A Gated Recurrent Convolutional Neural Network for Noise Robust Spoofing Detection

Alejandro Gomez-Alanis, Antonio M. Peinado, *Senior Member, IEEE,* Jose A. Gonzalez, and Angel M. Gomez

***Abstract*—This work deals with the issue of the spoofing attack identification required for secure automatic speaker verification (ASV) systems. This topic has received increased attention in recent years and, accordingly, a number of countermeasures has been developed. Despite these anti-spoofing techniques have been successfully applied in clean scenarios, it has been shown that they perform poorly in noisy environments. In this work, we aim at improving the performance of anti-spoofing detection for ASV in noisy scenarios. To achieve this, we first propose the use of Gated Recurrent Convolutional Neural Networks (GRCNNs) as a deep feature extractor to robustly represent speech signals as utterance-level embeddings, which are later used by a back-end recognizer for performing the final genuine/spoofed classification. Then, to further enhance the robustness of the system in noisy conditions, we propose the use of signal-to-noise masks (SNMs) along with the deep model. These masks inform the deep feature extractor about the regions of the input spectrogram-based features that are mostly affected by noise and, hence, should be neglected when computing the embeddings. To evaluate our proposals, experiments were carried out on the clean and noisy versions of the ASVspoof 2015 corpus. The experimental results show that our proposal clearly outperforms other methods recently proposed such as the popular CQCC + GMM system or other similar deep feature based systems for both seen and unseen noisy conditions and, even in clean conditions.**

***Index Terms*—Spoofing detection, noise robustness, speaker verification, deep learning, signal-to-noise masks.**

I. Introduction

AUTOMATIC Speaker Verification (ASV) aims to authenticate the identity claimed by a given individual based on the provided speech samples [1]. In recent years, this technology has gained an increased interest due to its commercial applications. As the importance of this technology grows, so does the concerns about its security. In ASV, an impostor could gain fraudulent access to the system by presenting speech resembling the voice of a genuine user. Four types of spoofing attacks have been identified [2]: (i) replay (i.e. using pre-recorded voice of the target user), (ii) impersonation (i.e. mimicking the voice of the target voice), and, also, either (iii) text-to-speech synthesis (TTS) or (iv) voice conversion (VC) systems to generate artificial speech resembling the voice of a legitimate user.

The main problem which is present in all anti-spoofing systems is that they have to be trained considering only a finite number of spoofing attacks despite the fact they may be exposed to other attacks in the evaluation phase, as it has been planned in the popular logical access spoofing detection challenge ASVspoof 2015 [3]. Thus, it is desirable that the system learns to detect not only the attacks observed in the training dataset, but also how to generalize to unseen attacks. To address this issue, deep feature extraction has been proposed in [4], where features are extracted from an inner layer of a deep neural network to represent every temporal frame of the voice signal, or even the whole utterance.

In recent years, deep neural networks have shown to be very effective for feature engineering in several speech-based applications, such as speech recognition [5], speech synthesis [6], speaker verification [7] and spoofing detection [8]. Their nonlinear modeling and discriminative capabilities make them not only a powerful back-end classifier [9], [10], but also advantageous for feature extraction [11]. The architecture of these deep feature extractors has shown to be determinant for the performance of the anti-spoofing system. Depending on the architecture employed, we can differentiate two types of deep features: (i) frame-level, and (ii) utterance-level (or spoofing identity vectors). Moreover, the nature of the speech features which are fed into the deep feature extractor can also determine the whole performance of the anti-spoofing system [12], [13]. Thus, we can find in the literature three types of speech features which have been successfully applied to spoofing detection: (i) magnitude based spectral features [14], (ii) phase based spectral features [15], and (iii) raw speech samples [16].

The extraction of deep features (embeddings) at a frame level has demonstrated to be effective in ASV [17] and spoofing detection [18]. Fig. 1 shows the diagram of a common anti-spoofing system based on frame-level deep feature extraction. Fully-connected deep neural networks (DNN) and convolutional neural networks (CNN) were used in [19] to obtain frame-level deep features, showing that convolutional layers have a powerful ability for detecting the artifacts produced by the speech vocoders used in TTS/VC systems even in noisy acoustic conditions, as they can be seen as filter banks whose filters are optimized for the specific task of spoofing detection [20].

These frame-level features must be combined into a single identity vector which characterizes the utterance. There are several ways to combine them, such as averaging [21], attentive statistics pooling [17], or the use of recurrent neural networks (RNNs) [8] as illustrated in Fig. 1. There is a mounting evidence that recurrent neural networks are powerful at extracting discriminative features to capture the temporal artifacts in the spoofed speech. For instance, in our previous work [8], we showed that an RNN, with cells based on

Alejandro Gomez-Alanis, Antonio M. Peinado, Jose A. Gonzalez and Angel M. Gomez are with the Department of Signal Processing, Telematics and Communications, University of Granada, Granada, 18071 Spain (e-mail: agomezalanis@ugr.es; amp@ugr.es; joseangl@ugr.es; amgg@ugr.es).

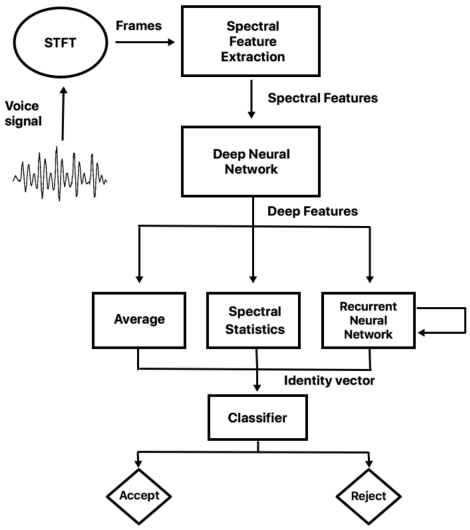

Fig. 1. Extraction of frame-level deep features and utterance-level identity vector for spoofing detection. Here, we consider 3 methods to extract the utterance-level identity vector from frame-level deep features: average (left), spectral statistics (middle), and recurrent neural network (right).

gated recurrent units (GRU), was able to model the long-term dependencies of the consecutive frame-level deep feature vectors for spoofing detection. Also, Long Short Term Memory networks (LSTMs) [21] and a combination of CNNs and recurrent neural networks (RNNs) [22] have been already successfully applied to extract utterance-level deep features.

Once the spoofing identity vector of the utterance has been extracted, a classifier must be used to decide between genuine and spoofed speech. Choosing a reliable classifier is particularly important given the unpredictable nature of the attacks in a practical system (it is unknown what kind of attack the perpetrator may use to access the verification system). The classifier must be selected accounting for the dimensionality and characteristics of the features. Standard classifiers such as Gaussian Mixture Models (GMMs), Linear Discriminant Analysis (LDA) and Support Vector Machines (SVMs) are often employed for this task.

While research on anti-spoofing has been mainly focused on systems operating on clean conditions, little work has been carried out considering the noise likely present in realistic situations. Although noise will be, in general, a cause for performance degradation, its effect varies according to the type of attack. Thus, replay recordings made in high noise conditions (noisier than bona fide speech) pose a lower threat to ASV than the recordings made in low noise conditions [23], so that high noisy replayed speech should be detected with relative ease. The noise introduced by the playback and recording devices may be even helpful to detect replay attacks and cannot be easily separated from the noise present in the acoustic environment. On the other hand, as shown in [24], VC/TTS spoofing countermeasure systems trained with clean speech perform poorly in noisy conditions and their performance decreases rapidly as the signal-to-noise ratio (SNR) worsens. This lack of robustness will be one of the main motivations of this work.

One of the first studies about the impact of noise on anti-spoofing systems was carried out in [25], where the robustness of several front-end features were evaluated under different noisy conditions. In [24] an anti-spoofing system based on neural networks was trained using different front-end features and tested under five additive noises and reverberant conditions. Also, [19] showed that deep feature extractors improve significantly the noise robustness of the spoofing detector when multi-condition training is used, since the nonlinear modeling capability of neural networks allows to learn features which are more invariant to the effects of noise. Furthermore, [19] also proposed the use of the mean noise vector, being justified as a mean to provide useful information about the noise to the neural network. More recently, we proposed to extend the deep feature extractor with information about the distorsion level of each temporal frame of the signal [8].

