TESIS DOCTORAL

Robust speaker verification systems based on deep neural networks

Universidad de Granada

Departamento de Teoría de la Señal, Telemática y Comunicaciones

Programa de Doctorado en Tecnologías de la Información y Comunicación

Autor:

Alejandro Gómez Alanís

Directores:

Antonio Miguel Peinado Herreros

José Andrés González López

Granada, noviembre de 2021

Editor: Universidad de Granada. Tesis Doctorales
Autor: Alejandro Gómez Alanís
ISBN: 978-84-1117-220-2
URI: http://hdl.handle.net/10481/72468

Agradecimientos

Esta tesis está dedicada a todas las personas que han hecho posible su realización, que me han apoyado y motivado durante este interesante camino.

En primer lugar quiero expresar mi más profunda gratitud a mis directores de tesis. A Antonio por su dedicación y pasión por nuestra investigación y trabajo. Por todo el conocimiento y experiencia que me ha transferido, y por todos los buenos ratos que hemos compartido a diario. A José Andrés por todos sus valiosos consejos y su ayuda en la definición y elaboración de todo el trabajo de la tesis doctoral, y sobre todo por apoyarme y motivarme para cumplir todos los objetivos. Antonio y José Andrés, me siento muy afortunado por haberos tenido como tutores y poder compartir estos años con vosotros. ¡Gracias!

Me gustaría agradecer a los miembros del grupo de investigación SigMAT: mis amigos Juanma, Amelia, Ángel, Victoria y José Luis. Muchas gracias por nuestras motivadoras conversaciones y salidas que tanto hemos disfrutado.

Y finalmente, una última y gran mención a mi familia y amigos que tanto me han apoyado para formarme y dedicarme a lo que más me gusta. En este trabajo podéis encontrar vuestra huella también.

GRACIAS A TODOS

Abstract

In a world becoming more and more digital, the need for robust authentication methods enabling the secured access to resources and systems is becoming crucial. In early stages, the identity management systems relied on cryptographic methods requiring the users to remember a password, store cards or even a combination of both to prove their identity. As opposed to these authentication methods, a more natural alternative for human identification/verification is that based on physiological (fingerprint, face, iris, etc) or behavioral (voice, gait, signature, etc) attributes of individuals known as biometrics. This Thesis is focused on voice biometric systems for human verification where the speech signal is employed for making a one-to-one comparison between the user's voice and all the enrolled voices stored in the database.

The main goal of this Thesis is the development of robust automatic speaker verification (ASV) systems which are able to detect the two main types of biometric attacks: (i) zero-effort attacks, where a non-enrolled speaker utters bonafide speech in order to try to gain access as an enrolled speaker; and (ii) spoofing attacks, where an impostor tries to gain fraudulent access by presenting speech resembling the voice of a genuine enrolled speaker. The vulnerability of ASV systems to malicious spoofing attacks is a serious concern nowadays, since an impostor can easily present a pre-recorded voice of an enrolled user (replay spoofing attack), generates artificial voice resembling the voice of an enrolled user (text-to-speech spoofing attack), or transform the voice recording of a given speaker so that it sounds as that from an enrolled speaker without changing the phonetic content of the recording (voice conversion spoofing attack). For making voice biometric systems more robust to this type of attacks, we propose the following contributions in this Thesis.

First, we have have dealt with the problem of spoofing attack detection for voice biometric systems. The main problem here is the lack of robustness and generalization across different databases. We addressed this issue by proposing a novel neural network architecture which can be used for detecting both logical and physical access spoofing attacks. The proposed convolutional RNN-based architecture is able to process the whole input utterance without cropping it or applying any post-processing combination of chunks. Moreover, since noisy acoustic scenarios can significantly degrade the performance of anti-spoofing systems, we have also proposed two noise-aware techniques based on the usage of masks which help to effectively reduce the performance degradation. Our best performing technique involves the computation and use of signal-to-noise masks that inform the DNN-based spoofing embedding extractor of the noise probability for each time-frequency bin in the input speech spectrogram.

Secondly, we also proposed new loss functions which can be effectively used by anti-spoofing and integration of ASV and anti-spoofing systems. We have proposed a new probabilistic loss function for supervised metric learning, where every training class is represented with a probability density function using all the samples of the mini-batch and is estimated through kernel density estimation. We can argue that each class is more accurately represented than in other popular loss functions. Moreover, the proposed loss function replaces the concept of distance between

embeddings in negative hard-mining techniques by the concept that an embedding belongs to a class with a given probability. This has the advantage of avoiding the selection of an appropiate distance measure and tuning extra hyper-parameters such as distance margins. Furthermore, we also propose a new loss function for integration systems based on the expected performance and spoofability curve (EPSC) which allows to optimize the voice biometric system in the operating range, instead of only one operating point, in which it is expected to work during evaluation. These proposals allow to improve significantly the performance of both anti-spoofing and complete voice biometric systems.

Third, we have studied the integration of ASV and anti-spoofing systems at the score-level and at the embedding-level. To avoid the integration of ASV and anti-spoofing systems at the score-level using scores computed separately, we proposed a new neural network architecture for integrating the systems at the embedding-level which exploits the fact that ASV and anti-spoofing systems share the bonafide speech subspace. Thus, the proposed integration system is able to model the three main biometric speech subspaces: bonafide speech, zero-effort attacks and spoofing attacks. Experimental results on the ASVspoof 2019 corpus show that the joint processing of the ASV and anti-spoofing embeddings with the proposed integration neural network clearly outperforms other state-of-the-art techniques trained and evaluated on the same conditions.

Finally, we have studied the robustness of the state-of-the-art voice biometric systems under the presence of adversarial spoofing attacks. Furthermore, we also proposed a new DNN-based generator network for this type of attacks which is trained using existing spoofing attacks and it can be used for finetuning the biometric system in order to make it more robust to adversarial spoofing attacks. Experimental results show that voice biometric systems are highly sensitive to adversarial spoofing attacks in both logical and physical access scenarios. Moreover, the proposed ABTN generator clearly outperforms other classical adversarial attacks techniques such as the fast gradient signed method (FGSM) and the projected gradient descent (PGD).

To conclude, we would like to highlight that our contributions successfully integrate the signal processing and deep learning methods for developing robust voice biometric systems. As a result, the systems proposed in this Thesis significantly outperform other state-of-the-art systems.

Resumen

En un mundo cada vez más digital, la necesidad de métodos de autenticación robustos que permitan el acceso seguro a los recursos y sistemas se está volviendo crucial. En las primeras etapas, los sistemas de gestión de identidad se basaban en métodos criptográficos que exigían a los usuarios recordar una contraseña, almacenar tarjetas o incluso una combinación de ambos para probar su identidad. A diferencia de estos métodos de autenticación, una alternativa más natural para la identificación / verificación humana es la basada en atributos fisiológicos (huellas dactilares, rostro, iris, etc.) o conductuales (voz, marcha, firma, etc.) de los individuos conocidos como biométricos. Esta tesis se centra en los sistemas biométricos de voz para la verificación humana donde la señal de voz se emplea para hacer una comparación uno a uno entre la voz del usuario y todas las voces registradas almacenadas en la base de datos.

El objetivo principal de esta Tesis es el desarrollo de sistemas robustos de verificación automática de locutores (ASV) que sean capaces de detectar los dos tipos principales de ataques biométricos: (i) ataques de esfuerzo cero, donde un hablante no inscrito pronuncia una frase para intentar ganar acceso como si fuese un locutor legítimo; y (ii) ataques de suplantación de identidad, en los que un impostor intenta ganar acceso fraudulento presentando una frase que se asemeja a la voz de un locutor legítimo genuino. La vulnerabilidad de los sistemas ASV a ataques de suplantación de identidad maliciosos es una preocupación seria hoy en día, ya que un impostor puede presentar fácilmente una voz pregrabada de un usuario inscrito (ataque de suplantación de reproducción), genera una voz artificial que se asemeja a la voz de un usuario inscrito (ataque de suplantación de síntesis de voz), o transformar la grabación de voz de un locutor dado para que suene como la de un locutor registrado sin cambiar el contenido fonético de la grabación (ataque de suplantación de conversión de voz). Para hacer los sistemas biométricos de voz más robustos a este tipo de ataques, proponemos las siguientes contribuciones en esta Tesis.

En primer lugar, hemos abordado el problema de la detección de ataques de suplantación de identidad para sistemas biométricos de voz. El principal problema aquí es la falta de solidez y generalización en diferentes bases de datos. Abordamos este problema proponiendo una nueva arquitectura de red neuronal que se puede utilizar para detectar ataques de suplantación de acceso tanto lógicos como físicos. La arquitectura convolucional basada en redes neuronales recurrentes (RNNs) propuesta es capaz de procesar toda la locución de entrada sin recortarla ni aplicar ninguna combinación de fragmentos de posprocesamiento. Además, dado que los escenarios acústicos ruidosos pueden degradar significativamente el rendimiento de los sistemas de anti-spoofing, también hemos propuesto dos técnicas de detección de ruido basadas en el uso de máscaras que ayudan a reducir eficazmente la degradación del rendimiento. Nuestra técnica de mejor rendimiento implica el cálculo y el uso de máscaras de señal a ruido que informan al extractor de características de suplantación de identidad basado en redes neuronales profundas (DNNs) de la probabilidad de ruido para cada intervalo de frecuencia de tiempo en el espectrograma de voz de entrada.

En segundo lugar, también hemos propuesto nuevas funciones de coste que se pueden utilizar

de forma eficaz para la detección de ataques de suplantación de identidad y para la integración de sistemas ASV y anti-spoofing. Hemos propuesto una nueva función de coste probabilística para el aprendizaje métrico supervisado, donde cada clase de entrenamiento se representa con una función de densidad de probabilidad utilizando todas las muestras del *batch* de entrenamiento y se estima mediante la estimación de la densidad del kernel. Podemos argumentar que cada clase está representada con mayor precisión que en otras funciones de coste populares. Además, la función de coste propuesta reemplaza el concepto de distancia entre *embeddings* en técnicas de minería dura negativa por el concepto de que un *embedding* pertenece a una clase con una probabilidad determinada. Esto tiene la ventaja de evitar la selección de una medida de distancia adecuada y ajustar hiperparámetros adicionales como los márgenes de distancia. Además, también proponemos una nueva función de coste para sistemas de integración basada en la curva de rendimiento esperado y *spoofability* (EPSC) que permite optimizar el sistema biométrico de voz en el rango operativo, en lugar de un solo punto operativo, en el que se espera que el sistema trabaje durante la evaluación. Estas propuestas permiten mejorar significativamente el rendimiento de los sistemas biométricos de voz tanto de anti-spoofing como completos.

En tercer lugar, hemos estudiado la integración de ASV y sistemas de anti-spoofing a nivel de *scores* y a nivel de *embeddings*. Para evitar la integración de ASV y sistemas de anti-spoofing a nivel de *scores* utilizando puntuaciones calculadas por separado, hemos propuesto una nueva arquitectura de red neuronal para integrar los sistemas a nivel de *embeddings* que explota el hecho de que el sistema de ASV y los sistemas de anti-spoofing comparten el subespacio de voz genuino. Por tanto, el sistema de integración propuesto es capaz de modelar los tres principales subespacios biométricos de la voz: voz auténtica, ataques de esfuerzo cero y ataques de suplantación de identidad. Los resultados experimentales en el corpus ASVspoof 2019 muestran que el procesamiento conjunto de ASV y los *embeddings* de anti-spoofing con la red neuronal de integración propuesta supera claramente a otras técnicas del estado del arte entrenadas y evaluadas en las mismas condiciones.

Finalmente, hemos estudiado la robustez de los sistemas biométricos de voz de última generación ante la presencia de ataques de suplantación de identidad adversarios. Además, también hemos propuesto una nueva red generadora basada en DNNs para este tipo de ataques que se entrena utilizando ataques de suplantación de identidad existentes y se puede utilizar para ajustar el sistema biométrico con el fin de hacerlo más robusto a los ataques de suplantación de identidad adversarios. Los resultados experimentales muestran que los sistemas biométricos de voz son muy sensibles a los ataques de suplantación de identidad adversarios en escenarios de acceso lógico y físico. Además, el generador propuesto supera claramente a otras técnicas clásicas de ataques adversarios, como el método rápido con signo de gradiente (FGSM) y el descenso de gradiente proyectado (PGD).

En conclusión, nos gustaría destacar que nuestras contribuciones integran con éxito los métodos de procesamiento de señales y aprendizaje profundo para desarrollar sistemas biométricos de voz robustos. Como resultado, los sistemas propuestos en esta Tesis superan significativamente a otros sistemas del estado del arte.

Contents

Chapter 1

Introduction

1.1 Background

In a world becoming more and more digital, the need for robust authentication methods enabling the secured access to resources and systems is becoming crucial. In early stages, the identity management systems relied on cryptographic methods requiring the users to remember a password, store cards or even a combination of both to prove their identity. Consequently, the users were flooded with passwords to gain access to resources that they require, such as bank transactions, email login, unblock the mobile phone, border control, etc.

As opposed to these authentication methods, a more natural alternative for human identification/verification is that based on physiological (fingerprint, face, iris, etc) or behavioral (voice, gait, signature, etc) attributes of individuals known as *biometrics* [1]. Despite the fact that biometrics is not still the perfect solution, it offers several advantages over knowledge and possesion based approaches since there is no need to remember anything, and biometric attributes cannot be lost, transferred or *easily* stolen. Thus, a biometric system is a pattern recognition system which matches the salient or discriminatory features of acquired signals (probe signal) with those of pre-stored signals (gallery signals). A biometric system can operate in one of the three following modes:

- VERIFICATION: the system validates or voids user's claim by making a one-to-one comparison between the submitted biometric signature and the enrolled biometric signature associated with that particular identity [2]. Example applications include computer logins, e-commerce, access control, user authentication on mobile phones, etc.

- IDENTIFICATION: the system tries to recognize the user by comparing the submitted biometric signature to all enrolled signatures stored in the database by making one-to-many comparisons without specific identity claim from the user [2]. Identification prevents an individual from using multiple identities. Example applications include issuance of ID cards, passports, driving licenses, etc.

- SCREENING: this is an extension to identification where the biometric system assures that a particular individual does not belong to a watch list of identities by performing one-to-many comparisons throughout the database [3]. Example applications include airport security, surveillance activities, etc.

This thesis is focused on voice biometric systems for human verification where the speech signal is employed for making a one-to-one comparison between the user's voice and all the enrolled voices stored in the database. This type of biometrics is known as automatic speaker verification (ASV) [4]. There are two main types of attacks for ASV systems:

- ZERO-EFFORT ATTACKS: a non-enrolled speaker utters *bonafide* speech in order to try to gain access as an enrolled speaker. This type of attacks is normally easily detected by the ASV system since the one-to-one comparison does not match any previous enrolled speaker.
- SPOOFING ATTACKS: an impostor tries to gain fraudulent access by presenting speech resembling the voice of a genuine enrolled speaker.

The vulnerability of ASV systems to malicious *spoofing* attacks is a serious concern nowadays [5]. Four types of *spoofing* attacks have been identified:

- IMPERSONATION: a person mimicks the voice of an enrolled user. This type of attack is normally easy to detect since the physiological attributes of the voices are different.
- REPLAY: a person presents a pre-recorded voice of an enrolled user. This type of attack is very difficult to detect since the artefacts introduced by the microphone and loudspeaker must be detected.
- TEXT-TO-SPEECH (TTS) SYNTHESIS: generates artificial voice resembling the voice of an enrolled user.
- VOICE CONVERSION (VC): aims to transform voice recordings of a given (source) speaker so that they sound as those from a target speaker without changing the phonetic content (message) of the recordings.

Moreover, the *spoofing* 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 *spoofing* signal is presented to or captured by the sensor, i.e., the microphone. Whilst, in the LA attack scenario, the sensor is by-passed and attacks are directly injected into the ASV system, typically generated using TTS or VC technologies. Anti-spoofing or presentation attack detection (PAD in ISO/IEC 30107 nomenclature [6]) systems for ASV have gained increased attention in recent years. This thesis is also mainly devoted to improving the performance of the state-of-the-art anti-spoofing systems.

This memory is divided in three chapters and is organized as follows. In Chapter 1, after this introductory part, a background overview of the three main research topics treated in this work is provided in the next subsections (1.1.1-1.1.6). In Section 1.2 we address the open problems and describe the starting hypotheses that justify the elaboration of this work. The objectives of this thesis are detailed in Section 1.3. In Section 1.4, the proposed techniques are briefly summarized. Chapter 2 consists of the publications that deal with the proposed objectives. Finally, the last Chapter is devoted to conclusions and it outlines some important aspects to be taken into account in the future work.

Introducción

En un mundo cada vez más digital, la necesidad de métodos de autenticación sólidos que permitan el acceso seguro a los recursos y sistemas se está volviendo crucial. En las primeras etapas, los sistemas de gestión de identidad se basaban en métodos criptográficos que exigían a los usuarios recordar una contraseña, almacenar tarjetas o incluso una combinación de ambos para probar su identidad. En consecuencia, los usuarios se vieron inundados de contraseñas para acceder a los recursos que requieren, como transacciones bancarias, inicio de sesión de correo electrónico, desbloqueo del teléfono móvil, control de fronteras, etc.

A diferencia de estos métodos de autenticación, una alternativa más natural para la identificación/verificación humana es la basada en atributos fisiológicos (huellas dactilares, rostro, iris, etc.) o conductuales (voz, marcha, firma, etc.) de personas conocida como biometría [1]. A pesar de que la biometría aún no es la solución perfecta, ofrece varias ventajas sobre los enfoques basados en el conocimiento y la posesión, ya que no hay necesidad de recordar nada y los atributos biométricos no se pueden perder, transferir o fácilmente robar. Por tanto, un sistema biométrico es un sistema de reconocimiento de patrones que hace coincidir las características destacadas o discriminatorias de las señales adquiridas (señal de sonda) con las de las señales pre-almacenadas (señales de galería). Un sistema biométrico puede funcionar en uno de los tres modos siguientes:

- Verificación: el sistema valida o anula la reclamación del usuario haciendo una comparación uno a uno entre la firma biométrica enviada y la firma biométrica registrada asociada con esa identidad en particular [2]. Algunas de sus aplicaciones son los inicios de sesión en ordenadores personales, comercio electrónico, control de acceso, autenticación de usuarios en teléfonos móviles, etc.

- Identificación: el sistema intenta reconocer al usuario comparando la firma biométrica enviada con todas las firmas registradas almacenadas en la base de datos haciendo comparaciones de uno a varios sin una declaración de identidad específica del usuario [2]. La identificación evita que un individuo use múltiples identidades. Algunas de sus aplicaciones son la emisión de tarjetas de identificación, pasaportes, permisos de conducir, etc.

- Cribado: esta es una extensión de la identificación donde el sistema biométrico asegura que un individuo en particular no pertenece a una lista de vigilancia de identidades al realizar comparaciones de uno a muchos en toda la base de datos [3]. Algunas de sus aplicaciones son la seguridad aeroportuaria, actividades de vigilancia, etc.

Esta tesis se centra en los sistemas biométricos de voz para verificación humana donde la señal de voz se emplea para realizar una comparación uno a uno entre la voz del usuario y todas las voces registradas en la base de datos. Este tipo de datos biométricos se conoce como verificación automática de locutores (ASV) [4]. Hay dos tipos principales de ataques para los sistemas ASV:

- Ataques de esfuerzo cero: un locutor no registrado pronuncia una frase filedigna para intentar obtener acceso como si fuese un locutor registrado. El sistema ASV suele detectar fácilmente este tipo de ataques, ya que la comparación uno a uno no coincide con ningún locutor registrado anteriormente.

- Ataques de suplantación de identidad: un impostor intenta obtener acceso fraudulento presentando una frase que se asemeja a la voz de un locutor genuino registrado.

La vulnerabilidad de los sistemas ASV a los ataques de suplantación de identidad es una preocupación seria hoy en día [5]. Se han identificado cuatro tipos de ataques de suplantación de identidad:

- Imitación: una persona imita la voz de un usuario registrado. Este tipo de ataque suele ser fácil de detectar ya que los atributos fisiológicos de las voces son diferentes.
- Repetición: una persona presenta una voz pregrabada de un usuario registrado. Este tipo de ataque es muy difícil de detectar ya que normalmente se deben detectar los artefactos introducidos por el micrófono y el altavoz.
- Síntesis de texto a voz (TTS): genera una voz artificial que se asemeja a la voz de un usuario registrado.
- Conversión de voz (VC): tiene como objetivo transformar las grabaciones de voz de un locutor dado (fuente) para que suenen como las de un locutor objetivo sin cambiar el contenido fonético (mensaje) de las grabaciones.

Además, los ataques de suplantación de identidad se pueden presentar al sistema ASV de acuerdo con dos escenarios diferentes: acceso lógico (LA) y acceso físico (PA). En el escenario de ataque de acceso físico, la señal de suplantación es presentada o capturada por el sensor, es decir, el micrófono. Mientras que en el escenario de ataque de acceso lógico, el sensor se pasa por alto y los ataques se inyectan directamente en el sistema ASV, generalmente generados mediante tecnologías TTS o VC. Los sistemas de anti-spoofing o de detección de ataques de presentación (PAD en la nomenclatura [6] ISO/IEC 30107) para ASV han ganado una mayor atención en los últimos años. Esta tesis también está dedicada principalmente a mejorar el rendimiento de los sistemas de anti-spoofing de última generación.

Esta memoria se divide en tres capítulos y se organiza de la siguiente manera. En el Capítulo 1, después de esta parte introductoria, se proporciona una descripción general de los tres principales temas de investigación tratados en este trabajo en las siguientes subsecciones (1.1.1 - 1.1.6). En la Sección 1.2 abordamos los problemas abiertos y describimos las hipótesis de partida que justifican la elaboración de este trabajo. Los objetivos de esta tesis se detallan en la Sección 1.3. En la Sección 1.4, se resumen brevemente las técnicas propuestas. El capítulo 2 consta de las publicaciones que tratan los objetivos propuestos. Finalmente, el último Capítulo está dedicado a las conclusiones y describe algunos aspectos importantes a tener en cuenta en el trabajo futuro.

1.1.1 Voice Spoofing Detection

Replay, TTS and VC *spoofing* attacks degrade the performance of ASV systems [7]. Boosted by fraud prevention in call-centers and securing our identities in other applications, a new research community working on voice anti-spoofing has emerged during the past few years as evidenced by the organization of multiple evaluation campaigns (challenges): (i) ASVspoof 2015 [8], which focused on LA attack scenarios (TTS and VC attacks); (ii) BTAS 2016 [9], which addressed both the detection of LA and PA-based attacks; (iii) ASVspoof 2017 [10], which focused on PA scenarios (real replay attacks) under noisy environments; (iv) ASVspoof 2019 [11], which addressed both the detection of LA-based attacks generated with the latest TTS and VC technologies, and simulated replay attacks under different reverberant conditions; and (v) ASVspoof 2021 [12], where apart from LA and PA scenarios, a new speech deepfake (DF) task has been introduced to reflect the

scenario in which an attacker has access to the voice data of a targeted victim, e.g. data posted to social media.

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 or it is a *spoofing* attack. There are two main machine learning models to detect *spoofed* speech [13]: (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 [14], linear frequency cepstral coefficients (LFCC) [15], constant Q cepstral coefficients (CQCC) [16], and raw speech samples [17]. In the last ASVspoof challenges [10]–[12], deep learning has shown to be the most effective approach to detect *spoofing* attacks.

The evaluation of standalone PAD systems is carried out in terms of the spoof protocol [18], which contains *bonafide* speech and *spoofing* attacks. Just like in ASV, the equal error rate (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 [19] 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 2021 challenge [12] in the LA and PA scenarios.

1.1.2 Deep Neural Networks for Voice Biometric Systems

Anti-spoofing systems must learn to detect not only the attacks observed in the training dataset, but also be able to generalize to unseen attacks. To address this issue, deep feature extraction was proposed in [20], where deep features (also known as embeddings) are extracted from an inner layer of a deep neural network (DNN) to represent every temporal frame of the voice signal, or even the whole utterance. The system attempts to determine whether the input speech signal is genuine or *spoofing*. To this end, the classifier make use of the embeddings extracted by a DNN. Depending on the architecture of the neural network, we can differentiate two types of deep features: (i) frame-level, and (ii) utterance-level. Moreover, the nature of the speech signal features which are fed into the deep feature extractor can also determine the whole performance of the anti-spoofing system [21], [22]. Thus, we can find in the literature three types of speech signal features which have been successfully applied to *spoofing* detection: (i) magnitude based spectral features [23], (ii) phase based spectral features [24], and (iii) raw speech samples [25].

The extraction of deep features (embeddings) at a frame level has demonstrated to be effective in both ASV [26] and spoofing detection [27]. For instance, x-vectors [28] have become very popular in ASV due to its good performance, superior to that of i-vectors [29]. Regarding anti-spoofing, DNNs and convolutional neural networks (CNNs) were used in [30] to obtain frame-level deep features, showing that convolutional layers have a powerful ability for detecting the artefacts caused by the speech vocoders used in TTS/VC systems. This is even possible in noisy conditions, as CNNs can be seen as filter banks whose filters are optimized for the specific task of *spoofing* detection [31]. In addition, a residual CNN architecture was also employed in [32] as a frame-level feature extractor for detecting replay attacks, and a Light-CNN [14] which employs Max-Feature-Map activations was the best system of the ASVspoof 2017 Challenge [10].

These frame-level features must be combined into a single identity vector which characterizes the whole utterance. There are several ways to combine them such as averaging [33], attentive statistics pooling [34], or by using recurrent neural networks (RNNs) [35]. There is an ample evidence that RNNs are powerful at extracting discriminative features to capture the temporal

artefacts in the *spoofed* speech. For instance, a combination of a CNN with an RNN based on gated recurrent unit (GRU) blocks was successfully applied in [35] to extract utterance-level deep features for detecting logical access attacks. Likewise, a combination of several fully connected layers with two long short-term memory (LSTM) blocks was proposed in [36] as an end-to-end system. In [14], a combination of a Light-CNN with an RNN was proposed to extract utterance-level embeddings for detecting replay attacks. In addition, a deep siamese neural network architecture based on convolutional layers was proposed in [37] as an utterance-level feature extractor to improve the performance of the previous systems at detecting replay attacks. More recently, end-to-end systems based on RNNs have also been proposed in [38], [39] for detecting replay attacks.

1.1.3 Voice Spoofing Detection Under Noisy and Reverberant Acoustic Scenarios

Research on anti-spoofing has been mainly focused on systems operating on clean conditions, while little work has been carried out considering the effects of noise on those systems (i.e. acoustic noise and/or reverberation) which will be likely present in realistic situations. Noise will be, in general, a cause for performance degradation, although its effects varies according to the type of attack. Thus, noise introduced by the playback and recording devices might be a hint of attack [10], but it cannot be easily separated from the noise present in the acoustic environment, which, in turn, may conceal those hints. For instance, as shown in [40], 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 is one of the motivations of this work.

One of the first works to study the impact of noise on antispoofing systems was [41], where the robustness of several front-end features were evaluated under different noisy conditions. In [40] 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, [30] showed that the anti-spoofing techniques based on deep feature extractors improve significantly when they are trained with noisy data (i.e. multicondition training), owing this improvement to the cabapility of neural networks to learn discriminative features which are more invariant to noise. Furthermore, in [30], robustness against noise was improved by means of noise aware training (NAT), in which a vector with the mean noise magnitude spectra is presented to the network.

