On Joint Optimization of Automatic Speaker Verification and Anti-spoofing in the Embedding Space

Alejandro Gomez-Alanis, Jose A. Gonzalez-Lopez, S. Pavankumar Dubagunta, *Student Member, IEEE,* Antonio M. Peinado, *Senior Member, IEEE,* Mathew Magimai.-Doss, *Member, IEEE.*

***Abstract*—Biometric systems are exposed to spoofing attacks which may compromise their security, and voice biometrics based on automatic speaker verification (ASV), is no exception. To increase the robustness against such attacks, anti-spoofing systems have been proposed for the detection of replay, synthesis and voice conversion-based attacks. However, most proposed anti-spoofing techniques are loosely integrated with the ASV system. In this work, we develop a new integration neural network which jointly processes the embeddings extracted from ASV and anti-spoofing systems in order to detect both zero-effort impostors and spoofing attacks. Moreover, we propose a new loss function based on the minimization of the area under the expected (AUE) performance and spoofability curve (EPSC), which allows us to optimize the integration neural network on the desired operating range in which the biometric system is expected to work. To evaluate our proposals, experiments were carried out on the recent ASVspoof 2019 corpus, including both logical access (LA) and physical access (PA) scenarios. The experimental results show that our proposal clearly outperforms some well-known techniques based on the integration at the score- and embedding-level. Specifically, our proposal achieves up to 23.62% and 22.03% relative equal error rate (EER) improvement over the best performing baseline in the LA and PA scenarios, respectively, as well as relative gains of 27.62% and 29.15% on the AUE metric.**

***Index Terms*—Automatic speaker verification (ASV), spoofing detection, embeddings, integration of ASV and anti-spoofing, expected performance and spoofability curve (EPSC).**

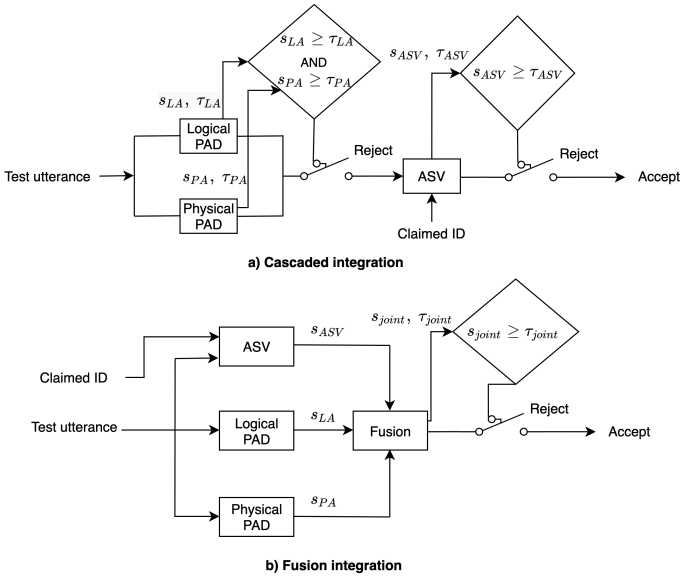

Fig. 1. Block diagram of two score-level integration systems: (a) cascaded (PAD preceeding ASV) integration; (b) fusion integration. s_{LA}, s_{PA}, s_{ASV} and s_{joint} denote the scores of the LA, PA, ASV and joint integration systems, respectively. Likewise, τ_{LA}, τ_{PA}, τ_{ASV} and τ_{joint} denote the thresholds of the same systems used for the decision of accepting or rejecting the test utterance.

I. Introduction

Biometric authentication [1] aims to authenticate the identity claimed by a given individual based on the samples measured from biological processes and/or organs (e.g., voice, face, and fingerprints). While the main biometric techniques can already handle noisy environments robustly [2], [3], their vulnerability to malicious *spoofing* attacks is still a serious concern nowadays [4], [5]. Our focus in this work is on spoofing detection for automatic speaker verification (ASV) [6], in which an impostor could gain fraudulent access to a system or resource (e.g., bank account) by presenting speech resembling the voice of a genuine user.

Four types of *spoofing* attacks have been identified [7]: (i) replay (i.e., using pre-recorded voice of the target user), (ii) impersonation (i.e., mimicking the voice of the target voice, where the twins fraud [8] is a specific form of the impersonation attack specially challenging), or either using (iii) text-to-speech synthesis (TTS) or (iv) voice conversion (VC) systems to generate artificial speech resembling the voice of a legitimate user. Moreover, these attacks can be presented to the ASV system according to two different scenarios: logical access (LA) and physical access (PA). In the PA attack scenario, the spoof signal is presented to or captured by the sensor, i.e., the microphone. Whilst, in the LA scenario, the sensor is by-passed and attacks are directly injected into the ASV system, normally generated using TTS or VC technologies.

Spoofing detection or presentation attack detection (PAD in ISO/IEC 30107 nomenclature [9]) for ASV has gained increased attention in recent years as evidenced by the organization of several evaluation campaigns (challenges): (i) ASVspoof 2015 [10], which focused on LA scenarios (TTS and VC attacks); (ii) BTAS 2016 [11], which addressed both the detection of LA and PA-based attacks; (iii) ASVspoof 2017

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

S. Pavankumar Dubagunta and Mathew Magimai.-Doss are with the Speech and Audio Processing group, Idiap Research Institute, Martigny 1920, Switzerland (e-mail: pavankumar.dubagunta@idiap.ch; mathew@idiap.ch).

[12], which focused on PA scenarios (real replay attacks) under noisy environments; and (iv) ASVspoof 2019 [13], which addressed both the detection of LA-based attacks generated with the latest TTS and VC technologies, and simulated replay attacks under different reverberant acoustic conditions.

While ASV and spoofing detection have been well studied separately so far, the integration of both systems still requires further research. This paper deals with this issue and proposes an embedding-level solution capable of achieving a significant improvement in terms of biometric authentication security. Fig. 1 shows two typical integration approaches for such systems: (a) cascaded or tandem integration in which PAD precedes ASV, or viceversa, and where utterances can be rejected by either the first or the second module; and (b) score fusion integration where the ASV and PAD scores are the inputs of a final classifier which assigns an unique score to the test utterance. In this work, however, we argue that this type of system integration is sub-optimal owing to the following reasons. First, these techniques calibrate the standalone and joint thresholds considering only one point of the *error rate* on the development set. However, it is difficult to predict the ideal operating point of the integration system, since the evaluation data is usually unseen and does not match the development data. Second, these systems typically handle two or three scores (ASV and PADs) obtained by independent classifiers, without exploiting the fact that ASV and PAD systems share the *bonafide* speech subspace. Recently, two joint ASV and anti-spoofing systems [14], [15] were studied in the i-vector [16] and x-vector [17] space, respectively. They obtained promising results, thus demonstrating the feasibility and advantage of a joint ASV/PAD decision at the embedding level.

Inspired by recent developments on deep learning methods, in which deep neural networks (DNNs) are used as powerful non-linear feature extraction front-ends to map variable-length sequences to fixed-dimensional embedding vectors, in this paper, we investigate on system integration at the embedding level. Specifically, we propose an embedding-level integration system based on a neural network whose parameters can be optimized in the range of operating points in which the biometric system is expected to work, which is easier to predict than a single calibration point. The main contributions of our work can be summarized as follows:

1) Integration Neural Network: We propose a new integration technique based on a DNN which processes three types of embeddings (ASV, LA, and PA) jointly. Due to the fact that embeddings extracted from ASV and PAD systems share the *bonafide* space (i.e., non-spoofed space of speech), the proposed system is able to exploit this fact in order to better discriminate between *bonafide* target speech and *zero-effort* impostors or *spoofing* attacks.

2) Loss Function: To train the integration neural network, we propose a new loss function which minimizes the area under the expected (AUE) [18] performance and spoofability curve (EPSC) [18]. This allows us to optimize the integration system in the operating range in which the biometric system is expected to work a priori.

3) Agnosticism: In order to be agnostic to the type of spoofing attacks that integration systems might encounter, we develop and evaluate different integration techniques under the presence of TTS, VC, and replay attacks. To the best of our knowledge, the existing integration techniques have only been trained to detect either TTS/VC or replay attacks. In addition, we compare the performance of the agnostic integration systems with the fusion of two similar non-agnostic integration systems which can only detect either LA- or PA-based attacks.

This paper is organized as follows. Section II outlines the ASV, PAD, and integration systems, as well as the metrics to evaluate them. Then, in Section III, we describe the proposed integration neural network and the new loss function specifically conceived to optimize it. After that, Section IV outlines the speech corpora, systems details, and metrics employed in the experiments. Then, Section V discusses the performance of the standalone ASV and PAD systems, in order to choose the best ones for building the integration systems which are evaluated in Section VI. Finally, we summarize the conclusions derived from this research in Section VII.

II. Background

This section briefly describes the existing standalone ASV and PAD approaches, as well as the metrics to evaluate them. Then, a detailed description of the existing integration systems and metrics are provided in Section II-C.

A. Standalone Automatic Speaker Verification (ASV) Systems

The goal of an ASV system is to determine whether a test utterance is produced by the claimed speaker $\mathcal{S}$ (hypothesis $\mathcal{H}_{\mathcal{S}}$) or not (hypothesis $\mathcal{H}_{\overline{\mathcal{S}}}$). The speaker information encoded in the utterance is typically represented as either i-vectors [16] or x-vectors [17]. In the verification stage, the i-vectors or x-vectors of the test and enrollment utterances are extracted, and then are usually mapped into a more discriminative subspace using linear discriminant analysis (LDA). Finally, the ASV score of each test utterance can be obtained using three main techniques:

1) Cosine scoring [19]: It does not require any training data. It uses the cosine distance to compute the score between the enrollment and test embeddings.

2) Probabilistic Linear Discriminant Analysis (PLDA) [20], [21]: This is a probabilistic framework able to model the intra- and inter-speaker variability. There are three types of PLDA models [22]: standard [20], simplified [23] and two-covariance [24]. All of them are trained using the expectation-maximization (EM) algorithm [25].

3) B-vector system [26]: This technique considers speaker verification as a binary classification problem. In particular, from the x-vectors $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ computed for each pair of utterances, a b-vector representing the relationship between $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ is computed as follows,

$$\boldsymbol{b} = \left[\boldsymbol{x}_1 \oplus \boldsymbol{x}_2, \boldsymbol{x}_1 \otimes \boldsymbol{x}_2, |\boldsymbol{x}_1 \ominus \boldsymbol{x}_2|\right], \tag{1}$$

where $\oplus$, $\otimes$ and $\ominus$ are the element-wise addition, multiplication, and subtraction operations, respectively. The b-vectors

computed from the dataset are fed to a binary DNN in order to classify them as positive or negative, i.e., determine whether the x-vectors $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are originated from the same or different speaker/s.