The search of solutions for the issues mentioned above is the main motivation of this work, whose major contributions can be summarized as follows:

1) Gated Recurrent Convolutional Neural Network: We propose the use of a new architecture which introduces several convolutional layers inside a gated recurrent unit (GRU) based RNN. Our goal is to combine the ability of the convolutional layers for extracting discriminative features at frame level with the capacity of RNNs for learning long-term dependencies of the subsequent deep features. The architecture is called Gated Recurrent Convolutional Neural Network (GRCNN), and although similar deep learning frameworks have been applied in learning video representations [26], audio tagging [27] and optical character recognition [28], to the best of our knowledge, this is the first time that this type of neural network is adapted to spoofing detection.

2) Combination of magnitude and phase spectrum: Typically, TTS and VC systems employ a vocoder which may introduce artifacts in both the magnitude and phase spectrums of the reconstructed signal. Most antispoofing systems use either only one type of features or a fusion of systems employing different types of features [25]. To take into account both the amplitude and phase information of the speech spectra, we propose the combination of magnitude and phase based features to feed the GRCNN. To the best of our knowledge, optimizing a single anti-spoofing system which is fed with both magnitude and phase based features has not been explored yet.

3) Signal-to-Noise Masks: To enhance the robustness of the anti-spoofing system against noise, we propose a new technique for estimating masks based on a deep learning framework. In a previous work [8], we demonstrated that applying classical SNR-based masks [29], [30] for spoofing detection obtain the best state-of-the-art results in noisy scenarios. Here, we improve the estimation of signal-to-noise masks by means of deep learning techniques.

II. Gated Recurrent Convolutional Neural Network

In this section we describe the details of the GRCNN architecture for spoofing detection. Based on our previous

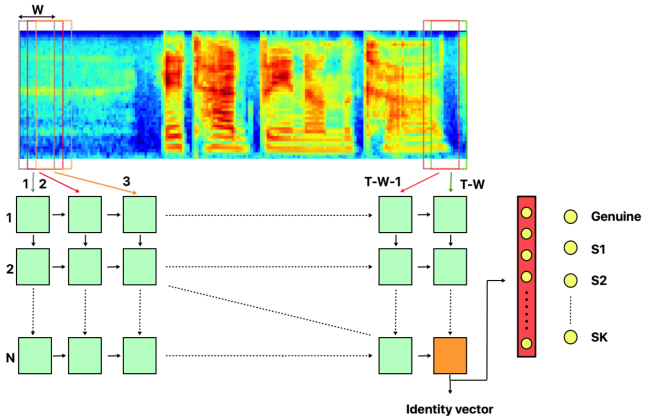

Fig. 2. Block diagram of the proposed utterance-level spoofing identity vector extractor. It consists of a Gated Recurrent Convolutional Neural Network (GRCNN) which processes the spectral features of a context of W consecutive frames in each time step through the N recurrent layers. The utterance has T temporal frames and the training set of the speech corpus has K spoofing attacks.

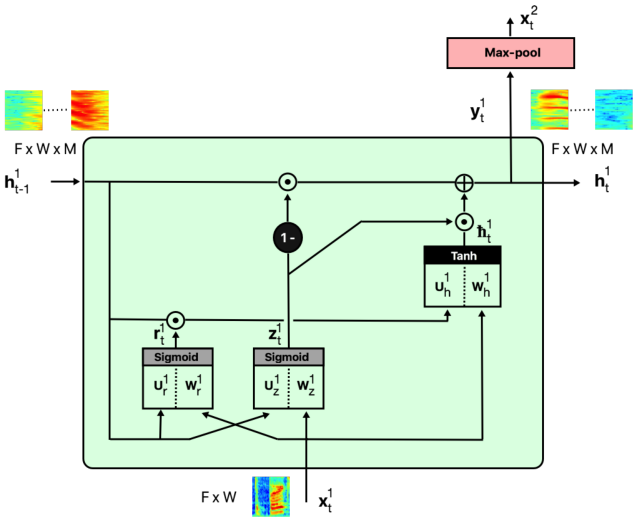

Fig. 3. Gated recurrent convolutional unit cell (GRCU) of the first recurrent layer. The input is 2-dimensional and may include several channels. The output consists of M 2-dimensional feature maps, which are passed to both the next layer and next step of the GRCNN.

work [8], our hypothesis is that introducing the convolutional layers of a CNN inside the cell of a recurrent neural network strive to: (1) extract discriminative features at frame level, (2) learn long-term dependencies, and (3) integrate the extraction of frame-level deep features and the utterance-level identity vector into a single network.

An RNN can process a sequential input with possibly variable lengths. It defines a recurrent hidden state whose activation at each time is dependent on that of the previous time. Specifically, given an input sequence $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_T)$, the RNN hidden state at time t is defined as $\mathbf{h}_t = \phi(\mathbf{h}_{t-1}, \mathbf{x}_t)$, where ϕ is a nonlinear activation function. Typical RNN architectures are the Long Short Term Memory (LSTM) [31] and the Gated Recurrent Unit (GRU) [32], which shows similar performance to LSTM but with lower memory and computational requirements [33]. In our work, we will use GRU structures as the basis for the GRCNN.

Unlike the RNN architectures mentioned above, the hidden state $\mathbf{h}_t$ of a GRCNN model is computed by convolving the current input features $\mathbf{x}_t^n$ and the previous state $\mathbf{h}_{t-1}^n$ with multiple convolutional filters ($n = 1, ..., N$ stands for the index identifying the network layer as remarked later in this section). Taking into account that most of the cues that enable the detection of spoofing attacks can be found in certain frequency bands [34], we embed such a prior in our neural network architecture by replacing the fully-connected operations in the GRU with convolutions. This has the potential advantage that more discriminative features can be extracted at the frame level [19].

The proposed feature extractor is shown in Fig. 2. At each time step, the GRCNN is fed with the set of spectral features corresponding to a context window of W frames. Therefore, the number of steps of the GRCNN is $T - W$, where T is the number of frames of the utterance being processed. Moreover, the GRCNN has N consecutive layers. This architecture acts as a classifier whose task consists of determining whether the utterance is either genuine or belongs to one of the K spoofing attacks of the training set $(S_1, S_2, ..., S_k)$. In order to do this, a fully connected layer is connected to the output of the last time step, followed by a softmax layer which contains $K+1$ neurons (one per class: genuine, $S_1, S_2, ..., S_k$). In fact, the state of the last time step represents the spoofing identity vector of the whole utterance.

Our GRCNN cell contains three different gates where each one includes 2 different single-layer convolutional nets with M filters in parallel. Each time step of the GRCNN plays the role of a frame-level deep feature extractor providing N state (feature) vectors for each context window of W consecutive frames, which are passed to the next time step of the GRCNN. The unit cell of each layer of the GRCNN is a gated recurrent convolutional unit (see Fig. 3; only first layer is shown), which applies a total of 6 convolution operations (4 for computing the update and reset gates, and 2 for computing the candidate activation). This results in an output volume $\mathbf{y}_t^n$ of $[F, W, M]$, where F is the number of frequency bins considered for the spectral features. After a max-pool downsampling, every $\mathbf{y}_t^n$ is fed into the following layer as $\mathbf{x}_t^{n+1}$ (details provided in Section IV).

The update gate at time step t, which is computed as

$$\mathbf{z}_t^n = \sigma(\mathbf{W}_z^n * \mathbf{x}_t^n + \mathbf{U}_z^n * \mathbf{h}_{t-1}^n), \quad (1)$$

determines which information from the previous frames needs to be passed along the next steps, avoiding the risk of the vanishing gradient problem [35]. The operator $*$ denotes a convolution operation. Similarly, the reset gate

$$\mathbf{r}_t^n = \sigma(\mathbf{W}_r^n * \mathbf{x}_t^n + \mathbf{U}_r^n * \mathbf{h}_{t-1}^n) \quad (2)$$

is used to decide whether or not to forget some information from the previous frames. These convolutional layers can be interpreted as filter banks which are trained and optimized to detect artifacts from the spoofed speech. The main advantage

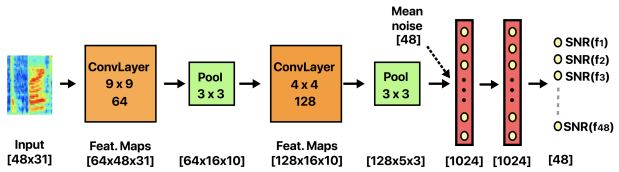

Fig. 4. Convolutional neural network for the estimation of the signal-to-noise ratio for each time-frequency bin. The utterance noise mean is concatenated with the output of the convolutional layers as a noise reference.

of employing these filters is the extraction of frame-level features at every time step which are more discriminative than those extracted by using fully connected units [22]. Finally, the third gate is the update activation,

$$\tilde{\mathbf{h}}_t^n = \delta(\mathbf{W}_h^n * \mathbf{x}_t^n + \mathbf{U}_h^n * (\mathbf{r}_t^n \odot \mathbf{h}_{t-1}^n)), \tag{3}$$

which uses the reset gate to store the relevant information from the past frames, removing firstly the non-relevant information through an element-wise multiplication with the previous state. In these equations, the functions $\sigma(\cdot)$ and $\delta(\cdot)$ are the sigmoid and tangent activation functions, and $\odot$ denotes an element-wise multiplication. The input $\mathbf{x}_t^1$ (dimension $F \times W$) represents a context of consecutive spectral features at time step t, and the model parameters $\mathbf{W}_z^n$, $\mathbf{W}_r^n$, $\mathbf{W}_h^n$ and $\mathbf{U}_z^n$, $\mathbf{U}_r^n$, $\mathbf{U}_h^n$ are the convolutional filters of the 6 different single-layer convolutional nets, which are shared in each time step of the GRCNN.

