearing your voice is not enough: An articulatory gesture based liveness detection for voice authentication," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Dallas, TX, USA, Oct. 2017, pp. 57–71.

[34] L. Blue, L. Vargas, and P. Traynor, "Hello, is it me You're looking for? Differentiating between human and electronic speakers for voice interface security," in *Proc. 11th ACM Conf. Secur. Privacy Wireless Mobile Netw.*, Stockholm, Sweden, Jun. 2018, pp. 123–133.

[35] J.-W. Jung, H.-S. Heo, J.-H. Kim, H.-J. Shim, and H.-J. Yu, "RawNet: Advanced end-to-end deep neural network using raw waveforms for text-independent speaker verification," in *Proc. INTERSPEECH*, Sep. 2019, pp. 1268–1272.

[36] H. Tak, J. Patino, M. Todisco, A. Nautsch, N. Evans, and A. Larcher, "End-to-end anti-spoofing with RawNet2," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2021, pp. 6369–6373.

[37] J.-W. Jung, H.-S. Heo, H. Tak, H.-J. Shim, J. S. Chung, B.-J. Lee, H.-J. Yu, and N. Evans, "AASIST: Audio anti-spoofing using integrated spectro-temporal graph attention networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2022, pp. 6367–6371.

[38] G. Montavon, W. Samek, and K.-R. Müller, "Methods for interpreting and understanding deep neural networks," *Digit. Signal Process.*, vol. 73, pp. 1–15, Feb. 2017.

[39] F. K. Došilovic, M. Brcic, and N. Hlupic, "Explainable artificial intelligence: A survey," in *Proc. 41st Int. Conv. Inf. Commun. Technol., Electron. Microelectron. (MIPRO)*, May 2018, pp. 0210–0215.

[40] S. Shukla, J. Prakash, and R. S. Guntur, "Replay attack detection with raw audio waves and deep learning framework," in *Proc. Int. Conf. Data Sci. Eng. (ICDSE)*, Sep. 2019, pp. 66–70.

[41] J. Yi, R. Fu, J. Tao, S. Nie, H. Ma, C. Wang, T. Wang, Z. Tian, Y. Bai, C. Fan, S. Liang, S. Wang, S. Zhang, X. Yan, L. Xu, Z. Wen, and H. Li, "ADD 2022: The first audio deep synthesis detection challenge," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Singapore, May 2022, pp. 9216–9220.

[42] I.-Y. Kwak, S. Choi, J. Yang, Y. Lee, and S. Oh, "CAU_KU team's submission to ADD 2022 challenge task 1: Low-quality fake audio detection through frequency feature masking," 2022, *arXiv:2202.04328*.

[43] I.-Y. Kwak, S. Choi, J. Yang, Y. Lee, S. Han, and S. Oh, "Low-quality fake audio detection through frequency feature masking," in *Proc. 1st Int. Workshop Deepfake Detection Audio Multimedia (DDAM)*. New York, NY, USA: Association for Computing Machinery, Oct. 2022, p. 917.

[44] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "Wav2vec 2.0: A framework for self-supervised learning of speech representations," in *Advances in Neural Information Processing Systems*, vol. 33. Red Hook, NY, USA: Curran Associates, 2020, pp. 12449–12460.

[45] J. M. Martin-Donas and A. Alvarez, "The vicomtech audio deepfake detection system based on wav2vec2 for the 2022 ADD challenge," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2022, pp. 9241–9245.

[46] G. Zhang, C. Yan, X. Ji, T. Zhang, T. Zhang, and W. Xu, "DolphinAttack: Inaudible voice commands," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur. (CCS)*, 2017, pp. 103–117.

[47] S. Wang, J. Cao, X. He, K. Sun, and Q. Li, "When the differences in frequency domain are compensated: Understanding and defeating modulated replay attacks on automatic speech recognition," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur. (CCS)*, 2020, pp. 1103–1119.

[48] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Salt Lake City, UT, USA, Jun. 2018, pp. 4510–4520.

IL-YOUP KWAK (Member, IEEE) received the Ph.D. degree in statistics from the University of Wisconsin-Madison, in 2014 and the second Ph.D. degree in computer science. He was a Postdoctoral Associate with the University of Minnesota-Twin Cities, from 2014 to 2017. He was with Samsung Research, from 2017 to 2019. Since 2019, he has been an Assistant Professor with the Department of Applied Statistics, Chung-Ang University. His research interests include deep learning applied to audio, AI security, and statistical genetics.

SUNGSU KWAG received the B.S. degree in software from Sungkyunkwan University. He is currently a Researcher of continuous and action-less human authentication with Samsung Research. His research interests include data-driven security and security for consumer devices.

JUNHEE LEE received the B.S. degree in computer science from Dongguk University. He is currently a Researcher of continuous and action-less human authentication with Samsung Research. His research interest includes data-driven security.

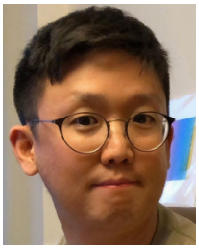
YOUNGBAE JEON received the B.S. degree in electrical and electronics engineering from Yonsei University and the Ph.D. degree in information security from Korea University. He is currently a Staff Engineer with Samsung Research. His research interests include information security applications based on signal processing and machine learning.

JEONGHWAN HWANG received the M.S. degree in information security from Korea University. His research interests include signal processing and probabilistic modeling.

HYO-JUNG CHOI received the B.S. degree in statistics from Duksung Women's University, Seoul, South Korea, in 2019, and the M.S. degree in statistics from Chung-Ang University, Seoul, in August 2021. Since 2021, she has been with CJ Logistics and with TES Logistics Technology Research Center. Her research interests include sequential data analysis, such as time-series and audio data.

JONG-HOON YANG received the B.S. degree in applied statistics and the M.S. degree in statistics from Chung-Ang University, Seoul, South Korea, in 2020 and March 2022, respectively. Since 2022, he has been with Tmaxsoft, as a Researcher. His research interests include text mining and deep learning applications in audio data.

SO-YUL HAN received the M.S. and Ph.D. degrees in statistics from Chung-Ang University, Seoul, South Korea, in February 2017 and August 2021, respectively. She was with the National Cancer Center, from 2017 to 2018. Her research interest includes deep learning applications in audio data.



JUN HO HUH received the Ph.D. degree in cybersecurity and trustworthy computing from the University of Oxford. He is currently a Principal Engineer with Samsung Research. His research interests include data-driven security and usable security, focusing on building usable biometric authentication systems for smartphones.

CHOONG-HOON LEE received the Ph.D. degree in computer science from KAIST, Daejeon, South Korea, in 2004. He was a Postdoctoral Associate and a Research Professor with KAIST, from 2004 to 2006. He is currently a Master (VP of Technology) with Samsung Research. His research interests include data driven security, authentication, and cybercrime prevention.

JI WON YOON received the Ph.D. degree in statistical signal processing with the University of Cambridge, U.K., in 2008. He was with the IBM Research Laboratory, from 2011 to 2012. Since 2021, he has been a Professor with the School of Cybersecurity, Korea University. His research interest includes all areas of intelligence.