The evaluation of an ASV system is done in terms of the licit protocol [27], which only contains speech uttered by *bonafide* target speakers and *zero-effort* impostors. The most common metric to evaluate an ASV system is the equal error rate (EER), which is the operating point at which the *false acceptance rate* (FAR) equals the *false rejection rate* (FRR). However, the EER metric does not account for the costs of missing target users and falsely accepting impostors, nor the prior probabilities of each. To take these costs and priors into account, the detection cost function (DCF) framework [28] has been endorsed by the National Institute of Standards and Technology (NIST) within the scope of the speaker recognition evaluation (SRE) campaigns [29]. The costs and priors have varied across the different NIST SRE campaigns, being DCF08 [30] and DCF10 [31] two of the most popular metrics. However, the DCF still only measures the performance at a single operating point. To address this issue, NIST included the evaluation of the area under the curve (AUC), which is a visualization model for the receiver operating characteristic (ROC) curve. Then, the detection error tradeoff (DET) [32] curve was developed as a non-linear version of the ROC. However, the speaker recognition and ASVspoof community favors another non-linear way of ROC such as the ROC's convex hull (ROCCH) [33]. The ROCCH is the expectation of all possible optimistic and pessimistic ROC estimates. It relates to the minDCF metric and is summarized by the minimum log-likelihood ratio cost metric (C_{llr}^{min}) [34]. The former one is commonly used to analyze how well an ASV system performs and is calibrated across all operating points. When the ROCCH-EER is optimized, the entire ROC profile optimizes due to convexity, but this does not necessarily hold for other optimizations based on other EER estimates. Also, in order to enable a more realistic comparison between systems as well as a better analysis of their respective expected performance, the expected performance curve (EPC) framework [35] developed the area under the expected (AUE) performance curve, which also allows to measure the performance of an ASV system for a wide range of operating points. Most of these metrics are used in our experiments for evaluating standalone ASV systems.

B. Standalone Presentation Attack Detection (PAD) Systems

Spoofing detection is a binary classification task which aims at differentiating spoofed speech from *bonafide* speech. For each test utterance, two hypotheses are computed: either it is *bonafide* speech $\mathcal{N}$ ($\mathcal{H}_{\mathcal{N}}$), or it is a spoofing attack ($\mathcal{H}_{\overline{\mathcal{N}}}$).

There are two main machine learning models to detect spoofed speech [36]: (i) Gaussian mixture models (GMMs) and (ii) neural networks (NNs). A wide range of features have been proposed to train these models, such as spectrogram [37], linear frequency cepstral coefficients (LFCC) [38], constant Q cepstral coefficients (CQCC) [39], and raw speech samples [40]. In the last ASVspoof challenges [12], [13], deep learning has shown to be the most effective approach to detect spoofing.

TABLE I
CLASSIFICATION OF TRIALS IN ASV AND PAD SYSTEMS. SYMBOL "-" MEANS THAT EITHER ASV HAS NO CAPABILITY TO REJECT SPOOFING IMPOSTOR TRIALS OR THAT PAD CANNOT MAKE A DISTINCTION BETWEEN ZERO-EFFORT IMPOSTOR AND GENUINE TARGET TRIALS.

Class	C_1	C_2	C_3
System / Trial	Genuine target	Genuine non-target	Spoof target
ASV	Positive	Negative	-
PAD	Positive	-	Negative
ASV + PAD	Positive	Negative	Negative

The evaluation of standalone PAD systems is carried out in terms of the spoof protocol [27], which contains *bonafide* speech and *spoofing* attacks. Just like ASV, the EER metric is typically used to evaluate standalone anti-spoofing systems, where *false rejection* happens when a *bonafide* speech utterance is detected as a spoofing attack, and *false acceptance* occurs when spoofed speech is detected as *bonafide* speech. Recently, the ASV-constrained minimum tandem detection cost function (min-tDCF) metric [41] was proposed to evaluate a PAD system given a fixed ASV system, considering the priors and costs of the different hypotheses. This was the primary metric used in the last ASVspoof 2019 challenge [13].

C. Integration Systems: Joint ASV and PAD

In the joint approach, each utterance has two attributes: (i) an indicator of the *bonafide* speech ($\mathcal{N}$), and (ii) an indicator of the target speaker ($\mathcal{S}$). Thus, the null hypothesis $\mathcal{H}_{(\mathcal{S},\mathcal{N})}$ is that the test utterance is *bonafide* speech uttered by the target speaker. In turn, the complementary hypotheses is a union of the other three classes:

$$\mathcal{H}_{\overline{(\mathcal{S},\mathcal{N})}} = \mathcal{H}_{(\overline{\mathcal{S}},\mathcal{N})} \cup \mathcal{H}_{(\mathcal{S},\overline{\mathcal{N}})} \cup \mathcal{H}_{(\overline{\mathcal{S}},\overline{\mathcal{N}})}, \tag{2}$$

where $(\overline{\mathcal{S}},\mathcal{N})$ represents *bonafide* speech uttered from a non-target speaker (*zero-effort* impostor), $(\mathcal{S},\overline{\mathcal{N}})$ corresponds to a spoofing attack, and $(\overline{\mathcal{S}},\overline{\mathcal{N}})$ represents spoofed speech from a non-target speaker. Normally, the latter case is not considered since it does not make sense in an authentication context. Table I defines the three types of trial that ASV and PAD systems may encounter: (i) *genuine target*, (ii) *genuine non-target* or *zero-effort impostor*, and (iii) *spoof target* trials. Also, Table I illustrates the ground-truth labels for each task and trial combination as well as the class names that we have defined.

The integration of ASV and PAD systems can be achieved at the score level (late fusion) [42] or at the model/feature level (early fusion) [14]. Most existing integration methods perform the integration at the score level, where dedicated classifiers are developed for ASV and PAD, and the scores computed by each independent system are combined. At this score-level integration, there are three main approaches:

1) Tandem or cascaded integration [42], [43], [44]: ASV and PAD systems can be cascaded in either order - PAD followed by ASV as shown in Fig. 1(a), or ASV followed by PAD. In order to estimate the performance of the integrated system, utterances rejected in the first module are assigned arbitrarily $-\infty$ scores and are thereby rejected automatically by the subsystem that follows. Thus, the cascaded approach

relies on three thresholds, τ_{ASV}, τ_{LA}, and τ_{PA}, applied to ASV and PAD (LA and PA) scores, respectively, as illustrated in Fig. 1.

2) Logistic regression fusion [44]: Logistic regression has been successfully employed for combining several PAD systems [45], [46] and speaker classifiers [47], [48] at the score level. The three scores s_{ASV}, s_{LA}, and s_{PA} from ASV and PAD (LA and PA) systems, respectively, can be fused inside the logistic function of a multinomial regression.

3) Gaussian back-end fusion [49]: For each ASV trial which belongs to class C_l, $l \in \{1, 2, 3\}$, a three-dimensional scores vector, $\boldsymbol{s} = [s_{ASV}, s_{LA}, s_{PA}]$, is obtained in order to model the conditional probability density of $\mathbf{s}$ using a multivariate Gaussian distribution. The scores are computed as the log-likelihood ratio between the null and complementary hypotheses, where the latter is represented as a two-component GMM with mixing weight $\alpha \in [0, 1]$, which determines the importance of classes C_2 and C_3.

On the other hand, the integration of ASV and PAD systems at the embedding level has not been fully explored by the scientific community. To the best of our knowledge, only two embedding-level integration techniques have been studied:

4) Two-stage PLDA [14]: This technique is composed of two stages. First, it trains a simplified PLDA [23] model using only the embeddings of the *bonafide* speech. Then, on the second stage, this technique estimates a new mean vector, adds a *spoofing channel* subspace, and trains it using only the embeddings of the *spoofed* speech.

5) Multi-task triplet TDNN [15]: This approach extracts embeddings that contain speaker identity and spoofing information using a multi-task time delay neural network (TDNN) [50] which is optimized using the triplet loss [51]. The dimension of these embeddings is then reduced using LDA, and the integration scores are obtained by fusing two PLDA models, one for ASV and the other one for anti-spoofing.

The evaluation of integration systems can be done in terms of EER, measured in either the licit (target speakers and *zero-effort* impostors), spoof (*bonafide* speech and spoofed speech) or joint (union of licit and spoof) scenario. However, the EER does not account for the costs of missing target users and falsely accepting zero-effort impostors or spoofing attacks, nor the prior probabilities of each. To take these costs and priors into account, the min-tDCF [41], [52] has been recently proposed as a metric for evaluating decision-level integration systems. Nevertheless, decision-level integration systems assume that there are two separate systems (ASV and PAD) with two different operating thresholds which make their own binary decisions independently. The decision-level integration system fuses their binary decision outputs in order to make the final binary decision. However, in this work we focus on score- and embedding-level integration systems which combine the scores/embeddings of ASV and PAD subsystems in order to provide one final score and handle one single threshold. Moreover, both the EER and min-tDCF metrics need that ASV and PAD operating points are set before evaluation. Thus, these metrics only measure the performance at a single operating point of the whole integration system, although the optimization of the ROCCH-EER ensures the optimization of the entire ROC due to convexity. Therefore, the ROCCH-EER can give us an idea of the overall performance of the integration system.

To allow the evaluation of integration systems across all operating points, an extension of the EPC framework was developed for evaluating integration systems, namely, the expected performance and spoofability (EPS) framework [18]. To enable this, it establishes a criteria for determining a decision threshold considering the cost of the two types of negative hypotheses as well as the cost of rejecting positives, by using two parameters: $\omega \in [0, 1]$, which denotes the relative cost of *spoofing* attacks with respect to *zero-effort* impostors; and $\beta \in [0, 1]$, which denotes the relative cost of the negative classes (*zero-effort* impostors and *spoofing* attacks) with respect to the positive class. The EPS framework plots the weighted error rate ($\text{WER}_{\omega,\beta}$) [18] with respect to one of the parameters ω or β, while the other one is fixed to a predefined value. It can be computed as [18],

$$\text{WER}_{\omega,\beta}(\tau^*_{\omega,\beta}) = \beta \cdot \text{FAR}_{\omega}(\tau^*_{\omega,\beta}) + (1 - \beta) \cdot \text{FRR}(\tau^*_{\omega,\beta}), \tag{3}$$

where FAR_{ω} is a weighted error rate for the two negative classes (ZFAR for *zero-effort* FAR and SFAR for *spoofing* FAR):

$$\text{FAR}_{\omega}(\tau) = \omega \cdot \text{SFAR}(\tau) + (1 - \omega) \cdot \text{ZFAR}(\tau), \tag{4}$$

and $\tau^*_{\omega,\beta}$ denotes the optimal classification threshold, which is chosen to minimize the weighted difference between FAR_{ω} and FRR on the development set:

$$\tau^*_{\omega,\beta} = \underset{\tau}{\text{argmin}} |\beta \cdot \text{FAR}_{\omega}(\tau) - (1 - \beta) \cdot \text{FRR}(\tau)|. \tag{5}$$

Using the WER function defined in (3), the global performance of the integrated biometric system can be computed as the area under the EPS (AUE) curve [18]. Normally, it is computed for a fixed β, which represents the average expected $\text{WER}_{\omega,\beta}$ for all values of ω:

$$\text{AUE}(\beta) = \int_0^1 \text{WER}_{\omega,\beta}(\tau^*_{\omega,\beta})\, d\omega. \tag{6}$$

This function allows the comparison between different biometric systems, with lower values indicating better performance (i.e., lower WER for the whole range of operating points). Moreover, the AUE could be also computed between certain bounds $a, b \in [0, 1]; a < b$, enabling to compare two systems depending on the required range of the varying parameter.