III. Noise Robustness: Signal-to-Noise Masks

Based on the deep feature framework for spoofing detection described in Section II, we propose a novel mask estimation technique to increase the noise robustness. Its goal is not noise reduction, but providing an estimation of the noise present in each time-frequency bin to the network.

As a first step towards an increased robustness, training a deep neural network with multi-condition data enables it to learn features which are more invariant to the effects of noise. In terms of feature engineering, the layers of a deep learning framework are optimized to learn discriminative features which are as invariant as possible for the acoustic conditions present in the training data. However, the testing acoustic conditions may meaningfully differ from the training ones. To overcome the mismatch between training and testing acoustic conditions, we propose the use of signal-to-noise masks (SNMs) in order to provide the GRCNN with information about the amount of noise present in each time-frequency bin of the signal spectrum. In order to do this, the SNM will be defined as a score from 0 to 1 indicating the relative amount of noise with respect to that of clean speech.

In our recent work [8] we proposed the use of masks, similar to those employed by missing data techniques, for spoofing detection, showing that this approach is better than appending a feature vector with the averaged noise of the utterance [19]. In [8], the masks were computed from the noise estimates obtained using a linear interpolation of the averaged noise spectra of the first and last $T = 10$ frames of the utterance (assuming that there is a short non-speech period at the beginning and the end of the utterance). This approach, however, performs poorly in highly non-stationary noise or when there is little noise at the beggining/end of the utterance. To address this issue, here we propose a new technique to estimate the SNR using a deep learning framework. The proposed system is the convolutional neural network shown in Fig. 4, whose output is the estimated SNR of each time-frequency bin corresponding to the frame which is being processed. The input is a context of W magnitude mel filterbank features (computed as indicated in [36] for the FE standard), centered at the frame being processed. Furthermore, the mean noise of the utterance, which is calculated averaging the first $T = 10$ frames, is concatenated with the output of the convolutional layers. This way, instead of providing the mean noise to the input of the CNN, we combine the advantages from the topographical feature based CNN and the assistance of the mean noise reference.

In the training phase, the instantaneous SNR target which is presented to the CNN for each temporal frame computed as,

$$\boldsymbol{SNR}(t,f) = 10 * log_{10}\left(\frac{\mathbf{X}(t,f)}{\mathbf{N}(t,f)}\right), \tag{4}$$

where the tuple (t, f) represents the time-frequency bin, and $\mathbf{X}$ and $\mathbf{N}$ are the (linear) filter bank outputs of the clean speech and noise, respectively[1]. To obtain the signal-to-noise mask target, this SNR is compressed in the range $[0, 1]$ using a sigmoid function centered at 0 dB:

$$\mathbf{m}_k = \frac{1}{1 + e^{-\boldsymbol{SNR}(t,f_k)}}. \tag{5}$$

The criterion used to train the CNN is the Binary Cross Entropy (BCE) between the target $\mathbf{m}_k$ and the output $\mathbf{z}_k$ of the network, that is,

$$\text{Loss} = \sum_{k=1}^{F} \mathbf{m}_k \cdot log(\sigma(\mathbf{z}_k)) + (1 - \mathbf{m}_k) \cdot log(1 - \sigma(\mathbf{z}_k)), \tag{6}$$

so that each frequency bin contributes equally to the loss function. Therefore, the mask $\mathbf{m}_k$ has the meaning of a SNR compressed in the interval $[0, 1]$. The use of the BCE function deserves some comments. First, it provides the masks with a probability sense. Moreover, it allows us to benefit from the power that neural networks have as statistical classifiers for the estimation of the SNR. Fig. 5 shows an example of the ideal mask and its estimation with the proposed technique for one sample waveform contaminated with an additive babble noise at 10 dB. The similarity between both masks clearly shows the suitability of the proposed CNN for this task.

To implement the noise-aware technique based on SNMs in the proposed GRCNN architecture of Section II, a second channel is appended to the input features $\mathbf{x}_t^1$. Therefore, the first layer cell units of the GRCNN are fed with two input channels (total dimension $2 \times F \times W$): (i) spectral features, and (ii) signal-to-noise mask. Thus, this training method has

[1]Note that filter bank outputs are managed here in a linear scale while they will be processed by the network in the logarithmic scale.

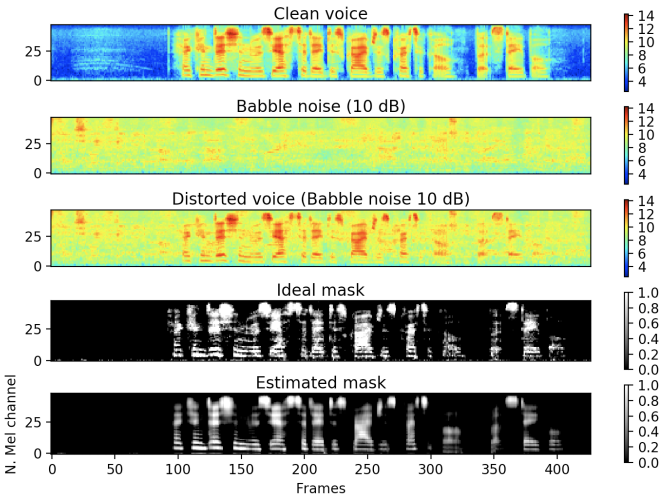

Fig. 5. Estimation of a noise mask for babble noise ($SNR = 10\,dB$): (a) original clean voice, (b) babble noise of 10 dB, (c) distorted voice with additive babble noise of 10 dB, (d) ideal noise mask, (e) estimated mask.

the advantage of optimizing the model parameters taking into account the reliability of every time-frequency bin.

IV. Experimental Framework

In order to evaluate the performance of our proposed techniques, the ASVspoof 2015 corpus [3], a well-known database containing data from different TTS and VC spoofing attacks, was employed. Also, a noisy version of this corpus [24] was also used to evaluate the robustness of the different proposals against noise. The detection of replay attacks is not covered in this work, since the noise introduced by the playback and recording devices may be even helpful to detect replay attacks. Details about the methodology followed for training and testing are given in this section.

A. Speech Corpus

As mentioned, we conducted experiments on two databases: (a) the automatic speaker verification spoofing 2015 (ASVspoof 2015) database, which contains TTS and VC based attacks in clean acoustic conditions; and (b) a noisy version of the ASVspoof 2015 corpus, which was artificially generated distorting the original signals with different noise types.

1) ASVspoof 2015 Clean Corpus: The clean ASVspoof 2015 corpus [3] is a standard data corpus for research on spoofing detection. It defines three data sets (training, development and evaluation), each one containing a mix of genuine and spoofed speech. The structure of these three data sets are shown in Table I. There is no overlap between speakers across training, development and evaluation sets.

Spoofing attacks were generated either by speech synthesis (TTS) or voice conversion (VC). A total of 10 types of spoofing attacks (S1 to S10) are defined: three of them are implemented by using speech synthesis (S3, S4 and S10), while the remaining seven ones (S1, S2, S5, S6, S7, S8 and S9) by means of different voice conversion systems. Attacks S1 to S5 are referred to as *known attacks*, since the training and development sets contain data for these types of attacks, while attacks S6 to S10 are referred to as *unknown attacks*, because they only appear in the evaluation set. More details about this corpus can be found in [3].

TABLE I
Structure of the ASVspoof 2015 data corpus [3]

Subset	# Speakers		# Utterances	
	Male	Female	Genuine	Spoofed
Training	10	15	3750	12,625
Development	15	20	3497	49,875
Evaluation	20	26	9404	184,000

2) ASVspoof 2015 Noisy Corpus: To evaluate the robustness of our system against noise, the noisy version of the ASVspoof 2015 corpus was also employed. This version was generated by artificially distorting the signals in the original clean corpus with different noise types at various signal-to-noise ratio (SNR) levels.