1.1.4 Loss Functions for Voice Biometric Neural Networks

DNN-based approaches have been widely explored for audio- [35] and video-based anti-spoofing [42], [43]. Three key points are important for building a DNN-based anti-spoofing system with generalization capabilities: (i) architecture, (ii) input features, and (iii) loss function. Multiple types of DNN architectures have been explored to this end, such as feed-forward DNNs [33], CNNs [30], RNNs [38], light convolutional neural networks (LCNNs) [14], central difference convolutional networks (CDCNs) [42], etc. Also, a wide range of features have been proposed to train these models. Normally, the DNN architecture is adapted to the dimension of the input features, and viceversa. However, the loss function employed to train the DNN is usually independent from architecture and input features.

Within the DNN-based anti-spoofing framework, several recent studies have focused on designing new loss functions in order to make neural networks more suitable for the specific task of biometrics:

- **Cross-entropy loss function**: it is also known as softmax loss, and is widely used to train DNNs for classification tasks. Typically, when the softmax loss function is used in ASV and anti-spoofing systems, embeddings are extracted from a middle or the last hidden layer of the DNN.

- **Additive margin loss function**: The additive margin (AM) softmax loss function [44] was proposed to replace the inner product operation of the cross-entropy loss function with the cosine similarity operation in order to widen the inter-class margin in the embedding space [45]. This loss function is a generalized version of the angular softmax loss [44]. Recently, this type of loss function has been successfully applied to anti-spoofing [46] and speaker verification systems [47], [48].

- **Generalized end-to-end loss function**: In the generalized end-to-end (GE2E) loss [49], which was originally proposed for ASV, each class (speaker) is represented by a centroid obtained averaging all the embeddings belonging to that class in the mini-batch. From those centroids, two loss functions were proposed in [49] which seek for minimizing the distance between the embeddings and their corresponding class centroids, while also maximizing the distance with the centroids from the other speakers. In anti-spoofing, the speakers can be replaced by *spoofing* attacks.

- **Siamese loss function**: The siamese architecture processes two utterances at once using the same neural network, obtains two embeddings and computes a loss based on an embedding distance. There are many siamese network variants reported in the literature for different applications, such as face recognition [50], person identification [51] and image recognition [52].

- **Triplet loss function**: The triplet network [53] is a neural network architecture which attempts to learn an embedding representation of a multi-class labeled dataset which favours a small distance between example pairs labeled as similar, and large distances for pairs labeled as dissimilar. However, unlike the siamese networks, this architecture works with triplets of embeddings. In particular, it defines a loss function which ensures that an *anchor* embedding of a certain class is closer to other *positive* samples than to any *negative* sample [54]. Recently, the triplet loss function has been successfully applied to train face verification systems [54], ASV systems [55], [56], and joint ASV and PAD systems [57].

1.1.5 Integration Systems for Voice Biometric Systems

The integration of ASV and PAD systems can be achieved at the score level (late fusion) [58] or at the model/feature level (early fusion) [59]. 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:

- **Tandem or cascaded integration [58], [60], [61]**: ASV and PAD systems can be cascaded in either order - PAD followed by ASV or viceversa. 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.

- **Logistic regression fusion [61]**: Logistic regression has been successfully employed for combining several PAD systems [30], [62] and speaker classifiers [63], [64] 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.

- **Gaussian back-end fusion [65]**: For each ASV trial which belongs to one of the three voice biometric classes, 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 the classes *zero-effort* and *spoofing*.

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:

- **Two-stage probabilistic linear discriminant analysis (PLDA) [59]**: This technique is composed of two stages. First, it trains a simplified PLDA [66] 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.

- **Multi-task triplet time-delay neural network (TDNN) [57]**: This approach extracts embeddings that contain speaker identity and spoofing information using a multi-task TDNN [67] which is optimized using the triplet loss [54]. The dimension of these embeddings is then reduced using linear discriminant analysis (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 [19], [68] 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. On the other hand, score- and embedding-level integration systems 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 receiver operating characteristic curve hull (ROCCH)-EER ensures the optimization of the entire receiver operating characteristic (ROC) curve 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 expected performance curve (EPC) framework was developed for evaluating integration systems,

namely, the expected performance and spoofability (EPS) framework [69]. To enable this, it establishes a criterion 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 ($\text{WER}_{\omega,\beta}$) [69] with respect to one of the parameters ω or β, while the other one is fixed to a predefined value. Thus, the global performance of the integrated biometric system can be computed as the area under the EPS (AUE) curve [69]. 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).

1.1.6 Adversarial Examples for Voice Biometric Systems

To make things more complex, different investigations [70], [71] have recently shown that PAD systems are also vulnerable to adversarial attacks [72]. They are attacks which can easily fool DNN-based models by perturbing benign samples in a way normally imperceptible to humans [73]. Adversarial attacks can be divided into two main categories: white-box and black-box attacks. In this work, we refer to white-box attacks as those where the attacker can access all the information of the victim model (i.e., model architecture and its weights). Likewise, we will use the term black-box for those attacks where the attacker does not know any information about the victim model but it can be queried multiple times in order to estimate a surrogate model (student) of the victim model (teacher), using the binary responses (acceptance/rejection) of the victim model as ground-truth labels.

Adversarial *spoofing* examples can be generated by adding a minimally perceptible perturbation to the input *spoofing* utterance in order to do a refinement of the *spoofing* attack. The perturbation is found by solving an optimization problem. There are two types of adversarial attacks: targeted and nontargeted attacks. Targeted attacks aim at maximizing the probability of a targeted class which is not the correct class, whereas nontargeted attacks aim at minimizing the probability of the correct class. In this work, we focus on targeted attacks, which aim to fool the PAD system by maximizing the probability of a targeted class (*bonafide*) different from the correct class (*spoof*).

1.2 Drawbacks of previous works

In this section, we outline the main drawbacks of previous investigations summarized in Section 1.1 and identify some related issues which deserve a further research.

- **Neural Network Architectures for Anti-spoofing Systems**: The best anti-spoofing systems so far (according to the results from ASVSpoof challenges [10]–[12]) are those based on convolutional neural networks (CNNs) which are able to extract very good deep features at the frame level [14]. There are usually two approaches to process a whole utterance with a CNN: (1) crop the utterance so that it has a fixed length of frames and process the whole cropped utterance with the CNN, and (2) divide the utterance into multiple chunks of the same length, process each chunk with the CNN, and finally average the softmax probability outputs of all chunks in order to obtain the final probability output of the whole utterance. However, the following issues should be addressed:

- Cropping an utterance can cause the loss of frames, normally from the end of utterance, which might have the keys to detect the *spoofing* artefacts. The cutting-off technique might not be necessary for training, since the development *spoofing* detection datasets can be created with utterances of the same length. However, it will be likely necessary when using the anti-spoofing system in real scenarios, where it is very difficult to know by default the duration of the utterances.
- The division of an utterance into multiple chunks is an alternative technique which avoids the issue of losing any acoustic frames from the utterance. However, most systems employing this technique average out the softmax probabilities for all chunks in order to obtain the final probability output of the whole utterance being *spoofing*. The average technique might not be optimal for *spoofing* detection when combining multiple outputs since it can compensate some *spoofing* detected chunks by some other genuine detected chunks. Furthermore, the averaging technique is a post-processing technique which does not form part of the anti-spoofing system which is optimized for the given training dataset.
- The CNNs employed for anti-spoofing systems are not designed to handle noisy acoustic scenarios where the *spoofing* detection task is more difficult.

- **Loss Functions of Voice Biometric Systems**: Voice biometric systems are normally formed by an ASV system and an anti-spoofing system. State-of-the-art systems for ASV and anti-spoofing are based on DNNs trained using the standard cross-entropy loss function. DNNs are also employed in state-of-the-art systems to extract embedding vectors, such as x-vectors [28], in order to use them in posterior stages of the biometric verification process. Within the DNN-based biometric framework, several recent studies have focused on designing new loss functions in order to make neural networks more suitable for the specific tasks of anti-spoofing [37], [74], ASV [49], [75] and/or their combination [57]. However, these studies do not usually address the following issues:

  - One particular characteristic of anti-spoofing systems is that embeddings extracted by DNNs should enable precise discrimination between *bonafide* speech and *spoofing* speech and, at the same time, they should be able to generalize well to unknown attacks which are not present in the development datasets. In other words, from a metric problem perspective [76], [77], the objective is to learn a meaningful embedding representation that keeps similar training instances closed to each other and the dissimilar instances far away on the embedding space. While specialized loss functions such as the triplet loss [53] address this issue, conventional loss functions such as the cross-entropy loss fall short in achieving this goal.
  - In supervised scenarios, as is the case of voice biometrics, metric learning aims to learn a representation which keeps close the embeddings belonging to the same class. In order to represent each class, different approaches have been investigated in the literature, such as representing each class by a centroid in the embedding space [49] or employing an anchor sample to represent the positive class [54]. In these examples, however, the training classes are not fully represented by all the samples of the mini-batch, but by a single embedding representation (i.e., either a centroid or an anchor sample), which may be suboptimal for distance learning.
  - Recent loss functions, such as the siamese [37], generalized-end-to-end [49] and triplet loss [55], [56] functions, are based on distance measures between embedding vectors.

However, the main issue of these new losses for working well for specific applications, such as voice biometrics, is to choose the most appropiate distance measure as well as tuning some more extra hyper-parameters such as a distance margin.

- Most voice biometric systems calibrate the ASV and anti-spoofing thresholds considering only one point of the error rate on the development dataset. However, it is normally 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.

- **Integration of ASV and Anti-spoofing Systems**: ASV and anti-spoofing systems have been mostly studied in isolation so far. In this regard, the integration of both subsystems still requires further research. Most integration systems in voice biometrics compute two scores: (1) ASV score, and (2) PAD score. In order to combine these two scores, one of the following approaches is normally used: (a) cascaded or tandem integration in which PAD precedes ASV, or viceversa, so that 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 a single score to the utterance. However, this type of integration based on independent scores might be suboptimal owing to one main reason:

  - Integration systems do not exploit the fact that ASV and PAD systems share the *bonafide* speech subspace, so that they could model it jointly in order to better discriminate between *bonafide* target speech and *zero-effort* impostors or *spoofing* attacks.

- **Adversarial Attacks**: Voice biometric systems usually consist of an ASV system which is able to detect *zero-effort* impostors, and by an anti-spoofing system which is able to detect *spoofing* attacks. Moreover, both subsystems are usually based on DNNs. Thus, these two subsystems should also be robust to adversarial attacks [72].

  - Adversarial attacks can easily fool DNN-based models by perturbing benign samples in a way normally imperceptible to humans [73]. Recent investigations [70], [71] have shown that anti-spoofing systems are very vulnerable to adversarial attacks.

1.3 Objectives

The main goal of this PhD thesis is to design and implement anti-spoofing and complete voice biometric systems that will outperform state-of-the-art techniques. In the following, we will specify the sub-objectives that enable us to accomplish the proposed challenge:

- Anti-spoofing architectures

  - **To avoid cropping the utterance or the post-processing averaging technique which does not form part of the optimization of the anti-spoofing system**. An RNN is proposed to either combine all the output predictions from the different chunks or directly process the whole utterance with a convolutional RNN-based anti-spoofing system.
  - **To handle noisy acoustic scenarios in spoofing detection**. Two noise awareness techniques are proposed for DNN-based anti-spoofing systems which are based on the usage of missing-data masks or signal-to-noise masks.

- Loss functions of voice biometric systems
  - **To avoid all the described issues detected in other loss functions for DNN-based biometric systems**. A new probabilistic loss function is proposed for supervised metric learning, where every training class is represented with a probability density function (pdf) which is estimated through kernel density estimation (KDE) [78]–[80] in each mini-batch. The mini-batches are formed so that all training classes are present in the mini-batch and are represented with the same number of samples. Since the proposed KDE based loss function estimates a pdf per class using all the samples of the mini-batch rather than representing each class with a sole point (centroid or anchor point), we can argue that each class is more accurately represented than in other popular loss functions. Moreover, the proposed loss function replaces the concept of distance between embeddings by the concept that an embedding belongs to a certain class with a given probability. This has the advantage of avoiding the selection of an appropiate distance measure and tuning extra hyper-parameters such as distance margins.
  - **To avoid the selection of an expected operating point for training a biometric system**. A new loss function is proposed for training an integration system which allows to optimize it in the operating range, instead of only one operating point, in which the biometric system is expected to work a priori.
- Integration of ASV and anti-spoofing systems
  - **To avoid the integration of ASV and anti-spoofing systems based on independent scores**. A new integration system is proposed for integrating the ASV and PAD systems in the embedding space in order to exploit the fact that ASV and PAD systems share the *bonafide* speech subspace.
- Adversarial Attacks
  - **To make the biometric system robust to adversarial attacks**. A new generator of *spoofing* adversarial attacks is proposed which fools both the ASV and PAD systems. These adversarial attacks can be used for re-training the biometric system and make it more robust.

1.4 Contributions

In this section we provide a summary of the main contributions of this thesis along with a brief discussion about the obtained results.

1.4.1 Neural Network Architectures for Voice Spoofing Detection Systems

In previous sections, we have presented the issues of current state-of-the-art anti-spoofing systems. In order to avoid cropping utterances, post-processing techniques of scores and the degradation in noisy acoustic scenarios, we propose some novel techniques.

1.4.1.1 Anti-spoofing Architectures Trained in Clean Conditions

In order to avoid the cropping of utterances and the post-processing techniques of scores in anti-spoofing systems based on DNNs or CNNs, we propose two techniques that aim to combine CNNs with RNNs so that the variable length utterances can be more appropiately processed.

- A CNN+RNN system which obtains a single embedding vector per utterance. It processes the whole utterance by dividing it into multiple chunks using overlapped context windows. Thus, it avoids the issue of applying any padding or cropping pre-processing technique so that all utterances have the same length.
- A hybrid LCNN [14], [81] and RNN which combines the ability of the LCNN for extracting discriminative features at frame level with the capacity of GRU-based RNNs for learning long-term dependencies of the subsequent deep features. The main difference with respect to the CNN+RNN system is that this hybrid architecture replaces the fully connected layers inside the recurrent cells with LCNN layers in order to:
  - Extract discriminative features at the frame level.
  - Learn long-term dependencies.
  - Integrate the extraction of frame-level deep features and the utterance-level embedding into a single network.

In terms of results, the proposed Light Convolutional Gated Recurrent Neural Network (LC-GRNN) architecture obtained one of the best single-system results in terms of EER and min-tDCF in the ASVspoof 2019 Challenge for both logical and physical access attack scenarios.

The conference papers associated to this part are:

- Alejandro Gomez-Alanis, A. M. Peinado, Jose A. Gonzalez and Angel M. Gomez, "Performance evaluation of front- and back-end techniques for ASV spoofing detection systems based on deep features", *Proc. IberSPEECH*, pp. 45-49, Barcelona, Spain, November 2018.
- Alejandro Gomez-Alanis, A. M. Peinado, Jose A. Gonzalez and Angel M. Gomez, "A Light Convolutional GRU-RNN Deep Feature Extractor for ASV Spoofing Detection", *Proc. Interspeech*, pp. 1068-1072, Graz, Austria, September 2019.

1.4.1.2 Anti-spoofing Architectures With Countermeasures for Noisy Acoustic Conditions

The performance of anti-spoofing systems is degraded when exposed to noisy and reverberant acoustic environments [41]. To address this issue, a noise robustness technique was proposed in [30] in which information about the acoustic noise (represented as an averaged embedding vector of the initial segment of the recording) is passed to the system.

The first technique which was proposed to countermeasure this problem was noise-aware training [30] using an averaged embedding vector of the initial noise of the utterance. However, this technique might be suboptimal due to the fact that it only characterizes the noise in the utterance as an average vector. Furthermore, it also requires the first frames of the utterance to be non-speech. Instead, in this thesis we propose the use of spectro-temporal missing-data masks that provide finer information about the degree each tempo-frequency bin of the speech spectrum to be distorted by noise.

- First, in order to have a finer grain detail about the reliability of each spectro-temporal region of the noisy utterance, we proposed the use of missing-data masks. This mask defines, for each spectral feature of the utterance, the probability that it is contaminated by noise. It is computed from the noise estimates obtained by using a linear interpolation of the averaged noise spectra of the first and last $T = 10$ frames of the utterance (assuming there is a short non-speech period at the beginning and at the end of the utterance). In terms of results, we show that appending the missing-data mask to the neural network performs better than appending an averaged noise embedding vector.

- The previous proposed approach, however, performs poorly in highly non-stationary noise or when there is little noise at the beginning/end of the utterance. To address this issue, we propose the extraction of signal-to-noise masks (SNMs) by estimating them with a CNN which is trained using a binary cross entropy (BCE) loss. In fact, the proposed SNM defines a family of parametric masks which include the well-known Ideal Binary Masks (IBMs) [82] and the Ideal Ratio Masks (IRMs) [83], since its hyperparameters are tunable.

The proposed techniques, based on the usage of masks for noise-aware training have been evaluated on a noisy version of the ASVspoof 2015 database [40], and it is shown that these techniques perform significantly better in terms of EER than using embedding vectors with the initial averaged noise of the utterance for noise-aware training.

The conference and journal papers associated to this part are:

- Alejandro Gomez-Alanis, A. M. Peinado, Jose A. Gonzalez and Angel M. Gomez, "A Deep Identity Representation for Noise Robust Spoofing Detection", *Proc. Interspeech*, pp. 676-680, Hyderabad, India, September 2018.

- Alejandro Gomez-Alanis, A. M. Peinado, Jose A. Gonzalez and Angel 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, December 2019.

1.4.2 Loss Functions for Voice Biometric Neural Networks

Training a DNN for speech anti-spoofing is a challenging task, since the DNN should be able to learn to differentiate *bonafide/spoofed* speech from the training dataset, but also being able to generalize well to unseen attacks. The standard cross-entropy loss function falls short in achieving this goal. Other loss functions have been proposed to avoid this issue such as the angular-softmax loss [84], [85]. In this thesis we also propose two loss functions for training biometric systems.

1.4.2.1 Loss Function for ASV-Spoofing Detection

In this work, we propose a new loss function based on kernel density estimation whose main characteristics are:

- Every training class is represented with a probability density function (pdf) which is estimated through kernel density estimation. Thus, all the samples of each class from the mini-batch are represented by a pdf rather than representing each class with a sole point (cendroid [49] or anchor point [54]).

- The training classes are fully represented by all the samples within the mini-batch, by estimating with KDE a pdf per class which places a probability mass at every embedding sample.
- The proposed loss function replaces the concept of distance between embeddings by the concept that an embedding vector belongs to a certain class with a given probability.
- We show that the proposed loss function is a generalization of both the GE2E loss [49] and the triplet loss [54].

Experimental results on the ASVspoof 2019 database show that the proposed loss function outperforms other conventional loss functions that have been used so far for training DNN-based anti-spoofing systems.

The journal paper associated to this part is:

- Alejandro Gomez-Alanis, Jose 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. 108530-108543, June 2020.

1.4.2.2 Loss Function for Biometric Systems

Most biometric integration systems calibrate the decision thresholds for the ASV and PAD systems considering only one point of the error rate on the development dataset. However, we believe that it is difficult to know a priori the ideal operating point in which the integration system is going to work, since the evaluation data is usually unseen and does not match the development data.

On the other hand, most integration systems of ASV and PAD systems are based on scores computed separately. However, this type of integration might be suboptimal since the integration system is not optimized exploiting the fact that ASV and PAD systems share the *bonafide* speech subspace.

In this thesis we have made the following two contributions in this regard:

- A new integration neural network which processed the embeddings extracted by the ASV and PAD systems jointly is proposed. Thus, the integration systems is able to model the three main biometric speech subspaces: *bonafide* speech, *zero-effort* attacks and *spoofing* attacks.
- To train the integration neural network, we proposed a new loss function which minimizes the area under the expected (AUE) [69] performance and spoofability curve (EPSC) [69]. This loss function allows to optimize the integration system in the operating range which it is expected to work a priori, instead of optimizing in one sole operating point.

Experimental results on the ASVspoof 2019 corpus show that the joint processing of the ASV and PAD embeddings with the proposed integration neural network clearly outperforms other state-of-the-art techniques trained on the same conditions. Specifically, our proposal achieves around 23.6% and 22.0% of relative EER improvement over the best performing baseline in logical and physical access scenarios, respectively, as well as relative gains around 27.6% and 29.2% on the AUE metric. Moreover, the proposed loss function also achieves up to 22.2% and 20.8% relative

joint EER improvement over the standard cross-entropy loss in both logical and physical access evaluation scenarios, respectively.

The journal paper associated to this part is:

- Alejandro Gomez-Alanis, Jose A. Gonzalez-Lopez and A. M. Peinado, "On Joint Optimization of Automatic Speaker Verification and Anti-spoofing in the Embedding Space", *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 1579-1593, November 2020.

1.4.3 Adversarial Examples for Voice Biometric Systems

Adversarial attacks can easily fool DNN-based models by perturbing benign samples in a way normally imperceptible to humans. Voice biometric systems have been shown to be specially sensitive to these attacks [70], [71].

In this work, we evaluate the robustness of a state-of-the-art voice biometric system under presence of adversarial *spoofing* attacks and propose a new generator which can be used in the future for re-training the system and make it more robust to this type of attacks. The main contributions of this work are:

- Investigate the robustness of voice biometric systems under the presence of adversarial *spoofing* attacks. Although these type of attacks have only being studied on logical access scenarios (TTS and VC based *spoofing* attacks), we also study on physical access scenarios (replay based *spoofing* attacks).
- Propose an adversarial biometrics transformation network (ABTN) for both white-box and black-box scenarios which is able to generate adversarial *spoofing* attacks in order to fool the PAD system without being detected by the ASV system.

Experimental results show that voice biometric systems are highly sensitive to adversarial *spoofing* attacks in both logical and physical access scenarios. Moreover, the proposed ABTN system clearly outperforms other popular adversarial attacks techniques such as the fast gradient signed method (FGSM) [86] and the projected gradient descent (PGD) [87] in both white-box and black-box scenarios.

The conference paper associated to this part is:

- Alejandro Gomez-Alanis, Jose A. Gonzalez-Lopez and A. M. Peinado, "Adversarial Transformation of Spoofing Attacks for Voice Biometrics", *Proc. IberSPEECH*, pp. 255-259, Valladolid, Spain, March 2021.

Chapter 2

Publications: Published and Accepted Papers

2.1 Neural Network Architectures for Anti-spoofing Systems

The papers associated to this part are:

2.1.1 Anti-spoofing Architectures Trained in Clean Acoustic Conditions

2.1.1.1 Performance Evaluation of Front- and Back-end Techniques for ASV Spoofing Detection Systems based on Deep Features

- Alejandro Gomez-Alanis, A. M. Peinado, Jose A. Gonzalez and Angel M. Gomez, "Performance evaluation of front- and back-end techniques for ASV spoofing detection systems based on deep features", *Proc. IberSPEECH*, pp. 45-49, Barcelona, Spain, November 2018.
    - Status: Published.

Performance evaluation of front- and back-end techniques for ASV spoofing detection systems based on deep features

Alejandro Gomez-Alanis[1], Antonio M. Peinado[1], Jose A. Gonzalez[2], and Angel M. Gomez[1]

[1]University of Granada, Granada, Spain
[2]University of Malaga, Malaga, Spain
{agomezalanis,amp,amgg}@ugr.es, jgonzalez@uma.es

Abstract

As Automatic Speaker Verification (ASV) becomes more popular, so do the ways impostors can use to gain illegal access to speech-based biometric systems. For instance, impostors can use Text-to-Speech (TTS) and Voice Conversion (VC) techniques to generate speech acoustics resembling the voice of a genuine user and, hence, gain fraudulent access to the system. To prevent this, a number of anti-spoofing countermeasures have been developed for detecting these high technology attacks. However, the detection of previously unforeseen spoofing attacks remains challenging. To address this issue, in this work we perform an extensive empirical investigation on the speech features and back-end classifiers providing the best overall performance for an antispoofing system based on a deep learning framework. In this architecture, a deep neural network is used to extract a single identity spoofing vector per utterance from the speech features. Then, the extracted vectors are passed to a classifier in order to make the final detection decision. Experimental evaluation is carried out on the standard ASVSpoof2015 data corpus. The results show that classical FBANK features and Linear Discriminant Analysis (LDA) obtain the best performance for the proposed system.

Index Terms: Automatic speaker verification, spoofing detection, deep neural networks, features, classifier.

1. Introduction

Automatic Speaker Verification (ASV) aims to authenticate the identity claimed by a given individual [1]. However, most ASV systems are vulnerable to spoofing attacks, in which an impostor try to gain fraudulent access to the system by presenting to the ASV system speech acoustics 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 aim of this work is to develop robust anti-spoofing countermeasures for either VC or TTS based attacks.

The performance of anti-spoofing systems can meaningfully vary depending on the voice features used to feed them. Due to this, voice features have attracted the attention of a number of researchers [8, 9, 10]. However, anti-spoofing systems based on neural networks usually use classical voice features, such as FBANKs, and to the best of our knowledge, the new popular CQCC features have not been employed yet to feed these types of systems.

In the last years, the technique of deep features extraction have been explored to obtain more discriminative and effective features for spoofing detection [6, 7, 11]. This technique consists of employing deep neural networks in the front-end of the anti-spoofing system which are fed by speech features, so that the deep features extracted by the neural network are passed to a classifier in order to make the final detection decision (genuine or spoof). The core idea is to take advantage of the nonlinear modeling and discriminative capabilites of deep neural networks which have shown to be suitable for feature engineering [3], not only for spoofing detection, but also for speech recognition [4], speaker recognition [3], and speech synthesis [5].

In this work, we compare the performance of different features and back-ends in anti-spoofing system which extracts deep features [6] in order to detect VC and TTS attacks. This anti-spoofing system employs a convolutional neural network (CNN) plus a recurrent neural network (RNN) and gets a single spoofing identity representation per utterance. Although a similar comparison has already been studied in [7], our study presents three important differences: (1) our anti-spoofing system employs a CNN to extract convolutional features at the speech frame level, (2) we compare the performance of classical features, such as FBANKs and MFCCs, with the performance of the recent popular CQCC features [8], and (3) we combinate different features and classifiers in order to find the combination which offers the best performance.

This paper is organized as follows. Section 2 describes the features and back-ends we are going to compare in a CNN + RNN anti-spoofing system. Then, in Section 3, we outline the speech corpora, the network training, and the performance evaluation details. Section 4 discusses the results of the different features and back-ends in the deep neural network based anti-spoofing system. Finally, we present the conclusions derived from this research in Section 5.

2. System description

This section is devoted to the description of the anti-spoofing system. First, Section 2.1 describes different voice features: FBANK, MFCC and CQCC. The neural network architecture for deep feature extraction is detailed in Section 2.2. Furthermore, Section 2.3 describes different classifiers (back-ends): Linear Discriminant Analysis (LDA), Support Vector Machine (SVM), and One-Class Support Vector Machine (One-Class SVM).

2.1. Speech features

As demonstrated in [11], traditional log MEL filterbank features (FBANK) are effective for detecting spoofing attacks with systems based on neural networks. These features are obtained by passing the Short Time Fourier Transform (STFT) magnitude spectrum through a Mel-filterbank and applying a log opera-

tion. However, FBANK features are usually high-correlated. One way to decorrelate these features is to apply the Discrete Cosine Transform (DCT) to get the classical Mel Frequency Cepstral Coefficient (MFCC) features.