III. Proposed integration technique

In this section, we propose a new early-integration technique based on a DNN which processes embeddings computed by ASV and PAD systems jointly. As embeddings extracted by ASV and PAD systems share the *bonafide* subspace, the proposed system exploits this fact in order to better discriminate between *bonafide* target speech and *zero-effort* impostors or *spoofing* attacks. Moreover, we propose a new loss function

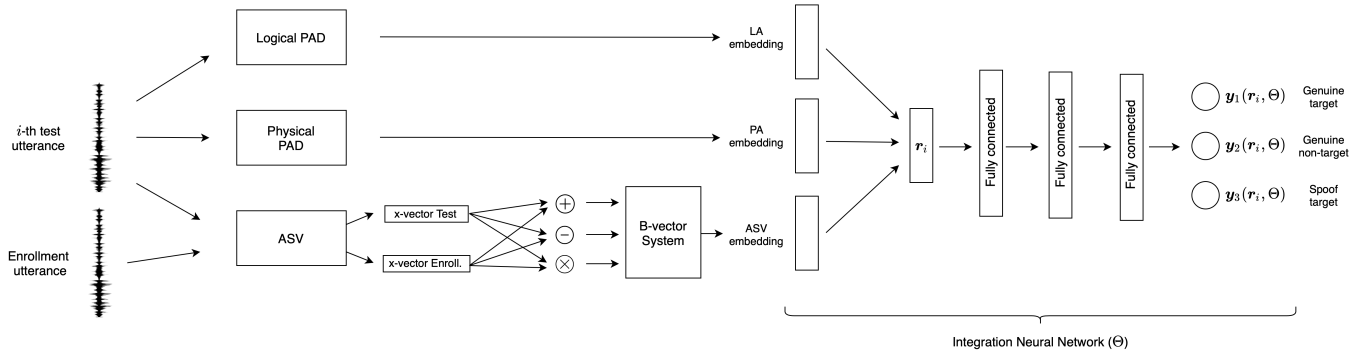

Fig. 2. Proposed integration neural network framework. System overview for classifying a pair of enrollment and test utterances into one of the three integration classes: C_1 (target genuine), C_2 (genuine non-target) and C_3 (spoof target). The LA and PA spoofing embeddings are extracted from the STFT features of the i-th test utterance, while the x-vectors (extracted from MFCC features) of the enrollment and test utterances are combined into a single ASV embedding. These three vectors are concatenated into a single input vector ($\boldsymbol{r}_i$) for the integration neural network (Θ).

to train the integration neural network which minimizes the AUE, given in Eq. (6), in order to optimize the integration system in the desired operating range in which the biometric system is expected to work.

A. Integration Neural Network

The diagram system of the proposed integration is depicted in Fig. 2, where the input for feeding the integration neural network is formed by the concatenation of three embeddings: LA, PA and ASV embeddings. The proposed approach is agnostic about the type of spoofing attack (TTS, VC or replay attacks) that it might encounter, since it is composed of two independent PAD systems for detecting LA and PA-based attacks, respectively. Thus, the LA and PA embeddings are directly extracted from the spectral features of the test utterance using these two PAD systems. In addition, the single ASV embedding combines the speaker information of both the enrollment and test utterances since it is extracted from the last fully connected layer of a b-vector system (described in Section II-A3), which contains information about whether the test and enrollment utterances are uttered by the same speaker or not. As detailed in Section IV, the ASV system is based on x-vectors [17] and processes the Mel-frequency cepstral coefficients (MFCCs) features of the enrollment and test utterances, while the PAD systems are based on a Light Convolutional Gated Recurrent Neural Network (LC-GRNN) [53] which processes the short time Fourier transform (STFT) based features of the test utterance. As can be seen in Fig. 2, the architecture of the integration neural network consists of three fully connected layers and one output layer made up of three neurons whose values represent the likelihood of the test utterance belonging to each one of the three integration classes defined in Table I: (i) C_1 (genuine target), (ii) C_2 (genuine non-target), and (iii) C_3 (spoof target).

B. Loss function

The proposed integration neural network can be trained as a multiclass classifier using the softmax function in tandem with the negative log-likelihood (NLL), which results in the classical Cross-Entropy (CE) loss function:

$$L_{CE}(\Theta) = -\log \frac{\exp(\boldsymbol{y}_l(\boldsymbol{r}, \Theta))}{\sum_{k=1}^{K} \exp(\boldsymbol{y}_k(\boldsymbol{r}, \Theta))}, \tag{7}$$

where $K = 3$ is the number of integration classes, Θ represents the parameters of the integration neural network, $\boldsymbol{r}$ is the input sample (concatenation of the three input embeddings which are fed to the integration neural network), and $\boldsymbol{y}_k(\boldsymbol{r}, \Theta)$ denotes the k-th component of the three dimensional output vector of the neural network.

However, we want to build a loss function which better fits the biometrics problem, as other works have successfully done for different speech processing tasks such as ASV [54], [55], anti-spoofing [5], and keyword spotting [56]. Specifically, we would like to optimize the parameters of the integration neural network in the desired operating range in which the biometric system is expected to work. To do so, we propose a new loss function based on the EPS framework [18] described in Section II-C, which minimizes the AUE for a specific range of operating points. In order to to minimize the AUE numerically, we compute the sum of $\text{WER}_{\omega,\beta}$ over a range of points of $\omega_j \in [0, 1]$:

$$L_{\text{AUE}}(\beta, \Theta, \tau) = \sum_{\omega_j} \Big[\beta\omega_j \cdot \hat{\text{SFAR}}(\Theta, \tau) + \beta(1-\omega_j) \cdot \hat{\text{ZFAR}}(\Theta, \tau)\Big] + (1-\beta) \cdot \hat{\text{FRR}}(\Theta, \tau), \tag{8}$$

where τ is the decision threshold for accepting or rejecting a trial $\boldsymbol{r}_i$ as genuine target, and Θ denotes the model parameters.

The integration neural network in Fig. 2 computes three scores $\boldsymbol{y}_l(\boldsymbol{r}_i, \Theta),\ l \in \{1, 2, 3\}$ in the output (softmax) layer for each input embedding $\boldsymbol{r}_i$, one for each of the three integration classes. Thus, for N pairs of enrollment and training utterances per batch, the $\hat{\text{FRR}}(\Theta, \tau)$ can be determined empirically by the average number of times that either the genuine target training utterances (that is, $\boldsymbol{r}_i \in C_1$) get positive scores ($\boldsymbol{y}_1(\boldsymbol{r}_i, \Theta)$) smaller than the decision threshold (τ), or when any of their two negative scores ($\boldsymbol{y}_2(\boldsymbol{r}_i, \Theta)$ or $\boldsymbol{y}_3(\boldsymbol{r}_i, \Theta)$) is greater than the decision threshold. The latter case is a logical OR function which can be implemented in a soft way as,

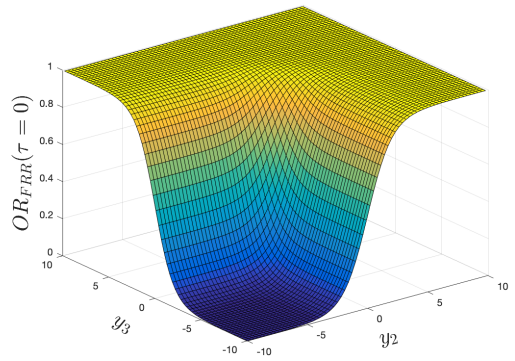

Fig. 3. Logical OR_{FRR} function, $\text{OR}_{\text{FRR}}(\tau = 0) = (\sigma(\boldsymbol{y}_2)+\sigma(\boldsymbol{y}_3))/(1+\sigma(\boldsymbol{y}_2)\sigma(\boldsymbol{y}_3))$, which is softly activated when any of the two negative scores ($\boldsymbol{y}_2$ or $\boldsymbol{y}_3$) is greater than the decision threshold of $\tau = 0$. $\boldsymbol{y}_2$ denotes the score of the genuine non-target class. $\boldsymbol{y}_3$ denotes the score of the spoof target class.

$$\text{OR}_{\text{FRR}}(\Theta, \tau, \boldsymbol{r}_i) = \frac{\sigma(\boldsymbol{y}_2(\boldsymbol{r}_i, \Theta) - \tau) + \sigma(\boldsymbol{y}_3(\boldsymbol{r}_i, \Theta) - \tau)}{1 + \sigma(\boldsymbol{y}_2(\boldsymbol{r}_i, \Theta) - \tau)\sigma(\boldsymbol{y}_3(\boldsymbol{r}_i, \Theta) - \tau)}, \tag{9}$$

where $\sigma(\cdot)$ denotes the sigmoid function which replaces the step function $u(x)$ to make the expression differentiable. Note also that the sigmoid function centered in τ represents the probability that the i-th utterance of the mini-batch with output value $\boldsymbol{y}_l(\boldsymbol{r}_i, \Theta)$ belongs to class C_l. Thus, the output range of OR_{FRR} is $[0, 1]$. Fig. 3 depicts the logical OR_{FRR} function when $\tau = 0$. Therefore, the FRR can be expressed as,

$$\hat{\text{FRR}}(\Theta, \tau) = \frac{1}{2N_1} \sum_{\boldsymbol{r}_i \in C_1} \sigma(\tau - \boldsymbol{y}_1(\boldsymbol{r}_i, \Theta)) + \text{OR}_{\text{FRR}}(\Theta, \tau, \boldsymbol{r}_i), \tag{10}$$

where $N_l,\ l \in \{1, 2, 3\}$ is the number of training utterances of the class C_l present in the current mini-batch. In the same way, the $\hat{\text{ZFAR}}(\Theta, \tau)$ and $\hat{\text{SFAR}}(\Theta, \tau)$ can be determined by the average number of times that the positive scores of *zero-effort* ($\boldsymbol{y}_2(\boldsymbol{r}_i, \Theta)$, with $\boldsymbol{r}_i \in C_2$) and *spoofing* ($\boldsymbol{y}_3(\boldsymbol{r}_i, \Theta)$, with $\boldsymbol{r}_i \in C_3$) training utterances, respectively, are smaller than the decision threshold (τ), or when their negative score ($\boldsymbol{y}_1(\boldsymbol{r}_i, \Theta)$) is greater than the decision threshold. Therefore, these error rates can be approximated as,

$$\hat{\text{ZFAR}}(\Theta, \tau) = \frac{1}{2N_2} \sum_{\boldsymbol{r}_i \in C_2} \sigma(\tau - \boldsymbol{y}_2(\boldsymbol{r}_i, \Theta)) + \sigma(\boldsymbol{y}_1(\boldsymbol{r}_i, \Theta) - \tau), \tag{11}$$

$$\hat{\text{SFAR}}(\Theta, \tau) = \frac{1}{2N_3} \sum_{\boldsymbol{r}_i \in C_3} \sigma(\tau - \boldsymbol{y}_3(\boldsymbol{r}_i, \Theta)) + \sigma(\boldsymbol{y}_1(\boldsymbol{r}_i, \Theta) - \tau). \tag{12}$$

The three error rates contain a $1/2$ factor due to the addition of two errors, in order to contribute with a 1 factor to the error rate when both are activated, i.e., when the positive and negative scores of a training sample $\boldsymbol{r}_i$ are smaller and greater than the decision threshold (τ), respectively. Moreover, it is worth noticing that τ is optimized as part of the system parameters, and that the training stage is carried out using subsets of $N = N_1 + N_2 + N_3$ samples per training batch. Unlike multi-task triplet loss [15], there is no negative mining involved, so the training process for minimizing the AUE loss (8) has a similar efficiency and convergence speed to Cross-Entropy (7)[1].

IV. Experimental setup

This section first describes the speech corpora used for the evaluation of the integration systems described in this paper. Then, Sections IV-B, IV-C and IV-D outline the details and training process of the ASV, PAD and integration systems, respectively. Finally, the performance metrics employed to evaluate the standalone and integration systems are discussed.