A total of 5 additive noise types (*white noise*, *babble*, *volvo*, *street* and *café*) were added to the clean signals at three SNR levels (20, 10 and 0 dB). Three *reverberant* scenarios were also considered by convolving the clean signals with three room impulse responses (RIR) with different T60 values (0.3, 0.6 and 0.9s). Thus, in total, 18 different noisy conditions (15 additive noises and 3 reverberant conditions) were finally considered.

As suggested in [19], data in the noisy corpus was divided into *seen* and *unseen* conditions for further realism. The *seen condition* consists of *white*, *babble* and *street* noises, and the 3 *reverberant* conditions, which are present in the training, development and evaluation datasets. On the other hand, the *unseen condition* contains *café* and *volvo* noises, which are only present in the evaluation set. Another aspect to take into account is that *white* and *volvo* noises are stationary noises, while *babble*, *street* and *café* are non-stationary. This division allows us to analyze stationary and non-stationary noises in both *seen* and *unseen* conditions. More details about this corpus are given in [24].

B. System

This section details the methodology followed to train our proposed system based on a gated recurrent convolutional neural network:

1) Spectral analysis: Speech signals were analyzed using an analysis window of 25 ms lenght with 10 ms frame shift. The size of the context window is $W = 31$ frames, and the number of filters used to get the spectral features is $F = 48$ filters. In contrast to [19] and [21], we used a 48-dim static spectral features (detailed below) without delta and acceleration coefficients, since we realized that the context window of 31 frames already covers the correlations between consecutive frames. Therefore, a higher spectral resolution is achieved while the size of the spectral feature vector is smaller than in [21].

Two kinds of spectral features are extracted to feed the network: (i) traditional log filter bank features (FBANKs) [36] which only contain information about the amplitude of the

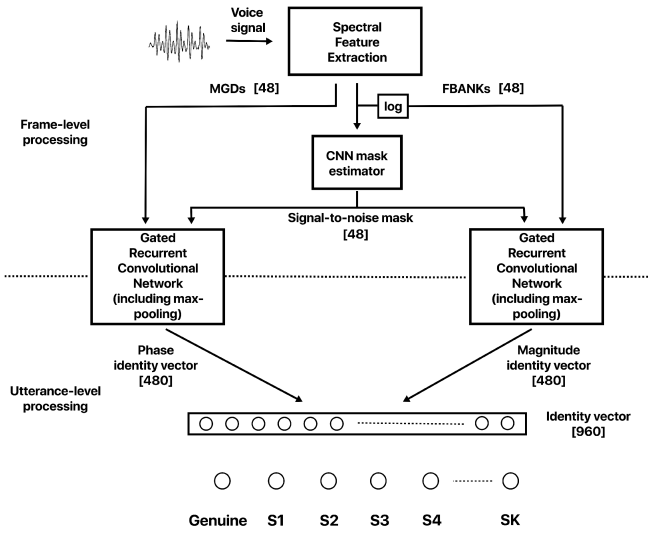

Fig. 6. Proposed architecture for the extraction of the utterance-level spoofing identity vector. Two independent gated recurrent convolutional nets extract the magnitude and phase identity vectors, which are stacked into a single spoofing identity vector of 960 components.

speech signal, and (ii) modified group delay features (MGDs) [37] which carry phase-related information. The core idea is to provide the network with both amplitude and phase information of the utterance. In addition, signal-to-noise masks are appended to the input features as a second channel. In the ASVspoof 2015 clean corpus, the SNMs are not employed, as all the utterances are completely clean. The CNN presented in Section III for SNMs estimation has been trained (independently from the anti-spoofing system) using the genuine data from the training set of the noisy version ASVspoof 2015 corpus. To this end, the ideal or target mask of every utterance is calculated using (4), and the optimizing criterion is the binary cross entropy presented in (6).

2) Spoofing identity vector extraction: FBANKs and MGDs (along with the corresponding SNMs) are processed by two independent gated recurrent convolutional networks, and their outputs of 480 components are stacked into one single spoofing identity vector of 960 components. As shown in Fig. 6, this identity vector is passed to a fully connected layer of 960 neurons, whose output is then passed to a softmax layer to carry out the classification of the utterance into the genuine class or into one of the K spoofing attacks present in the training set. Therefore, the parameters of the two parallel gated recurrent convolutional layers are optimized jointly, being each set of layers specialized in processing either the amplitude or the phase based features.

Each GRCNN has $N = 2$ layers where the first recurrent layer has $M_0 = 16$ convolutional filters of size 9×9 and the second recurrent layer has $M_1 = 32$ convolutional filters of size 5×5. As shown in Fig. 3, there are 6 single-layer convolutional nets inside each gated recurrent convolutional unit cell, and although they have the same number of filters, they are totally independent (do not share any weights). Moreover, a max pooling filter of size 3×3 is applied between the 2 recurrent layers in order to reduce the size of the deep feature maps. In this way, the output of the first GRCNN layer is a volume of size $[16 \times 48 \times 31]$, which is reduced to $[16 \times 16 \times 10]$ after this max pooling operation. Then, the second layer of the GRCNN employs 32 feature maps resulting in an output volume of $[32 \times 16 \times 10]$. After that, a second max pooling operation is applied to the output of the second layer to reduce the final volume to $[32 \times 5 \times 3]$. In both pooling layers, we employ a stride of 3 and valid padding. Finally, the 32 feature maps of size $[5 \times 3]$ are flattened to make up a deep feature vector of 480 components.

3) Training setup and toolkits: The proposed deep learning framework was trained using the Adam optimizer [38] with a learning rate of $3 \cdot 10^{-4}$. Also, early stopping was applied to stop the training process when no improvement of the cross entropy across the validation set is obtained after five epochs. To prevent the problem of over-fitting, a fixed 30% dropout was applied in the convolutional layers. All the specified hyperparameters of the system were optimized using the validation set of the data corpus. The Pytorch toolkit [39] was employed to implement the deep learning frameworks. On the other hand, the FBANK features were obtained using the HTK toolkit [40] and they were normalized in mean and variance, while the MGD features were obtained using the Covarep toolkit [41].

4) Classifier: After the extraction of the spoofing identity vector which represents every utterance, these can be used with different back-end classifiers. The objective of the classifier is to assign a score indicating whether the utterance is genuine or spoofed. In this work, some popular classifiers in ASV are compared for spoofing detection: (i) support vector machine (SVM), (ii) one-class support vector machine (One-Class SVM [42]), which is trained using only genuine speech data, (iii) linear discriminant analysis (LDA), which projects the spoofing identity vectors onto $K-1$ dimensions and uses only the genuine class for scoring in the evaluation phase, and (iv) its probabilistic version (PLDA).

We also performed some preliminary experiments with a GMM classifier with unsuccessful results. This could be potentially due to a combination of two factors: (i) curse of dimensionality and (b) insufficient data for robust parameter estimation, as there is only one spoofing identity vector per utterance.

C. Performance Metric

The equal error rate (EER) is used to evaluate the system performance. It was computed using the Bosaris toolkit [43]. As described in the ASVspoof 2015 challenge evaluation plan [3], the EER was computed independently for each spoofing algorithm and then the average EER across all attacks was used. Similarly, the different noisy conditions of the Noisy ASVspoof 2015 corpus were evaluated individually to obtain the EER for each scenario.

V. Results

This section presents an experimental evaluation of the proposed techniques. First, section V-A evaluates the proposed GRCNN architecture on clean conditions using the original

TABLE II
COMPARISON OF CLASSIFIERS AND SPECTRAL FEATURES USING THE PROPOSED GRCNN SYSTEM ON THE ASVSPOOF 2015 EVALUATION CLEAN DATA SET OF IN TERMS OF (%) EER

Classifier	Features	Known	Unknown	
			S6 - S9	S10
SVM-One	FBANK	0.18	0.53	5.45
	MGD	0.32	0.59	6.35
	FBANK + MGD	0.12	0.55	4.42
SVM	FBANK	0.22	0.53	5.78
	MGD	0.41	0.55	6.15
	FBANK + MGD	0.18	0.46	4.21
LDA	FBANK	0.14	0.42	4.75
	MGD	0.31	0.43	5.34
	FBANK + MGD	0.08	0.09	3.11
PLDA	FBANK	0.06	0.33	4.45
	MGD	0.26	0.36	5.12
	FBANK + MGD	**0.02**	**0.09**	**2.51**

TABLE III
COMPARISON OF THE FUSION OF SEPARATED FBANK + GRCNN AND MGD + GRCNN SYSTEMS WITH THE JOINT FBANK + MGD + GRCNN SYSTEM ON THE ASVSPOOF 2015 EVALUATION CLEAN DATA SET IN TERMS OF (%) EER

System	Known	Unknown	
		S6 - S9	S10
Fusion Scores SVM-One	0.14	0.45	5.13
FBANK + MGD + SVM-One	0.12	0.55	4.42
Fusion Scores SVM	0.20	0.42	5.44
FBANK + MGD + SVM	0.18	0.46	4.21
Fusion Scores LDA	0.12	0.37	4.51
FBANK + MGD + LDA	0.08	0.09	3.11
Fusion Scores PLDA	0.04	0.24	4.21
FBANK + MGD + PLDA	**0.02**	**0.09**	**2.51**

ASVspoof 2015 corpus. Then, Section V-B is devoted to evaluate the noise robustness of the system with the proposed SNMs estimation technique on the noisy version of the ASVspoof 2015 database.