In [8], CQCC features are proposed for spoofing detection, which are obtained using the Constant Q Transform (CQT). The Q factor is a measure of the selectivity of each filter and is defined as the ratio between the center frequency and the bandwidth of the filter. In contrast to the STFT, whose Q factor increases when moving from low to high frequencies as the bandwidth is the same for all filters, the bandwidth of the filters employed in the CQT is not constant, and this results in getting a higher frequency resolution for low frequencies and a higher temporal resolution for high frequencies. In this manner, the CQCC features try to imitate the human perception system which is known to approximate a constant Q factor between 500Hz and 20kHz [20].

In this work, we employ the classical FBANK and MFCC features, as well as the popular CQCC features, to feed the anti-spoofing system.

2.2. Front-end

The front-end architecture of the anti-spoofing system is shown in Fig. 1. A context window of W frames (centered at the frame being processed) is used to obtain the input signal spectral features which are fed into the system. Then, the CNN provides a deep feature vector per window, and all deep features vectors of the considered utterance are processed by the RNN which computes an embedding vector for the whole utterance. We call this the spoofing identity vector. Since the front-end is trained to perform utterance-level classification of the attacks, this embedding vector should provide more discriminative information for spoofing detection than the raw speech features.

In this architecture, the CNN plays the role of a frame-level deep feature extractor providing one feature vector for each context window of W frames. In order to this, the CNN acts as a classifier whose task consists of determining whether the input feature are either genuine or belong to one of the K spoofing attacks (S1, S2, ..., SK) present in the training set. This CNN uses 2 convolutional and pooling layers as feature extractors, followed by 2 fully connected layers with a softmax layer of $K+1$ neurons as classification layer. To prevent overfitting, we used an annealed dropout training procedure [17]. In annealed dropout, the dropout probability of the nodes in the network is decreased as training progresses. In this work, the annealed function reduces the dropout rate from an initial rate $prob[0]$ to zero over N steps with constant rate. The dropout probability $prob[t]$ at epoch t is given as:

$$prob[t] = max\Big(0, 1 - \frac{t}{N}\Big) prob[0]. \quad (1)$$

As shown in Fig. 1, the deep features obtained from the CNN are fed into an RNN, which computes the anti-spoofing identity vector of the utterance. The main advantage of using an RNN, based on gated recurrent units (GRU) [16], is its ability for learning the long-term dependencies of the subsequent deep feature vectors. Finally, a fully connected layer containing $K+1$ neurons (one per class: genuine, S1, S2, ..., SK) is connected to the output of the last time step, followed by a softmax layer. The state of the last time step represents the single identity spoofing vector of the whole utterance.

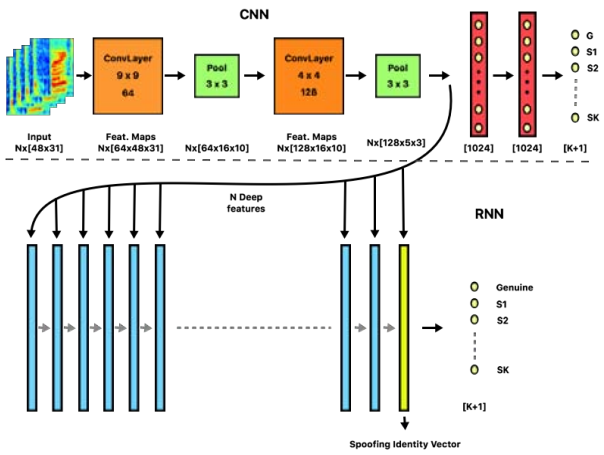

Figure 1: *Front-end architecture of the anti-spoofing system which extracts a spoofing identity vector per utterance (N represents the number of context windows per utterance). This system is proposed in [6].*

2.3. Back-end

After deep feature extraction, every utterance is represented by a single spoofing identity vector. A back-end classifier is then applied on these vectors to do the final detection decision. In this section three classifiers will be tested: LDA, SVM and one-class SVM.

A. Linear Discriminant Analysis

LDA assumes that each class density can be modeled as a multivariate Gaussian

$$N(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \frac{1}{(2\pi)^{\frac{p}{2}}|\boldsymbol{\Sigma}_k|^{\frac{1}{2}}} e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_k)}, \quad (2)$$

where $\boldsymbol{\Sigma}_k$ and $\boldsymbol{\mu}_k$ is the covariance and mean for class k, and p is the dimension of the identity vectors. Moreover, the LDA model assumes every class shares the same covariance, that is, $\boldsymbol{\Sigma}_k = \boldsymbol{\Sigma}, \forall k$. The goal of LDA is to find a transformation which maximizes the distance between classes while minimizing the spreading within each class. This can be formulated as a diagonalization problem where matrix $\boldsymbol{\Sigma}_b\boldsymbol{\Sigma}$ ($\boldsymbol{\Sigma}_b$ is the between-class covariance) is diagonalized, so the transformation can be built from the resulting eigenvectors.

Our LDA classifier uses $K+1$ classes which represent genuine speech and the K known spoofing attacks considered in the training set. In this way, the LDA assigns a genuine speech confidence score to each utterance, which is then used for binary decision (spoof or genuine) during the evaluation.

B. Support Vector Machine

A support vector machine (SVM) separates data points in a high dimensional space defined by a kernel function. In this manner, we first obtain a binary function that describes the probability density function where the genuine data lives. This function returns +1 in the small region corresponding to the genuine speech data and -1 elsewhere. Thus, the core idea of SVM is to estimate the hyperplane with the largest separation margin between the two classes.

Table 1: *Structure of the ASVspoof2015 data corpus divided by the training, development and evaluation sets [14].*

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

In this work, this classifier is used to classify the spoofing identity vectors obtained by the front-end system, where +1 indicates genuine speech and -1 indicates spoofed speech.

C. One-Class Support Vector Machine

Complex classifiers may overfit the training spoofed data. To create a spoof-independent system, we also test a derivative model that can only be trained on genuine speech data. This is a type of one-class SVMs [12], usually employed to find abnormal data. This was first tried in spoofing detection with phase-based features in [13]. This kind of SVM is also applied here to classify the spoofing identity vectors, and only genuine speech data has been used to train the one-class SVM model.

3. Experimental framework

To evaluate the performance of several features and back-ends in an anti-spoofing system based on neural networks, the ASVspoof 2015 dataset [14], a standard data corpus for research on spoofing detection, was employed. Details about the methodology followed for training and testing are also given in this section.

3.1. Speech corpus

The ASVspoof 2015 corpus [14] defines three datasets (training, development and evaluation), each one containing a mix of genuine and spoofed speech. The structure of these three datasets are shown in Table 1. Spoofing attacks were generated either by TTS or VC. A total of 10 types of spoofing attacks (S1 to S10) are defined: three of them are implemented using TTS (S3, S4 and S10), and the remaining seven ones (S1, S2, S5, S6, S7, S8 and S9) using different VC 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 [14].

3.2. Spectral Analysis

The frame window size is 25 ms with 10 ms of frame shift. Moreover, the size of the context window is $W = 31$ frames, and the number of filters used to get the spectral features is $M = 48$ filters. In contrast to [7] and [11], we use a 48-dim static spectral features without delta and acceleration coefficients, as we have realized that the context window of 31 frames is already exploiting the correlations between consecutive frames. Therefore, a higher spectral resolution is achieved while the size of the spectral feature vector is smaller than in [7].

3.3. Training

The CNN and RNN networks are trained using Adam optimizer [18]. As there are $K = 5$ known spoofing attacks in the data corpus, the softmax layer of both CNN and RNN contains $K + 1 = 6$ neurons (one per class). The two fully connected layers of the CNN have 1024 sigmoid neurons, and the layer of the RNN has 1920 GRUs, which is the length of the identity spoofing vector of the whole utterance. To prevent the problem of overfitting, the initial dropout probabilities are 50% and 40% from the first to the last fully connected layer, respectively. Also, early stopping is applied in order to stop the training process when no improvement of the cross entropy is obtained after 15 iterations. All the specified parameters of the system have been optimized using the validation set of the data corpus [14].

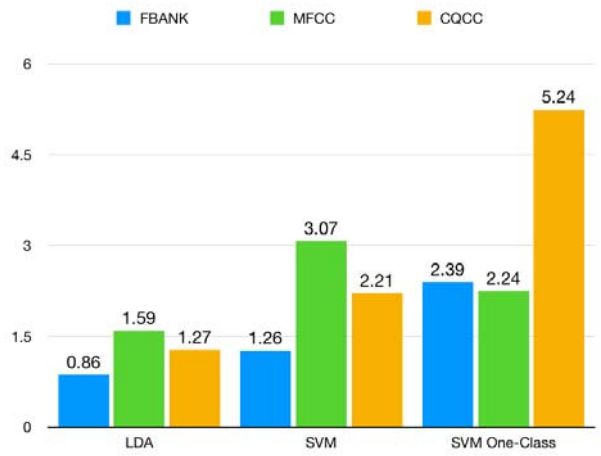

Figure 2: *Comparison of average EERs (%) between known and unknown spoofing attacks on evaluation dataset for different features and back-ends, including FBANK, MFCC, CQCC, LDA, SVM and SVM One-Class.*

3.4. Performance evaluation

The equal error rate (EER) is used to evaluate the system performance. As described in the ASVspoof 2015 challenge evaluation plan [14], the EER was computed independently for each spoofing algorithm and then the average EER across all attacks was used. To compute the average EER, we used the Bosaris toolkit [15].

4. Results

4.1. Comparison of features and back-ends

Table 2 shows the detailed results of the different features (FBANK, MFCC and CQCC) and classifiers (LDA, SVM and SVM One-Class) in the described CNN + RNN anti-spoofing system. Furthermore, a summary of these results is shown in Fig. 2. The best performance is obtained with the combination of FBANK features and the LDA classifier. In average, the FBANK features obtain the best performance independently of the back-end, although MFCC features perform better on the SVM One-Class considering all the attacks. The CQCC features achieve the best average performance in the known attacks with LDA and SVM back-ends, but these two combinations perform very poorly in the S10 attack.

Regarding the back-ends, the LDA outperforms the other 2 classifiers in the known and unknown attacks. Moreover, the binary SVM classifier performs much better than SVM One-Class using FBANK and CQCC features.

Table 2: *Comparison on evaluation dataset for each spoofing attack in terms of (%) EER*

Features	Back-end	Known Attacks						Unknown Attacks						Total
		S1	S2	S3	S4	S5	Avg.	S6	S7	S8	S9	S10	Avg.	Avg.
FBANK	LDA	**0.01**	0.09	**0.00**	**0.00**	0.11	0.04	0.66	0.21	**0.00**	0.36	**7.16**	**1.68**	**0.86**
	SVM	0.03	0.13	**0.00**	0.01	0.22	0.08	0.77	0.34	0.18	0.48	10.46	2.44	1.26
	SVMOne	0.36	2.07	0.17	0.12	4.37	1.42	5.44	1.34	0.34	1.53	8.23	3.38	2.40
MFCC	LDA	0.06	0.08	**0.00**	**0.00**	0.06	0.04	0.11	**0.12**	**0.00**	0.05	15.43	3.14	1.59
	SVM	0.05	0.19	0.01	0.01	0.23	0.10	0.22	0.21	0.05	0.15	29.58	6.04	3.07
	SVMOne	0.43	1.97	0.12	0.12	2.11	0.95	3.38	2.07	0.06	1.03	11.09	3.53	2.24
CQCC	LDA	0.04	0.04	**0.00**	**0.00**	0.04	0.02	0.13	0.51	0.05	0.08	11.76	2.51	1.27
	SVM	0.03	**0.01**	0.01	0.01	**0.02**	**0.01**	**0.06**	0.37	0.07	**0.02**	21.52	4.41	2.21
	SVMOne	1.72	6.14	0.49	0.47	7.34	3.23	10.13	9.67	1.39	6.50	8.54	7.25	5.24

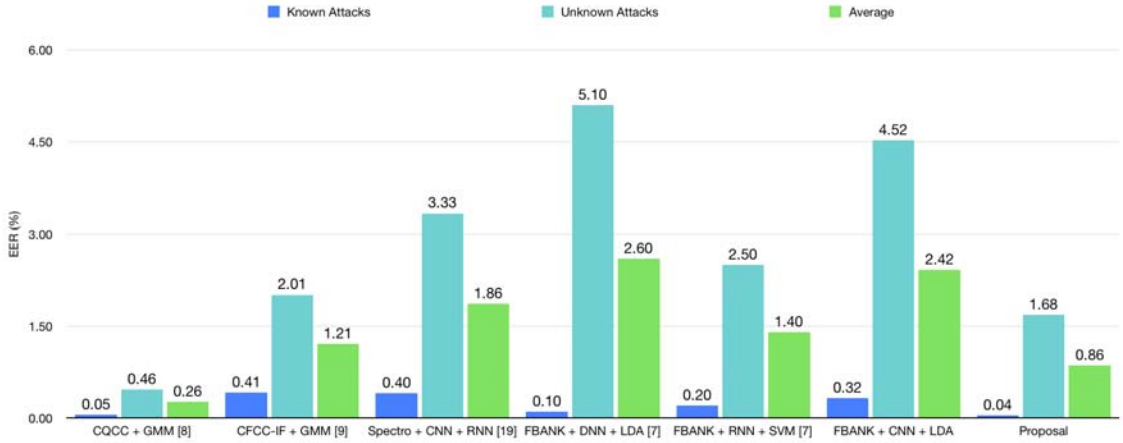

Figure 3: *Comparison on evaluation dataset for known and unknown spoofing attacks in terms of average (%) EER*

According to these results, we propose an anti-spoofing system which employs FBANK features, a CNN + RNN architecture to get the spoofing identity vector of an utterance, and an LDA classifier to make the final detection decision (spoof or genuine).

4.2. Comparative performance

A comparison of our proposal with other popular techniques from the literature are presented in Fig. 3.

The first two systems CQCC + GMM [8] and CFCC-IF + GMM [9] employ the features which perform best for spoofing detection using a Gaussian Mixture Model (GMM) as back-end. The other four systems are the most popular anti-spoofing systems based on deep learning frameworks. The FBANK + CNN + LDA system has been proposed in [11], but as its performance is not provided in this reference for the clean scenario, we have evaluated instead our proposed system removing the RNN and averaging the deep features for getting the spoofing identity vector of the utterance as in [11].

The CQCC + GMM system achieves the best average performance, although our proposed system (FBANK + CNN + RNN + LDA) achieves the best results for the known attacks. Compared to the rest of deep learning systems (Spectro + CNN + RNN [19], FBANK + DNN + LDA [7], FBANK + RNN + SVM [7] and FBANK + CNN + LDA), our proposal outperforms all of them for the known and unknown attacks. In particular, the result of our proposal for the S10 attack is quite noteworthy. Furthermore, our proposed system also achieves a lower EER in almost all attacks than the CFCC-IF system [9], performing 0.45% better on average when considering all the attacks.

Despite that the CQCC + GMM system outperforms all the systems in a clean condition training scenario, our previous work [6] and reference [11] demonstrate that CQCC + GMM suffers from a drastic performance reduction in noisy scenarios. In this way, our proposed system, based on a deep learning framework, outperforms CQCC + GMM in the clean evaluation dataset when using multi-condition training, which is a more realistic scenario.

5. Conclusions

This paper has evaluated different features and classifiers in order to find the combination which offers the best performance for an anti-spoofing system based on a deep learning framework. The experimental results have shown that FBANK features and an LDA obtain the best performance for systems based on the extraction of deep features, rather than the popular CQCC features and other types of classifiers, such as binary SVM and SVM One-Class. Furthermore, the proposed system (FBANK + CNN + RNN + LDA) outperforms the rest of deep learning systems of the literature.

6. Acknowledgements

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.

7. References

[1] Z. Wu, N. Evans, T. Kinnunen, J. Yamagishi, F. Alegre, and H. Li, "Spoofing and countermeasures for speaker verification: A survey," in *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] Liu, Y., Qian, Y., Chen, N., Fu, T., Zhang, Y., Yu, K., "Deep feature for text-dependent speaker verification", in *Speech Communication*, vol. 13, pp. 1–13, 2015.

[4] Grézl, F., Karafiát, M., Kontár, S., Cernocky, J., "Probabilistic and bottle-neck feature for LVCSR of meetings," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2007, pp. 757–760.

[5] Wu, Z., King, S., "Improving trajectory modelling for dnn-based speech synthesis by using stacked bottleneck features and minimum trajectory error training," in *IEEE/ACM Trans. on Audio, Speech and Language Processing*, vol. 24, pp. 1255–1265, 2016.

[6] Alejandro Gomez-Alanis, Antonio M. Peinado, Jose A. Gonzalez, and Angel M. Gomez, "A Deep Identity Representation for Noise Robust Spoofing Detection," in *Proc. InterSpeech*, 2018.

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

[8] M. Todisco, H. Delgado, and N. Evans, "A new feature for automatic speaker verification anti-spoofing: Constant Q cepstral coefficients," *Proc. Odyssey*, 2016, pp. 249–252.

[9] 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, pp. 2062–2066.

[10] Muckenhirn, H., Korshunov, P., Magimai-Doss, M., Marcel, S., "Long-Term Spectral Statistics for Voice Presentation Attack Detection," in *IEEE/ACM Trans. on Audio, Speech and Language Processing*, vol. 25, pp. 2098–2111, 2017.

[11] 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.

[12] Scholkopf, B., Williamson, R. C., Smola, A. J., et al., "Support vector method for novelty detection," in *Proc. NIPS*, 2000, pp. 582–588.

[13] Jesús Villalba, Antonio Miguel, Alfonso Ortega, and Eduardo Lleida, "Spoofing detection with dnn and one-class svm for the asvspoof 2015 challenge," in *Proc. InterSpeech*, 2015, pp. 2067–2071.

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

[15] N. Brümmer and E. deVilliers, "The BOSARIS toolkit: Theory, algorithms and code for surviving the new DCF," in *NIST SRE11 Speaker Recognition Workshop*, Atlanta, Georgia, USA, Dec. 2011, pp. 1–23.

[16] Kyunghyun Cho, et al., "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation," *Proc. Empirical Methods in Natural Language Processing*, 2014, pp. 1724–1734.

[17] S. Rennie, V. Goel, and S. Thomas, "Annealed dropout training of deep networks," in *Proc. Spoken Language Technology Workshop*, 2014, pp. 159–164.

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

[19] Chunlei Zhang, Chengzhu Yu, and John 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.

[20] B. C. J. Moore, "An Introduction to the Psychology of Hearing", BRILL, 2003.

2.1.1.2 A Light Convolutional GRU-RNN Deep Feature Extractor for ASV Spoofing Detection

- Alejandro Gomez-Alanis, A. M. Peinado, Jose A. Gonzalez and Angel M. Gomez, "A Light Convolutional GRU-RNN Deep Feature Extractor for ASV Spoofing Detection", *Proc. Interspeech*, pp. 1068-1072, Graz, Austria, September 2019.
    - Status: Published.

A Light Convolutional GRU-RNN Deep Feature Extractor for ASV Spoofing Detection

Alejandro Gomez-Alanis[1], Antonio M. Peinado[1], Jose A. Gonzalez[2], and Angel M. Gomez[1]

[1]University of Granada, Granada, Spain
[2]University of Malaga, Malaga, Spain
{agomezalanis,amp,amgg}@ugr.es, j.gonzalez@uma.es

Abstract

The aim of this work is to develop a single anti-spoofing system which can be applied to effectively detect all the types of spoofing attacks considered in the ASVspoof 2019 Challenge: text-to-speech, voice conversion and replay based attacks. To achieve this, we propose the use of a Light Convolutional Gated Recurrent Neural Network (LC-GRNN) as a deep feature extractor to robustly represent speech signals as utterance-level embeddings, which are later used by a back-end recognizer which performs the final genuine/spoofed classification. This novel architecture combines the ability of light convolutional layers for extracting discriminative features at frame level with the capacity of gated recurrent unit based RNNs for learning long-term dependencies of the subsequent deep features. The proposed system has been presented as a contribution to the ASVspoof 2019 Challenge, and the results show a significant improvement in comparison with the baseline systems. Moreover, experiments were also carried out on the ASVspoof 2015 and 2017 corpora, and the results indicate that our proposal clearly outperforms other popular methods recently proposed and other similar deep feature based systems.

Index Terms: spoofing detection, automatic speaker verification, deep learning, ASVspoof.

1. Introduction

Automatic Speaker Verification (ASV) aims to authenticate the identity claimed by a given individual based on the provided speech samples [1]. This technology has gained significant interest in recent years due to its commercial applications. As the importance of this technology grows, so does the concerns about its security. 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 aim of this work is the development of a common framework aiming at detecting different spoofing attacks, namely TTS, VC and replay attacks.

One popular approach for spoofing detection is to deploy machine learning techniques in order to learn to discriminate genuine vs. spoofed speech, using a training dataset. It is desirable that the anti-spoofing system learns to detect not only the attacks observed in the training dataset, but also be able to generalize to unseen attacks. To address this issue, deep feature extraction has been proposed in [3], where feature embeddings 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.

Deep neural networks have shown to be very effective for feature engineering in several speech-based applications [4]. Their nonlinear modeling and discriminative capabilities make them not only a powerful back-end classifier [5, 6], but also advantageous for feature extraction [7]. The architecture of these deep feature extractors has shown to be determinant for the performance of the anti-spoofing system.

This paper presents a novel neural network architecture for ASV-based spoofing detection. We propose a hybrid light convolutional neural network (LCNN) [8] plus recurrent neural network (RNN) architecture which combines the ability of the LCNNs for extracting discriminative features at frame level with the capacity of gated recurrent unit (GRU) based RNNs for learning long-term dependencies of the subsequent deep features. The resulting architecture will be referred to as Light Convolutional Gated Recurrent Neural Network (LC-GRNN). Despite the fact that similar deep learning frameworks have been applied in learning video representations [9], audio tagging [10] and optical character recognition [11], to the best of our knowledge our work constitutes the first adaptation of such architecture to the problem of spoofing detection.

Our system has participated in the ASVspoof 2019 Challenge, whose results will be presented at a special session of Interspeech 2019, to address the issue of detecting: (i) logical access attacks (generated by TTS or VC algorithms), and (ii) physical access attacks (replay). Furthermore, we also evaluate our proposal on the ASVspoof 2015 [12] and 2017 [13] datasets in order to provide a comparison with other state-of-the-art systems.

This paper is organized as follows. Section 2 describes the proposed deep feature extractor employed along the work. Then, in Section 3, we outline the speech corpora, the network training and the system details. Section 4 discusses the performance of our system on the ASVspoof 2015 [12], 2017 [13] and 2019 [14] databases. Finally, we summarize the conclusions derived from this research in Section 5.

2. System Description

In our previous work [15], we proposed a hybrid CNN-RNN architecture to compute spoofing embeddings at utterance level. When evaluated on standard spoofing databases, our architecture was able to outperform other similar deep feature extractors which average frame-level features for getting the spoofing identity vector of the utterance.

Since our preliminary system of [15], our work has focused on building a better integration of convolutional and recurrent layers. In particular, in this work we take one step forward and propose to replace the fully connected layers inside the recurrent cells with LCNN layers in order to: (1) extract discriminative features at frame level, (2) learn long-term dependencies,

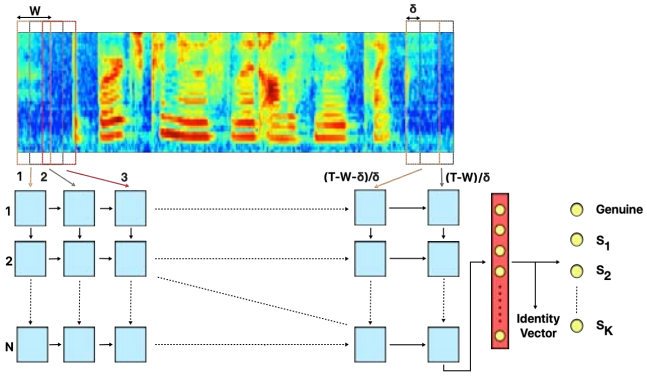

Figure 1: *Block diagram of the proposed LC-GRNN utterance-level identity vector extractor.*

and (3) integrate the extraction of frame-level deep features and the utterance-level identity vector into a single network. The Light term of the convolutional layers stands for the usage of Max-Feature-Map (MFM) activations. These are applied to reduce the dimension of the output and obtain more discriminative feature maps, as it has been shown for face recognition [8] and replay spoofing detection [16].

A block diagram of the proposed LC-GRNN architecture is shown in Fig. 1. At each time step, the LC-GRNN processes a context window of W consecutive frames. This context window moves forward δ frames on every time step[1], so that the total number of time steps of the LC-GRNN is $(T-W)/\delta$, where T is the number of frames of the utterance being processed. Moreover, the LC-GRNN has N recurrent layers. This architecture acts as a classifier whose task is to determine whether the input utterance is either genuine or belongs to one of the K spoofing attacks included in the training set $(S_1, S_2, ..., S_K)$. In order to do this, the output of the last time step and last recurrent layer is fed to a fully connected layer with MFM activation to obtain the spoofing identity vector of the whole utterance. During the training phase, this identity vector is finally passed through another fully connected layer with softmax activation of $K+1$ neurons to discriminate between the genuine and the K spoofing classes.

Unlike classical RNNs, the hidden state $\mathbf{h}_t^n$ ($t = 1, ..., (T-W)/\delta$; $n = 1, ..., N$) of the LC-GRNN model is computed by convolving the current input features $\mathbf{x}_t^n$ and the previous state $\mathbf{h}_{t-1}^n$ with multiple filters. Due to the fact that most of the cues that enable the detection of spoofing attacks can be found in certain frequency bands [17], we embed such a prior in our deep feature extractor 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 [18].

As shown in Fig. 2, similarly to a GRU cell, our LC-GRU cell defines three gates, each one implemented by means of a LCNN. Each LCNN block in Fig. 2 consists of either one or two LCNN layers as shown in Fig. 3. Every LCNN layer performs convolutions (one or two) followed by an MFM operation intended to reduce the output feature maps by a $1/2$ factor via

[1]Instead of moving forward the context window one frame on every time step, we propose to move it δ frames ($\delta <$ W) in order to reduce the processing time, while maintaining the classification performance.

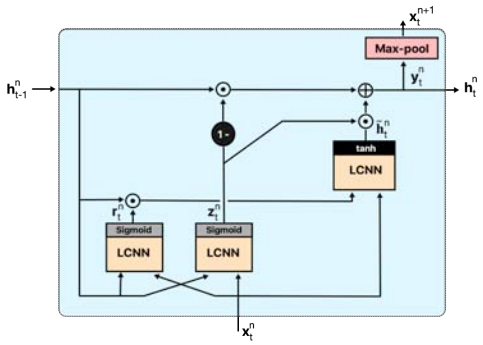

Figure 2: *Light Convolutional Gated Recurrent Unit cell (LC-GRU).*

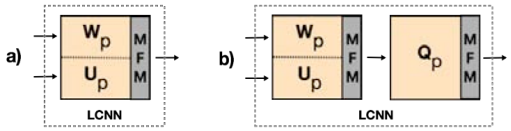

Figure 3: *Possible LCNN block configurations inside the LC-GRU cell. a) 1-layer LCNN, b) 2-layer LCNN. ($p = r, z, \tilde{h}$).*

a competitive relationship [8]. In this way, each time step of the LC-GRNN plays the role of a frame-level deep feature extractor providing N state (feature) vectors for each context window of W consecutive frames.