A. Speech Corpora

We conducted experiments on the ASVspoof 2019 database [13] which encompasses two partitions for the assessment of LA and PA scenarios. A summary of their composition in terms of speakers and number of utterances is presented in Table II. The LA database contains 17 attacks generated with state-of-the-art TTS and VC technologies, where only six of them are *known attacks* (six logical attacks for training). On the other hand, the *bonafide* and spoofed data in the PA database were generated according to a simulation of their presentation to the microphone of an ASV system within a reverberant acoustic condition. It includes a total of nine replay configurations, comprising three categories of attacker-to-speaker recording distances and three categories of loudspeaker quality, so that we considered nine types of replay attacks for training.

The ASVspoof 2019 database includes protocols for assessing the performance of anti-spoofing, ASV and integration systems. In the context of anti-spoofing, both target and non-target utterances are considered as *bonafide*. Regarding ASV systems, the development and evaluation partitions include protocols for both ASV tasks: enrollment and evaluation. In the context of integration, the PAD and ASV protocols are combined in order to evaluate integration systems. A full review of all these protocols can be found in [57].

Thus, we employed the ASVspoof 2019 database for training the standalone anti-spoofing system, as well as for training the integration systems (using the *bonafide* utterances for the *target* and *non-target* classes, and the spoofed utterances for the *spoof* class). Over 9 million utterance pairs (training/enrollment) extracted from the training sets of the ASVspoof 2019 database were employed to train the integration systems, considering a balanced representation for the three classes presented in Table I: (i) *genuine target*, (ii) *genuine non-target*, and (iii) *spoof target*.

[1]The computational times for training the integration neural network using the CE and AUE based loss functions were 18.5 and 19.2 hours, respectively, on an Ubuntu system with an i7-6850K CPU (3.60 GHz), 32 GB RAM, and a Titan X GPU of 12 GB.

TABLE II
STRUCTURE OF THE ASVSPOOF2019 DATA CORPUS DIVIDED BY THE TRAINING, DEVELOPMENT AND EVALUATION SETS [13].

Subset	#speakers		#utterances			
	Male	Female	Logical Access		Physical Access	
			Bonafide	Spoof	Bonafide	Spoof
Training	8	12	2,580	22,800	5,400	22,800
Development	4	6	2,548	22,296	5,400	24,300
Evaluation	21	27	7,355	63,882	18,090	116,640

On the other hand, we also employed the Voxceleb2 [58] database to train the TDNN x-vector model, which contains over 1 million utterances for over 6,000 speakers, extracted from videos uploaded to YouTube. Moreover, the development set of the Voxceleb1 [59] database, which includes a total of 1,231 training speakers, was combined with the *bonafide* training sets of the ASVspoof 2019 database in order to train the PLDA and b-vector ASV scoring systems. The latter dataset allows us to make an environment adaptation for the PLDA and b-vector systems. All the training details are discussed in the following.

B. Standalone ASV Systems Description

The ASV system is based on x-vectors [17] extracted from MFCC features, and we used the Voxceleb2 [58] database to train the TDNN model using the Kaldi [60] recipe [61]. To train the ASV scoring systems, we extracted the x-vectors (512 components) from the training set of the Voxceleb1 [59] database and from the *bonafide* training sets of the ASVspoof 2019 [13] database. Then, we reduced the dimension of the x-vectors from 512 to 200 components using LDA, and we fed them to the following ASV scoring systems:

1) Cosine scoring: This system does not require any training. The score was obtained as the cosine distance between the enrollment and test embeddings.

2) PLDA: We trained three different types of PLDA models: (i) standard [20], (ii) simplifed [23], and (iii) two-covariance [24]. We used the Bob toolkit [62].

3) B-vector system: The input is the concatenation of two embeddings from enrollment and test utterances. It is formed by five fully connected layers of size $[1024, 1024, 1024, 512, 128]$ with leaky ReLU activations, batch normalization and dropout of 50%, and one output linear layer composed of two neurons representing the positive and negative classes. The ASV score was obtained from the positive class of the softmax output, which corresponds to the probability of belonging the two input embeddings to the same speaker.

C. Standalone PAD Systems Description

The anti-spoofing system employed in this work is also based on embeddings extraction, and it has been one of the ten top performing single systems of the ASVspoof 2019 [13] challenge. The architecture is called LC-GRNN [53], and it is based on one of our recent works [2] (see also [53] for a detailed description of the LC-GRNN architecture). The LC-GRNN processes the STFT features from the utterance and extracts one utterance-level embedding of 64 components.

We developed two independent PAD systems, one for detecting LA-based attacks and the other for the detection of PA-based attacks. To train each of them, we used the ASVspoof 2019 [13] LA and PA training sets, respectively. Then, the embeddings of 64 components computed by the LC-GRNN network were post-processed by different scoring techniques, which obtain the PAD scores indicating the likelihood of the utterances being genuine or spoofed. We employed five state-of-the-art scoring techniques: (i) Support Vector Machine (SVM), (ii) Gaussian Mixture Model (GMM), (iii) LDA, (iv) PLDA, and (v) softmax scoring. The latter obtains the PAD score directly from the genuine class of the LC-GRNN softmax output, which corresponds to the probability of the utterance being genuine. In contrast, the other four classifiers train a specific model using the embedding vectors extracted by the LC-GRNN.

D. Integration Systems Description

We evaluated several score- and embedding-level integration systems. The score-level integration systems are: *Tandem Spoof - ASV*, *Tandem ASV - Spoof*, *Logistic regression fusion* and *Gaussian back-end fusion*. Whilst, the embedding-level integration systems are: *Two-stage PLDA*, *Multi-task triplet TDNN* and the proposed *Integration neural network*. A description of these sytems is provided below.

The score-level integration systems and the proposed integration neural network share the same standalone ASV and PADs systems (described in Sections IV-B and IV-C, respectively) in order to make a fair comparison between them. Whilst, the *Two-stage PLDA* only needs to use the x-vector based ASV system, and the *Multi-task triplet TDNN* trains a multi-task TDNN for ASV and anti-spoofing jointly, as described in Section IV-D6.

All the integration systems were trained using the scores or embeddings extracted from the Voxceleb1 database and the *bonafide* training data of the ASVspoof 2019 database.

1) Tandem Spoof - ASV [49]: This system is depicted in Fig. 1(a), where the two PAD systems precede the ASV system. This is the same scenario as the ASVspoof 2019 challenge [13]. In this system the decision to whether the test utterance is rejected is based on two PAD thresholds (τ_{LA} and τ_{PA}). We computed these thresholds using the ROCCH-EER as the reference metric evaluated on the LA and PA development sets of the ASVspoof 2019 database, respectively. Specifically, the value of these thresholds are $\tau_{LA} = 0.2948$ and $\tau_{PA} = 0.8572$. Thus, if any utterance gets a PA or LA score smaller than these thresholds, it is automatically rejected by the integration system with a score of $-\infty$, and otherwise it is assigned the ASV score.

2) Tandem ASV - Spoof [42], [43], [44]: This system is similar to the tandem Spoof - ASV, with the difference that the ASV system precedes the two PAD systems. In this system the decision to whether the test utterance is rejected is based on the ASV threshold (τ_{ASV}). We computed this threshold

using the ROCCH-EER as the reference metric evaluated on the joint *bonafide* data of the LA and PA development sets, obtaining $\tau_{ASV} = 0.6007$. Thus, if any utterance gets an ASV score smaller than this threshold, it is automatically rejected by the integration system with a score of $-\infty$, and otherwise it is assigned the smallest score between the LA and PA scores.

3) Logistic regression fusion [44]: We trained a multiclass logistic regression classifier using the three classes defined in Table I. The optimization was done using the Limited Memory Broyden-Fletcher-Gordfarb-Shanno (LM-BFGS) algorithm [63]. The optimized regression coefficients for each class are: *genuine target* ($\boldsymbol{\beta}_1 = [0.0750, -6.3119, -4.3502]$), *genuine non-target* ($\boldsymbol{\beta}_2 = [-0.0767, -3.3821, -3.9767]$), and *spoof target* ($\boldsymbol{\beta}_3 = [0.0013, 9.6941, 8.3269]$). We used the Scikit Learn toolkit [64].

4) Gaussian back-end fusion [49]: We estimated a multivariate Gaussian distribution for each one of the three integration classes. Then, we obtained the best mixing weight $\alpha = 0.58$ from development data.

5) Two-stage PLDA [14]: We replaced the i-vectors from the original work [14] by x-vectors. Thus, the first stage of the system was trained using the x-vectors from the *bonafide* data of the Voxceleb1 [59] and ASVspoof 2019 [13] databases (1,231 speakers). Then, the second stage was trained using the x-vectors from the spoofed training data of the ASVspoof 2019 database, including VC, TTS and replay attacks. We used the Bob toolkit [62].

6) Multi-task triplet TDNN [15]: A multi-task TDNN was fed with 57-dimension MFCCs and 90-dimension CQCCs, including their first and second order delta features, and was trained using the triplet loss function. Then, LDA was used to reduce the dimension of the extracted embeddings to 200. After that, two PLDA models, one for ASV and the other one for PAD, were trained using the reduced embeddings. Finally, the integration scores were obtained from the fused discrimination of the two PLDA models.

7) Proposed integration neural network: The input to the integration neural network is the concatenation of three embeddings (as depicted in Fig. 2): ASV, LA and PA embeddings. The LA and PA embeddings (64 components) are computed by the LC-GRNN network described in Section IV-C. The ASV embedding is extracted from the last fully connected layer of the b-vector system described in Section IV-B3 (128 components). Thus, the three embeddings are flattened to make up an input vector ($\boldsymbol{r}$) of 256 components.

The model of the integration neural network contains 3 fully connected layers of 256 neurons with leaky ReLU activations and batch normalization. The last layer consists of three neurons which correspond to each one of the three integration classes: (i) *genuine target*, (ii) *genuine non-target*, and (iii) *spoof target*. It was trained using the Adam optimizer [65] with a learning rate of $3 \cdot 10^{-4}$ and a batch size of 50,000 pairs of enrollment and training embeddings. Also, early stopping was applied to stop the training process when no improvement of the loss across the validation set was obtained. To prevent the problem of over-fitting, a fixed 50% dropout was applied in the fully connected layers. The Pytorch toolkit [66] was employed to implement the deep learning framework.

E. Performance Metrics

The standalone ASV systems were evaluated in terms of pooled EER, AUE [35], C_{llr}^{min} [34], as well as NIST 2008 (*DCF08* [30]) and NIST 2010 (*DCF10* [31]) minimum detection costs. Likewise, the evaluation of the standalone anti-spoofing systems was done in terms of pooled EER. Once we had the ASV and PAD scores, we also evaluated the ASV-constrained min-tDCF [41] for both the LA and PA scenarios, separately. These metrics (EER and min-tDCF) have been evaluated using the optimal threshold for each metric, as the ASVspoof 2019 challenge did for evaluating every anti-spoofing system.

To evaluate the robustness of the integration systems against attacks, we computed the ZFAR (*zero-effort* FAR) and SFAR (*spoofing* FAR) at the threshold when the FRR of the system equals 1%, as done in the previous work on two-stage PLDA approach [14]. Furthermore, we evaluated the estimated EER using the ROCCH (when FRR is equal to FAR), the area under the EPS (AUE) curve [18] and the DET [32] curves. The main objective of the integration system is to reduce the AUE as much as possible in the range of operating points in which is expected to work. We defined three working operating points, setting $\beta = \{0.2, 0.5, 0.8\}$ in order to make more emphasis on either FRR or FAR.