A. Architecture evaluation on clean speech

Table II shows the EER results obtained using different input features and classifiers with the GRCNN model on the clean ASVspoof 2015 database. The results for the known and unknown attacks of the evaluation set are presented separately. Moreover, the results for the unknown S10 attack are also presented, since this is the most difficult attack to detect.

In addition to our GRCNN model using as input both the FBANK and MGD features, we evaluated a fusion of separated FBANK + GRCNN and MGD + GRCNN systems. The rationale of this comparison is to determine whether the joint FBANK + MGD + GRCNN system, as shown in Fig. 6, can exploit better the input information than a fusion. The results of this evaluation are shown in Table III. The fusion is performed by normalizing the individual scores to zero mean and unit variance using the pre-computed mean and standard variance which are estimated on the training set. Finally, the weighted average of the two scores obtained by the individual systems is calculated for the detection decision.

As can be seen, the best result is obtained using a PLDA classifier and employing FBANKs and MGDs jointly as input features. The combination of FBANK and MGD features to feed the deep feature extractor obtains the best performance independently of the classifier, outperforming the fusion of the individual systems FBANK + GRCNN and MGD + GRCNN. This can be explained by the fact that the proposed GRCNN is optimized using the magnitude and phase information of the signal, being able to detect different artifacts of the spoofing attacks from correlations detected between both types of features. In the case of only employing one type of spectral features to feed the deep feature extractor, FBANKs outperform MGD features independently of the classifier, although the difference is not really significant. This indicates that although it is important to use the magnitude of the signal spectrum to detect spoofing attacks, the phase also provides meaningful information about the artifacts present in the spoofed speech.

Regarding the classifiers, PLDA yields the lowest EER independently of the input spectral features. Furthermore, LDA outperforms SVM and SVM-One for all attacks. There are significant differences of performance depending on the final classifier, but in general none of these perform very poorly in the S10 attack in comparison with the results obtained in the challenge ASVspoof 2015.

Based on the results from Tables II and III, in the rest of the evaluation we will use the GRCNN architecture jointly employing both FBANK and MGD features, and a PLDA classifier to make the final detection decision (spoofed or genuine speech). Table IV compares the performance of our proposal with other relevant anti-spoofing systems from the literature in the clean version of the ASVspoof 2015 corpus. The 4 first systems are based on deep learning frameworks, whereas the remaining 3 systems are based on the extraction of features specifically developed to detect spoofing attacks (CFCC-IF [44], CQCC [45] and LTSS [46]). We can observe that all systems achieve low EERs on the attacks S1 to S9. The main source of error is the S10 attack, for which we can observe meaningful differences of performance.

Compared to the deep learning based systems (Spectro + CNN + RNN [22], Best DNN [21], Best RNN [21] and FBANK + CNN + RNN [8]), our proposal outperforms all of them for the known and unknown attacks. It can be observed that the EER in the S10 attack is significantly lower than in those systems, independently of the input features and classifier employed. The performance for the S10 attack is only outperformed by the CQCC + GMM and LTSS + MLP which, to the best of our knowledge, have obtained the best overall average performance. However, compared to these systems, our proposal performs 0.03 and 0.07% better on average in the known attacks, as well as 0.18 and 0.02% better on average in the unknown S6-S9 attacks, respectively. Furthermore, we will see in the next subsection that our GRCNN-based system can also outperform CQCC + GMM and LTSS + MLP for S10 attack in clean conditions when multi-condition training is applied.

TABLE IV
COMPARISON WITH OTHER SYSTEMS OF THE LITERATURE ON THE ASVSPOOF 2015 EVALUATION CLEAN DATA SET IN TERMS OF (%) EER

System	Known Attacks						Unknown Attacks						Total
	S1	S2	S3	S4	S5	S1-S5	S6	S7	S8	S9	S10	S6-S10	Avg.
Spectro + CNN + RNN [22]	0.16	0.50	0.03	0.03	1.38	0.40	0.85	0.91	0.03	0.59	14.27	3.33	1.86
Best DNN [21]	**0.00**	0.10	**0.00**	**0.00**	0.20	0.10	0.20	**0.00**	**0.00**	**0.00**	25.5	5.10	2.60
Best RNN [21]	**0.00**	0.90	**0.00**	**0.00**	0.30	0.20	0.80	0.50	**0.00**	0.70	10.70	2.50	1.40
FBANK + CNN + RNN [8]	0.00	0.08	**0.00**	**0.00**	0.07	0.03	0.22	0.10	0.08	0.13	9.34	1.97	1.00
CFCC-IF [44]	0.10	0.86	**0.00**	**0.00**	1.08	0.41	0.85	0.24	0.14	0.35	8.49	2.01	1.21
CQCC + GMM [45]	**0.00**	0.10	**0.00**	**0.00**	0.13	0.05	**0.10**	0.06	1.03	0.05	**1.07**	0.46	0.26
LTSS + MLP [46]	0.01	0.15	**0.00**	**0.00**	0.35	0.10	0.29	0.05	0.04	0.07	1.56	**0.40**	**0.25**
FBANK + MGD + GRCNN + PLDA	**0.00**	**0.06**	**0.00**	**0.00**	**0.06**	**0.02**	0.12	0.08	0.09	0.03	2.51	0.57	0.30

B. Noise robustness evaluation

1) Evaluation on Seen Conditions: Table V presents the per-attack results of the proposed anti-spoofing system on the seen conditions of the noisy ASVspoof 2015 corpus evaluation set. Multi-condition training is applied in order to get high level features that are more robust against noise. Moreover, the proposed signal-to-noise masks of Section III are also employed to mitigate the effects of noise.

For all types of noisy conditions, it can be observed that the EER is higher when the signal-to-noise ratio decreases. When the noise power increases, the artifacts present in the spoofed signal are more difficult to detect, as those artifacts can be concealed by the noise. *Babble* and *street* are the most challenging seen conditions for detecting attacks, as they present non-stationary noises. Moreover, reverberation is the distorsion type which is easier to counteract, independently of the three types of impulse responses.

A very noticeable result obtained from these experiments under multi-condition training is the excellent performance of the proposed GRCNN under clean conditions. In fact, it practically equals the performance of the CQCC + GMM system (shown in table IV) in case of multi-condition training, and outperforms it if SNM masks are also employed. Specifically, our proposal performs 0.04% and 0.15% better in the known and unknown attacks, respectively (including SNMs). This result suggests that the variability introduced by the noise employed for the multi-condition training increases the generalization capability of the proposed network architecture. Furthermore, it also suggests that the SNM masks make the GRCNN focus on the spectral regions where speech is present.

2) Evaluation on Unseen Conditions: Although it is possible to collect multiple noise types for training and optimize the model using multi-condition training, there would still be many unseen noisy scenarios in real applications. Accordingly, to validate the effectiveness and generalization capability of the proposed approach, an evaluation on unseen noisy scenarios is performed, and the detailed results of different training techniques (clean, multi-condition and signal-to-noise masks training) are shown in Table VI. Additionally, Fig. 7 shows a box plot of the averaged EERs obtained by these training techniques on the unseen noisy evaluation scenarios.

First of all, in order to assess the impact of noisy environments, a baseline test using the clean model (without SNMs) is performed. In this case, the GRCNN is trained just using the clean ASVspoof 2015 corpus, and then used to extract deep features. It is observed that the clean-condition training technique only yields good performance in the matched clean data. However, in the case of testing with noisy data, large performance drops are observed due to the existing mismatch.