The update $\mathbf{z}_t^n$ and reset $\mathbf{r}_t^n$ gates determine which information from the previous frames needs to be passed along the next time steps, avoiding the risk of the vanishing gradient problem [19]. In the case of a single-layer LCNN block (Fig. 3a), they are computed as

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

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

where the operator $*$ denotes a convolution operation. These convolutional layers can be interpreted as filter banks which are trained and optimized to detect artifacts from the spoofed speech. The main advantage 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 [20]. Similarly, the update activation gate

$$\tilde{\mathbf{h}}_t^n = tanh(\text{MFM}(\mathbf{W}_{\tilde{h}}^n * \mathbf{x}_t^n + \mathbf{U}_{\tilde{h}}^n * (\mathbf{r}_t^n \odot \mathbf{h}_{t-1}^n))), \quad (3)$$

uses the reset gate to store the relevant information from the past frames, removing firstly the non-relevant information through an element-wise multiplication (denoted as $\odot$) with the previous state. In these equations, MFM$(\cdot)$ is the Max-Feature-Map function, and $\sigma(\cdot)$, $tanh(\cdot)$ are the activation functions of the gates. The model parameters $\mathbf{W}_z^n$, $\mathbf{W}_r^n$, $\mathbf{W}_{\tilde{h}}^n$, $\mathbf{U}_z^n$, $\mathbf{U}_r^n$, $\mathbf{U}_{\tilde{h}}^n$, and $\mathbf{Q}_z^n$, $\mathbf{Q}_r^n$, $\mathbf{Q}_{\tilde{h}}^n$ are the filters of the 3 described LCNNs, which are shared in each time step of the LC-GRNN.

3. Experimental Framework

This section briefly describes the databases employed in our experiments, as well as the details of the proposed system.

Table 1: *LC-GRNN architecture ($p = r, z, \tilde{h}$).*

LC-GRNN	Type	Filter / Stride	Output
Layer 1	Conv ($\mathbf{W}_p^1$, $\mathbf{U}_p^1$)	5×5 / 1×1	$16 \times 256 \times 32$
	MFM	-	$8 \times 256 \times 32$
	MaxPool	2×1 / 2×1	$8 \times 128 \times 32$
Layer 2	Conv ($\mathbf{W}_p^2$, $\mathbf{U}_p^2$)	1×1 / 1×1	$16 \times 128 \times 32$
	MFM	-	$8 \times 128 \times 32$
	Conv ($\mathbf{Q}_p^2$)	3×3 / 1×1	$32 \times 128 \times 32$
	MFM	-	$16 \times 128 \times 32$
	MaxPool	2×1 / 2×1	$16 \times 64 \times 32$
Layer 3	Conv ($\mathbf{W}_p^3$, $\mathbf{U}_p^3$)	1×1 / 1×1	$32 \times 64 \times 32$
	MFM	-	$16 \times 64 \times 32$
	Conv ($\mathbf{Q}_p^3$)	3×3 / 1×1	$16 \times 64 \times 32$
	MFM	-	$8 \times 64 \times 32$
	MaxPool	2×1 / 2×1	$8 \times 32 \times 32$
-	FC1	-	512×2
	MFM	-	512
-	FC2	-	$K + 1$

3.1. Speech Corpora

We evaluated the proposed anti-spoofing system on both logical access (LA) and physical access (PA) attacks with the corpora described below.

3.1.1. Logical Access

A total of 10 and 19 TTS/VC attacks were generated for the ASVspoof 2015 [12] and 2019 LA [14] databases, respectively. However, only $K = 5$ and $K = 6$ attacks have been employed for training the ASVspoof 2015 and 2019 LA models, respectively, since the rest of attacks belong to the evaluation sets (unknown attacks).

3.1.2. Physical Access

The replay attacks of the ASVspoof 2017 version 1 [13] were generated using 3 categories (low, medium, high) of recording and playback devices. For a balanced training, we have considered $K = 4$ types of replay attacks as a result of combining low/medium and high qualities of both playback and recording devices.

On the other hand, ASVspoof 2019 PA [14] database includes a total of 9 different replay configurations, comprising 3 categories of attacker-to-speaker recording distances, and 3 categories of loudspeaker quality. The evaluation data was generated with different randomly acoustic and replay configurations. Each replay configuration has been considered a different type of replay attack, so that $K = 9$ replay attacks have been used for training.

3.2. System

This section details the methodology followed to train our proposed system. First, speech signals were segmented using a Blackman analysis window of 16 ms length with 4 ms of frame shift. Log magnitude spectrogram features with $F = 256$ bins were obtained to feed the proposed deep feature extractor described in Section 2. It processes context windows of $W = 32$ frames with a shift of $\delta = 12$ frames. For every experiment on each of the 4 described speech corpora, the system was trained employing only the training set of the corresponding database.

Table 1 shows a summary of the employed LC-GRNN architecture. It is composed by $N = 3$ recurrent layers, where each one has different light convolutional layers followed by a max pooling operation which reduces the frequency dimension. The LCNN block architectures are inspired by the ones proposed in [8, 16], which were able to extract very discriminative features. Once all the frame-level context windows are processed by the convolutional and recurrent layers, 8 feature maps of size 32×32 are flattened to make up a feature vector of 8192 components. Then, this vector is fed to a fully connected layer (FC1) with MFM activation to obtain the spoofing identity vector of the utterance of 512 components.

The proposed deep feature extractor was trained using the Adam optimizer [21] with a learning rate of $3 \cdot 10^{-4}$, early stopping, 60% dropout (FC1), and normalizing the input in mean and variance. All the specified hyperparameters were optimized using the development sets of the ASVspoof 2019 data corpora.

For the back-end, we evaluated three different classifiers: support vector machine (SVM), linear discriminant analysis (LDA), and its probabilistic version (PLDA). The objective of the classifier is to assign a score indicating whether the utterance is genuine or spoofed. In some models, we also applied a posterior normalization of the scores. Provided the prior of the different classes is uniform, the normalized score of the spoofing identity vector $\mathbf{x}$ is

$$p(genuine|\mathbf{x}) = log \frac{p(\mathbf{x}|genuine)}{\sum_{j=1}^{K+1} exp(p(\mathbf{x}|j))}, \qquad (4)$$

where $p(\mathbf{x}|j)$ is the log posterior predictive probability of the spoofing identity vector $\mathbf{x}$ given class j ($j = 1, ..., K + 1$, including the genuine class).

4. Results

4.1. Results on ASVspoof 2015

Table 2 compares the performance of our proposed anti-spoofing system with different state-of-the-art and deep feature extractor based systems on the ASVspoof 2015 database. Our proposed system with PLDA classifier achieves the best performance in the known and unknown attacks, outperforming other state-of-the-art systems such as the CQCC + GMM [22] and LTSS + MLP [23]. It clearly outperforms other similar deep feature extractors of the literature such as the CNN + RNN [20] and FBANK + Best RNN [24], as well as our previous system FBANK + CNN + RNN [15]. In particular, the performance of our proposals for S10 attack is quite meaningful. Regarding the classifiers employed to score the spoofing identity vectors extracted by our proposed LC-GRNN, PLDA without scoring normalization yields the lowest Equal Error Rate (EER), outperforming the other 2 classifiers (SVM and LDA) in the S10 attack.

4.2. Results on ASVspoof 2017

Table 3 shows a comparison of the performance of our anti-spoofing system with different state-of-the-art single systems on the ASVspoof 2017 database. Our proposed system with PLDA and scoring normalization achieves the best performance. It outperforms other state-of-the-art single systems such as the SCMC + GMM [25], LCNN + GMM [16] and CNN + RNN [16], which were presented to the ASVspoof 2017 Challenge. In fact, our proposal achieves an EER 0.32% lower than the Siamese CNN + GMM [26], which is one of the state-of-the-art single deep feature extractors for this corpus.

Table 2: *Comparison between classifiers and with other systems on evaluation set of ASVspoof 2015 in terms of (%) EER*

System	Known	Unknown	
		S6 - S9	S10
Spectro + CNN + RNN [20]	0.40	0.60	14.27
FBANK + Best RNN [24]	0.20	0.50	10.70
FBANK + CNN + RNN [15]	0.03	0.13	9.34
CQCC + GMM [22]	0.05	0.31	1.07
LTSS + MLP [23]	0.10	0.11	1.56
LC-GRNN + SVM	**0.00**	**0.00**	1.01
LC-GRNN + LDA	**0.00**	0.01	0.82
LC-GRNN + PLDA (Norm.)	**0.00**	0.03	2.83
LC-GRNN + PLDA	**0.00**	**0.00**	**0.69**

Table 3: *Comparison between classifiers and with other systems on ASVspoof 2017 database in terms of (%) EER*

System	Development	Evaluation
Baseline: CQCC + GMM	10.35	30.60
SCMC + GMM [25]	9.32	11.49
LCNN + GMM [16]	4.53	7.37
CNN + RNN [16]	7.51	10.69
Siamese CNN + GMM [26]	-	6.40
LC-GRNN + SVM	4.62	8.12
LC-GRNN + LDA	4.10	7.53
LC-GRNN + PLDA	3.42	6.35
LC-GRNN + PLDA (Norm.)	**3.26**	**6.08**

Regarding the scoring normalization, we can conclude that it is beneficial when the nature of the unseen attacks is similar to the seen ones, such as the different replay configurations considered for training the ASVspoof 2017 model. However, the genuine class can be easily mistaken for an attack when it is generated with a new technique not seen during training (as it happens with the S10 attack based on MaryTTS [27] in the ASVspoof 2015 corpus).

4.3. Results on ASVspoof 2019 LA

Because of the good results obtained on the ASVspoof 2015 database at detecting logical access attacks, we decided to submit the LC-GRNN + PLDA as primary and single system, the LC-GRNN + LDA as contrastive 1 system, and the LC-GRNN + SVM as contrastive 2 system, to the ASVspoof 2019 Logical Access Challenge [14]. We can only compare their performance with the baseline systems because the participating systems have not been published yet.

Table 4 shows the results on the ASVspoof 2019 LA database obtained by the baseline systems (CQCC + GMM [22] and LFCC + GMM [28]) and our submitted systems. The proposed LC-GRNN system outperforms the baseline systems on the development and evaluation sets independently of the scoring classifier. Specifically, our contrastive 1 system (LC-GRNN + LDA) achieves a relative 22.37% and 34.38% better performance than CQCC + GMM and LFCC + GMM on the evaluation set, respectively. It is worth noticing that although our contrastive 1 system outperforms our primary/single system (LC-GRNN + PLDA), the difference of EER is only 0.06%. Taking into account that PLDA only performed 0.01% better on the overall EER of ASVspoof 2015 evaluation set, we can conclude that LDA and PLDA classifiers have a similar performance at scoring the LA spoofing identity vectors extracted by the proposed LC-GRNN system.

Table 4: *Results on ASVspoof 2019 Logical Access in terms of min-tDCF and EER (%)*

System	min-tDCF		EER (%)	
	Dev.	Eval.	Dev.	Eval.
Baseline 1: CQCC + GMM	0.0123	0.2366	0.43	9.57
Baseline 2: LFCC + GMM	0.0663	0.2116	2.71	8.09
LC-GRNN + SVM	0.0002	0.1873	0.01	7.12
LC-GRNN + PLDA	**0.0000**	0.1552	**0.00**	6.34
LC-GRNN + LDA	**0.0000**	**0.1523**	**0.00**	**6.28**

Table 5: *Results on ASVspoof 2019 Physical Access in terms of min-tDCF and EER (%)*

System	min-tDCF		EER (%)	
	Dev.	Eval.	Dev.	Eval.
Baseline 1: CQCC + GMM	0.1953	0.2454	9.87	11.04
Baseline 2: LFCC + GMM	0.2554	0.3017	11.96	13.54
LC-GRNN + LDA	0.0469	0.0946	1.59	3.49
LC-GRNN + PLDA	0.0306	0.0747	1.18	2.68
LC-GRNN + PLDA (Norm.)	**0.0203**	**0.0614**	**0.73**	**2.23**

4.4. Results on ASVspoof 2019 PA

According to the results obtained on the ASVspoof 2017 database at detecting replay attacks, we decided to submit the LC-GRNN + PLDA with scoring normalization as primary and single system, the LC-GRNN + PLDA without normalization as contrastive 1 system, and the LC-GRNN + LDA as contrastive 2 system, to the ASVspoof 2019 Physical Access Challenge [14].

Table 5 shows the results on the ASVspoof 2019 PA database obtained by the baseline systems and our submitted systems. The proposed LC-GRNN system clearly outperforms the baseline systems on the development and evaluation sets independently of the scoring classifier. Specifically, our primary/single system (LC-GRNN + PLDA with scoring normalization) achieves a relative 74.69% and 79.80% better performance than CQCC + GMM and LFCC + GMM on the evaluation set, respectively.

5. Conclusions

This paper has proposed a novel technique for the extraction of utterance-level identity vectors for an efficient detection of TTS/VC and replay attacks. In our system, a gated recurrent unit based RNN learns long-term dependencies of the subsequent deep features, while several integrated light convolutional neural networks extract discriminative features at frame level. This proposal has been submitted as a single system to the ASVspoof 2019 Challenge [14].

The results show that our proposed system notably outperforms the baseline systems of the ASVspoof 2019 challenges (CQCC + GMM and LFCC + GMM). Moreover, it also yields very remarkable results as single system on the ASVspoof 2015 and 2017 databases, outperforming other popular methods such as the CQCC + GMM [22] and even the fusion of systems (winner of the 2017 challenge) presented in [16].

6. Acknowledgements

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, as well as the organizers of the ASVspoof 2019 Challenge.

7. 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] 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.

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

[5] 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.

[6] 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.

[7] 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.

[8] Xiang Wu, Ran He, Zhenan Sun and Tieniu Tan, "A Light CNN for Deep Face Representation with Noisy Labels," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 11, pp. 2884–2896, 2018.

[9] 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.

[10] 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.

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

[12] 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.

[13] 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.

[14] H. Delgado, M. Todisco, Md Sahidullah, N. Evans, T. Kinnunen, K. A. Lee, J. Yamigishi, et al. (2019, March). ASVspoof 2019: Automatic Speaker Verification Spoofing and Countermeasures Challenge. [Online] Available: http://www.asvspoof.org

[15] 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.

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

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

[18] 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.

[19] 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.

[20] 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.

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

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

[23] 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.

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

[25] R. Font, J. M. Espin, and M. J. Cano, "Experimental analysis of features for replay attack detection–Results on the ASVspoof 2017 Challenge," *Proc. Interspeech*, 2017.

[26] K. Sriskandaraja, V. Sethu, and E. Ambikairajah, "Deep Siamese Architecture Based Replay Detection for Secure Voice Biometric," *Proc. Interspeech*, 2018.

[27] DFKI's Language Technology Lab and Multimodal Speech Processing, MMCI (2019, March). The Mary Text-to-Speech System (MaryTTS). [Online] Available: http://mary.dfki.de

[28] M. Sahidullah, T. Kinnunen, and C. Hanilçi, "A comparison of features for synthetic speech detection," *Proc. Interspeech*, 2015.

2.1.2 Anti-spoofing Architectures with Countermeasures for Noisy Acoustic Conditions

2.1.2.1 A Deep Identity Representation for Noise Robust Spoofing Detection

- Alejandro Gomez-Alanis, A. M. Peinado, Jose A. Gonzalez and Angel M. Gomez, "A Deep Identity Representation for Noise Robust Spoofing Detection", *Proc. Interspeech*, pp. 676-680, Hyderabad, India, September 2018.
    - Status: Published.

A Deep Identity Representation for Noise Robust Spoofing Detection

Alejandro Gomez-Alanis[1], Antonio M. Peinado[1], Jose A. Gonzalez[2], and Angel M. Gomez[1]

[1]University of Granada, Granada, Spain
[2]University of Malaga, Malaga, Spain

{agomezalanis,amp,amgg}@ugr.es, jgonzalez@lcc.uma.es

Abstract

The issue of the spoofing attacks which may affect automatic speaker verification systems (ASVs) has recently received an increased attention, so that a number of countermeasures have been developed for detecting high technology attacks such as speech synthesis and voice conversion. However, the performance of anti-spoofing systems degrades significantly in noisy conditions. To address this issue, we propose a deep learning framework to extract spoofing identity vectors, as well as the use of soft missing-data masks. The proposed feature extraction employs a convolutional neural network (CNN) plus a recurrent neural network (RNN) in order to provide a single deep feature vector per utterance. Thus, the CNN is treated as a convolutional feature extractor that operates at the frame level. On top of the CNN outputs, the RNN is employed to obtain a single spoofing identity representation of the whole utterance. Experimental evaluation is carried out on both a clean and a noisy version of the ASVSpoof2015 corpus. The experimental results show that our proposals clearly outperforms other methods recently proposed such as the popular CQCC+GMM system or other similar deep feature systems for both seen and unseen noisy conditions.

Index Terms: Spoofing detection, noise robustness, speaker verification, deep learning, missing-data masks.

1. Introduction

In recent years, automatic speaker verification (ASV) [1, 2, 3] 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 unauthorized access to a system by using spoofing attacks [4]. For ASV, four types of spoofing attacks have been identified [5]: (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. In this work, we are interested in providing anti-spoofing measures against spoofing attacks based on either VC or TTS.

As shown in [5], state-of-the-art ASV systems are highly vulnerable to TTS/VC based spoofing attacks. Thus, the development of anti-spoofing techniques is a subject that has recently attracted the attention of a number of researchers [4, 5]. Broadly speaking, these techniques attempt to identify synthetic speech by detecting the artifacts produced by the speech vocoders used in TTS/VC systems. For instance, a popular approach attempts to detect the phase artifacts introduced by minimum-phase vocoders [8]. Although these countermeasures have been successfully applied in clean conditions, they are known to fail when the attacks are deployed in noisy scenarios. As shown in [10], the performance of the spoofing countermeasures trained on clean conditions is significantly degraded in noisy scenarios and this deterioration increases as the signal-to-noise ratio (SNR) decreases. Thus, providing robust anti-spoofing techniques against noisy conditions is also becoming a key issue.

The literature about ASV anti-spoofing in noisy conditions is scarce due to the novelty of this area. One of the first studies was carried out in [11], where the robustness of various front-end features were evaluated under different noisy conditions. In [10], a neural network was trained as an anti-spoofing detection system, and several front-end features were tested under five additive noises and reverberant conditions. Also, the use of frame-level deep features were proposed and evaluated in [12], being justified as a mean to extract useful information for spoofing detection from the noise corrupted spectral features. These deep features were extracted using several neural network architectures.

In this work, we propose a CNN+RNN system to get a single spoofing identity representation per utterance, which is robust to noisy and reverberant conditions. Although a CNN+RNN model was firstly proposed in [20] for anti-spoofing, our proposed system presents four important differences: (1) it uses context windows to avoid applying a padding or cropping method to the input of the system, (2) the CNN and RNN are not optimized simultaneously, (3) it introduces noise features for an increased noise robustness, and (4) the CNN+RNN framework is not used as the final classifier. In contrast to the DNN and CNN systems proposed in [12], our spoofing identity representation is not obtained by averaging the frame-level deep features of an utterance. Instead of that, we propose a recurrent layer which is fed with the outputs of the CNN in order to learn long-term dependencies. Furthermore, we propose a novel methodology for noise awareness based on the use of missing-data masks [6, 7], which define the reliability of the spectro-temporal regions in the noisy spectrum.

This paper is organized as follows. Section 2 describes the proposed (CNN+RNN) deep feature extractor and the LDA back-end employed along the work. Then, in Section 3, we outline the speech corpora, the network training, and the performance evaluation details. Section 4 discusses the results of our system under clean and noisy scenarios, and shows a comparison with other relevant anti-spoofing systems. Finally, we present the conclusions derived from this research in Section 5.

2. System description

This section is devoted to the description of the proposed anti-spoofing feature extraction procedure. First, Section 2.1 describes the different front-end stages: input spectral feature extraction, frame-level CNN deep feature extraction, and RNN (utterance-level) identity feature extraction. The linear discriminant analysis (LDA) classifier employed as back-end is detailed in Section 2.2. A block diagram of the proposed feature extrac-

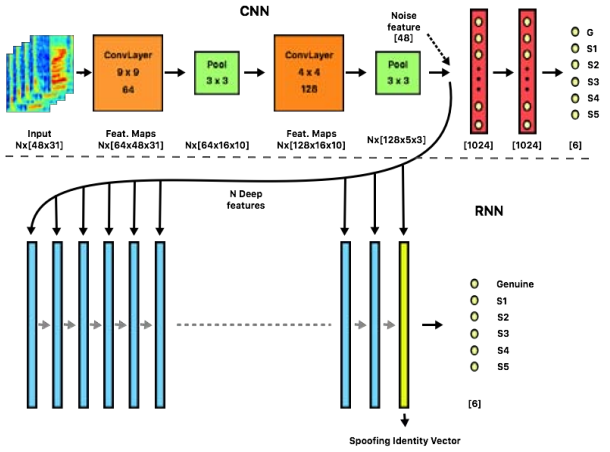

Figure 1: *Deep learning framework to extract a spoofing identity representation per utterance (N represents the number of context windows per utterance).*

tion system is shown in Fig. 1. All deep models are implemented with the Tensorflow toolkit [14].

2.1. Front-end

The proposed CNN+RNN front-end system provides a single spoofing identity representation of the whole utterance as shown in Fig. 1. The frame window size is 25 ms with 10 ms of frame shift. A context window of 31 frames (centered at the frame being processed) is used to obtain the input signal spectral features which are fed into the system. Then, the CNN provides a deep feature vector per window, and all deep features vectors of the considered utterance are processed by the RNN in order to obtain the spoofing identity vector of the utterance.

As demonstrated in [12], traditional log MEL filterbank features (FBANK) are more robust to noise than the recently proposed constant Q cepstral coefficients (CQCCs) [16]. Thus, we have also adopted FBANK features. In contrast to [12] and [15], we use a 48-dim static FBANK without delta and acceleration coefficients, as we have realized that the context window of 31 frames is already exploiting the correlations between successive frames. Therefore, a higher spectral resolution is achieved while the size of the spectral feature vector is smaller than in [12]. The result of this processing block is a feature matrix of size $[48 \times 31]$ per frame. The FBANK features are obtained using the HTK toolkit [18]. Mean and variance normalization is applied to the resulting FBANK parameters.

In our architecture, the CNN plays the role of a frame-level deep feature extractor providing one feature vector for each context window of 31 frames. In order to do this, the CNN acts as a classifier whose task consists of determining whether the input features are either genuine or belong to one of the 5 spoofing attacks (S1, S2, S3, S4 or S5) present in the training set. Our CNN uses 2 convolutional and pooling layers as feature extractors, followed by 2 fully connected layers of 1024 sigmoid neurons with a softmax layer of 6 neurons as classification layer. To prevent the problem of overfitting, 50 % and 40 % dropout is applied to the 2 fully connected layers, respectively.

The first convolutional layer obtains 64 feature maps using 9×9 filters. This results in a volume of size $[64 \times 48 \times 31]$. Then, the pooling layer performs a downsampling operation along the spatial dimensions (width=31, height=48) using 3×3 filters, resulting in a smaller volume of $[64 \times 6 \times 10]$. The second convolutional layer obtains 128 features using 4×4 filters, which results in a volume of $[128 \times 16 \times 10]$. After that, a second pooling layer with 3×3 filters reduces the final volume to $[128 \times 5 \times 3]$. In the two pooling layers, we use a stride of 3 and a VALID padding. Finally, the 128 features of size $[5 \times 3]$ are concatenated to make up a deep feature vector of 1920 components.

As shown in Fig. 1, the deep features obtained from CNN are fed into an RNN, which computes the anti-spoofing identity vector for the utterance. The advantage of using an RNN is its ability for learning the long-term dependencies of the subsequent deep feature vectors. The activation function of the RNN is a gated recurrent unit (GRU) [19]. Finally, a fully connected layer containing 6 neurons (one per class: genuine, S1, S2, S3, S4 and S5) is connected to the output of the last time step, followed by a softmax layer. The state of the last time step represents the single deep identity spoofing vector of the whole utterance.

In addition to multi-condition training, in this work we evaluate two types of noise features aimed at improving the robustness against noise of our anti-spoofing detection methods. First, noise-aware training (NAT) is implemented by using a noise code per utterance, which is computed by averaging the 48 spectral features of the first 10 frames of the utterance. Second, in order to have a more finer grain detail about the reliability of each spectro-temporal region of the noisy utterance, we propose the use of soft masks [6, 7]. Each mask defines, for each spectral feature, the probability that this feature is contaminated by noise. To compute the mask, noise is firstly estimated for each frame by linearly interpolating two independent noise estimates computed by averaging the first and last $N = 10$ frames of each utterance. Next, the SNR is estimated from the original noisy features and the noise estimates. A sigmoid function is finally applied to the SNR values to compress them between the $[0, 1]$ range in order to obtain the missing-data masks. In both cases (NAT and missing-data masks), the noise features are appended to the output of the convolutional layers of the CNN, which results in deep features of 1968 components as shown in Fig. 1. Finally, this augmented deep feature vector is fed into the RNN.

2.2. Back-end

As shown in [15], a linear discriminant analysis (LDA) back-end achieves the best performance for anti-spoofing in comparison with other state-of-the-art techniques. In general, LDA classification has shown a high performance for a variety of tasks [21, 22]. Thus, we will employ an LDA back-end to assign a genuine speech confidence score to each utterance. Our LDA classifier uses 6 classes which represent genuine speech and the five known spoofing attacks considered in the training set. The genuine class score is the only used for decision.

3. Experimental framework

In order to evaluate the performance of our proposed techniques, the ASVspoof 2015 corpus [9], a well-known database containing data from different spoofing attacks under clean conditions, was employed. Also, a noisy version of this corpus [10] was also considered to evaluate the robustness of the different proposals against noise. Details about the methodology followed for training and testing are given in this section.

3.1. Speech corpus

The clean ASVspoof 2015 corpus [9] defines three datasets (training, development and evaluation), each one containing a mix of genuine and spoofed speech. Spoofing attacks were generated either by TTS or VC. A total of 10 types of spoofing attacks (S1 to S10) are defined: three of them are implemented using speech synthesis (S3, S4 and S10), and the remaining seven ones (S1, S2, S5, S6, S7, S8 and S9) using 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 [9].

In order to evaluate the robustness of our proposals against noise, the noisy version of the ASVspoof 2015 corpus (described in [10]) 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. In particular, 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 [12], 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 development sets. More details about this corpus are given in [10, 12].

3.2. Training

As mentioned in the previous section (front-end description), FBANK spectral features, extracted from a 48-filter Mel-filterbank, are used to represent the speech signal. These features were normalized in mean and variance.

In the clean scenario, the original ASVspoof 2015 corpus [9] is used for training and evaluation. Here, our anti-spoofing system does not append any noise features at the output of the convolutional layers.

In the noisy scenario, the noisy version of the ASVspoof 2015 corpus [10] is used. Here, multi-condition training is applied in order to get higher level features that are more robust against noise. Furthermore, noise features are appended to the output of the convolutional layers in order to increase this robustness. We have tested two types of noise features: (1) noise aware training (NAT), and (2) soft noise masks (MASK). This augmented feature vector is fed into both the upper layers of the CNN and the RNN.

In both scenarios, the separately CNN and RNN are trained using Adam optimizer [23]. Also, early stopping is applied in order to stop the training process when no improvement is obtained after ten iterations.

3.3. Performance evaluation

The equal error rate (EER) is used to evaluate the system performance. As described in the ASVspoof 2015 challenge evaluation plan [9], the EER was computed independently for each spoofing algorithm and then the average EER across all attacks was used. To compute the average EER, we used the Bosaris toolkit [13].