V. Standalone Systems Results

This section presents the experimental results from the evaluation of the standalone systems on the ASVspoof 2019 corpus. First, Section V-A evaluates the different ASV standalone systems on the licit scenario (using only *zero-effort* attacks, i.e., genuine speech from non-target users) of the LA and PA development sets. Then, Section V-B is devoted to the evaluation of the PAD systems. We employed the development sets to choose the best standalone systems, in order to use them in the integration systems evaluated in Section VI.

A. Standalone ASV results

In order to choose the best-performing ASV system for being used later in the integration systems, this section compares the performance of the ASV scoring techniques described in Section II-A, namely: cosine scoring, b-vector system, and three versions of PLDA (standard, simplified and two-covariance). As mentioned above, the experiments are conducted using the *zero-effort* impostor data from the development set of ASVspoof 2019.

Table III presents the EER, C_{llr}^{min}, DCF08, DCF10 and AUE metrics achieved by the standalone ASV systems evaluated on the licit scenario. It can be seen that the standard version of the PLDA yields the best performance in terms of AUE, C_{llr}^{min} and EER. In general, the PLDA classifier outperforms the cosine scoring and b-vector systems irrespective of the PLDA version on both the LA and PA scenarios.

Fig. 4 shows the curves obtained for the WER_β metric defined in (3) as a function of β (relative cost of the negative classes, i.e., *zero-effort* impostors and *spoofing* attacks, with respect to the genuine target class) for the three types of ASV scoring techniques. The parameter ω, which controls the

TABLE III
RESULTS OF THE X-VECTOR BASED ASV SYSTEM WITH DIFFERENT SCORING TECHNIQUES ON ASVSPOOF 2019 LOGICAL AND PHYSICAL ACCESS DEVELOPMENT LICIT SCENARIOS IN TERMS OF EER (%), AUE, C_{llr}^{min}, NIST DCF08 AND NIST DCF10.

System	Logical Access Development Set					Physical Access Development Set				
	DCF08	DCF10	C_{llr}^{min}	EER (%)	AUE	DCF08	DCF10	C_{llr}^{min}	EER (%)	AUE
Cosine	4.5430	0.0749	0.3425	10.25	0.1488	7.1068	0.0940	0.5353	16.48	0.2368
b-vector	2.3361	**0.0396**	0.1917	5.95	0.0819	4.1876	0.0677	0.2986	8.96	0.1311
Standard PLDA	**1.4680**	0.0422	**0.1192**	**3.44**	**0.0504**	**2.5543**	0.0491	**0.2035**	**6.33**	**0.0926**
Simplified PLDA	**1.4680**	0.0426	0.1199	**3.44**	0.0506	2.5742	0.0496	0.2062	6.41	0.0937
Two-cov PLDA	1.6163	0.0401	0.1266	3.54	0.0531	3.0286	**0.0464**	0.2485	7.64	0.1123

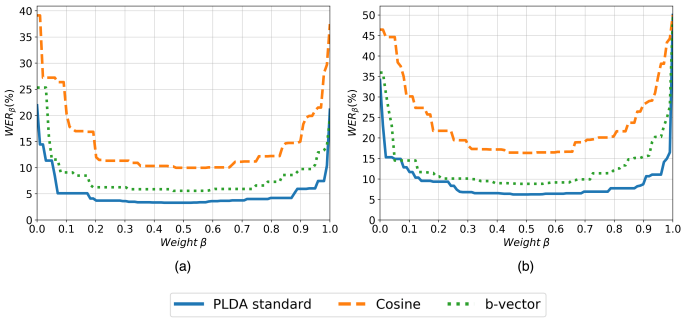

Fig. 4. WER_β for the following ASV scoring systems: cosine, b-vector and PLDA standard. The EPC is evaluated on the development licit scenarios ($\omega = 0$) of the ASVspoof 2019 sets. (a) Logical Access. (b) Physical Access.

relative cost of the error rate related to *spoofing* attacks, is set to 0 since we are evaluating them on the licit scenario. It can be observed that the PLDA outperforms the cosine scoring and b-vector systems for almost the whole range of β on both the LA and PA datasets. However, being an essential component of our proposed integration network (described in Section III), the b-vector system approaches the performance of the PLDA technique, and has the advantage that it can be easily integrated in a DNN to compute embeddings. Based on these development results, in the rest of the evaluation we will use the standard PLDA as the standalone ASV scoring system for the score-level integration systems.

B. Standalone anti-spoofing results

In this section, we evaluate the LC-GRNN-based anti-spoofing systems with different back-end classifiers. The objective is to compare their performance in order to choose the best PAD scores for the score-level integration systems.

We first evaluated the anti-spoofing systems with different back-end classifiers (SVM, GMM, LDA, PLDA and softmax scoring technique) on the development sets of the ASVspoof 2019 database in terms of EER. Since the types of attacks in the development set are seen during training, all the techniques yielded an EER close to or equal to 0.0%. The best scoring technique was the softmax scoring, achieving an EER of 0.0% in both the LA and PA datasets. Thus, in the rest of the evaluation we will use the softmax scores of the standalone anti-spoofing system for the score-level integration systems.

For the sake of completeness, we also evaluated them on the evaluation set which also contains *unknown spoofing* attacks. Table IV reports the EER of the PAD scoring systems

TABLE IV
RESULTS OF THE STANDALONE LC-GRNN BASED ANTI-SPOOFING SYSTEM WITH DIFFERENT BACK-END CLASSIFIERS ON ASVSPOOF 2019 LOGICAL AND PHYSICAL ACCESS EVALUATION SETS IN TERMS OF POOLED EER (%) AND MIN-TDCF.

Classifier	Logical Access Test Set		Physical Access Test Set	
	EER (%)	min-tDCF	EER (%)	min-tDCF
SVM	7.12	0.1763	3.07	0.0817
GMM	7.55	0.1912	4.09	0.1264
LDA	6.28	0.1372	3.49	0.0865
PLDA	6.34	0.1403	2.23	0.0578
Softmax	**6.21**	**0.1355**	**2.10**	**0.0553**

evaluated on the LA and PA evaluation sets. The softmax scoring technique also outperforms the rest of classifiers in terms of EER, although the PLDA classifier achieves a similar performance. The SVM, GMM and LDA classifiers achieve higher EERs than PLDA and softmax scoring. Table IV also shows the min-tDCF metric obtained when joining the best standalone ASV scores (standard PLDA) evaluated in Section V-A with the PAD scores of the different back-end classifiers. It can be seen that the softmax scoring technique also outperforms the rest of classifiers (SVM, GMM, LDA and PLDA) in terms of min-tDCF. According to the results of the ASVspoof 2019 Challenge [13], the performance of this single system is comparable to the best fusion/ensemble systems and it is among the best single systems on both the LA and PA scenarios reflected in [13].

VI. INTEGRATION SYSTEMS RESULTS

In this section, we evaluate our proposed integration system and compare it with other state-of-the-art score- and embedding-level integration systems at different operating points which put more emphasis on either FAR or FRR. The integration protocols employed to evaluate them are defined in the ASVspoof 2019 database [57].

A. Comparison of agnostic integration systems

Table V reports the EER, ZFAR and SFAR values obtained on the LA and PA evaluation sets of the ASVspoof 2019 database for different types of agnostic integration systems, i.e., systems which are able to handle both LA- and PA-based attacks. The EERs are evaluated in three scenarios: (i) licit scenario (considering only *zero-effort* impostor attacks), (ii) spoof scenario (considering only *spoofing* attacks), and (iii) joint scenario (considering both *zero-effort* impostor and *spoofing* attacks). For the sake of comparison with ASV

TABLE V
RESULTS ON ASVSPOOF 2019 LOGICAL AND PHYSICAL ACCESS EVALUATION SCENARIOS IN TERMS OF EER (%), ZFAR (%) AND SFAR (%).

System	Logical Access Test Set					Physical Access Test Set				
	Licit EER (%)	Spoof EER (%)	Joint EER (%)	ZFAR (%)	SFAR (%)	Licit EER (%)	Spoof EER (%)	Joint EER (%)	ZFAR (%)	SFAR (%)
ASV: b-vector System	2.93	41.73	31.36	6.75	79.86	6.61	41.79	27.69	25.51	97.22
ASV: Standard PLDA	2.16	38.49	29.29	4.34	77.10	**5.02**	38.62	25.43	**20.10**	95.36
Tandem ASV-Spoof	2.32	12.29	10.52	100.00	70.90	5.66	5.74	6.78	100.00	27.36
Tandem Spoof-ASV	3.76	8.51	7.67	99.24	78.83	15.49	8.56	14.93	100.00	96.51
Logistic Regression	3.42	14.82	11.46	11.79	40.22	12.40	7.04	10.53	82.59	35.66
Gaussian Fusion	3.39	15.21	11.68	7.53	37.10	9.74	4.71	8.21	64.24	42.31
Two-stage PLDA	**2.05**	36.91	28.40	**3.91**	75.85	5.29	38.36	25.43	22.87	95.42
Multi-task Triplet TDNN	3.55	8.66	7.92	8.99	22.55	7.66	3.45	6.50	54.13	22.13
Integration Network (CE)	3.18	8.75	7.56	8.52	21.43	7.35	3.56	6.42	50.18	19.51
Integration Network (AUE)	3.01	**7.82**	**6.05**	7.53	**18.10**	6.98	**3.08**	**5.21**	31.29	**14.24**

TABLE VI
RESULTS ON ASVSPOOF 2019 LOGICAL AND PHYSICAL ACCESS EVALUATION SCENARIOS IN TERMS OF AUE FOR DIFFERENT β OPERATING POINTS.

System	Logical Access Test Set			Physical Access Test Set		
	AUE ($\beta = 0.5$)	AUE ($\beta = 0.8$)	AUE ($\beta = 0.2$)	AUE ($\beta = 0.5$)	AUE ($\beta = 0.8$)	AUE ($\beta = 0.2$)
ASV: b-vector System	0.2130	0.1790	0.0964	0.2549	0.1767	0.1281
ASV: Standard PLDA	0.1966	0.1768	0.0930	0.2312	0.1733	0.1214
Tandem ASV-Spoof	0.1243	0.1805	0.0663	0.0570	0.0511	0.0550
Tandem Spoof-ASV	0.0787	0.0816	0.0547	0.1061	0.0588	0.1408
Logistic Regression	0.0917	0.0894	0.0569	0.0977	0.0599	0.0912
Gaussian Fusion	0.0945	0.1023	0.0771	0.0763	0.0565	0.0792
Two-stage PLDA	0.1920	0.1771	0.0929	0.2332	0.1730	0.1240
Multi-task Triplet TDNN	0.0754	0.0857	0.0442	0.0566	0.0473	0.0654
Integration Neural Network (CE)	0.0753	0.0757	0.0412	0.0656	0.0493	0.0584
Integration Neural Network (AUE)	**0.0571**	**0.0558**	**0.0361**	**0.0422**	**0.0365**	**0.0359**

systems, the first two systems correspond to the b-vector and standard PLDA standalone ASV systems, which achieve the best performance in terms of ZFAR along with the two-stage PLDA integration system. This is not surprising since the ASV systems are trained to detect only *zero-effort* impostor trials, and the two-stage PLDA integration system includes a similar ASV system in its first stage. However, these three techniques are the worst in terms of spoof and joint EERs, as they are not able to detect *spoofing* attacks effectively. In fact, they are fed with x-vectors which only contain speaker information, but not *spoofing* information. In this way, the joint EER of the standard PLDA ASV system drastically degrades when considering *spoofing* attacks from 2.16 and 5.02% to 29.29 and 25.43% in the LA and PA scenarios, respectively.