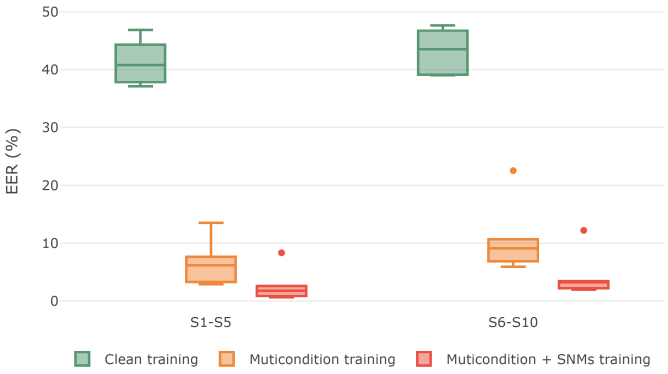

Fig. 7. Box plot of averaged EERs (%) for unseen noisy evaluation scenarios employing different training techniques. Box edges are at 25% and 75% quantiles.

Then, multi-condition training (without SNMs) using the seen noise data (*white*, *babble*, *street* and *reverberation*) is evaluated. Compared to the clean-condition training, the performance of the system is dramatically improved in all noisy conditions. The EERs are decreased more than 30% in all unseen conditions. This is due to the invariant effects across different acoustic conditions which the deep features learned from the multi-condition training.

After that, multi-condition training and the proposed SNMs are employed to feed the GRCNN. Compared to simple multi-condition training, the performance of the system is meaningfully higher in all unseen noisy conditions. In fact, the averaged EERs are decreased 5.28% and 8.52% for the known and unknown attacks, respectively. These results show the benefits of providing the neural network with information about the noise present in each time-frequency bin, so that it can discriminate which bins are more reliable to detect spoofing artifacts.

3) Comparison with other systems: Table VII compares the proposed approach (GRCNN + MASK-2) with five different systems on the noisy version of the ASVspoof 2015 database: CQCC + GMM evaluated in [19], CNN + NAT [8], CNN +

TABLE V
PERFORMANCE ACHIEVED WITH MULTI-CONDITION TRAINING AND SIGNAL-TO-NOISE MASKS FOR THE SEEN SCENARIOS OF THE ASVSPOOF 2015 EVALUATION NOISY DATA SET IN TERMS OF (%) EER

Evaluated Condition	Known Attacks						Unknown Attacks						Total
	S1	S2	S3	S4	S5	S1-S5	S6	S7	S8	S9	S10	S6-S10	Avg.
Clean	0.00	0.02	0.00	0.00	0.03	0.01	0.02	0.01	0.00	0.00	1.51	0.31	0.16
White (SNR = 20 dB)	0.04	0.92	0.00	0.00	0.96	0.38	0.21	0.08	0.04	0.11	6.84	1.46	0.92
White (SNR = 10 dB)	0.29	1.57	0.00	0.00	2.58	0.89	1.14	0.43	0.08	0.19	8.82	2.13	1.51
White (SNR = 0 dB)	2.23	6.12	0.17	0.17	7.91	3.32	4.13	3.89	0.32	1.92	17.41	5.54	4.43
Babble (SNR = 20 dB)	0.03	2.11	0.00	0.00	2.71	0.97	0.18	0.10	0.04	0.09	7.12	1.51	1.24
Babble (SNR = 10 dB)	0.47	5.04	0.00	0.00	5.89	2.28	0.92	0.42	0.06	0.19	9.78	2.59	2.44
Babble (SNR = 0 dB)	4.12	12.54	0.56	0.56	14.62	6.68	5.62	4.88	0.48	2.69	21.56	7.05	6.87
Street (SNR = 20 dB)	0.04	3.11	0.00	0.00	3.30	1.29	0.28	0.15	0.09	0.12	8.24	1.78	1.54
Street (SNR = 10 dB)	0.42	4.83	0.00	0.00	5.70	2.21	1.10	0.52	0.07	0.24	10.12	2.41	2.31
Street (SNR = 0 dB)	3.58	9.34	0.35	0.35	12.98	5.32	1.89	4.15	0.56	2.56	20.24	5.88	5.60
Reverberation (T60 = 0.3 s)	0.00	0.06	0.00	0.00	0.15	0.22	0.18	0.07	0.06	0.05	4.22	0.92	0.57
Reverberation (T60 = 0.6 s)	0.00	0.20	0.00	0.00	0.19	0.10	0.37	0.18	0.15	0.23	4.53	1.09	0.60
Reverberation (T60 = 0.9 s)	0.08	1.12	0.00	0.00	0.78	0.40	0.82	0.34	0.21	0.31	5.09	1.35	0.88
Avg. EER seen conditions	0.94	3.91	0.36	0.36	4.81	2.00	1.40	1.27	0.18	0.73	10.33	2.81	2.41

TABLE VI
COMPARISON OF DIFFERENT TRAINING TECHNIQUES FOR THE CLEAN AND UNSEEN SCENARIOS OF THE ASVSPOOF 2015 EVALUATION NOISY DATA SET IN TERMS OF (%) EER

Evaluated Condition	Clean-condition Training			Multi-condition Training			Multi-condition + SNMs Training		
	Known	Unknown	Avg.	Known	Unknown	Avg.	Known	Unknown	Avg.
Clean	0.02	0.57	0.30	0.02	0.48	0.25	0.01	0.31	0.16
Cafe (SNR = 20 dB)	37.82	39.04	38.43	3.29	6.87	5.08	1.27	2.21	1.74
Cafe (SNR = 10 dB)	37.12	42.84	39.98	6.32	10.67	8.50	2.31	3.42	2.87
Cafe (SNR = 0 dB)	46.85	47.62	47.24	13.51	22.54	18.03	8.34	12.21	10.28
Volvo (SNR = 20 dB)	38.78	39.14	38.96	2.89	5.93	4.41	0.65	1.96	1.31
Volvo (SNR = 10 dB)	42.79	44.21	43.50	6.04	8.68	7.36	0.87	3.02	1.95
Volvo (SNR = 0 dB)	44.31	46.73	45.52	7.65	9.58	8.62	2.61	3.42	3.02
Avg. EER unseen conditions	41.28	43.26	42.27	6.62	10.71	8.67	1.34	2.19	1.76

TABLE VII
COMPARISON OF DIFFERENT TECHNIQUES ON THE ASVSPOOF 2015 EVALUATION NOISY DATA SET IN TERMS OF AVERAGE (%) EER USING MULTI-CONDITION TRAINING