4. Results

The results by our proposal and other techniques from the literature are presented below.

4.1. Clean scenario

Table 1 shows a comparison of the performance of different anti-spoofing systems in the clean version of ASVspoof 2015 database. The FBANK+CNN+LDA system has already been proposed in [12], but as its performance is not provided in this reference for the clean scenario, we have evaluated instead our proposed system removing the RNN and averaging the deep features for getting the identity spoofing vector of the utterance as in [12]. The CQCC+GMM system achieves the best average performance, although our proposed system (FBANK+CNN+RNN+LDA) achieves the best results for the known attacks. Compared to the rest of deep learning systems (Spectro+CNN+RNN [20], Best DNN [15], Best RNN [15] and FBANK+CNN+LDA), our proposal outperforms all of them in the known and unknown attacks. Particularly noteworthy is the result of our system in the S10 attack. The CFCC-IF system [17] achieves a lower EER in the S10 attack than our proposed system, but our proposal performs 0.21 % better on average when considering all the attacks.

4.2. Noisy scenario

Table 2 compares the performance of five different systems on the noisy version of the ASVspoof 2015 database. Multi-condition training was used in all cases. In the table, the performance of NAT and MASK techniques for noise awareness is also evaluated. The last four systems use FBANK as input features and an LDA as final classifier. For the sake of clarity, these two acronyms have been removed.

As shown in Table 2, when multi-condition training is used, our CNN+MASK+RNN system achieves the best overall performance in the clean condition, even outperforming CQCC+GMM, which was the best system in Table 1. Furthermore, the use of the RNN decreases the total average EER from 1.09 % and 0.93 % to 0.59 % and 0.47 % when using NAT and MASK techniques, respectively. This result shows the importance of getting the identity spoofing representation of an utterance using a recurrent layer to learn the long-term dependencies, instead of averaging the deep features as in [12].

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 CNN+MASK+RNN system achieves the best results with an overall relative improvement of 26.6 % compared to CQCC+GMM. Moreover, the use of the proposed MASK noise features provides the best robustness against noise outperforming NAT in both models (CNN and CNN+RNN). Specifically, it reduces a 0.6% and 0.3% the total average EER, respectively.

The CQCC+GMM performs again very poorly in the unseen noise conditions when compared to our proposals. As in the seen noisy conditions, the MASK noise feature obtains significantly better results than NAT and so does the hybrid CNN+RNN model in comparison with the CNN model.

To sum up, the results under noisy conditions show that our two proposals (RNN for utterance-level identity representation and MASK noise awareness) significantly improve the perfor-

Table 1: *Comparison on evaluation clean dataset for each spoofing attack in terms of (%) EER*

System	Known Attacks						Unknown Attacks						Total
	S1	S2	S3	S4	S5	Avg.	S6	S7	S8	S9	S10	Avg.	Avg.
CQCC + GMM [16]	**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**
Spectro + CNN + RNN [20]	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 [15]	**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 [15]	**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
CFCC-IF [17]	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
FBANK + CNN + LDA	0.02	1.07	**0.00**	**0.00**	0.51	0.32	1.03	0.44	0.05	0.51	20.57	4.52	2.42
FBANK + CNN + RNN + LDA	**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

Table 2: *Comparison on evaluation noisy dataset in terms of average (%) EER using multi-condition training*

Eval. Condition	CQCC + GMM [12]			CNN + NAT			CNN + MASK			CNN + NAT + RNN			CNN + MASK + RNN		
	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	0.12	1.74	0.93	0.04	1.13	0.59	**0.03**	**0.90**	**0.47**
white_snr_20	46.8	44.6	45.7	1.7	4.3	3.0	1.4	3.9	2.7	1.1	2.9	2.0	**0.8**	**2.5**	**1.7**
white_snr_10	48.9	48.1	48.5	3.2	5.1	4.4	2.7	4.6	3.7	**2.1**	3.7	**2.9**	2.3	**3.4**	**2.9**
white_snr_0	49.3	48.9	49.1	7.9	10.0	9.0	7.2	9.3	8.3	6.9	9.1	8.0	**5.9**	**8.6**	**7.3**
babble_snr_20	18.2	18.3	18.3	3.1	4.6	3.9	2.9	4.1	3.5	2.5	**3.7**	**3.1**	**2.3**	3.9	**3.1**
babble_snr_10	33.9	33.6	33.8	5.7	6.7	6.2	5.2	5.9	5.6	4.1	4.8	4.5	**3.7**	**4.5**	**4.1**
babble_snr_0	44.6	44.0	44.3	12.9	14.7	13.8	12.1	13.6	12.9	10.1	11.7	10.9	**9.5**	**10.6**	**10.1**
street_snr_20	22.7	22.3	22.5	3.9	5.1	4.5	2.7	4.2	3.5	2.1	3.5	2.8	**1.9**	**3.1**	**2.5**
street_snr_10	37.5	36.3	36.9	6.1	7.5	6.8	5.1	6.7	5.9	4.6	5.7	5.2	**4.1**	**5.4**	**4.8**
street_snr_0	46.1	45.4	45.8	11.1	13.7	12.4	10.1	12.4	11.3	9.1	10.8	10.0	**8.7**	**9.9**	**9.3**
reverberation_0.3	8.4	9.3	8.9	1.3	2.1	1.7	1.5	2.2	1.9	1.2	**1.8**	1.5	**1.1**	1.9	**1.5**
reverberation_0.6	10.6	7.8	9.2	1.6	2.0	1.8	1.5	2.1	1.8	**1.4**	1.7	**1.6**	1.6	**1.5**	**1.6**
reverberation_0.9	7.6	6.9	7.3	1.5	1.9	1.7	1.4	1.7	1.6	1.2	**1.5**	**1.4**	**1.1**	1.6	**1.4**
Avg. Seen Noise	31.2	30.5	30.8	5.0	6.5	5.8	4.5	5.9	5.2	3.9	5.1	4.5	**3.6**	**4.7**	**4.2**
cafe_snr_20	30.7	30.1	30.4	2.9	5.3	4.1	2.7	5.4	4.1	1.9	4.7	3.3	**1.8**	**4.5**	**3.2**
cafe_snr_10	42.1	41.3	41.7	5.6	8.1	6.9	5.3	7.8	6.6	4.7	6.1	5.4	**4.5**	**5.7**	**5.1**
cafe_snr_0	49.8	47.1	47.3	13.5	20.0	16.8	12.4	18.7	15.6	10.7	15.4	13.1	**10.1**	**14.3**	**12.2**
volvo_snr_20	0.9	**2.7**	**1.8**	1.0	3.7	2.4	0.9	3.4	2.2	**0.7**	3.1	1.9	0.8	3.0	1.9
volvo_snr_10	4.3	5.6	4.9	2.4	4.9	3.7	2.1	4.5	3.3	1.7	3.6	2.7	**1.5**	**3.4**	**2.5**
volvo_snr_0	13.0	13.0	13.0	3.7	5.0	4.4	3.4	4.7	4.1	3.1	3.7	3.4	**2.7**	**3.5**	**3.1**
Avg. Unseen Noise	23.1	23.3	23.2	4.9	7.8	6.4	4.5	7.4	6.0	3.8	6.1	5.0	**3.6**	**5.7**	**4.7**

mance in both seen and unseen noisy conditions with respect the two reference techniques (CQCC+GMM, CNN+NAT). It must be taken into account that although CNN+NAT is the best isolated deep feature extraction proposed in [12], this reference also proposes a combination of DNN, CNN, RNN and NAT for frame-level feature extraction that outperforms CNN+NAT. However, this combination is not directly comparable with our CNN+MASK+RNN since it is a fusion of techniques unlike our proposal. Despite this, it is worth mentioning that this combination can only outperform our best proposal in the case of seen noises but not in the case of the unseen ones. Finally, it is worth noticing that, although the CQCC+GMM system has been proved to get the best state-of-the-art results using the clean ASVspoof 2015 database, our FBANK+CNN+MASK+RNN+LDA system gets a better performance in the clean evaluation dataset when using multi-condition training.

5. Conclusions

This paper has proposed a novel technique for the extraction of deep identity features for an efficient detection of spoofing attacks in clean and noisy environments. In our system, a CNN+RNN hybrid architecture is employed to embed the utterances as a single vector, providing information about whether the utterance is genuine or spoofed. Furthermore, to increase the noise robustness of our anti-spoofing detector, a soft missing-data mask technique has been proposed.

Our system has been evaluated on the ASVspoof 2015 clean corpus and on a distorted version of the same corpus, including both additive noise and reverberation. The experimental results have shown that our best proposal outperforms the CQCC+GMM system (baseline of the ASVspoof 2017 challenge [24]) and the best isolated deep feature extractor proposed in [12] (CNN+NAT) for both seen and unseen distorted conditions, respectively.

In the future, we plan to integrate other noise mask estimation techniques in the deep feature extraction procedure in order to obtain further improvements in noisy conditions. Also, we will investigate the incorporation of phase-based features that could complete the signal information lost by the FBANK features.

6. Acknowledgements

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.

7. References

[1] D. A. Reynolds, "Speaker identification and verification using Gaussian mixture speaker models," *Speech Communication*, vol. 17, no. 1-2, pp. 91–108, 1995.

[2] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted Gaussian mixture models," *Digital Signal Processing*, vol. 10, no. 1-3, pp. 19–41, 2000.

[3] N. Dehak, P. J. 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. 130–153, 2011.

[4] 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, no. 4, pp. 788-798, 2015.

[5] 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.

[6] 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.

[7] J. P. Barker, L. Josifovski, M. P. Cooke, and P. D. Green, "Soft decisions in missing data techniques for robust automatic speech recognition," in *Proc. ICSLP*, 2000, pp. 373–376.

[8] J. Sanchez, I. Saratxaga, I. Hernaez, E. Navas, and D. Erro, "A cross-vocoder study of speaker independent synthetic speech detection using phase information," in *Proc. InterSpeech*, 2014, pp. 1663–1667.

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

[10] X. Tian, Z.Wu, X. Xiao, E. S. Chng, and H. Li, "An investigation of spoofing speech detection under additive noise and reverberant condition," in *Proc. InterSpeech*, 2016, pp. 1715–1719.

[11] 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," in *Speech Communication*, vol. 85, pp. 83–97, 2016.

[12] 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.

[13] N. Brümmer and E. deVilliers, "The BOSARIS toolkit: Theory, algorithms and code for surviving the new DCF," in *NIST SRE11 Speaker Recognition Workshop*, Atlanta, Georgia, USA, Dec. 2011, pp. 1–23.

[14] Martín Abadi, Ashish Agarwal, Paul Barham, et al. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

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

[16] M. Todisco, H. Delgado, and N. Evans, "A new feature for automatic speaker verification anti-spoofing: Constant Q cepstral coefficients," *Proc. Odyssey*, 2016, pp. 249–252.

[17] 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, pp. 2062–2066.

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

[19] Kyunghyun Cho, et al., "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation," *Proc. Empirical Methods in Natural Language Processing*, 2014, pp. 1724–1734.

[20] Chunlei Zhang, Chengzhu Yu, and John 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.

[21] Q. Jin and A. Waibel, "Application of LDA to speaker recognition," *Proc. Interspeech*, 2010, pp. 250–253.

[22] M. McLaren and D. Van Leeuwen, "Source-normalised-and-weighted LDA for robust speaker recognition using i-vectors," *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2011, pp. 5456–5459.

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

[24] Tommi Kinnunen, Md Sahidullah, Héctor Delgado, et al. "The ASVspoof 2017 Challenge: Asessing the Limits of Replay Spoofing Attack Detection," *Proc. Interspeech*, 2017, pp. 2–6.

2.1.2.2 A Gated Recurrent Convolutional Neural Network for Robust Spoofing Detection

- Alejandro Gomez-Alanis, A. M. Peinado, Jose A. Gonzalez and Angel 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, December 2019.
    - Status: Published.
    - Impact Factor (JCR 2019): 3.398
    - Subject Category: Acoustics. Ranking 4/32 (D1).
    - Subject Category: Engineering, Electrical & Electronic. Ranking 66/273 (Q1).
    - Awards: Best student journal paper of the Red Temática en Tecnologías del Habla - Edition 2020.

A Gated Recurrent Convolutional Neural Network for Robust Spoofing Detection

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

***Abstract*—Automatic speaker verification (ASV) systems are exposed to spoofing attacks which may compromise their security. While anti-spoofing techniques have been mainly studied for clean scenarios, it has also been shown that they perform poorly in noisy environments. In this work, we aim at improving the performance of spoofing detection for ASV in clean and 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 the final genuine/spoofed classification. Then, to enhance the robustness of the system in noisy conditions, we propose the use of signal-to-noise masks (SNMs) as new input features to inform the anti-spoofing system about the time-frequency regions of the input spectral 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 for detecting logical access attacks, as well as on the ASVspoof 2017 database to detect replay attacks. Additional results are provided for the ASVspoof 2019 corpus, including both logical and physical scenarios. The experimental results show that our proposal clearly outperforms some well-known methods based on classical features and other similar deep feature based systems for both clean and noisy conditions.**

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

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[1] [2]. 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 [3]: (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.

Anti-spoofing systems must learn to detect not only the attacks observed in the training dataset, but also be able to generalize to unseen attacks. To address this issue, deep feature extraction was proposed in [4], where features are extracted from an inner layer of a deep neural network (DNN) to represent every temporal frame of the voice signal, or even the whole utterance. Fig. 1 illustrates this idea. The system attempts to determine whether the input speech signal is genuine or spoofed. To this end, the classifier make use of the deep features (embeddings) extracted by a DNN. Depending on the architecture of the neural network, 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 [5], [6]. 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 [7], (ii) phase based spectral features [8], and (iii) raw speech samples [9].

The extraction of deep features (embeddings) at a frame level has demonstrated to be effective in both ASV [10] and spoofing detection [11]. For instance, x-vectors [12] have become very popular in ASV due to its good performance, superior to that of i-vectors. Regarding anti-spoofing, DNNs and convolutional neural networks (CNNs) were used in [13] to obtain frame-level deep features, showing that convolutional layers have a powerful ability for detecting the artifacts caused by the speech vocoders used in TTS/VC systems. This is even possible in noisy conditions, as CNNs can be seen as filter banks whose filters are optimized for the specific task of spoofing detection [14]. In addition, a residual CNN architecture was also employed in [15] as a frame-level feature extractor for detecting replay attacks, and a Light-CNN [16] which employs Max-Feature-Map activations was the best system of the ASVspoof 2017 Challenge [17].

These frame-level features must be combined into a single identity vector which characterizes the whole utterance. There are several ways to combine them, as depicted in Fig. 1, such as averaging [18], attentive statistics pooling [19], or by using recurrent neural networks (RNNs) [20]. There is an ample evidence that RNNs are powerful at extracting discriminative features to capture the temporal artifacts in the spoofed speech. For instance, a combination of a CNN with an RNN based on gated recurrent unit (GRU) blocks was successfully applied in [21], [22] to extract utterance-level deep features for detecting logical access attacks. Likewise, a combination of several fully connected layers with two long short-term memory (LSTM) blocks was proposed in [23] as an end-to-end system. In [16], a combination of a Light-CNN with an RNN was proposed to extract utterance-level embeddings for detecting replay attacks.

The authors 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).

[1]https://www.nuance.com/omni-channel-customer-engagement/security/fraud-detection.html

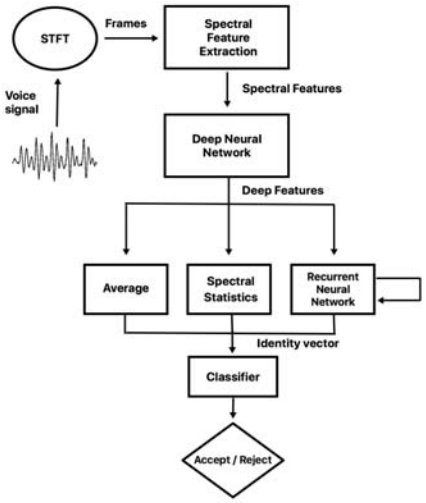

Fig. 1. Extraction of frame-level deep features and utterance-level identity vector for spoofing detection. Here, we consider three methods to extract the utterance-level identity vector from frame-level deep features: average (left), spectral statistics (middle), and recurrent neural network (right).

In addition, a deep siamese neural network architecture based on convolutional layers was proposed in [24] as an utterance-level feature extractor to improve the performance of the previous systems at detecting replay attacks. More recently, end-to-end systems based on RNNs have also been proposed in [25], [26] for detecting replay attacks.

In contrast to end-to-end anti-spoofing systems, spoofing detectors based on the extraction of deep features need to use a classifier 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 those based on Gaussian Mixture Models (GMMs), Linear Discriminant Analysis (LDA) and Support Vector Machines (SVMs) are often employed for this task.

On the other hand, research on anti-spoofing has been mainly focused on systems operating on clean conditions, while little work has been carried out considering the noise (i.e. acoustic noise and/or reverberation) which will be likely present in realistic situations. Noise will be, in general, a cause for performance degradation, although its effect varies according to the type of attack. Thus, the noise introduced by the playback and recording devices might be a hint of attack [17], but it cannot be easily separated from the noise present in the acoustic environment, which, in turn, may conceal those hints. In addition, as shown in [27], 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 is one of the main motivations of this work.

One of the first works to study the impact of noise on antispoofing systems was carried out in [28], where the robustness of several front-end features were evaluated under different noisy conditions. In [27] 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, [13] showed that the anti-spoofing techniques based on deep feature extractors improve significantly when they are trained with noisy data (i.e. multicondition training), owing this improvement to the cabapility of neural networks to learn discriminative features which are more invariant to noise. Furthermore, in [13], the system's robustness against noise was improved by means of noise aware training (NAT), in which a vector with the mean noise magnitude spectra is presented to the network. This idea was further refined in our previous work [20] where, rather than just using the noise mean, a missing-data mask, informing about the reliability of each time-frequency bin, was used.

In this work we propose an anti-spoofing system which can be deployed to detect various types of spoofing attacks and, at the same time, is noise robust. In particular, the main contributions of our work can be summarized as follows:

1) Gated Recurrent Convolutional Neural Network: We propose the use of a new architecture called Gated Recurrent Convolutional Neural Network (GRCNN), as a deep feature extractor for spoofing detection. In a GRCNN, the dense layers inside the GRU cells are replaced with convolutions in order to extract more discriminative features. Thus, we expect 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. Although similar deep learning frameworks have been applied in learning video representations [29], audio tagging [30] and optical character recognition [31], to the best of our knowledge, this is the first time that GRCNNs are adapted to spoofing detection.

2) Signal-to-Noise Masks: To enhance the robustness of anti-spoofing systems against noise, we propose a new technique for estimating masks based on a deep learning framework. In a previous work [20], we demonstrated that applying classical SNR-based masks [32], [33] for spoofing detection obtains the best state-of-the-art results in noisy scenarios. Here, we improve the estimation of signal-to-noise mask (SNM) features by means of deep learning techniques. Moreover, we carry out multiple experiments to choose a suitable configuration and training procedure for the proposed SNM feature extraction, and we evaluate the SNM features in several noisy anti-spoofing databases.

This work also provides two more minor contributions. First, we evaluate the use of both magnitude and phase based features and show that a single anti-spoofing system using both types of features outperforms a fusion of independent systems, each working with a feature type. Second, we establish different classes of replay attacks by combining low, medium and high qualities of both playback and recording devices in order to extract more discriminative utterance-level embeddings. This network arrangement has been shown to be more effective than that of using just two classes (attack, genuine).

This paper is organized as follows. Section II describes

the proposed deep feature extractor employed along the work. Then, in Section III, we describe the proposed SNM features for noise robust spoofing detection. Section IV outlines the speech corpora, the network training and the system details. After that, Sections V, VI and VII discuss the performance of our system for detecting logical access and replay attacks under clean and noisy acoustic conditions. Finally, we summarize the conclusions derived from this research in Section VIII.

II. Gated Recurrent Convolutional Neural Network

In this section we describe the details of the GRCNN architecture for spoofing detection. Based on our previous work [20], our hypothesis is that replacing the standard dense operations within GRU cells with convolutional layers strives 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 those based on LSTM [34] and GRU [35] cells. The latter one shows similar performance to LSTM but with lower memory and computational requirements [36]. 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 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 [37], we replace the fully-connected operations in the GRU with convolutional layers, thus allowing the network to focus on these frequency bands through filter optimization. This change in network architecture is what mainly differences our proposal from the RNNs proposed in other works [16], [21], [23], [25]. The advantage of this modification is that more discriminative features can be extracted at the frame level [13], as shown later in the experimental results Sections (V and VI).

The proposed feature extractor with N recurrent layers 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 consecutive frames. This context window moves forward δ frames on every time step[2], so that the total number of time steps of the GRCNN is $(T-W)/\delta$, where T is the number of frames of the utterance being processed. This architecture acts as a classifier whose task consists of determining whether the

[2]Instead of moving forward the context window one frame on every time step, we propose to move it δ frames (δ < W) in order to reduce the processing time, while maintaining the classification performance.

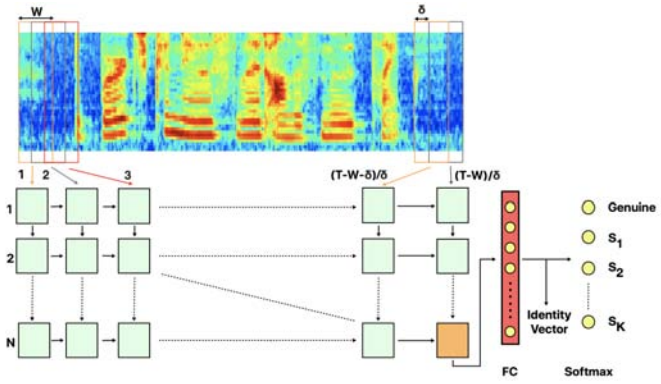

Fig. 2. Block diagram of the proposed utterance-level spoofing identity vector extractor. It consists of a Gated Recurrent Convolutional Neural Network (GRCNN) which processes the spectral features of a context of W consecutive frames in each time step through the N recurrent layers. The utterance has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

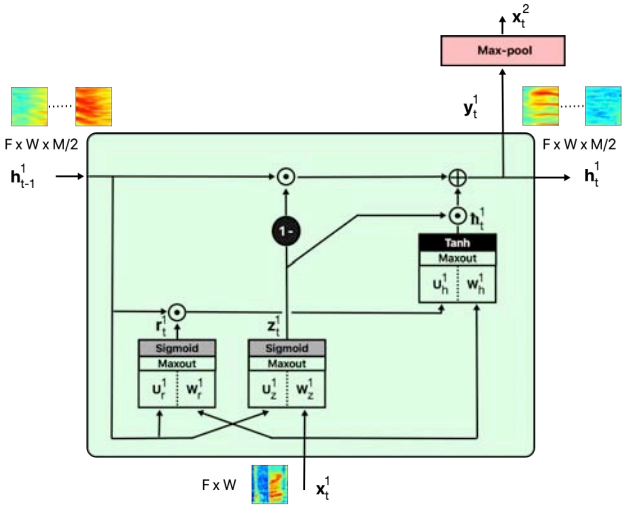

Fig. 3. Gated recurrent convolutional unit cell (GRCU) of the first recurrent layer. The input is 2-dimensional and may include several channels. The output consists of M 2-dimensional feature maps, which are passed to both the next layer and next step of the GRCNN.

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, the output of the last time step and last recurrent layer is fed to a fully connected (FC) layer with maxout activation to obtain the spoofing identity vector of the whole utterance. During the training phase, this identity vector is finally passed through another fully connected layer with softmax activation of $K+1$ neurons to discriminate between the genuine and the K spoofing classes.

Each block in Fig. 2 represents a gated recurrent convolutional unit cell (GRCU). As shown in Fig. 3, similarly to a GRU cell, the GRCU defines three gates, each one implemented by means of 2 single-layer CNNs of M filters. Every CNN performs convolutions followed by a maxout activation intended to reduce the output feature maps by a $1/2$ factor [38]. In this way, the GRCNN plays the role of a

frame-level deep feature extractor in each time step, providing N state (feature) vectors for each context window of W consecutive frames. As can be seen, each GRCU cell applies a total of 6 convolutional 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 dimension $[F, W, M/2]$, 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

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

determines which information from the previous frames needs to be passed along the next steps, avoiding the risk of the vanishing gradient problem [39]. The operator $*$ denotes a convolution operation. Similarly, the reset gate

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

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 of employing these filters is the extraction of frame-level features at every time step. These are more discriminative than those extracted by using fully connected units [21]. Finally, the third gate is the update activation,

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

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 the previous equations, the functions $\sigma(\cdot)$ and $\tanh(\cdot)$ are respectively the sigmoid and hyperbolic tangent activation functions of the gates, while $\odot$ 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 filters of the single-layer CNNs, which are shared by all time steps of the GRCNN.

III. Noise Robustness: Signal-to-Noise Masks

In an attempt to improve the robustness against noise of our anti-spoofing system, in this section we propose a novel mask estimation technique which aims at identifying those regions of the speech spectrum which are less reliable (i.e. more corrupted by noise). Unlike speech enhancement, where the goal is to reduce the amount of noise which is present in the speech signal, our goal is to provide an estimation of the noise present in each time-frequency bin to the GRCNN, in order to extract more robust embeddings from the noisy signals.

As a first step towards an increased robustness, training a DNN 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 to 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. To do this, each component of 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.

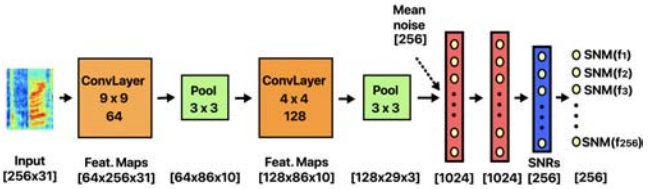

Fig. 4. Convolutional neural network for the estimation of the signal-to-noise masks for each time-frequency bin. The utterance noise mean is concatenated with the output of the convolutional layers as a noise reference.

In our recent work [20] 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 [13]. In [20], the masks were computed from the noise estimates obtained by 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 at 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 which estimates the SNMs using a deep learning framework. The proposed system is the CNN shown in Fig. 4, which provides the estimated SNR of each time-frequency bin corresponding to the frame which is being processed. The input is a context of W STFT features, 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, is computed as,

$$\boldsymbol{SNR}(t,f) = 20 \cdot log_{10}\left(\frac{\mathbf{X}(t,f)}{\mathbf{N}(t,f)}\right), \quad (4)$$

where the tuple (t, f) represents the time-frequency bin, and $\mathbf{X}$ and $\mathbf{N}$ are the magnitude STFT outputs of the clean speech and noise, respectively. To obtain the SNM target, this SNR is

compressed in the range $[0, 1]$ using a tunable sigmoid function centered at β dB

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

where α controls the slope of the sigmoid, and β corresponds to the threshold commonly used to define the Ideal Binary Masks (IBMs) [40]. In fact, the SNM mask corresponds to the IBM when $\alpha \to \infty$. By substituting the $\boldsymbol{SNR}(t, f)$ of Eq. (4) in Eq. (5), the proposed mask is given by

$$\mathbf{m}_k = \frac{\mathbf{X}^\gamma}{\mathbf{X}^\gamma + e^{\alpha\beta} \cdot \mathbf{N}^\gamma}, \quad (6)$$

where $\gamma = 20\alpha/log(10)$. This target mask also corresponds to the Ideal Ratio Mask (IRM) [41] when $\beta = 0$ dB and $\alpha = 0.23$. In this way, Eq. (6) defines a family of parametric masks which include the IBM and IRM masks, since both parameters, α and β, are tunable.