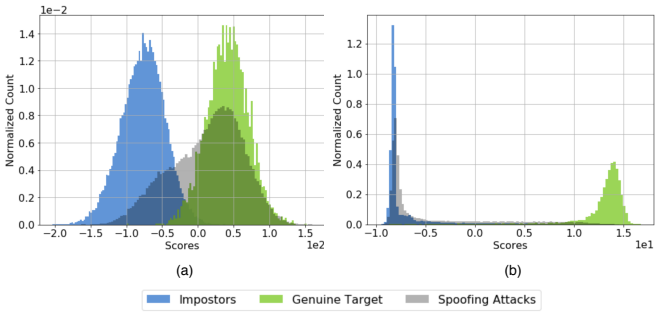

Fig. 5. Scores distribution of genuine target accesses, genuine non-target or impostors accesses, and *spoofing* attacks, evaluated in the logical access dataset. (a) ASV system (PLDA standard). (b) Integration Neural Network (AUE).

The proposed integration neural network achieves the best joint EER and SFAR in the LA and PA scenarios, irrespective of the loss function employed for optimizing it (CE or AUE). These results show the effectiveness of our proposal, outperforming other classical score-level integration techniques, such as logistic regression, Gaussian fusion and cascaded or tandem systems. Moreover, our proposal also outperforms the other two embedding-based integration systems: (i) two-stage PLDA, and (ii) multi-task triplet TDNN. Only the two-stage PLDA system outperforms the proposed integration neural network in terms of ZFAR and licit EER. This is due to the fact that the two-stage PLDA system is only able to detect *zero-effort* impostors effectively, with a similar behaviour to the standalone ASV systems. In general, all the integration systems with the exception of two-stage PLDA suffer from a performance degradation of the licit EER with respect to their corresponding standalone ASV systems. This could be expected since the integration systems normally have a trade-off between detecting *zero-effort* impostor attacks and *spoofing* attacks. On the other hand, the proposed loss function, which minimizes the AUE, outperforms the classical cross-entropy (CE), achieving an absolute reduction of 1.51% and 1.21% joint EER in the LA and PA scenarios, respectively.

Fig. 5 shows the score distribution of the proposed integration system and the ASV system with PLDA scoring,

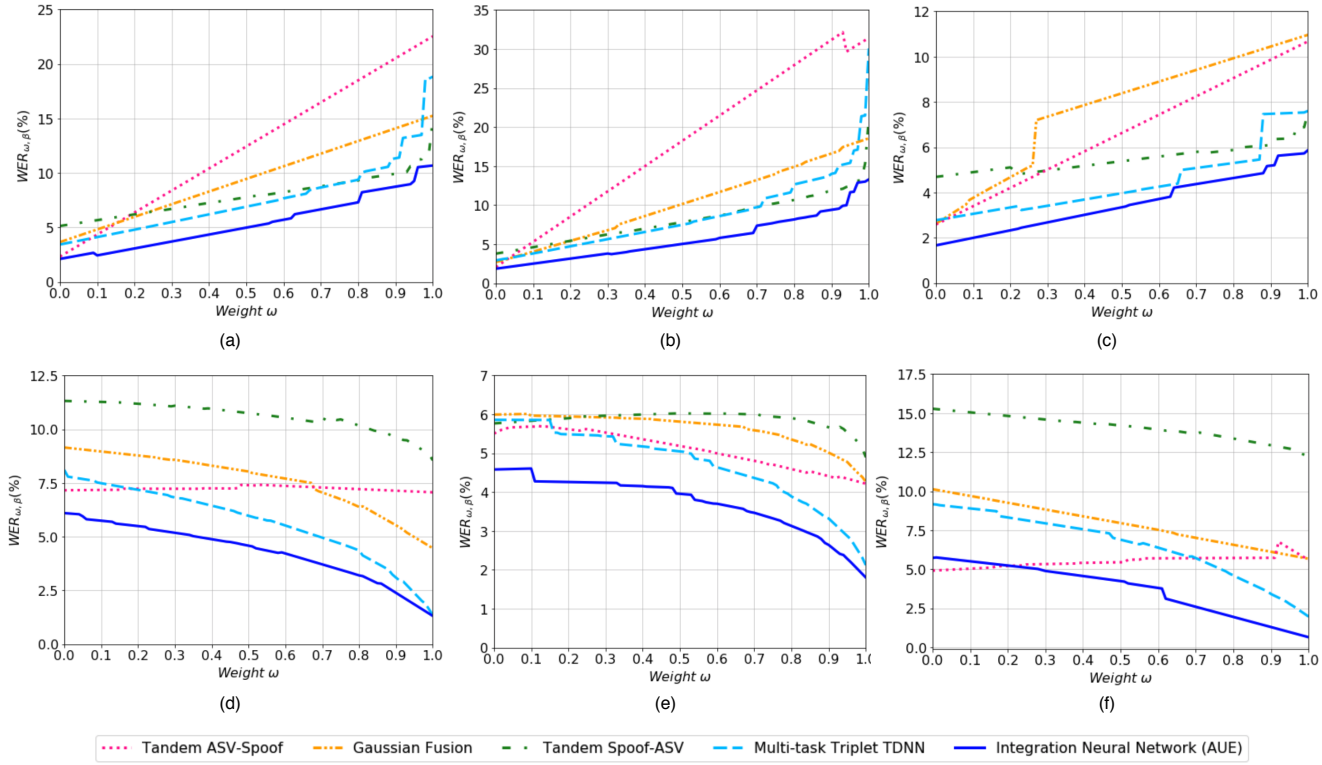

Fig. 6. Expected Performance and Spoofability Curves (EPSC) of different ASV and integration systems evaluated at different operating points and datasets. (a) Logical Access ($\beta = 0.5$). (b) Logical Access ($\beta = 0.8$). (c) Logical Access ($\beta = 0.2$). (d) Physical Access ($\beta = 0.5$). (e) Physical Access ($\beta = 0.8$). (f) Physical Access ($\beta = 0.2$).

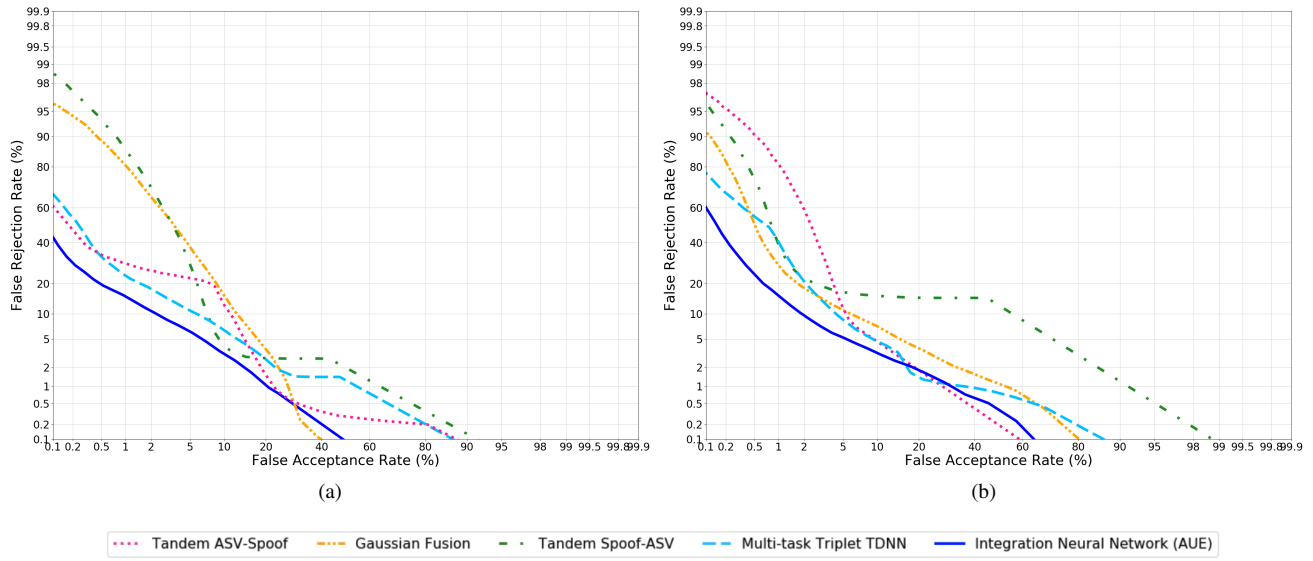

Fig. 7. Detection Error Tradeoff (DET) curves of different integration systems evaluated in the evaluation datasets of the ASVspoof 2019 database: (a) Logical Access; and (b) Physical Access.

evaluated in the LA test dataset. The scores are divided into three classes: (i) genuine target, (ii) genuine non-target or *zero-effort* impostors, and (iii) *spoofing* attacks. As can be seen, the ASV system (Fig. 5a) is only able to differentiate between genuine target and *zero-effort* impostor accesses, while the integration system is also able to effectively detect *spoofing* attacks (Fig. 5b).

Table VI shows the AUE of the same systems evaluated in the LA and PA scenarios at different operating points. Fig. 6 and 7 depict the EPS and DET curves, respectively, of the proposed integration neural network which minimizes the AUE and the other four better integration techniques (Gaussian fusion, ASV/Spoof tandems and multi-task triplet TDNN). If β is set close to 1, the biometric system gives

more importance to detecting false alarms than false rejections, and the contrary occurs when β is set close to 0. As can be seen, the standalone ASV systems and the two-stage PLDA integration system are the ones which obtain the worst WER in all scenarios, since (as mentioned before) they are not able to detect *spoofing* attacks. The performance of the tandem Spoof-ASV system is very remarkable in the LA evaluation, although it is degraded considerably in the PA evaluation. On the contrary, the tandem ASV-Spoof achieves small AUEs in the PA evaluation, but they are increased considerably in the LA evaluation. These differences of performance can be due to the difficult calibration of these systems for choosing the ASV and *spoofing* thresholds, so that they may be better adapted for detecting LA attacks than PA attacks, and viceversa.

On the other hand, the logistic regression and Gaussian fusion integration techniques have a similar performance at the different operating points, so that logistic regression slightly outperforms Gaussian fusion in the LA evaluation, and the contrary occurs in the PA evaluation. Similarly to the results reported in Table V, we can see in Fig. 6 that the proposed integration neural network achieves the smallest WER in almost the whole range of ω at the three β operating points considered in the evaluation of the LA and PA scenarios, and therefore obtains the best AUE in all scenarios. There is only one case in which the tandem ASV-Spoof outperforms our proposal in the PA scenario for $\beta = 0.2$ and low values of ω. This could be attributed to the fact that in this range the WER gives much more importance to *zero-effort* accesses than to *spoofing* attacks, and the tandem ASV-Spoof contains a PLDA scoring based ASV system in its first stage which obtains a higher performance than b-vector system, as previously shown in Table III. Similarly, we can see in Fig. 7, which shows the DET curves for different integration systems, that the proposed integration neural network outperforms the other integration techniques in almost the whole range of operating points. Moreover, we can see in Table VI that the AUE loss function (8) outperforms the classical cross-entropy (7) in all scenarios, demonstrating the effectiveness of the proposed loss function for integration systems.