Evaluated Condition	CQCC + GMM [19]			CNN + NAT [8]			CNN + MASK-1 + RNN [8]			nat-DNN + nat-CNN + nat-RNN [19]			GRCNN + MASK-2		
	Kn.	Un.	Avg.	Kn.	Un.	Avg.	Kn.	Un.	Avg.	Kn.	Un.	Avg.	Kn.	Un.	Avg.
Clean	0.10	0.90	0.50	0.14	2.03	1.09	0.03	0.90	0.47	**0.00**	1.30	0.70	0.01	**0.31**	**0.16**
White (SNR = 20 dB)	46.8	44.6	45.7	1.7	4.3	3.0	0.8	2.5	1.7	**0.4**	2.8	1.6	**0.4**	**1.5**	**0.9**
White (SNR = 10 dB)	48.9	48.1	48.5	3.2	5.1	4.4	2.3	3.4	2.9	1.2	3.2	2.2	**0.9**	**2.1**	**1.5**
White (SNR = 0 dB)	49.3	48.9	49.1	7.9	10.0	9.0	5.9	8.6	7.3	3.8	7.2	5.5	**3.3**	**5.5**	**4.4**
Babble (SNR = 20 dB)	18.2	18.3	18.3	3.1	4.6	3.9	2.3	3.9	3.1	1.1	2.7	1.9	**1.0**	**1.5**	**1.2**
Babble (SNR = 10 dB)	33.9	33.6	33.8	5.7	6.7	6.2	3.7	4.5	4.1	3.4	4.0	3.7	**2.3**	**2.6**	**2.4**
Babble (SNR = 0 dB)	44.6	44.0	44.3	12.9	14.7	13.8	9.5	10.6	10.1	7.3	10.0	8.6	**6.7**	**7.1**	**6.9**
Street (SNR = 20 dB)	22.7	22.3	22.5	3.9	5.1	4.5	1.9	3.1	2.5	2.0	3.3	2.6	**1.3**	**1.8**	**1.5**
Street (SNR = 10 dB)	37.5	36.3	36.9	6.1	7.5	6.8	4.1	5.4	4.8	3.7	4.4	4.1	**2.2**	**2.4**	**2.3**
Street (SNR = 0 dB)	46.1	45.4	45.8	11.1	13.7	12.4	8.7	9.9	9.3	6.3	9.0	7.7	**5.3**	**5.9**	**5.6**
Reverberation (T60 = 0.3 s)	8.4	9.3	8.9	1.3	2.1	1.7	1.1	1.9	1.5	**0.2**	1.3	0.7	**0.2**	**0.9**	**0.6**
Reverberation (T60 = 0.6 s)	10.6	7.8	9.2	1.6	2.0	1.8	1.6	1.5	1.6	0.3	1.2	0.8	**0.1**	**1.1**	**0.6**
Reverberation (T60 = 0.9 s)	7.6	6.9	7.3	1.5	1.9	1.7	1.1	1.6	1.4	0.5	**1.4**	**0.9**	**0.4**	**1.4**	**0.9**
Avg. EER Seen Conditions	31.2	30.5	30.8	5.0	6.5	5.8	3.6	4.7	4.2	2.5	4.2	3.4	**2.0**	**2.8**	**2.4**
Cafe (SNR = 20 dB)	30.7	30.1	30.4	2.9	5.3	4.1	1.8	4.5	3.2	3.0	4.6	3.8	**1.3**	**2.2**	**1.7**
Cafe (SNR = 10 dB)	42.1	41.3	41.7	5.6	8.1	6.9	4.5	5.7	5.1	5.5	7.4	6.4	**2.3**	**3.4**	**2.9**
Cafe (SNR = 0 dB)	49.8	47.1	47.3	13.5	20.0	16.8	10.1	14.3	12.2	12.9	18.4	15.6	**8.3**	**12.2**	**10.3**
Volvo (SNR = 20 dB)	0.9	2.7	1.8	1.0	3.7	2.4	0.8	3.0	1.9	0.8	2.8	1.8	**0.7**	**2.0**	**1.3**
Volvo (SNR = 10 dB)	4.3	5.6	4.9	2.4	4.9	3.7	1.5	3.4	2.5	2.3	4.0	3.2	**0.9**	**3.0**	**2.0**
Volvo (SNR = 0 dB)	13.0	13.0	13.0	3.7	5.0	4.4	2.7	3.5	3.1	3.7	4.7	4.2	**2.6**	**3.4**	**3.0**
Avg. EER Unseen Conditions	23.1	23.3	23.2	4.9	7.8	6.4	3.6	5.7	4.7	4.7	7.0	5.8	**1.3**	**2.2**	**1.8**

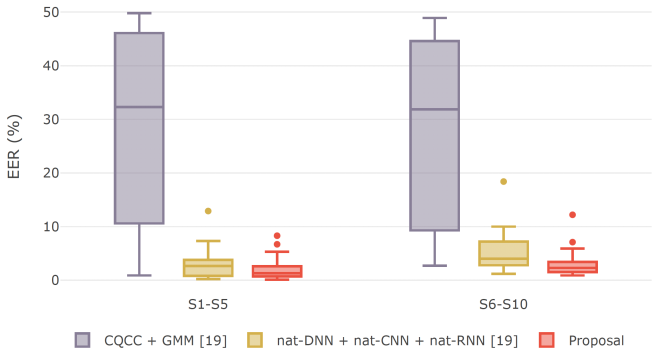

Fig. 8. Box plot of averaged EERs (%) for all noisy evaluation scenarios obtained by: (i) CQCC + GMM as evaluated in [19], (ii) a combination of a DNN, CNN and BLSTM [19], and (iii) our proposal. Box edges are at 25% and 75% quantiles.

MASK-1 + RNN [8], and a combination of three different neural networks (DNN, CNN and BLSTM) [19]. The terms MASK-1 and MASK-2 refer to different signal-to-noise mask estimation techniques, being MASK-1 the technique proposed in [8], and MASK-2 is the mask estimation technique proposed here in Section III. The CNN + NAT system was proposed in [19], but as its performance is not provided in this reference for the seen conditions, we evaluated it on all conditions in our previous work [8]. The term NAT stands for Noise-Aware Training, in which a mean noise vector of the utterance is appended to the input features. Additionally, Fig. 8 shows a boxplot of the averaged EERs obtained by CQCC + GMM, the fusion of systems proposed in [19] and our proposed system on all noisy evaluation scenarios.

When multi-condition training is used, our proposed GRCNN + MASK-2 system achieves the best overall performance in the clean condition, even outperforming CQCC + GMM, which was the best system in Table IV. Compared to CQCC + GMM and the fusion of systems proposed in [19], it achieves a 0.34% and 0.54% better overall EER, respectively.

When evaluated under noisy conditions, the CQCC + GMM system performs very poorly even for the seen noises (those used for multi-condition training). On the contrary, our proposed system achieves the best results with an overall relative improvement of 28.4% compared to CQCC + GMM. Moreover, although CNN + NAT and our previous proposal CNN + MASK-1 + RNN already improved the performance on all noisy conditions compared to CQCC + GMM, the proposed system outperforms both of them in 3.4% and 1.8% on the averaged EER of seen conditions, respectively.

Despite the fact that the combination of systems proposed in [19] is not directly comparable with our GRCNN + MASK-2, since, unlike our proposal, it is a fusion of techniques, it is worth mentioning that our system achieves better performance in both seen and unseen distorted conditions. This indicates that the proposed GRCNN achieves a better utterance-level representation than averaging the frame-level deep features to extract the spoofing identity vector of the utterance. In addition, the proposed mask estimation technique is better than extracting the mean noise vector of the utterance in order to provide the neural network with information about the noise present in the utterance.

VI. Conclusion

In this paper we have proposed a novel technique for the extraction of deep identity features for an efficient detection of TTS and VC attacks in clean and noisy environments. In our system, a gated recurrent convolutional neural network is employed to integrate the extraction of discriminative features at frame level and the utterance-level identity vector into a single network, providing information about whether the utterance is genuine or spoofed. Moreover, the anti-spoofing system has been trained with magnitude and phase spectral features (FBANKs and MGDs), yielding better results than a fusion of single systems which are fed with one type of these features. Experimental results on the clean ASVspoof 2015 corpus have shown that the proposed architecture outperforms all the state-of-the-art deep learning systems to detect logic access attacks.

Furthermore, to increase the noise robustness of our anti-spoofing detector, a signal-to-noise mask estimation technique has been proposed. Our proposal has been evaluated on a distorted version of the ASVspoof 2015 corpus, including both additive and noisy reverberant scenarios. The experimental results have shown that our proposal obtains the best state-of-the-art results for this corpus, when using multi-condition training along with the proposed signal-to-noise masks, outperforming the state-of-the-art CQCC + GMM system (the best system for the clean ASVspoof 2015 corpus and baseline of the ASVspoof 2017 challenge [47]) and the fusion of deep feature extraction systems proposed in [19] for both clean and noisy conditions. Thus, as an additional result, we have found that the variability introduced in the multi-condition training increases the power of generalization of the proposed GRCNN, since the results obtained even in the evaluation clean condition dramatically improve with respect to those obtained with clean condition training.

In future work it would be worthwhile to investigate the integration of the ASV and anti-spoofing systems in order to study how the ASV system processes the noisy spoofed speech. Also, the proposed GRCNN architecture for spoofing detection should be also effective to detect replay attacks. Some modifications could be done to the architecture to ensure a successful detection, such as employing input features with a higher frequency resolution [48]. Finally, it would be interesting to explore other types of mask estimation techniques which do not require any knowledge of the corresponding clean signal of a given noisy utterance in order to train the model, which would allow us to collect more data for training.

Acknowledgment

This work has been supported by the Spanish MINECO/FEDER Project TEC2016-80141-P and the Spanish Ministry of Education through the National Program FPU (grant reference FPU16/05490). We also acknowledge the support of NVIDIA Corporation with the donation of a Titan X GPU. Moreover, we would like to thank Mr. Xiaohai

Tian from Nanyang Technological University, Singapore for sharing the noisy version of ASVspoof 2015 database.

REFERENCES

[1] Z. Wu, N. Evans, T. Kinnunen, J. Yamagishi, F. Alegre, and H. Li, "Spoofing and countermeasures for speaker verification: A survey," *Speech Communication*, vol. 66, pp. 130–153, 2015.

[2] Z. Wu et al., "Anti-spoofing for text-independent speaker verification: An initial database, comparison of countermeasures, and human performance," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 4, pp. 768–783, 2016.

[3] 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," *Proc. Interspeech*, 2015.

[4] N. Chen, Y. Qian, H. Dinkel, B. Chen and K. Yu, "Robust Deep Feature for Spoofing Detection - The SJTU System for ASVspoof 2015 Challenge," *Proc. Interspeech*, 2015.