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,

$$\text{Loss} = \sum_{k=1}^{F} \mathbf{m}_k \cdot log(\sigma(\mathbf{z}_k)) + (1 - \mathbf{m}_k) \cdot log(1 - \sigma(\mathbf{z}_k)), \quad (7)$$

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. Second, 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 a target 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. This way, the model parameters are optimized 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 [42], ASVspoof 2017 [17], [43] and ASVspoof 2019 [44] corpora, three well-known databases containing data from different types of spoofing attacks, were employed. Also, a noisy version of the ASVspoof 2015 corpus [27] was also considered to evaluate the robustness of the different proposals against noise. Details about the experimental methodology are presented in the following.

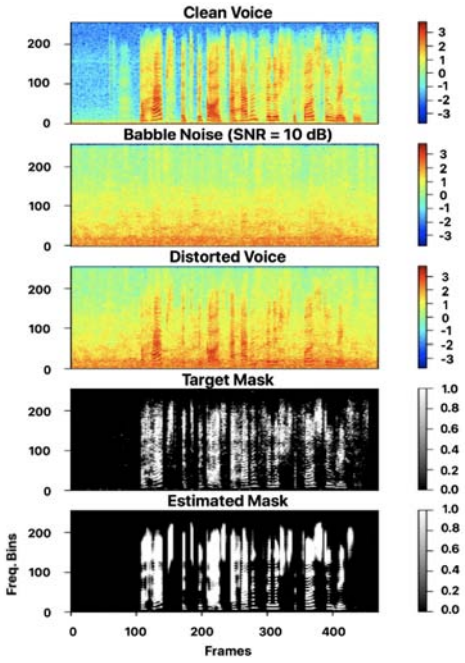

Fig. 5. Estimation of a noise mask for babble noise ($SNR = 10\,dB$): (a) original clean voice, (b) babble noise of 10 dB, (c) distorted voice with additive babble noise of 10 dB, (d) target mask with $\alpha = 0.25$ and $\beta = 5$ dB, (e) estimated mask.

TABLE I
Structure of the ASVspoof 2015 data corpus [42]

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

A. Speech Corpora

We conducted experiments on 3 databases: (a) the ASVspoof 2015 corpus [42]; (b) a noisy version of the ASVspoof 2015 corpus [27]; (c) the ASVspoof 2017 corpus [17], [43]; and (d) the ASVspoof 2019 corpus [44].

1) ASVspoof 2015 Corpus [42]: This is a standard data corpus for research on logical access attacks 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 TTS or VC techniques. A total of 10 types of spoofing attacks (S1 to S10) are defined: three of them are implemented by using TTS (S3, S4 and S10), while the remaining seven ones (S1, S2, S5, S6, S7, S8 and S9) by means of different VC 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.

2) ASVspoof 2015 Noisy Corpus [27]: To evaluate the robustness of our system against noise, a 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 SNR levels.

A total of 5 additive noise types (*white*, *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 [13], 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 the performance with stationary and non-stationary noises in both *seen* and *unseen* conditions. More details about this corpus are given in [27].

3) ASVspoof 2017 Corpus [17], [43]: This is a standard data corpus for research on replay spoofing attacks detection. It defines three datasets (training, development and evaluation), where each one is a collection of *bona fide* and *spoofed* utterances. A summary of their composition is presented in Table II. This corpus has two versions, version 1 and 2, where version 2 fixes a number of data anomalies that were found to potentially influence the results [43]. These mostly involve the presence of zero-valued samples in the periods of silence of the genuine utterances, which can be specially exploited by approaches that include some form of temporal attention mechanism.

The corpus contains data for 177 different replay sessions and 61 distinct replay configurations (RCs), where each RC comprises one playback device, one acoustic environment and one recording device. Furthermore, the data was collected in a total of 26 different environments, which correspond to the physical space in which the original speech data is replayed and re-recorded. Variations between them include different types and levels of ambient noise and reverberation. There are 6 types of environments: two of them contain high ambient noise (*balcony* and *cantine*), other 2 contain medium ambient noise levels (*home* and *office*), and the other three are low noise conditions (*anechoic room*, *studio* and *analog wire*).

When training our GRCNN for replay attack detection, we considered $K = 5$ conditions: genuine speech and 4 fidelity conditions, as a result of combining low/medium and high qualities of the playback and recording devices reported in [43].

4) ASVspoof 2019 Corpus [44]: This database encompasses two partitions for the assessment of logical access (LA) and physical access (PA) scenarios. Both LA and PA databases are themselves partitioned into three datasets, namely training, development and evaluation. The three partitions are disjoint in terms of speakers and the recording conditions for all source data are identical. While the training and development sets contain spoofing attacks generated with the same algorithms/conditions (*known attacks*), the attacks of the evaluation set were generated with different algorithms/conditions (*unknown attacks*).

TABLE II
STRUCTURE OF THE ASVSPOOF2017 DATA CORPUS DIVIDED BY THE TRAINING, DEVELOPMENT AND EVALUATION SETS [17].

Subset	# Speakers	# Replay Sessions	# Replay Config	# Utterances Bona Fide	# Utterances Replay
Training	10	6	3	1507	1507
Development	8	10	10	760	950
Evaluation	24	161	57	1298	12008
Total	42	177	61	3565	14465

TABLE III
GRCNN ARCHITECTURE ($p = r, z, \tilde{h}$).

GRCNN	Type	Filter / Stride	Output
Layer 1	Conv ($\mathbf{W}_p^1$, $\mathbf{U}_p^1$)	5×5 / 1×1	$16 \times 256 \times 32$
	Maxout	-	$8 \times 256 \times 32$
	MaxPool	2×1 / 2×1	$8 \times 128 \times 32$
Layer 2	Conv ($\mathbf{W}_p^2$, $\mathbf{U}_p^2$)	3×3 / 1×1	$32 \times 128 \times 32$
	Maxout	-	$16 \times 128 \times 32$
	MaxPool	2×1 / 2×1	$16 \times 64 \times 32$
Layer 3	Conv ($\mathbf{W}_p^3$, $\mathbf{U}_p^3$)	3×3 / 1×1	$16 \times 64 \times 32$
	Maxout	-	$8 \times 64 \times 32$
	MaxPool	2×1 / 2×1	$8 \times 32 \times 32$
-	FC	-	480×2
	Maxout	-	480

The LA database contains 17 attacks generated with the latest TTS and VC technologies, where only six of them are *known attacks*. On the other hand, the bona fide 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 9 replay configurations, comprising 3 categories of attacker-to-speaker recording distances and 3 categories of loudspeaker quality, so that we considered $K = 9$ replay attacks for training.

B. System description

This section provides a detailed description of the implemented system:

1) Spectral analysis: Speech signals were analyzed using a Blackman analysis window of 25 ms lenght with 10 ms of frame shift. Log magnitude spectrogram and modified group delay (MGD) [45] features with $F = 256$ bins were obtained to feed the proposed deep feature extractor described in Section II. The core idea is to provide the network with both amplitude and phase information. The log magnitude spectrogram features were normalized using the mean and variance of the whole training set. The MGD features were obtained using the Covarep toolkit [46] with $\gamma = 0.3$ and $\alpha = 0.1$, which were optimized using the validation set of the data corpora.

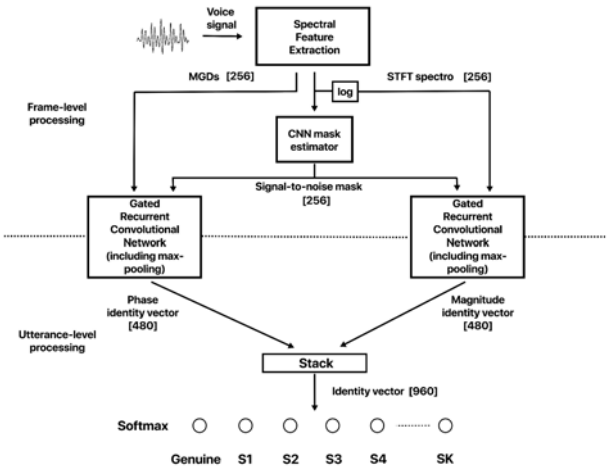

Fig. 6. Proposed architecture for the extraction of the utterance-level spoofing identity vector. Two independent gated recurrent convolutional nets extract the magnitude and phase identity vectors, which are stacked into a single spoofing identity vector of 960 components.

2) Spoofing identity vector extraction: Log spectrogram and MGD features (along with the corresponding SNM features) are processed by two parallel and independent GRCNNs. Table III shows a summary of the employed GRCNN architecture. Each GRCNN is composed by $N = 3$ recurrent layers, where each gate of the GRCU cell consists of a single-layer CNN with maxout activation of a $1/2$ factor. As shown in Fig. 3, there are 6 single-layer CNNs inside each GRCU 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 recurrent layers in order to reduce the size of the deep feature maps.

Once all the frame-level context windows are processed by the convolutional and recurrent layers of the GRCNN, 8 feature maps of size 32×32 are flattened to make up a feature vector of 8192 components. Then, this vector is fed to a fully connected layer with maxout activation to obtain the utterance-level embedding of the GRCNN (480 components). As shown in Fig. 6, the two utterance-level embeddings obtained from magnitude and phase spectrum are stacked into one single spoofing identity vector of 960 components. This identity vector 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 GRCNNs are optimized jointly, being each set of layers specialized in processing either the magnitude or the phase based features.

3) SNM features: In order to achieve noise robustness, signal-to-noise masks were appended to the input features as a second channel. The CNN presented in Section III for SNM estimation was independently trained using the genuine data from the training set of the noisy version ASVspoof 2015 corpus. To this end, the target mask of every utterance was calculated using (4), and the optimizing criterion was the binary cross entropy presented in (7). The SNM features were not employed in the clean ASVspoof 2015 database experiments, since all the utterances are completely clean.

4) Training setup and toolkits: The proposed deep learning framework was trained using the Adam optimizer [47] with a learning rate of $3{\cdot}10^{-4}$ and a batch size of 32 utterances. 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 overfitting, a fixed 30% dropout was applied in the convolutional layers, as well as a 60% dropout in the fully connected layer (FC) of the two GRCNNs. All the specified hyperparameters of the system were optimized using the validation set of the data corpora. The Pytorch toolkit [48] was employed to implement the deep learning framework.

5) Classifier: After the extraction of the spoofing identity vector for each 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 [49]), 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, (iv) its probabilistic version (PLDA), and (v) gaussian mixture model (GMM) with log-likelihood ratio.

C. Performance Metric

The equal error rate (EER) is used to evaluate the system performance. It was computed using the Bosaris toolkit [50]. For the ASVspoof 2015 corpus, the EER was computed independently for each spoofing algorithm and then the average EER across all attacks was obtained, as it is described in the ASVspoof 2015 challenge evaluation plan [42]. Similarly, the different noisy conditions of the noisy ASVspoof 2015 corpus were evaluated individually to obtain the EER for each scenario. For the ASVspoof 2017 corpus, a single EER was computed for the development and evaluation sets.

V. Results on ASVspoof 2015

This section presents the experimental results from the evaluation of the proposed techniques on the ASVspoof 2015 corpus. First, section V-A evaluates the proposed GRCNN architecture on clean conditions. Then, Section V-B is devoted to evaluate the noise robustness of the system with the proposed SNM features on the noisy version of the ASVspoof 2015 database.

A. Architecture evaluation on clean speech

Table IV 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 shown separately. Moreover, the results for the unknown S10 attack are also shown separately, since this one has proven to be the most difficult attack to detect for automatic anti-spoofing systems.

TABLE IV
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	STFT	0.08	0.12	2.65
	MGD	0.30	0.52	5.21
	STFT + MGD	0.10	0.06	2.33
SVM	STFT	0.07	0.11	2.24
	MGD	0.39	0.55	5.02
	STFT + MGD	0.12	0.14	1.85
GMM	STFT	0.04	0.08	1.82
	MGD	0.27	0.45	4.52
	STFT + MGD	0.02	0.05	1.12
LDA	STFT	0.01	0.11	0.82
	MGD	0.28	0.48	3.74
	STFT + MGD	0.01	0.04	0.28
PLDA	STFT	0.00	0.05	0.75
	MGD	0.21	0.41	3.32
	STFT + MGD	**0.00**	**0.00**	**0.21**

TABLE V
COMPARISON OF THE FUSION OF SEPARATED STFT + GRCNN AND MGD + GRCNN SYSTEMS WITH THE JOINT STFT + 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	0.12	0.09	3.12
STFT + MGD + SVM-One	0.10	0.06	2.33
Fusion Scores SVM	0.14	0.20	2.17
STFT + MGD + SVM	0.12	0.14	1.85
Fusion Scores GMM	0.10	0.12	1.58
STFT + MGD + GMM	0.02	0.05	1.12
Fusion Scores LDA	0.09	0.08	1.06
STFT + MGD + LDA	0.01	0.04	0.28
Fusion Scores PLDA	0.04	0.07	0.98
STFT + MGD + PLDA	**0.00**	**0.00**	**0.21**

In addition to our own model using both the magnitude STFT and phase MGD features, we evaluated a fusion of separated STFT + GRCNN and MGD + GRCNN systems. The rationale of this comparison is to determine whether the joint STFT + 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 V. The fusion is performed by normalizing the individual scores to zero mean and unit variance using the pre-computed mean and standard variance for the training set. Finally, the weighted average of the two scores obtained by the individual systems is calculated for the detection decision.

The best result is obtained using a PLDA classifier and employing magnitude STFT and MGDs jointly as input features. When only one type of features is used, STFTs outperform MGD features irrespective of the classifier. The combination of magnitude STFT and MGD features obtains the best performance independently of the classifier, outperforming the fusion of the individual systems STFT + 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, thus being able to detect different artifacts of the spoofing attacks from correlations detected between both types of features. These results indicate that although magnitude spectrum contains the most important information 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 irrespective of the input spectral features. Furthermore, LDA outperforms SVM, GMM and SVM-One for all attacks. There are relevant 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 [42].

Based on the results from Tables IV and V, in the rest of the evaluation we will use the GRCNN architecture jointly employing both STFT and MGD features, and a PLDA classifier. Table VI 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 first 4 systems employ RNNs to extract utterance-level embeddings, whereas the remaining 6 systems are based on the extraction of features specifically developed to detect spoofing attacks (CFCC-IF [51], CQCC [52], LTSS [53], CQCC(A) + APGDF(A) + FFV(SD) [54], eCQCC-A [55]), or on a new scoring method (CQCC + DNN-HLL [56]). We can observe that the main source of error for most of the systems is the S10 attack, for which we can observe meaningful differences of performance.

Compared to the RNN based systems (Spectro + CNN + RNN [21], MFCC + LSTM [23], Best RNN [18] and FBANK + CNN + RNN [20]), our proposal outperforms all of them for the known and unknown attacks. It can be observed that the EER in the S10 attack is much lower than that of those systems, irrespective of the input features and classifier employed. These results show the significant improvement of performance of the proposed GRCNN as a deep utterance-level extractor. Regarding the other 6 systems, our proposal also outperforms all of them in the known and unknown attacks, and even in the S10 attack. The average EER for all ten attacks is reduced up to 0.02 %, which, to the best of our knowledge, is the best performance among all the published results.

B. Noise robustness evaluation

1) Evaluation on Seen Conditions: Fig. 7 presents the results of the proposed anti-spoofing system on the seen conditions of the noisy ASVspoof 2015 corpus. Multi-condition training and SNM features are employed to increase the robustness of the system against noise. Results are shown for different configurations of the α and β parameters used to obtain the SNM features (see eq. (5)). As can be seen, the configuration which better mitigates the effects of noise is $\alpha = 0.25$ and $\beta = 5$ dB. Moreover, the slope of the sigmoid α seems to make more differences than the threshold β, due to the fact that α controls the smoothness of the mask.

Fig. 8 compares the performance obtained when the SNM features are trained using the BCE loss (as indicated in Section III) or the Mean Square Error (MSE) loss, which is the loss

TABLE VI
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 [21]	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
MFCC + LSTM [23]	2.20	3.40	0.00	0.20	3.50	0.54	3.90	2.40	0.00	2.80	10.70	3.96	2.91
Best RNN [18]	**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 [20]	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 [51]	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 [52]	**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 [53]	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
CQCC(A) + APGDF(A) + FFV(SD) [54]	**0.00**	0.02	**0.00**	**0.00**	**0.00**	**0.00**	0.10	0.02	**0.00**	0.02	0.30	0.09	0.05
eCQCC-A [55]	**0.00**	0.01	**0.00**	**0.00**	0.01	**0.00**	0.01	**0.00**	**0.00**	**0.00**	0.30	0.06	0.03
CQCC + DNN-HLL [56]	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	0.19	**0.00**	0.26	0.09	0.05
STFT + MGD + GRCNN + PLDA	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.21**	**0.04**	**0.02**

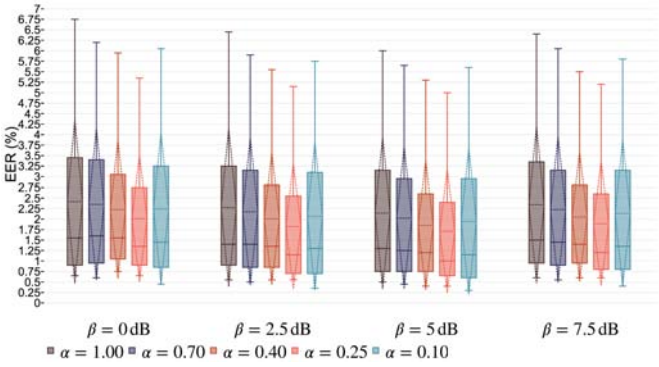

Fig. 7. Box plot of averaged EERs (%) for all seen noisy evaluation scenarios employing different SNM configurations. Box edges are at 25% and 75% quantiles. Dashed lines represent the mean and standard deviation values.

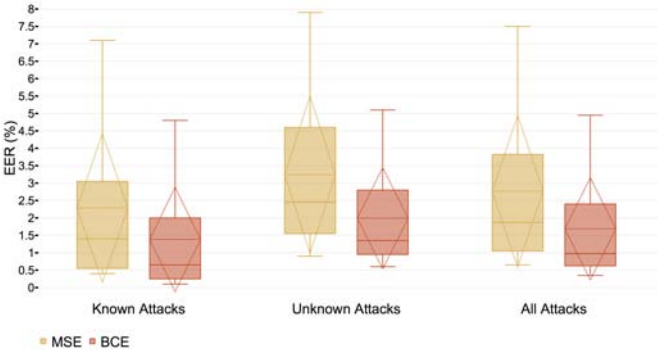

Fig. 8. Box plot of averaged EERs (%) for all seen noisy evaluation scenarios using either the BCE or MSE criterion for training the SNM features with $\alpha = 0.25$ and $\beta = 5$ dB. Box edges are at 25% and 75% quantiles. Dashed lines represent the mean and standard deviation values.

function commonly used for IRM mask estimation [41]. It can be seen that BCE clearly outperforms MSE for both the known and unknown attacks. According to these experiments, the rest of this work will employ the BCE criterion for training the SNM features with a configuration of $\alpha = 0.25$ and $\beta = 5$ dB.

Table VII presents the per-attack results of the proposed anti-spoofing system on the seen conditions of the noisy ASVspoof 2015 corpus evaluation set. As expected, the EER is higher when the SNR decreases. When the noise power increases, the artifacts present in the spoofed signal are more difficult to detect, as they can be concealed by the noise. *Babble* and *street* are the most challenging noises, since they are non-stationary and resemble genuine speech. On the other hand, reverberation is the distorsion type which is easier to counteract, irrespective of the reverberation level.

Comparing the results from Tables VI and VII, we can see that the performance obtained by our proposal in clean conditions is even better when multi-condition training and SNM features are employed. 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 features act as a simple attentional mechanism which make the GRCNN network to focus on the spectral regions where speech is dominant.

2) Evaluation on Unseen Conditions: Although it is possible to collect multiple noise types for training and optimize the model using multi-condition training, in real life applications there would still be many unseen noisy scenarios. Accordingly, to validate the effectiveness and generalization capability of the proposed approach, we conducted an evaluation on unseen noisy scenarios. The detailed results of different training techniques (clean, multi-condition training and SNM features) are shown in Table VIII.

First of all, in order to assess the impact of noisy environments, a baseline test using the clean model (without SNM features) 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.

Then, multi-condition training (without SNM features) 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

TABLE VII
PERFORMANCE ACHIEVED WITH MULTI-CONDITION TRAINING AND SNM FEATURES ($\alpha = 0.25$ AND $\beta = 5\ dB$) 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	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.10	0.02	0.01
White (SNR = 20 dB)	0.00	0.46	0.00	0.00	0.69	0.23	0.28	0.00	0.00	0.15	4.72	1.03	0.63
White (SNR = 10 dB)	0.08	1.21	0.00	0.00	1.81	0.62	1.06	0.38	0.00	0.20	6.01	1.53	1.08
White (SNR = 0 dB)	0.89	4.46	0.15	0.15	6.45	2.42	3.77	2.72	0.26	1.64	10.81	3.84	3.13
Babble (SNR = 20 dB)	0.00	0.79	0.00	0.00	1.11	0.38	0.15	0.06	0.00	0.06	4.33	0.92	0.65
Babble (SNR = 10 dB)	0.24	3.29	0.00	0.00	4.42	1.59	0.80	0.36	0.10	0.15	7.79	1.84	1.72
Babble (SNR = 0 dB)	2.64	8.50	0.36	0.36	12.24	4.82	4.15	3.64	0.32	1.78	15.46	5.07	4.95
Street (SNR = 20 dB)	0.00	1.24	0.00	0.00	2.36	0.72	0.22	0.07	0.00	0.10	5.26	1.13	0.93
Street (SNR = 10 dB)	0.21	3.08	0.00	0.00	4.26	1.51	0.97	0.42	0.00	0.21	6.95	1.71	1.61
Street (SNR = 0 dB)	2.11	6.88	0.29	0.29	9.88	3.89	1.05	3.06	0.24	2.22	15.08	4.33	4.11
Reverberation (T60 = 0.3 s)	0.00	0.23	0.00	0.00	0.32	0.11	0.12	0.02	0.00	0.08	2.98	0.64	0.38
Reverberation (T60 = 0.6 s)	0.00	0.30	0.00	0.00	0.40	0.14	0.32	0.15	0.09	0.20	3.59	0.87	0.51
Reverberation (T60 = 0.9 s)	0.10	0.59	0.00	0.00	0.96	0.33	0.69	0.28	0.26	0.35	4.62	1.24	0.79
Avg. EER seen conditions	0.52	2.59	0.07	0.07	3.74	1.40	1.13	0.93	0.10	0.60	7.30	2.01	1.71

TABLE VIII
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 + SNM Features		
	Known	Unknown	Avg.	Known	Unknown	Avg.	Known	Unknown	Avg.
Clean	0.00	0.04	0.02	0.00	0.03	0.02	0.00	0.02	0.01
Cafe (SNR = 20 dB)	35.28	38.12	36.70	2.96	4.86	3.91	1.12	1.81	1.47
Cafe (SNR = 10 dB)	37.04	42.23	39.64	5.95	8.12	7.04	1.98	3.21	2.60
Cafe (SNR = 0 dB)	44.25	47.12	45.69	12.89	18.31	15.60	7.83	11.19	9.51
Volvo (SNR = 20 dB)	34.79	38.34	36.82	2.05	3.52	2.79	0.49	1.84	1.17
Volvo (SNR = 10 dB)	37.78	42.67	40.23	4.69	5.76	5.23	0.77	2.62	1.70
Volvo (SNR = 0 dB)	42.56	45.64	44.10	6.85	8.11	7.48	2.42	3.11	2.77
Avg. EER unseen conditions	38.62	42.35	40.53	5.90	8.11	7.01	2.44	3.96	3.20

from the multi-condition training.

After that, multi-condition training and the proposed SNM features 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 3.46% and 4.15% for the known and unknown attacks, respectively. These results show the robustness provided by the proposed SNM features.

3) Evaluation on All Conditions: To further evaluate the robustness of the proposed SNM features in the practical case where any acoustic condition and type of attack are (a priori) possible, a single pooled EER (evaluated over all possible acoustic conditions, clean, seen and unseen, as well as over all possible attacks) was also obtained when using clean training, multi-condition training and multi-condition training along with SNM features (as in Table VIII). The corresponding pooled EERs are 39.12%, 6.93% and 2.85%, respectively. These results are in line with those of Tables VII and VIII, so that they confirm the benefits of providing the neural network with information about the noise present in each time-frequency bin.

4) Comparison with Other Systems: Table IX compares the proposed approach (GRCNN + MASK-2) with five systems on the noisy version of the ASVspoof 2015 database: CQCC + GMM evaluated in [13], CNN + NAT [20], CNN + MASK-1 + RNN [20], and a combination of three different neural networks (DNN, CNN and BLSTM) [13]. The terms MASK-1 and MASK-2 refer to different signal-to-noise mask estimation techniques, being MASK-1 the technique proposed in [20], in which the SNM masks are computed from SNR estimates computed from the beginning and final segments of the utterance, and MASK-2 is the mask estimation technique

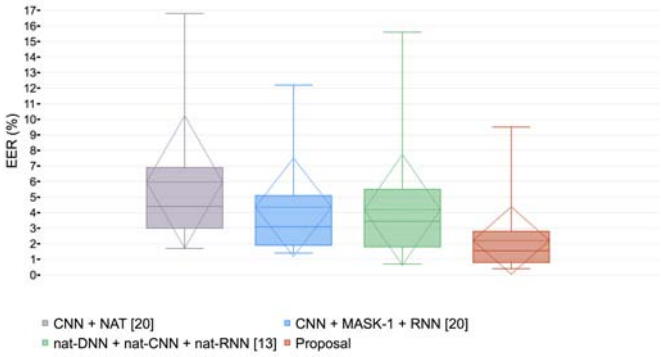

Fig. 9. Box plot of averaged EERs (%) for all noisy evaluation scenarios obtained by: (ii) CNN + NAT [20], (ii) a combination of a DNN, CNN and BLSTM [13], (iii) CNN + MASK-1 + RNN [20], and (iv) our proposal. Box edges are at 25% and 75% quantiles. Dashed lines represent the mean and standard deviation values.

TABLE IX
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 [13]			CNN + NAT [20]			CNN + MASK-1 + RNN [20]			nat-DNN + nat-CNN + nat-RNN [13]			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	**0.00**	**0.02**	**0.01**
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	**1.4**	**2.0**	**1.7**
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	**2.4**	**4.0**	**3.2**

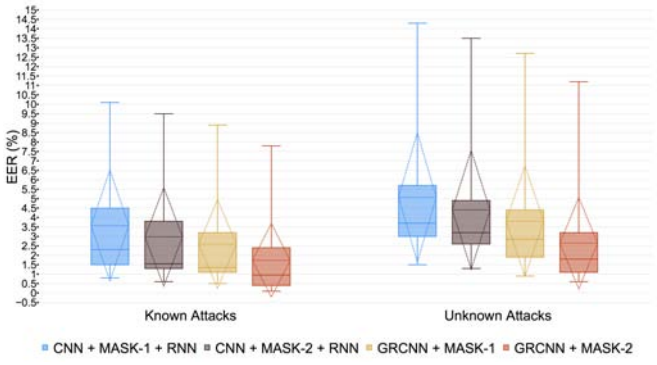

Fig. 10. Box plot of averaged EERs (%) for all noisy evaluation scenarios obtained by: (i) CNN + MASK-1 + RNN [20], (ii) CNN + MASK-2 + RNN, (iii) GRCNN + MASK-1, and (iv) GRCNN + MASK-2. Box edges are at 25% and 75% quantiles. Dashed lines represent the mean and standard deviation values.

proposed here in Section III. The CNN + NAT system was proposed in [13], but as its performance was not provided for the seen conditions, we evaluated it on all conditions [20]. 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. 9 shows a boxplot of the averaged EERs across all noisy conditions, where we can see the statistical differences between the 4 more relevant techniques on the noisy ASVspoof 2015 database.