B. Comparison between agnostic and fusion of non-agnostic integration systems

In order to evaluate the agnosticism to the type of *spoofing* attacks (LA or PA-based attacks) that we considered in all the integration systems evaluated in the previous section, we compare the performance of the agnostic integration systems, which are able to detect both types of *spoofing* attacks, with the performance of the fusion of two similar non-agnostic integration systems, where each one can only detect either LA or PA-based attacks. In the latter case, the two non-agnostic integration systems share the same ASV system, but they only contain one module of anti-spoofing trained for detecting either LA or PA-based attacks. For the sake of simplicity, the fusion of these two non-agnostic integration systems is based on a logistic regression. Fig. 8 and 9 show the joint EERs of these systems for the LA and PA evaluation scenarios, respectively. As can be seen, all the agnostic integration

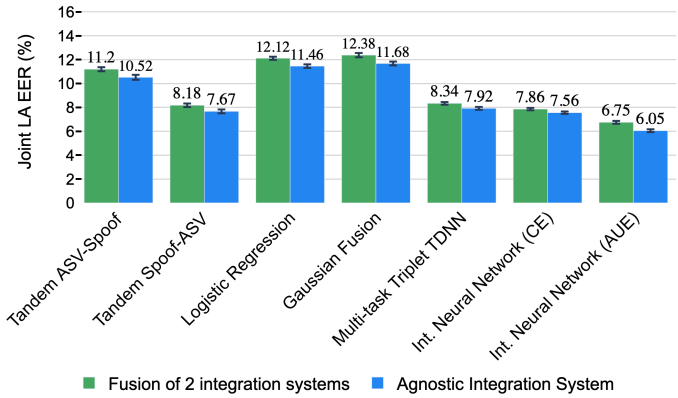

Fig. 8. Averaged joint EERs (%) evaluated in the LA test scenario of the agnostic and fusion of 2 integration systems. Mean intervals are presented at 95% of confidence.

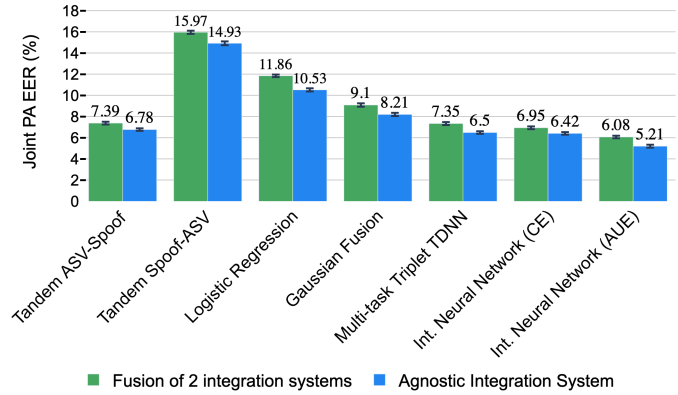

Fig. 9. Averaged joint EERs (%) evaluated in the PA test scenario of the agnostic and fusion of 2 integration systems. Mean intervals are presented at 95% of confidence.

systems outperform the fusion of the two non-agnostic integration systems in both the LA and PA evaluation scenarios. This can be due to achieving a better generalization when training the agnostic integration system with both LA and PA embeddings. Although the difference in terms of joint EER between the two types of integration systems is under 1.33% in all cases, the agnostic integration system always obtains a better performance. Moreover, the proposed integration neural network trained with the AUE loss leads to the best integration system in both cases (agnostic and fusion of non-agnostic integration systems). These results reveal the suitability of the agnostic approach for real scenarios, where the biometric system does not know about the type of *spoofing* attack that it might encounter.

VII. Conclusion

In this paper, we proposed a new integration neural network which jointly processes the embeddings extracted by ASV and anti-spoofing systems in order to detect whether the test utterance is *bonafide* and belongs to the claimed speaker. Furthermore, a new loss function which minimizes the area under the expected (AUE) performance and spoofability curve (EPSC) was proposed to optimize the integration neural network on the

operating range in which the biometric system is expected to work. The proposed approach and the other techniques were trained and evaluated using the LA and PA datasets of the ASVspoof 2019 corpus. Experimental results have shown that the joint processing of the ASV and PAD embeddings with the proposed integration neural network clearly outperforms other state-of-the-art integration techniques, trained on the same conditions. Specifically, our proposal achieves up to 23.62% and 22.03% relative equal error rate (EER) improvement over the best performing baseline (multitask triplet TDNN [15]) in the LA and PA scenarios, respectively, as well as relative gains of 27.62% and 29.15% on the AUE metric. Moreover, the proposed loss function also achieves up to 22.19% and 20.81% relative joint EER improvement over the classical cross-entropy (CE) loss in both the LA and PA evaluation scenarios, respectively.

To the best of our knowledge, most of the existing integration systems from the literature have only been trained and evaluated to detect either LA- or PA-based attacks. In this work, we also adapted and evaluated them for detecting TTS, VC and replay attacks, so that they are agnostic to the type of *spoofing* attack which they might encounter. In addition, we concluded that training a unique integration system for detecting LA- and PA-based attacks (agnostic integration system) is better than fusing two similar non-agnostic integration systems, where each one can only detect either LA- or PA-based attacks.

The proposed approach validated the feasibility of the joint processing of ASV and anti-spoofing embeddings with an integration neural network. One of the limitations of this work is that we only used one database of spoofing attacks for evaluating the integration systems. As future work, we will explore a cross-database evaluation of the integration systems in order to study their generalization between different datasets [67]. We also envision that the proposed integration neural network and loss function can be effectively used in other biometrics applications, taking into account that its hyper-parameters should be adapted according to the new biometrics system.

ACKNOWLEDGMENTS

This work has been partially supported by the Spanish MINECO/FEDER Project PID2019-104206GB-I00 and the HASLER Foundation (https://haslerstiftung.ch/) project FLOSS. Alejandro Gomez-Alanis holds a FPU fellowship from the Spanish Ministry of Education (FPU16/05490). Jose A. Gonzalez-Lopez holds a Juan de la Cierva-Incorporación fellowship from the Spanish Ministry of Science, Innovation and Universities (IJCI-2017-32926). We also acknowledge the support of NVIDIA Corporation with the donation of a Titan X GPU.

REFERENCES

[1] A. K. Jain, A. Ross, and S. Pankanti, "Biometrics: A tool for information security," *IEEE Transactions on Information Forensics and Security*, vol. 1, no. 2, pp. 125–143, 2006.

[2] A. Gomez-Alanis, A. M. Peinado, J. A. Gonzalez, and A. M. Gomez, "A gated recurrent convolutional neural network for robust spoofing detection," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 12, pp. 1985–1999, 2019.

[3] ——, "A deep identity representation for noise robust spoofing detection," in *Proc. Interspeech*, Hyderabad, India, 2018, pp. 676–680.

[4] Z. Wu, N. W. D. 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, 2014.

[5] A. Gomez-Alanis, J. A. Gonzalez-Lopez, and A. M. Peinado, "A kernel density estimation based loss function and its application to ASV-spoofing detection," *IEEE Access*, vol. 8, pp. 108 530–108 543, 2020.

[6] R. Naika, "An overview of automatic speaker verification system," *Advances in Intelligent Systems and Computing*, vol. 673, 2018.

[7] Z. Wu, P. L. D. Leon, C. Demiroglu, A. Khodabakhsh, S. King, Z.-H. Ling, D. Saito, B. Stewart, T. Toda, M. Wester, and J. Yamagishi, "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, pp. 768–783, 2016.

[8] A. K. Jain, S. Prabhakar, and S. Pankanti, "On the similarity of identical twin fingerprints," *Pattern Recognition*, vol. 35, no. 11, pp. 2653–2663, 2002.

[9] "Presentation attack detection." [Online]. Available: https://www.iso.org/standard/67381.html

[10] Z. Wu, T. Kinnunen, N. W. D. 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, 2015, pp. 2037–2041.

[11] P. Korshunov, S. Marcel, and H. M. et al., "Overview of BTAS 2016 speaker anti-spoofing competition," in *Proc. IEEE International Conference on Biometrics Theory, Applications and Systems (BTAS)*, Niagara Falls, NY, USA, 2016, pp. 1–6.

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

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

[14] A. Sizov, E. el Khoury, T. Kinnunen, Z. Wu, and S. Marcel, "Joint speaker verification and antispoofing in the i-vector space," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 4, pp. 821–832, 2015.

[15] J. Li, M. Sun, X. Zhang, and Y. Wang, "Joint decision of anti-spoofing and automatic speaker verification by multi-task learning with contrastive loss," *IEEE Access*, vol. 8, pp. 7907–7915, 2020.

[16] J. H. L. Hansen and T. Hasan, "Speaker recognition by machines and humans: A tutorial review," *IEEE Signal Processing Magazine*, vol. 32, no. 6, pp. 74–99, 2015.

[17] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust DNN embeddings for speaker recognition," Calgary, Alberta, Canada, 2018, pp. 5329–5333.

[18] I. Chingovska, A. Anjos, and S. Marcel, "Biometrics evaluation under spoofing attacks," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 12, pp. 2264–2276, 2014.

[19] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.

[20] S. Prince and J. H. Elder, "Probabilistic linear discriminant analysis for inferences about identity," in *Proc. IEEE International Conference on Computer Vision*, Rio de Janeiro, Brazil, 2007, pp. 1–8.

[21] S. Ioffe, "Probabilistic linear discriminant analysis," in *Proc. European Conference on Computer Vision (ECCV)*, Graz, Austria, 2006, pp. 531–542.

[22] A. Sizov, K.-A. Lee, and T. Kinnunen, "Unifying probabilistic linear discriminant analysis variants in biometric authentication," in *Proc. Structural and Syntactic Pattern Recognition*, Berlin, Heidelberg, 2014, pp. 464–475.

[23] P. Kenny, "Bayesian speaker verification with heavy-tailed priors," in *Proc. Odyssey*, Brno, Czech Republic, 2010.

[24] N. Brümmer and E. de Villiers, "The speaker partitioning problem," in *Proc. Odyssey*, Brno, Czech Republic, 2010.

[25] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society, Series B*, vol. 39, no. 1, pp. 1–38, 1977.

[26] H.-S. Lee, Y. Tso, Y.-F. Chang, H.-M. Wang, and S.-K. Jeng, "Speaker verification using kernel-based binary classifiers with binary operation derived features," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Florence, Italy, 2014, pp. 1660–1664.

[27] I. Chingovska, A. Anjos, and S. Marcel, "Anti-spoofing in action: Joint operation with a verification system," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, NW Washington, DC, USA, 2013, pp. 98–104.

[28] D. A. van Leeuwen and N. Brümmer, "An introduction to application-independent evaluation of speaker recognition systems," in *Speaker Classification I: Fundamentals and Methods*, Berlin, Germany, 2007, pp. 330–353.

[29] G. R. Doddington, M. A. Przybocki, A. F. Martin, and D. A. Reynolds, "The NIST speaker recognition evaluation - overview, methodology, systems, results, perspective," *Speech Communication*, vol. 31, pp. 225–254, 2000.

[30] A. F. Martin and C. S. Greenberg, "NIST 2008 speaker recognition evaluation: Performance across telephone and room microphone channels," in *Proc. Interspeech*, Brighton, United Kingdom, 2009, pp. 2579–2582.

[31] C. S. Greenberg, A. F. Martin, B. Barr, and G. R. Doddington, "Report on performance results in the NIST 2010 speaker recognition evaluation," in *Proc. Interspeech*, Florence, Italy, 2011, pp. 261–164.