[5] F. Grezl, M. Karafiat, S. Kontar, and J. Cernocky, "Probabilistic and bottle-neck features for Ivcsr of meetings," *Proc. International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2007.

[6] Z. Wu, and S. King, "Improving trajectory modeling for dnn-based speech synthesis by using stacked bottleneck features and minimum trajectory error training," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 7, pp. 1255–1265, 2016.

[7] S. Yadav, and A. Rai, "Learning Discriminative Features for Speaker Identification and Verification," *Proc. Interspeech*, 2018.

[8] A. Gomez-Alanis, A.M. Peinado, J.A. Gonzalez, and A.M. Gomez, "A Deep Identity Representation for Noise Robust Spoofing Detection," *Proc. Interspeech*, 2018.

[9] X. Tian, Z. Wu, X. Xiao, E.S. Chng, and H. Li, "Spoofing detection from a feature representation perspective," *Proc. International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016.

[10] C. Zhang, S. Ranjan, M. Nandwana, Q. Zhang, A. Misra, G. Liu, F. Kelly, and J.H., "Joint information from nonlinear features for spoofing detection," *Proc. International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016.

[11] A. Gomez-Alanis, A.M. Peinado, J.A. Gonzalez, and A.M. Gomez, "Performance evaluation of front- and back-end techniques for ASV spoofing detection systems based on deep features," *Proc. Iberspeech*, 2018.

[12] X. Xiao, X. Tian, S. Du, H. Hu, E.S. Chng, and H. Li, "Spoofing Detection Using High Dimensional Magnitude and Phase Features: the NTU System for ASVspoof 2015 Challenge," *Proc. Interspeech*, 2015.

[13] Y. Liu, Y. Tian, L. He, J. Liu, M.T. Johnson, "Simultaneous Utilization of Spectral Magnitude and Phase Information to Extract Supervectors for Speaker Verification Anti-Spoofing," *Proc. Interspeech*, 2015.

[14] X. Wang, Y. Xiao, and X. Zhu, "Feature selection based on CQCCs for automatic speaker verification spoofing," *Proc. Interspeech*, 2017.

[15] S. Jelil, R.K. Das, S.R.M. Prasanna, and R. Sinha, "Spoof Detection Using Source, Instantaneous Frequency and Cepstral Features," *Proc. Interspeech*, 2017.

[16] H. Muckenhirn, M. Magimai-Doss, and S. Marcel, "End-to-End Convolutional Neural Network-based Voice Presentation Attack Detection," *Proc. IEEE International Joint Conference on Biometrics (IJCB)*, 2017.

[17] K. Okabe, T. Koshinaka, and K. Shinoda, "Attentive Statistics Pooling for Deep Speaker Embedding," *Proc. Interspeech*, 2018.

[18] J. Yang, C. You, and Q. He, "Feature with Complementary of Statistics and Principal Information for Spoofing Detection," *Proc. Interspeech*, 2018.

[19] Y. Qian, N. Chen, H. Dinkel, and Z. Wu, "Deep Feature Engineering for Noise Robust Spoofing Detection," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 10, pp. 1942–1955, 2017.

[20] J. Huang, J. Li, and Y. Gong, "An Analysis of Convolutional Neural Networks for Speech Recognition," *Proc. International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015.

[21] Y. Qian, N. Chen, and K. Yu, "Deep features for automatic spoofing detection," *Speech Communication*, vol. 85, pp. 43–52, 2016.

[22] C. Zhang, C. Yu, and J. H. L. Hansen, "An investigation of Deep-Learning Frameworks for Speaker Verification Antispoofing," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, pp. 684–694, 2017.

[23] H. Delgado, M. Todisco, Md Sahidullah, N. Evans, T. Kinnunen, K. A. Lee, and J. Yamigishi, "ASVSpoof 2017 Version 2.0: meta-data analysis and baseline enhancements," *Proc. Odyssey*, 2018.

[24] X. Tian, Z.Wu, X. Xiao, E. S. Chng, and H. Li, "An investigation of spoofing speech detection under additive noise and reverberant condition," *Proc. Interspeech*, 2016.

[25] C. Hanilci, T. Kinnunen, M. Sahidullah, and A. Sizov, "Spoofing detection goes noisy: An analysis of synthetic speech detection in the presence of additive noise," *Speech Communication*, vol. 85, pp. 83–97, 2016.

[26] N. Ballas, L. Yao, C. Pal, and A. Courville, "Delving Deeper into Convolutional Networks for Learning Video Representations," *Proc. International Conference on Learning Representations (ICLR)*, 2016.

[27] Y. Xu, Q. Kong, Q. Huang, W. Wang, and M. Plumbley, "Convolutional Gated Recurrent Neural Network Incorporating Spatial Features for Audio Tagging," *Proc. International Joint Conference on Neural Networks (IJCNN)*, 2017.

[28] J. Wang, and X. Hu, "Gated Recurrent Convolutional Neural Network for OCR," *Proc. Neural Information Processing System (NIPS)*, 2017.

[29] M. P. Cooke, P. Green, L. Josifovski, and A. Vizinho, "Robust automatic speech recognition with missing and unreliable acoustic data," *Speech Communication*, vol. 34, pp. 267–285, 2001.

[30] J. P. Barker, L. Josifovski, M. P. Cooke, and P. D. Green, "Soft decisions in missing data techniques for robust automatic speech recognition," *Proc. ICSLP*, 2000.

[31] S. Hochreiter, and J. Schmidhuber, "Long Short-Term Memory," *Journal of Neural Computation*, vol. 9, pp. 1735–1780, 1997.

[32] K. Cho, B. Van Merrienböer, C. Gulcehre, D. Bahdanau, F. Bouhgares, H. Schwenk, and Y. Bengio , "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation," *Proc. Empirical Methods in Natural Language Processing (EMNPL)*, 2014.

[33] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling," *arXiv:1412.3555*, 2014.

[34] M. Witkowski, S. Zacprzak, P. Zelasko, K. Kowalczyk, and J. Galka, "Audio Replay Attack Detection Using High-Frequency Features," *Proc. Interspeech*, 2017.

[35] Y. Bengio, P. Simard, and P. Frascani, "Learning long-term dependencies with gradient descent is difficult," *IEEE Transactions on Neural Networks*, vol. 5, pp. 157–166, 1994.

[36] A.M. Peinado and J.C. Segura, "Speech Recognition Over Digital Channels: Robustness and Standards," Wiley, 2006, pp. 207–208.

[37] H. Murthy and V. Gadde, "The modified group delay function and its application to phoneme recognition," *Proc. International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2003.

[38] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv:1412.6890*, 2014.

[39] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Dasmaison, L. Antiga and A. Lerer, "Automatic Differentiation in Pytorch", *Proc. Neural Information Processing Systems (NIPS)*, 2017.

[40] S. Young, et al. The HTK Book, Version 3.4. Cambridge University Engineering Department (2006).

[41] G. Degottex, J. Kane, T. Drugman, T. Raitio and S. Scherer, "COVAREP - A collaborative voice analysis repository for speech technologies", *Proc. International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014.

[42] Scholkopf, B., Williamson, R. C., Smola, A. J., et al., "Support vector method for novelty detection," *Proc. Neural Information Processing System (NIPS)*, 2000.

[43] N. Brümmer and E. deVilliers, "The BOSARIS toolkit: Theory, algorithms and code for surviving the new DCF," *Proc. NIST SRE11 Speaker Recognition Workshop*, 2011.

[44] T. B. Patel and H. A. Patil, "Combining evidences from mel cepstral, cochlear filter cepstral and instantaneous frequency features for detection of natural vs. spoofed speech," *Proc. Interspeech*, 2015.

[45] M. Todisco, H. Delgado, and N. Evans, "A new feature for automatic speaker verification anti-spoofing: Constant Q cepstral coefficients," *Proc. Odyssey*, 2016.

[46] H. Muckenhirn, P. Korshunov, M. Magimai-Doss, and S. Marcel, "Long-Term Spectral Statistics for Voice Presentation Attack Detection," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 11, pp. 2098–2111, 2017.

[47] H. Delgado, M. Todisco, Md Sahidullah, N. Evans, T. Kinnunen, K. A. Lee, and J. Yamigishi, "The ASVspoof 2017 Challenge: Asessing the Limits of Replay Spoofing Attack Detection," *Proc. Interspeech*, 2017.

[48] G. Lavrentyeva, S. Novoselov, E. Malykh, A. Kozlov, O. Kudashev, and V. Shchemelinin, "Audio replay attack detection with deep learning frameworks," *Proc. Interspeech*, 2017.