When multi-condition training is used, our proposed GRCNN + MASK-2 system achieves the best overall performance in the clean condition. Compared to CQCC + GMM and the fusion of systems proposed in [13], it achieves a 0.49% and 0.69% better overall EER, respectively. If we evaluate the systems 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 absolute improvement of 29.1% 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 4.1% and 2.5% on the averaged EER of seen conditions, respectively.

In order to compare the effectiveness of the proposed mask estimation technique (MASK-2) for obtaining the SNM features, Fig. 10 shows the performance of our previous system (CNN + RNN) proposed in [20] and the GRCNN proposed in this work, feeding them with SNM features obtained using either the MASK-1 or MASK-2 estimation technique. It can be observed that MASK-2 performs much better for both systems (CNN + RNN and GRCNN) than MASK-1. Specifically, MASK-2 yields a 0.6% and 0.67% lower averaged EER than MASK-1 for the CNN + RNN system in the known and unknown attacks, respectively. Moreover, it also performs 0.86% and 1.16% better than MASK-1 for the GRCNN system in the known and unknown attacks, respectively. These latter results confirm the effectiveness of the proposed technique for obtaining the SNM features in comparison with a classical SNR-based masks estimation technique [32], [33].

Finally, despite the fact that the combination of systems proposed in [13] 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 outperforms them in both seen and unseen distorted conditions. This suggests 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 a better approach than extracting the mean noise vector of the utterance when providing the neural network with information about the noise present in the utterance.

VI. RESULTS ON ASVSPOOF 2017

This section presents the experimental results obtained on the ASVspoof 2017 replay attacks database. Table X compares the performance of our proposal with other relevant anti-spoofing systems of the literature on the two versions of this database.

First, we compare the performance of our system when the GRCNN is trained employing either $K = 2$ classes (genuine and spoofed speech) or $K = 5$ target classes (genuine speech and 4 different types of replay attacks as a result of combining low/medium and high qualities of both playback and recording devices). It can be observed that considering different types of replay attacks is better for extracting discriminative utterance-level embeddings than considering an unique class for the whole set of replay attacks. Specifically, the improvement of performance on the evaluation set is 0.86% and 1.01% when employing STFT and MGD features, as well as 0.95% and 0.64% when using STFT, MGD and SNM features on the versions 1 and 2 of the database, respectively. This indicates that the proposed GRCNN deep feature extractor learns to generalize better to unseen attacks when it is trained considering multiple types of replay attacks.

It can also be seen that SNM features are beneficial for detecting replay attacks. In particular, an absolute reduction in EER of 1.03% and 1.32% ($K = 2$), as well as 1.12%

TABLE X
COMPARISON WITH OTHER SYSTEMS OF THE LITERATURE ON THE ASVSPOOF 2017 CORPUS (VERSIONS 1 AND 2) IN TERMS OF (%) EER

System	V1	V2
CQCC + GMM (CMVN) [43], [63]	12.24	11.41
LCNN + GMM [16], [59]	7.37	16.08
CNN + RNN [16]	10.69	-
Fusion (LCNN, $SVM_{i\text{-}vect}$, CNN + RNN) [16]	6.73	-
Siamese CNN + GMM [24]	6.40	-
FBANK + GRU [25]	9.81	-
Hybrid Feature + DenseNet + LSTM [26]	8.84	-
GD-ResNet-18 with Attention [57]	**0.00**	-
AF(Sigmoid)-DRN(ReLU) [59]	-	**8.99**
i-vectors (cosine similarity) [59]	-	14.77
Evolving-RNN [60]	-	18.20
qDFTspec [61]	-	11.43
CQNCC-D + DNN [62]	-	10.31
STFT + MGD + GRCNN + PLDA ($K = 2$)	7.22	10.28
STFT + MGD + GRCNN + PLDA ($K = 5$)	6.36	9.27
STFT + MGD + SNM + GRCNN + PLDA ($K = 2$)	6.19	8.96
STFT + MGD + SNM + GRCNN + PLDA ($K = 5$)	**5.24**	**8.32**

and 0.95% ($K = 5$) is obtained when the SNM features are used on the versions 1 and 2 of the database, repectively. This indicates that SNM features help the anti-spoofing system to differentiate between the artifacts introduced by the replay attacks and the noise present in the utterance due to the acoustic environment, which is one of the main challenges of this database since the level of acoustic noise is high [17].

Compared to other relevant systems of the literature, our proposal is only outperfomed by the GD-ResNet-18 [57] which obtains a perfect EER of 0% on the evaluation set of the version 1 database. Nevertheless, the comparison with this system is not fair since it employs 2 pretrained networks on the Imagenet dataset [58] for building a visual attention mechanism, which is particularly beneficial for taking advantage of the issues of the version 1 database [43]. Apart from that, our proposal consisting of a GRCNN and SNM features yields a lower EER than the rest of systems using either $K = 2$ or $K = 5$ training classes, on both versions of the database. Particularly noticeable is the superiority of our system with respect to the RNN based systems (CNN + RNN [16], FBANK + GRU [25], Hybrid Feature + DenseNet + LSTM [26] and Evolving-RNN [60]), demonstrating that the proposed GRCNN architecture is a better approach for detecting replay attacks.

VII. ADDITIONAL RESULTS: ASVSPOOF 2019

This section evaluates the proposed anti-spoofing system (i.e. a GRCNN network using STFT, MGD and SNM input features) on the just published ASVspoof 2019 corpus [44] in order to validate its effectiveness at detecting the recent logical and physical access attacks.

Table XI compares the performance of our proposal on the ASVspoof 2019 database with other relevant single systems with already available descriptions [44], [64], [65], [66]. In addition to the EER, minimum normalized tandem detection cost function (min-tDCF) [67] is also used for performance measuring. As can be seen, our proposed system outperforms the baseline anti-spoofing sytems releases with the database (CQCC + GMM and LFCC + GMM), as well as the other top performing single systems, in both the logical and physical access scenarios. Specifically, our proposal yields a 4.24 % and 9.95 % lower pooled EER than the best baseline systems of the LA and PA scenarios, respectively. In addition, it achieves a 0.68 % and 0.14 % better pooled EER than the best single systems proposed in [64] (FFT + LCNN and CQT + LCNN) for both the LA and PA scenarios, respectively.

TABLE XI
RESULTS ON ASVSPOOF 2019 LOGICAL AND PHYSICAL ACCESS SCENARIOS IN TERMS OF MIN-TDCF AND EER (%)

System	Logical Access		Physical Access	
	min-tDCF	EER (%)	min-tDCF	EER (%)
CQCC + GMM [44]	0.2366	9.57	0.2454	11.04
LFCC + GMM [44]	0.2116	8.09	0.3016	13.54
LFCC + LCNN [64]	0.1000	5.06	0.1053	4.60
FFT + SENet34 [65]	0.2160	11.75	0.0360	1.29
FFT + LCNN [64]	0.1028	4.53	-	-
CQT + LCNN [64]	-	-	0.0295	1.23
End-to-End DNN [66]	-	-	0.1255	4.79
Proposal	**0.0952**	**3.85**	**0.0234**	**1.09**

According to the results of the ASVspoof 2019 Challenge [44], the performance of our single system is comparable to the best fusion/ensemble systems. This shows that our proposed single system is among the state-of-the-art systems at detecting the recent attacks based on the latest technologies.

VIII. CONCLUSION

In this paper we proposed a novel technique for the extraction of deep identity features for an efficient detection of TTS, VC and replay attacks in clean and noisy environments. In our system, a Gated Recurrent Neural Network (GRCNN) 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, we have shown that an anti-spoofing system trained with magnitude and phase spectral features (magnitude of the STFT and MGDs) yields better results than a fusion of single systems fed with each type of these features. As an additional result, we have found that, in the case of replay attacks, the proposed system extracts more discriminative features when training is performed by considering the different configurations of replay attacks as multiple classes. Experimental results on the ASVspoof 2015 and 2017 databases have shown that the proposed architecture outperforms, to the best of our knowledge, all the state-of-the-art systems at detecting logical access attacks, and it is among the best systems at detecting replay attacks. In addition, the experimental results on the recent ASVspoof 2019 corpus show that our proposed single system also performs well at detecting the attacks based on the latest technologies.

To increase the noise robustness of our anti-spoofing detector, a signal-to-noise (SNM) mask estimation technique was also proposed. Our proposal was evaluated on a distorted version of the ASVspoof 2015 corpus, including both additive

and noisy reverberant scenarios, as well as on the ASVspoof 2017 corpus. The experimental results have shown that SNM features are more effective than NAT techniques for detecting logical access attacks under noisy environments, and they are also useful for detecting replay attacks under noisy and reverberant conditions. In fact, our proposal obtains the best state-of-the-art results for the noisy version of the ASVspoof 2015 database, when using multi-condition training along with the proposed SNM features, outperforming the fusion of deep feature extraction systems proposed in [13]. 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 even the results obtained in the evaluation clean condition improve with respect to those obtained with clean condition training.

As 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, a study of different architectures for obtaining the proposed SNM features could be done in order to improve the performance of the estimator. Finally, it would be interesting to explore other types of masks which also employ the phase information of the signal, as well as other types of estimation techniques which do not require any knowledge of the clean signal corresponding to a given noisy utterance in order to train the SNM model, which, in turn, 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 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] M. Algabri, H. Mathkour, M. A. Bencherif, M. Alsulaiman, and M. A. Mekhtiche, "Automatic Speaker Recognition for Mobile Forensic Applications," *Mobile Information Systems*, 2017.

[3] 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.

[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] 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.

[6] 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.

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

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

[9] 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.

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

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

[12] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust DNN Embeddings for Speaker Recognition," *Proc. International Conference on Acoustics, Speech and Signal Processing (ICASSP)*," 2018.

[13] 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.

[14] 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.

[15] W. Cai, D. Cai, W. Liu, and M. Li, "Countermeasures for Automatic Speaker Verification Replay Spoofing Attack: On Data Augmentation, Feature Representation, Classification and Fusion," *Proc. Interspeech*, 2017.

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

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

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

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

[20] 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.

[21] 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.

[22] 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.

[23] S. Scardapane, L. Stoffl, F. Rohrbein, A. Uncini, "On the use of deep recurrent neural networks for detecting audio spoofing attacks," *Proc. International Joint Conference on Neural Networks (IJCNN)*, 2017.

[24] K. Sriskandarja, V. Sethu, and E. Ambikairajah, "Deep Siamese Architecture Based Replay Detection for Secure Voice Biometric," *Proc. Interspeech*, 2018.

[25] Z. Chen, W. Zhang, Z. Xie, X. Su, and D. Chen, "Recurrent Neural Networks for Automatic Replay Spoofing Attack Detection," *Proc. International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018.

[26] L. Huang, and C.M. Pun, "Audio Replay Spoof Attack Detection Using Segment-Based Hybrid Feature and Densenet-LSTM Network," *Proc. International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019.

[27] 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.

[28] 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.

[29] 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.

[30] 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.

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

[32] 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.

[33] J. P. Barker, L. Josifovski, M. P. Cooke, and P. D. Green, "Soft decisions in missing data techniques for robust automatic speech recognition," *Proc. International Conference on Spoken Language Processing (ICSLP)*, 2000.
[34] S. Hochreiter, and J. Schmidhuber, "Long Short-Term Memory," *Journal of Neural Computation*, vol. 9, pp. 1735–1780, 1997.
[35] 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.
[36] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling," *Proc. NIPS Workshop on Deep Learning*, 2014.
[37] M. Witkowski, S. Zacprzak, P. Zelasko, K. Kowalczyk, and J. Galka, "Audio Replay Attack Detection Using High-Frequency Features," *Proc. Interspeech*, 2017.
[38] I.J. Goodfellow, D. Warde-Farley, M. Mirza, A.C. Courville, and Y. Bengio, "Maxout networks," *Proc. International Conference on Machine Learning*, 2013.
[39] 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.
[40] D.L. Wang, U. Kjems, M.S. Perdersen, J.B. Boldt, and T. Lunner, "Speech intelligibility in background noise with ideal binary time-frequency masking," *Journal of Acoustical Society of America*, vol. 125, pp. 2336–2347, 2009.
[41] D.L. Wang, and J. Cheng, "Supervised Speech Separation Based on Deep Learning: An Overview," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 10, pp. 1702–1726, 2018.
[42] 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.
[43] H. Delgado, M. Todisco, Md Sahidullah, N. Evans, T. Kinnunen, K.A. Lee, and J. Yamagishi, "ASVspoof 2017 Version 2.0: meta-data analysis and baseline enhancements," *Proc. Odyssey*, 2018.
[44] 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," *arxiv:1904.05441v2*, 2019.
[45] 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.
[46] 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.
[47] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *Proc. International Conference on Learning Representations (ICLR)*, 2015.
[48] 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.
[49] Scholkopf, B., Williamson, R. C., Smola, A. J., et al., "Support vector method for novelty detection," *Proc. Neural Information Processing System (NIPS)*, 2000.
[50] N. Brümmer and E. deVilliers, "The BOSARIS toolkit: Theory, algorithms and code for surviving the new DCF," *Proc. NIST SRE11 Speaker Recognition Workshop*, 2011.
[51] 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.
[52] M. Todisco, H. Delgado, and N. Evans, "A new feature for automatic speaker verification anti-spoofing: Constant Q cepstral coefficients," *Proc. Odyssey*, 2016.
[53] 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.
[54] M. Pal, D. Paul, and G. Saha, "Synthetic speech detection using fundamental frequency variation and spectral features," *Computer Speech and Language*, vol. 48, pp. 31–50, 2018.
[55] J. Yang, R.K. Das, and H. Li, "Extended Constant-Q Cepstral Coefficients for Detection of Spoofing Attacks," *Proc. Asia-Pacific Signal and Information Processing Association (APSIPA)*, 2018.
[56] H. Yu, Z. Ta, Z. Ma, R. Martin, and J. Guo, "Spoofing Detection in Automatic Speaker Verification Systems Using DNN Classifiers and Dynamic Acoustic Features," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 10, pp. 4633–4644, 2018.
[57] F. Tom, M. Jain, and P. Dey, "End-To-End Audio Replay Attack Detection Using Deep Convolutional Networks with Attention," *Proc. Interspeech*, 2018.
[58] J. Deng, W. Dong, R. Socher, L.J. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.
[59] C. Lai, A. Abad, K. Richmond, J. Yamagishi, N. Dehak, and S. King, "Attentive Filtering Networks For Audio Replay Attack Detection," *Proc. International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019.
[60] G. Valenti, H. Delgado, M. Todisco, N. Evans, and L. Pilati, "An end-to-end spoofing countermeasure for automatic speaker verification using evolving recurrent neural networks," *Proc. Odyssey*, 2018.
[61] Md.J. Alam, G. Bhattacharya, and P. Kenny, "Boosting the performance of spoofing detection systems on replay attacks using q-logarithm domain feature normalization," *Proc. Odyssey*, 2018.
[62] J. Yang and R.K. Das, "Low frequency frame-wise normalization over constant-Q transform for playback speech detection," *Digital Signal Processing*, 2019.
[63] M. Todisco, H. Delgado, K.A. Lee, Md Sahidullah, N. Evans, T. Kinnunen, and J. Yamagishi, "Integrated Presentation Attack Detection and Automatic Speaker Verification: Common Features and Gaussian Back-end Fusion," *Proc. Interspeech*, 2018.
[64] G. Lavrentyeva, S. Novoselov, A. Tseren, M. Volkova, A. Gorlanov, and A. Kozlov, "STC Antispoofing Systems for the ASVSpoof2019 Challenge," *arxiv:1904.05576v1*, 2019.
[65] C. Lai, N. Chen, J. Villalba, and N. Dehak, "ASSERT: Anti-Spoofing with Squeeze-Excitation and Residual neTworks", *arxiv:1904.01120v1*, 2019.
[66] J. Jung, H. Shim, H. Heo, and H.J. Yu, "Replay attack detection with complementary high-resolution information using end-to-end DNN for the ASVspoof 2019 Challenge," *arxiv:1904.10134v1*, 2019.
[67] 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," *Proc. Odyssey*, 2018.

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 for developing automatic statistical tools. Since late 2017 he holds an FPU fellowship for pursuing the Ph.D. degree with the Deparment of Signal Theory, Telematics and Communications at the University of Granada. His research activities focus on the processing, modelling, and classification of speech for human-oriented applications.

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 for Inisel as 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 has 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 (antispoofing for automatic speaker verification, speech enhancement, and robust speech recognition and transmission), image processing and proteomic signal processing. He has served as a Reviewer for a number of international journals and conferences, as evaluator for project and grant proposals, and as a Member of the technical program committee of several international conferences.

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.

Angel M. Gomez received the M.Sc. and Ph.D. degrees in Computer Science from the University of Granada, Spain, in 2001 and 2006, respectively. In 2002 he joined the Deparment of Signal Theory, Telematics, and Communications, University of Granada, where he is a member of the research group on Signal Processing, Multimedia Transmission and Speech/Audio Technologies (SigMAT). Currently he is an associate professor at the University of Granada. His research interests include human speech processing and machine learning.

2.2 Loss Functions for Neural Networks

2.2.1 Loss Function for ASV-Spoofing Detection

2.2.1.1 A Kernel Density Estimation Based Loss Function and Its Application to ASV-Spoofing Detection

- Alejandro Gomez-Alanis, Jose 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. 108530-108543, June 2020.
    - Status: Published.
    - Impact Factor (JCR 2020): 3.367
    - Subject Category: Engineering, Electrical & Electronic. Ranking 94/273 (Q2).
    - Subject Category: Computer Science, Information Systems. Ranking 65/162 (Q2).
    - Subject Category: Telecommunications. Ranking 36/91 (Q2).

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Digital Object Identifier

A Kernel Density Estimation Based Loss Function and Its Application to ASV-Spoofing Detection

ALEJANDRO GOMEZ-ALANIS[1], JOSE A. GONZALEZ-LOPEZ[1], AND ANTONIO M. PEINADO.[1], (Senior Member, IEEE)

[1]Department of Signal Processing, Telematics and Communications, University of Granada, Granada, 18071 Spain (e-mail: agomezalanis@ugr.es; joseangl@ugr.es; amp@ugr.es)

Corresponding author: Alejandro Gomez-Alanis (e-mail: agomezalanis@ugr.es).

ABSTRACT Biometric systems are exposed to spoofing attacks which may compromise their security, and voice biometrics, also known as automatic speaker verification (ASV), is no exception. Replay, synthesis and voice conversion attacks cause false acceptances that can be detected by anti-spoofing systems. Recently, deep neural networks (DNNs) which extract embedding vectors have shown superior performance than conventional systems in both ASV and anti-spoofing tasks. In this work, we develop a new concept of loss function for training DNNs which is based on kernel density estimation (KDE) techniques. The proposed loss functions estimate the probability density function (pdf) of every training class in each mini-batch, and compute a log likelihood matrix between the embedding vectors and pdfs of all training classes within the mini-batch in order to obtain the KDE-based loss. To evaluate our proposal for spoofing detection, experiments were carried out on the recent ASVspoof 2019 corpus, including both logical and physical access scenarios. The experimental results show that training a DNN based anti-spoofing system with our proposed loss functions clearly outperforms the performance of the same system being trained with other well-known loss functions. Moreover, the results also show that the proposed loss functions are effective for different types of neural network architectures.

INDEX TERMS Spoofing detection, kernel density estimation, loss function, deep learning, automatic speaker verification.

I. INTRODUCTION

BIOMETRIC authentication [1] aims to authenticate the identity claimed by a given individual based on samples measured from biological processes and/or organs (e.g., voice, fingerprint, face, etc). Voice biometrics, in particular, is an emerging form of biometric authentication with potential advantages given its hands-free, liveliness and dynamic nature. Automatic speaker verification (ASV) [2] is the conventional way to put voice biometrics into practical usage. ASV techniques verify the claimed identity of a given speaker by recording her/his voice, extracting voiceprints from the voice recordings, and deciding whether the speaker is who s/he claims to be based on the extracted voiceprints and a set of pre-stored voiceprints from enrolled users.

However, the vulnerability of ASV systems to malicious attacks is a serious concern nowadays [3]. Our focus in this work is on spoofing detection for ASV, where an impostor could gain fraudulent bypass to the authentication system by presenting speech resembling the voice of a genuine user. Four types of spoofing attacks have been identified [4]: (i) impersonation (i.e., mimicking the voice of a target speaker), (ii) replay (i.e., using pre-recorded voice of a target user), 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.

Spoofing detection or presentation attack detection (PAD in ISO/IEC 30107 nomenclature [5]) for ASV has become a hot research topic in recent years as evidenced by the organization of several evaluation campaigns (challenges) in this specific topic: (i) ASVspoof 2015 [6], which focused on logical access (LA) attacks (TTS and VC); (ii) ASVspoof 2017 [7], which focused on physical access (PA) attacks (replay attacks) under noisy environments; and (iii) ASVspoof 2019 [8], which addressed both the detection of LA attacks generated with the latest TTS and VC technologies, and simulated replay attacks under different reverberant acoustic

conditions. One of the main conclusions withdrawn from these challenges is that the use of deep neural networks (DNNs) for the extraction of spoofing-aware embedding vectors outperforms other conventional approaches for ASV anti-spoofing [9]–[13].

Within the DNN-based anti-spoofing framework, several recent studies have focused on designing new loss functions in order to make NNs more suitable for the specific tasks of anti-spoofing [14], ASV [15], [16] and/or their combination [17]. However, these studies do not usually address the following three issues. First, one particular characteristic of anti-spoofing applications, which is shared with ASV systems, is that embeddings extracted by DNNs should enable precise discrimination between *bona fide* speech and spoofed speech and, at the same time, they should be able to generalize well to unknown attacks that are not present in the training dataset. In other words, from a metric learning problem perspective [18]–[20], the goal is to learn a meaningful embedding representation that keeps similar training instances close to each other and the dissimilar instances far away on the embedding space. While specialized loss functions as the triplet network [18] specifically address this issue, conventional losses (e.g., softmax) fall short in achieving this goal. Second, in a supervised scenario, as is the case for DNN-based anti-spoofing detection, metric learning aims to learn a representation which keeps close the embeddings belonging to the same class. To represent each class, different representations have been investigated in the literature, such as representing each class by a centroid in the embedding space [15] or employing an anchor sample to represent the positive class [21]. In these representations, however, the training classes are not fully represented by all the samples in the mini-batch, but by a single embedding representation (i.e., either a centroid or an anchor sample), which may be suboptimal for distance learning. Third, recent loss functions, such as the siamese [14], generalized-end-to-end (GE2E) [15] and triplet loss [21] functions, are based on distance measures between embedding vectors. However, it is not straightforward to select the most appropiate distance measure as well as the embedding normalization technique. Moreover, these loss functions typically require the usage of an extra hyper-parameter called *margin* which is difficult to optimize.

To address all these issues, we propose a new probabilistic loss function for supervised metric learning, where every training class is represented with a probability density function (pdf) which is estimated through kernel density estimation (KDE) [22], [23] in each mini-batch. The mini-batches are formed so that all training classes are present in the mini-batch and are represented with the same number of samples. Due to the fact that KDE techniques place a probability mass at every sample, we can argue that each class is more accurately represented than in previous approaches, since KDE estimates a pdf per class using all the samples of the mini-batch rather than representing each class with a sole point (centroid or anchor point). Thus, we replace the concept of distance between embeddings by the concept that an embedding belongs to a certain class with a given probability. This has the advantage of avoiding the selection of an appropiate distance measure as well as an embedding normalization technique. Although the experiments supporting these aforementioned advantages of the proposed loss functions are focused on ASV anti-spoofing, they could be applied to different classification tasks.

This paper is organized as follows. Section II outlines the most popular loss functions used to train DNNs for developing ASV and anti-spoofing systems. Then, in Section III, we describe the proposed loss functions based on KDE. Section IV describes the speech corpora, neural networks and loss functions which are then evaluated in Section V for spoofing detection. Finally, we summarize the conclusions derived from this research in Section VI.

II. RELATED WORK

This section describes several loss functions that can be used in the context of distance metric learning in order to learn a meaningful embedding representation for the data samples assuming that the target labels are available a priori (i.e., supervised scenario). Some of these functions have already been successfully applied to either ASV or anti-spoofing.

In this section we use the following notation: $\boldsymbol{e}_{ji}$ denotes the embedding (output of a hidden layer of the DNN) of the i-th utterance of the class j, M is the number of utterances per class in the mini-batch, and N is the number of classes of the training set. In addition, we consider that every mini-batch is composed of $N \times M$ utterances. In anti-spoofing, the number of classes N is usually the number of training spoofing attacks plus the genuine class.

A. CROSS ENTROPY LOSS FUNCTION

The cross entropy loss, also known as softmax loss, is widely used to train DNNs for classification tasks. Typically, when the softmax loss function is used in ASV and anti-spoofing systems, embeddings are extracted from a middle or the last hidden layer of the DNN. Assuming this latter case, where embeddings are extracted from the last hidden layer of the DNN, the softmax loss function can be expressed as,

$$\mathcal{L}_{\text{softmax}} = \sum_{j=1}^{N}\sum_{i=1}^{M} -\log\frac{\exp(\boldsymbol{w}_j^T\boldsymbol{e}_{ji}+b_j)}{\sum_{k=1}^{N}\exp(\boldsymbol{w}_k^T\boldsymbol{e}_{ji}+b_k)}, \quad (1)$$

where $\boldsymbol{w} = [\boldsymbol{w}_1, ..., \boldsymbol{w}_N]$ and $\boldsymbol{b} = [b_1, ..., b_N]$ are the weight matrix and bias vector of the output layer, respectively.

B. ADDITIVE MARGIN LOSS FUNCTION

The additive margin (AM) softmax loss function [24] was proposed to replace the inner product operation of the softmax loss function in Eq. (1) with the cosine similarity operation in order to widen the inter-class margin in the embedding space [25]. The AM softmax loss function can be expressed as,

$$\mathcal{L}_{\text{AM}} = \sum_{j=1}^{N}\sum_{i=1}^{M} -\log\frac{\exp(s\cdot(cos(\boldsymbol{w}_j,\boldsymbol{e}_{ji})-m))}{\exp(s\cdot(cos(\boldsymbol{w}_j,\boldsymbol{e}_{ji})-m))+r_{ji}}, \tag{2}$$

$$r_{ji} = \sum_{\substack{k=1\\k\neq i}}^{N}\exp(s\cdot cos(\boldsymbol{w}_k,\boldsymbol{e}_{ji})), \tag{3}$$

where m is an additional margin and s is a scaling factor for stabilizing training. This loss function is a generalized version of the angular softmax loss [24]. Recently, this type of loss function has been successfully applied to anti-spoofing [26] and speaker verification systems [27], [28].