[32] A. F. Martin, G. R. Doddington, T. Kamm, M. Ordowski, and M. A. Przybocki, "The DET curve in assessment of detection task performance," in *Proc. Eurospeech*, Rhodes, Greece, 1997.

[33] T. Fawcett and A. Niculescu-Mizil, "Pav and the roc convex hull," *Machine Learning*, vol. 68, pp. 97–106, 2007.

[34] N. Brümmer and J. Du Preez, "Application-independent evaluation of speaker detection," *Computer Speech & Language*, vol. 20, no. 2-3, pp. 230–275, 2006.

[35] S. Bengio, J. Mariéthoz, and M. Keller, "The expected performance curve," in *Proc. International Conference on Machine Learning (ICML)*, Bonn, Germany, 2005.

[36] 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," in *Proc. Iberspeech*, Barcelona, Spain, 2018, pp. 45–49.

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

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

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

[40] H. Muckenhirn, M. Magimai-Doss, and S. Marcel, "End-to-end convolutional neural network-based voice presentation attack detection," Denver, Colorado, USA, 2017, pp. 335–341.

[41] T. Kinnunen, H. Delgado, N. Evans, K. A. Lee, V. Vestman, A. Nautsch, M. Todisco, X. Wang, M. Sahidullah, J. Yamagishi, and D. A. Reynolds, "Tandem assessment of spoofing countermeasures and automatic speaker verification: Fundamentals," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 2195–2210, 2020.

[42] P. L. D. Leon, M. Pucher, J. Yamagishi, I. Hernáez, and I. Saratxaga, "Evaluation of speaker verification security and detection of HMM-based synthetic speech," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 8, pp. 2280–2290, 2012.

[43] F. Alegre, A. Amehraye, and N. Evans, "Spoofing countermeasures to protect automatic speaker verification from voice conversion," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Vancouver, Canada, 2013, pp. 3068–3072.

[44] M. Sahidullah, H. Delgado, M. Todisco, H. Yu, T. Kinnunen, N. W. D. Evans, and Z.-H. Tan, "Integrated spoofing countermeasures and automatic speaker verification: An evaluation on ASVspoof 2015," in *Proc. Interspeech*, San Francisco, CA, USA, 2016, pp. 1700–1704.

[45] 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, p. 1942–1955, 2017.

[46] R. Font, J. M. Espín, and M. J. Cano, "Experimental analysis of features for replay attack detection - results on the ASVspoof 2017 challenge," in *Proc. Interspeech*, Stockholm, Sweden, 2017, pp. 7–11.

[47] S. Pigeon, P. Druyts, and P. Verlinde, "Applying logistic regression to the fusion of the NIST'99 1-speaker submissions," *Digital Signal Processing*, vol. 10, pp. 237–248, 2000.

[48] N. Brummer, L. Burget, J. Cernocky, O. Glembek, F. Grezl, M. Karafiat, D. A. van Leeuwen, P. Matejka, P. Schwarz, and A. Strasheim, "Fusion of heterogeneous speaker recognition systems in the STBU submission for the NIST speaker recognition evaluation 2006," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 7, pp. 2072–2084, 2007.

[49] M. Todisco, H. Delgado, K.-A. Lee, M. Sahidullah, N. W. D. Evans, T. Kinnunen, and J. Yamagishi, "Integrated presentation attack detection and automatic speaker verification: Common features and Gaussian back-end fusion," in *Proc. Interspeech*, Hyderabad, India, 2018, pp. 77–81.

[50] A. H. Waibel, T. Hanazawa, G. E. Hinton, K. Shikano, and K. J. Lang, "Phoneme recognition using time-delay neural networks," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, pp. 328–339, 1989.

[51] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proc. IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, Massachusetts, USA, 2015, pp. 815–823.

[52] T. Kinnunen, K. Lee, H. Delgado, N. Evans, M. Todisco, M. Sahidullah, J. Yamagishi, and D. Reynolds, "t-DCF: A detection cost function for the tandem assessment of spoofing countermeasures and automatic speaker verification," in *Proc. Odyssey*, Les Sables d'Olonne, France, 2018, pp. 312–319.

[53] A. Gomez-Alanis, A. M. Peinado, J. A. Gonzalez, and A. M. Gomez, "A light convolutional GRU-RNN deep feature extractor for ASV spoofing detection," in *Proc. Interspeech*, Graz, Austria, 2019, pp. 1068–1072.

[54] V. Mingote, A. Miguel, A. Ortega, and E. Lleida, "Optimization of the area under the ROC curve using neural network supervectors for text-dependent speaker verification," *Computer Speech and Language*, vol. 63, p. 101078, 2020.

[55] Z. Bai, X. Zhang, and J. Chen, "Speaker verification by partial auc optimization with mahalanobis distance metric learning," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 1533–1548, 2020.

[56] S. Panchapagesan, M. Sun, A. Khare, S. Matsoukas, A. Mandal, B. Hoffmeister, and S. Vitaladevuni, "Multi-task learning and weighted cross-entropy for DNN-based keyword spotting," in *Proc. Interspeech*, San Francisco, CA, USA, 2016, pp. 760–764.

[57] X. Wang, J. Yamagishi, M. Todisco, H. Delgado, A. Nautsch, and N. E. et al., "ASVspoof 2019: a large-scale public database of synthetized, converted and replayed speech," *Computer Speech and Language*, p. 101114, 2020.

[58] J. S. Chung, A. Nagrani, and A. Zisserman, "Voxceleb2: Deep speaker recognition," in *Proc. Interspeech*, Hyderabad, India, 2018, pp. 1086–1090.

[59] A. Nagrani, J. S. Chung, and A. Zisserman, "Voxceleb: A large-scale speaker identification dataset," in *Proc. Interspeech*, Stockholm, Sweden, 2017, pp. 2616–2620.

[60] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. K. Goel, M. Hannemann, P. Motlícek, Y. Qian, P. Schwarz, J. Silovský, G. Stemmer, and K. Veselý, "The Kaldi speech recognition toolkit," in *Proc. IEEE Workshop on Automatic Speech Recognition and Understanding*, Hilton Waikoloa Village, Big Island, Hawaii, US, 2011.

[61] "SRE16 xvector model." [Online]. Available: http://kaldi-asr.org/models/m3

[62] A. Anjos, L. E. Shafey, R. Wallace, M. Günther, C. McCool, and S. Marcel, "Bob: A free signal processing and machine learning toolbox for researchers," in *Proc. ACM on Multimedia Systems (ACMMM)*, Nara, Japan, 2012.

[63] R. Fletcher, "Practical methods of optimization; (2nd ed.)," *John Wiley & Sons*, 1987.

[64] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[65] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. International Conference on Learning Representations (ICLR)*, San Diego, CA, USA, 2015.

[66] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. Devito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in PyTorch," in *Proc. Neural Information Processing Systems (NIPS)*, Long Beach, CA, USA, 2017, pp. 1–4.

[67] P. Korshunov and S. Marcel, "Cross-database evaluation of audio-based spoofing detection systems," in *Proc. Interspeech*, San Francisco, CA, USA, 2016, pp. 1705–1709.

Alejandro Gomez-Alanis was born in Granada, Spain, in 1994. He received the B.Sc. and M.Sc. degrees in telecommunications engineering from the University of Granada, Spain, in 2016 and 2018, respectively. In 2016 and 2017 he worked as a Software Engineer at Seplin and Strivelabs for developing automatic statistical tools. Since late 2017 he holds an FPU fellowship for pursuing the Ph.D. degree with the Department of Signal Theory, Telematics and Communications at the University of Granada working on speech biometrics. His research activities focus on the processing, modelling, and classification of speech for human-oriented applications.

Jose A. Gonzalez received the B.Sc. and Ph.D. degrees in computer science, both from the University of Granada, Granada, Spain, in 2006 and 2013, respectively. He then spent four years as a Postdoctoral Research Associate at the University of Sheffield, Sheffield, U.K., working on silent speech technology with special focus on speech synthesis from speech-related biosignals. In late 2017 he took up a Lectureship at the Department of Languages and Computer Sciences, University of Malaga, Spain. Since 2019 he holds a Juan de la Cierva - Incorporacion fellowship at the University of Granada, working on silent speech interfaces and speech biometrics. His research activities focus on the processing, modelling, and classification of speech for human-centered applications. He has authored or co-authored more than 60 papers in these areas.

S. Pavankumar Dubagunta is a Research Assistant at Idiap Research Institute and a PhD candidate in Electrical Engineering at École polytechnique fédérale de Lausanne (EPFL). His thesis focuses primarily on developing methods to learn features from raw waveform of speech in a task dependent manner for Automatic Speech Assessment. He has about four years of industry experience in Speech Recognition and Audio Processing: as a Research Intern at Google, as a Senior Speech Engineer at Interactive Intelligence (now Genesys Telecom Labs) and as a Lead Engineer at Samsung RD Institute India. Prior to that, he received his Master of Science by Research in Electrical Engineering for his work on Feature Normalisation for Robust Speech Recognition, from Indian Institute of Technology Madras. He holds a Bachelor of Engineering, in Electronics and Communication Engineering, from Andhra University. His interests are in the areas of Speech/Audio Processing, Deep Learning and Signal Processing.

Antonio M. Peinado (M'95–SM'05) received the M.S. and Ph.D. degrees in physics (electronics specialty) from the University of Granada, Granada, Spain, in 1987 and 1994, respectively. In 1988, he worked with Inisel as a Quality Control Engineer. Since 1988, he has been with the University of Granada, where he has led several research projects related to signal processing and transmission. In 1989, he was a Consultant with the Speech Research Department, AT&T Bell Labs, Murray Hill, NJ, USA, and, in 2018, a Visiting Scholar with the Language Technologies Institute of CMU, Pittsburgh, PA, USA. He has held the positions of an Associate Professor from 1996 to 2010 and a Full Professor since 2010 with the Department of Signal Theory, Networking and Communications, University of Granada, where he is currently the Head of the research group on Signal Processing, Multimedia Transmission and Speech/Audio Technologies. He authored numerous publications in international journals and conferences, and has co-authored the book entitled Speech Recognition Over Digital Channels (New York, NY, USA: Wiley, 2006). His current research interests are focused on several speech technologies (anti-spoofing for automatic speaker verification, speech enhancement, and robust speech recognition and transmission), image processing and proteomic signal processing. Prof. Peinado has been a reviewer for a number of international journals and conferences, an evaluator for project and grant proposals, and a Member of the technical program committee of several international conferences.

Mathew Magimai.-Doss(S'03, M'05) received the Bachelor of Engineering (B.E.) in Instrumentation and Control Engineering from the University of Madras, India in 1996; the Master of Science (M.S.) by Research in Computer Science and Engineering from the Indian Institute of Technology, Madras,India in 1999; the PreDoctoral diploma and the Docteur ès Sciences (Ph.D.) from the Ecole polytechnique fédérale de Lausanne (EPFL), Switzerland in 2000 and 2005, respectively. He was a postdoctoral fellow at the International Computer Science Institute (ICSI), Berkeley, USA from April 2006 till March 2007. He is now a Senior Researcher at the Idiap Research Institute, Martigny, Switzerland. He is also a lecturer at EPFL. His main research interest lies in signal processing, statistical pattern recognition, artificial neural networks and computational linguistics with applications to speech and audio processing and multimodal signal processing. He is a Senior Area Editor of the IEEE Signal Processing Letters. He is also an Associate Editor of the IEEE/ACM Transactions on Audio, Speech, and Language Processing.