C. GENERALIZED END-TO-END LOSS FUNCTION

In the generalized end-to-end (GE2E) loss, which was originally proposed for ASV, each class (speaker) is represented by a centroid obtained averaging all the embeddings belonging to that class in the mini-batch. From those centroids, two loss functions were proposed in [15] which seek for minimizing the distance between the embeddings and their corresponding class centroids, while also maximizing the distance with the centroids from the other speakers. In anti-spoofing, the speakers are replaced by attacks. The distance between the embedding of the i-th utterance of the j-th attack ($\boldsymbol{e}_{ji}$) and the centroid of the k-th attack ($\hat{\boldsymbol{c}}_k$), is computed as:

$$\boldsymbol{S}_{ji,k} = \omega\cdot cos(\boldsymbol{e}_{ji},\hat{\boldsymbol{c}}_k)+b, \tag{4}$$

where ω and b are learnable parameters for score scaling and shifting, $\boldsymbol{S}$ is the similarity matrix, and the centroid embedding is computed by averaging the embeddings of each attack:

$$\hat{\boldsymbol{c}}_k = \frac{1}{M}\sum_{i=1}^{M}\boldsymbol{e}_{ki}. \tag{5}$$

The GE2E loss function consists of two losses which are computed using the values of the similarity matrix $\boldsymbol{S}$: (i) softmax loss, and (ii) contrast loss. The softmax loss of the embedding $\boldsymbol{e}_{ji}$ is expressed as follows,

$$\mathcal{L}_{\text{GE2E-softmax}}(\boldsymbol{e}_{ji}) = -\boldsymbol{S}_{ji,j} + \log\sum_{k=1}^{N}\exp(\boldsymbol{S}_{ji,k}). \tag{6}$$

Likewise, the contrast loss of the embedding $\boldsymbol{e}_{ji}$ is computed as,

$$\mathcal{L}_{\text{GE2E-contrast}}(\boldsymbol{e}_{ji}) = 1-\sigma(\boldsymbol{S}_{ji,j}) + \max_{\substack{1\leq k\leq N\\k\neq j}}\sigma(\boldsymbol{S}_{ji,k}), \tag{7}$$

where $\sigma(x)$ is the sigmoid function. This contrast loss function deserves some comments. For every utterance, exactly two components are added to the loss: (i) a positive component, which is associated with a positive match between the embedding $\boldsymbol{e}_{ji}$ and its true class centroid $\hat{\boldsymbol{c}}_j$; and (ii) a negative component, which is associated with a negative match between the embedding $\boldsymbol{e}_{ji}$ and the centroid $\hat{\boldsymbol{c}}_k$ with the highest similarity among all false class centroids.

Combining equations (6) and (7), the final GE2E loss function is the sum of the two losses over the similarity matrix:

$$\mathcal{L}_{\text{GE2E}} = \sum_{j=1}^{N}\sum_{i=1}^{M}\Big[\mathcal{L}_{\text{GE2E-softmax}}(\boldsymbol{e}_{ji}) + \mathcal{L}_{\text{GE2E-contrast}}(\boldsymbol{e}_{ji})\Big]. \tag{8}$$

D. SIAMESE LOSS FUNCTION

The siamese architecture processes two utterances at once using the same neural network, obtains two embeddings $\boldsymbol{e}_{ji}$ and $\boldsymbol{e}_{k\sim}$, and computes a loss based on the embedding distance:

$$\mathcal{L}_{\text{siamese}} = \sum_{j=1}^{N}\sum_{i=1}^{M}\delta_{jk}\cdot D(\boldsymbol{e}_{ji},\boldsymbol{e}_{k\sim}) + (1-\delta_{jk})\cdot\max(m, D(\boldsymbol{e}_{ji},\boldsymbol{e}_{k\sim})), \tag{9}$$

where $\mathbf{e}_{k\sim}$ denotes any embedding of the class k, $\delta_{jk}\in\{0,1\}$ is a label which indicates whether the embeddings $\mathbf{e}_{ji}$ and $\mathbf{e}_{k\sim}$ belong to the same class (i.e., when $k=j$), $D(\mathbf{e}_{ji},\mathbf{e}_{k\sim})$ is any distance measure between $\mathbf{e}_{ji}$ and $\mathbf{e}_{k\sim}$, and m is a hyper-parameter distance margin. There are many siamese network variants reported in the literature for different applications, such as face recognition [29], person identification [30], image recognition [31], etc.

E. TRIPLET LOSS FUNCTION

The triplet network [18] is a neural network architecture which attempts to learn an embedding representation of a multi-class labeled dataset which favours a small distance between example pairs labeled as similar, and large distances for pairs labeled as dissimilar. However, unlike the siamese networks, this architecture works with triplets of embeddings. In particular, it defines a loss function which ensures that an *anchor* embedding ($\boldsymbol{e}_{ji}$) of class j is closer to other *positive* samples ($\boldsymbol{e}_{jp}$, $p\neq i$) than to any *negative* sample ($\boldsymbol{e}_{n\sim}$, $n\neq j$) [21]. Thus, if we consider a batch size of $N\times M$ utterances, the triplet loss which is minimized is:

$$\mathcal{L}_{\text{triplet}} = \sum_{j=1}^{N}\sum_{i=1}^{M}\max\Big[\|\boldsymbol{e}_{ji}-\boldsymbol{e}_{jp}\|_2^2-\|\boldsymbol{e}_{ji}-\boldsymbol{e}_{n\sim}\|_2^2+\alpha,0\Big], \tag{10}$$

where α is a margin which is enforced between the positive and negative distances. Thus, given an anchor embedding $\boldsymbol{e}_{ji}$, its corresponding triplet $(\boldsymbol{e}_{ji},\boldsymbol{e}_{jp},\boldsymbol{e}_{n\sim})$ will be built with a *hard positive* embedding $\boldsymbol{e}_{jp}$ and a *hard negative* embedding $\boldsymbol{e}_{n\sim}$ such that indices p and n are selected according to the

following criteria: $p = argmax_{r \neq i} \ \|\boldsymbol{e}_{ji} - \boldsymbol{e}_{jr}\|_2^2$, and $n = argmin_{s \neq j} \ \|\boldsymbol{e}_{ji} - \boldsymbol{e}_{s\sim}\|_2^2$.

Recently, the triplet loss function has been successfully applied to train face verification systems [21], ASV systems [32], [33], and joint ASV and PAD systems [17].

III. KERNEL DENSITY ESTIMATION LOSS FUNCTION

In this section we describe the proposed loss functions based on KDE for training DNN-based embedding extraction systems. Section III-A describes the computation of the log likelihood matrix employed by all the proposed losses. After that, the proposed KDE-based loss functions are described in Section III-B.

A. KDE-BASED LOG LIKELIHOOD MATRIX

Similarly to the GE2E loss method described in Section II-C, every mini-batch consists of $N \times M$ utterances from the N different training classes (genuine class and $N-1$ spoofing attacks), and each class is represented with M utterances. Thus, each utterance i $(1 \leq i \leq M)$ from the training class j $(1 \leq j \leq N)$, represented by its sequence of feature vectors $\boldsymbol{X}_{ji}$, is fed into a neural network in order to obtain the embedding vector $\boldsymbol{e}_{ji} = g(\boldsymbol{X}_{ji}; \boldsymbol{\Theta})$, where $\boldsymbol{\Theta}$ represents all the parameters of the neural network.

Let the embedding vectors from the k-th training class $\boldsymbol{e}_{k1}, ..., \boldsymbol{e}_{kM} \in \mathcal{R}^q$ be independent and identically distributed random samples from an unknown distribution $f_k(\boldsymbol{e})$. The estimation of its multivariate pdf using KDE [22], [23] is given by,

$$\hat{f}_k(\boldsymbol{e}) = \frac{1}{M}\sum_{m=1}^{M} \frac{1}{\det(\boldsymbol{H}_k)} K\big(\boldsymbol{H}_k^{-1}(\boldsymbol{e} - \boldsymbol{e}_{km})\big) = \frac{1}{M}\sum_{m=1}^{M} K_{\boldsymbol{H}_k}\big(\boldsymbol{e} - \boldsymbol{e}_{km}\big), \quad (11)$$

where $K(\cdot)$ is the kernel function, $\boldsymbol{H}_k$ is a nonsigular and symmetric bandwidth matrix [34], [35], and $K_{\boldsymbol{H}_k}(\boldsymbol{u}) = K(\boldsymbol{H}_k^{-1}\boldsymbol{u})/\det(\boldsymbol{H}_k)$. A range of kernel functions are commonly used, such as uniform, triangular, Gaussian and Epanechnikov [36]. For instance, the probability density function with a Gaussian kernel (that is, $K_{\boldsymbol{H}_k}(\boldsymbol{u}) = \mathcal{N}(\boldsymbol{u}; \boldsymbol{0}, \boldsymbol{\Sigma}_k)$) can be computed as,

$$\hat{f}_k(\boldsymbol{e}) = \frac{1}{M}\sum_{m=1}^{M} \mathcal{N}(\boldsymbol{e}; \boldsymbol{\mu}_k = \boldsymbol{e}_{km}, \boldsymbol{\Sigma}_k = \sigma_k^2 \cdot \boldsymbol{I}), \quad (12)$$

where $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ are the mean vector and covariance matrix of the Gaussian distribution $\mathcal{N}(\cdot)$, $\boldsymbol{I}$ is the identity matrix, and σ_k^2 represents the bandwidth of the KDE model for class k. Every class has its corresponding bandwidth, which is a learnable parameter that is constrained to be positive ($\sigma_k^2 > 0$). In this way, the kernel density estimator $\hat{f}_k(\boldsymbol{e})$ places a probability mass at each observation embedding $\boldsymbol{e}_{km}$ according to a Gaussian probability model.

Once all the probability density functions of the considered mini-batch have been built, they are evaluated for every embedding belonging to that mini-batch. That is, all possible $\hat{f}_k(\boldsymbol{e}_{ji})$ $(k, j = 1, ...N; i = 1, ..., M)$ are computed. Then, these probabilities are arranged in the following log-likelihood matrix:

$$\mathbf{L}_{ji,k} = \begin{cases} \log\Big(\frac{1}{M}\sum\limits_{m=1}^{M} K_{\boldsymbol{H}_k}\big(\boldsymbol{e}_{ji} - \boldsymbol{e}_{km}\big)\Big) & k \neq j \\ \log\Big(\frac{1}{M-1}\sum\limits_{\substack{m=1 \\ m\neq i}}^{M} K_{\boldsymbol{H}_k}\big(\boldsymbol{e}_{ji} - \boldsymbol{e}_{km}\big)\Big) & k = j \end{cases}. \quad (13)$$

To avoid trivial solutions and make training stable, the embedding vector $\boldsymbol{e}_{ji}$ is removed when estimating the density function of the true class (i.e., when $k = j$ in Eq. (13)). Fig. 1 illustrates the whole process for obtaining the log likelihood matrix with input features, embedding vectors and likelihoods from different training classes (genuine and spoofing attacks), represented by different colors.

From the log likelihood matrix $\mathbf{L}_{ji,k}$ in Eq. (13), we strive to achieve two goals simultaneously during the DNN training. First, we aim at maximizing the probability of each embedding vector $\mathbf{e}_{ji}$ belonging to its class j, that is,

$$\underset{\boldsymbol{\Theta}}{\text{maximize}} \quad \mathbf{L}_{ji,j} = \log \hat{f}_j(\boldsymbol{e}_{ji}), \quad (14)$$

where $\boldsymbol{\Theta}$ are the neural network parameters to be optimized in the training stage. At the same time, the probability of each embedding vector $\boldsymbol{e}_{ji}$ belonging to the rest of the classes should be minimized:

$$\underset{\boldsymbol{\Theta}}{\text{minimize}} \quad \mathbf{L}_{ji,k} = \log \hat{f}_k(\boldsymbol{e}_{ji}) \ (k \neq j). \quad (15)$$

In other words, as depicted in Fig. 1, we strive to find the optimum set of weights $\boldsymbol{\Theta}$ that results in large log likelihood values for red cells in the figure and small values for the blue cells in the figure. We achieve these two simultaneous goals by means of three alternative loss functions, as described in the next section.

B. KDE-BASED LOSS FUNCTIONS

There are several ways to implement the requirements described above. In this section, we describe three alternative losses to achieve our goal during the training of the neural network: softmax, contrast and triplet KDE based losses.

1) KDE-Softmax Loss

As described in Section II-A, the softmax function is typically used in tandem with the negative log-likelihood (NLL), such that: $L(\mathbf{y}) = -\log(\text{softmax}(\mathbf{y}))$. The output of the softmax function can be interpreted as the probabilities that a certain set of features belong to a certain class, which is combined with the NLL in order to build the popular cross-entropy or softmax loss.

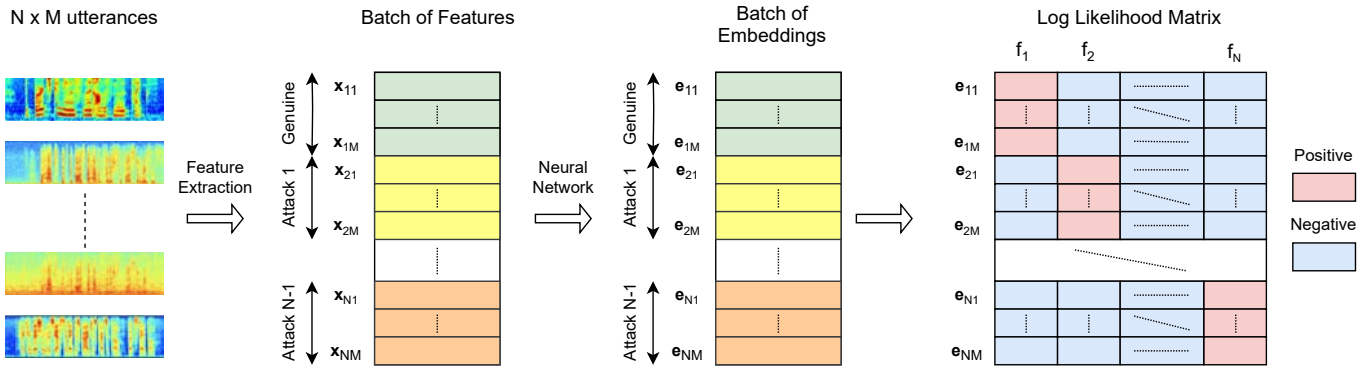

FIGURE 1. System overview for computing the log likelihood matrix of a mini-batch of $N \times M$ utterances.

The softmax loss can be directly applied to KDE using the log likelihood matrix, such that:

$$\mathcal{L}_{\text{KDE-softmax}} = \sum_{j=1}^{N}\sum_{i=1}^{M}\Big[-\mathbf{L}_{ji,j} + \log\sum_{k=1}^{N}\exp\big(\mathbf{L}_{ji,k}\big)\Big]. \quad (16)$$

This loss function tries to increase the probability of each embedding belonging to its true class, while minimizing the probability of the embedding belonging to the rest of the classes.

2) KDE-Contrast Loss

The contrast loss is formed by two terms: (i) a positive term, which is the probability of the embedding $\boldsymbol{e}_{ji}$ belonging to the true class; and (ii) a hard negative term, which is the highest probability of that embedding belonging to any of the negative classes, that is,

$$\mathcal{L}_{\text{KDE-contrast}} = \sum_{j=1}^{N}\sum_{i=1}^{M}\max\Big[\big(-\mathbf{L}_{ji,j} + \max_{\substack{1\le k\le N\\ k\neq j}}\mathbf{L}_{ji,k}\big), 0\Big]. \quad (17)$$

3) KDE-Triplet Loss

In the following we describe the adaptation of the triplet loss to our KDE-based framework. Similarly to the triplet loss, we want to find an embedding representation that, for a given anchor embedding $\boldsymbol{e}_{ji}$, the probability of such embedding to the positive class j is large, whereas the probability of a negative exemplar of belonging to the same class is small. While this loss is motivated in [21] in the context of nearest-neighbour classification [37], here the quadratic distances are replaced by log likelihoods.

This loss tries to ensure that an embedding vector $\boldsymbol{e}_{ji}$ (*anchor*) of a specific class j (*positive class*) obtains a higher probability of belonging to that class than any other embedding vector $\boldsymbol{e}_{n\sim}$ (*negative*) from other class ($n \neq j$). In this way, the triplet is formed by: (i) an *anchor embedding* $\boldsymbol{e}_{ji}$, (ii) a *negative embedding* $\boldsymbol{e}_{n\sim}$, and (iii) a *positive* estimated density function $\hat{f}_j$.

Thus, this loss tries to ensure

$$\hat{f}_j(\boldsymbol{e}_{n\sim}) + \alpha < \hat{f}_j(\boldsymbol{e}_{ji}), \quad (18)$$

where α is a margin that is enforced between the true and false positive probabilities. Using the log likelihood matrix of Eq. (13), the loss which is minimized is

$$\mathcal{L}_{\text{KDE-triplet}} = \sum_{j=1}^{N}\sum_{i=1}^{M}\max\Big[\mathbf{L}_{n\sim,j} - \mathbf{L}_{ji,j} + \alpha, 0\Big], \quad (19)$$

where α is a hyper-parameter margin which is enforced between the positive and negative likelihoods.

Generating all possible triplets would result in many of them being easily satisfied (i.e., fulfill constraint (18)). Thus, not all of them would contribute to the training, which might result in a slower convergence. Therefore, it is crucial to select hard triplets which do not fulfill constraint (18), and can therefore contribute to improving the model. As suggested in [21], instead of picking the hardest positives, we use all anchor-positive pairs within the mini-batch. In addition, [21] shows that selecting the hardest negatives can in practice lead to a bad local minima in training. In order to mitigate this, we select *semi-hard* negative exemplars which lie inside the margin α [21]:

$$\hat{f}_j(\boldsymbol{e}_{n\sim}) < \hat{f}_j(\boldsymbol{e}_{ji}) < \hat{f}_j(\boldsymbol{e}_{n\sim}) + \alpha. \quad (20)$$

4) Analysis and relation with other loss functions

From equations (16), (17) and (19), we can observe that the KDE softmax, contrast and triplet loss functions have in common the term $-\mathbf{L}_{ji,j}$, which aims at maximizing the probability of the embedding $\boldsymbol{e}_{ji}$ belonging to the estimated density function of the true class. The difference between these three loss functions lies in the penalization term, which tries to separate the positive class j from the rest of training classes. Specifically, the penalization term of these loss functions is:

- KDE-softmax loss: the sum of the likelihoods that the embedding vector $\boldsymbol{e}_{ji}$ belongs to all training classes.

- KDE-contrast loss: the highest log likelihood between the embedding vector $\boldsymbol{e}_{ji}$ and any negative class.
- KDE-triplet loss: the log likelihood that a negative embedding vector $\boldsymbol{e}_{n\sim}$ belongs to the anchor class j, plus a margin α.

If we combine the KDE softmax and contrast loss functions (Eqs. (16) and (17)), we can derive a probabilistic version of the GE2E loss described in Section II-C, which we call it as full kernel density estimation (FKDE) loss, that is,

$$\mathcal{L}_{\text{FKDE}} = \sum_{j=1}^{N}\sum_{i=1}^{M}\Big[\mathcal{L}_{\text{KDE-softmax}}(\boldsymbol{e}_{ji}) + \mathcal{L}_{\text{KDE-contrast}}(\boldsymbol{e}_{ji})\Big]. \tag{21}$$

However, while the G2E2 technique computes a cosine similarity matrix, our proposed FKDE loss computes a log likelihood matrix. Furthermore, the GE2E technique represents each class by means of a centroid, while our technique estimates a pdf for each class. From a clustering point of view, we argue that the latter is a superior and more informative representation.

On the other hand, the KDE-triplet loss function in (19) can be shown to be a generalization of the classical triplet loss in (10) when KDE with Gaussian kernel (GKDE) and diagonal covariance matrix is employed. In fact, if we only consider an embedding $\boldsymbol{e}_{ji}$ for estimating the probability density function in (12), and we introduce a positive index p such that $1 \leq p \leq M$, $p \neq i$, the GKDE triplet loss in (19) would become:

$$\mathcal{L} = \sum_{j=1}^{N}\sum_{i=1}^{M}\max\Big[\mathbf{L}_{n\sim,j(i)} - \mathbf{L}_{jp,j(i)} + \alpha, 0\Big] = \sum_{j=1}^{N}\sum_{i=1}^{M}\max\Big[\log\hat{f}_j^{(i)}(\boldsymbol{e}_{n\sim}) - \log\hat{f}_j^{(i)}(\boldsymbol{e}_{jp}) + \alpha, 0\Big], \tag{22}$$

where,

$$\log\hat{f}_j^{(i)}(\boldsymbol{e}) = \log\frac{\exp\big(-\frac{1}{2}(\boldsymbol{e}-\boldsymbol{e}_{ji})^T\boldsymbol{\Sigma}_j^{-1}(\boldsymbol{e}-\boldsymbol{e}_{ji})\big)}{(2\pi)^{q/2}|\boldsymbol{\Sigma}_j|^2}, \tag{23}$$

and q is the embedding size. Since we consider a diagonal covariance matrix $\boldsymbol{\Sigma}_j = \sigma_j^2 \cdot \boldsymbol{I}$, this log probability density function can be simplified to:

$$\log\hat{f}_j^{(i)}(\boldsymbol{e}) = -\frac{q}{2}\log(2\pi\sigma_j^2) - \frac{1}{2\sigma_j^2}\left\|\boldsymbol{e}-\boldsymbol{e}_{ji}\right\|_2^2. \tag{24}$$

Finally, if we consider a constant bandwidth for the GKDE $\sigma_j^2 = 1$, and substitute (24) into (22), the modified version of the proposed GKDE triplet loss equals the classical triplet loss function:

TABLE 1. Structure of the ASVspoof2019 data corpus divided by the training, development and evaluation sets [8].

Subset	#speakers		#utterances			
	Male	Female	Logical Access		Physical Access	
			Bona fide	Spoof	Bona fide	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

$$\mathcal{L} = \sum_{j=1}^{N}\sum_{i=1}^{M}\max\Big[\left\|\boldsymbol{e}_{ji}-\boldsymbol{e}_{jp}\right\|_2^2 - \left\|\boldsymbol{e}_{ji}-\boldsymbol{e}_{n\sim}\right\|_2^2 + \alpha, 0\Big]. \tag{25}$$

To sum up, the combination of the KDE softmax and contrast loss functions results in a probabilistic version of the GE2E loss. In addition, the GKDE triplet loss is a generalized version of the classical triplet loss.

IV. EXPERIMENTAL SETUP

This section is organized as follows. First, the speech corpora which was employed for the evaluation of the proposed techniques is described. Then, Section IV-B outlines the system configuration and network training. After that, Section IV-C provides the implementation details of the the loss functions that are evaluated, including our proposals and other well-known losses from the literature. Finally, the performance metrics employed to evaluate the performance of the anti-spoofing system are discussed.

A. SPEECH CORPORA

We conducted experiments on the recent ASVspoof 2019 database [8] which encompasses two partitions for the assessment of logical and physical access scenarios. A summary of their composition in terms of speakers and number of utterances is presented in Table 1.

1) ASVspoof 2019 Logical Access Corpus

The LA database contains *bona fide* speech and spoofed speech data generated using 17 TTS and VC systems. Six of these systems are designated as known attacks, with the other 11 being designated as unknown attacks. The training and development sets only contain known attacks, whereas the evaluation set contains 2 known and 11 unknown spoofing attacks. Among the 6 known attacks there are 2 VC systems and 4 TTS systems. VC systems use a neural-network-based and spectral-filtering-based approaches [38]. TTS systems use either waveform concatenation or neural-network-based speech synthesis using a conventional source-filter vocoder [39] or a WaveNet based vocoder [40]. The 11 unknown systems comprise 2 VC, 6 TTS and 3 hybrid TTS-VC systems and were implemented with various waveform generation methods including classical vocoding, GriffinLim [41], generative adversarial networks [42], neural waveform

models [43], waveform concatenation, waveform filtering [44], spectral filtering, and their combination.

2) ASVspoof 2019 Physical Access Corpus

The PA database contains *bona fide* speech and spoofed speech data generated according to a simulation of their presentation to the microphone of an ASV system within a reverberant acoustic environment. Training and development data is created by simulating 27 different acoustic and 9 different replay configurations. Acoustic configurations comprise an exhaustive combination of 3 categories of room sizes, 3 categories of reverberation and 3 categories of speaker-to-ASV microphone distances. Replay configurations comprise 3 categories of attacker-to-talker recording distances, and 3 categories of loudspeaker quality. Evaluation data is generated in the same manner as training and development data, albeit with different, random acoustic and replay configurations. Thus, the set of room sizes, levels of reverberation, speaker-to-ASV microphone distances, attacker-to-talker recording distances and loudspeaker qualities, are different from those of training and development.

B. SYSTEM DESCRIPTION

This section provides a detailed description of the implemented systems:

1) Spectral Analysis

Speech signals were analyzed using a Blackman analysis window of 25 ms length with 10 ms of frame shift. Log magnitude spectrogram features (STFT) with 256 frequency bins were obtained to feed the neural network. No normalization was applied to the input features.

We considered two techniques for obtaining an unified time-frequency (T-F) shape of features. First, we truncated the spectrum along the time axis with a fixed size of $T = 400$ frames in order to feed a convolutional neural network (CNN). During this procedure, short utterances were extended by repeating their contents if necessary to match the required length. Second, we used a sliding window approach of $W = 32$ frames with a shift of $\delta = 12$ frames in order to feed a RNN.

2) Light Convolutional Neural Network

A simplified version of the recently proposed Light Convolutional Neural Network (LCNN) [26] was employed in most of our experiments, which is an architecture that has demonstrated to be very effective to detect spoofed speech in the last two ASVspoof challenges [26], [45]. It was the best system of the ASVspoof 2017 challenge [45], and the best single system in the LA scenario of the ASVspoof 2019 challenge [26].

Table 2 details the architecture of the LCNN used in our experiments. In this model we truncated the spectrum of the utterances to a fixed size of $T = 400$ frames. As can be seen, the specific charasteristic of the LCNN architecture [10] is the usage of the Max-Feature-Map activation (MFM) which is based on the Maxout activation function [46]. Thus, the LCNN is composed of 7 convolutional layers with MFM activation, 4 max-pooling layers with kernel of size 2×2 and stride of size 2×2 in order to reduce both time and frequency dimension, 6 batch normalization layers in order

TABLE 2. LCNN architecture used in the experiments. MFM stands for Max Feature Map activation. FC stands for Fully Connected layer. "q" denotes the dimension of the embedding vectors extracted by the LCNN.

Type	Filter / Stride	Output size	# Parameters
Conv.	5×5 / 1×1	$256 \times 400 \times 16$	416
MFM	-	$256 \times 400 \times 8$	-
MaxPool	2×2 / 2×2	$128 \times 200 \times 8$	-
Batch Norm.	-	$128 \times 200 \times 8$	-
Conv.	1×1 / 1×1	$128 \times 200 \times 16$	144
MFM	-	$128 \times 200 \times 8$	-
Batch norm.	-	$128 \times 200 \times 8$	-
Conv.	3×3 / 1×1	$128 \times 200 \times 32$	2336
MFM	-	$128 \times 200 \times 16$	-
MaxPool	2×2 / 2×2	$64 \times 100 \times 16$	-
Batch norm.	-	$64 \times 100 \times 16$	-
Conv.	1×1 / 1×1	$64 \times 100 \times 32$	544
MFM	-	$64 \times 100 \times 16$	-
Batch norm.	-	$64 \times 100 \times 16$	-
Conv.	3×3 / 1×1	$64 \times 100 \times 32$	4640
MFM	-	$64 \times 100 \times 16$	-
MaxPool	2×2 / 2×2	$32 \times 50 \times 16$	-
Batch norm.	-	$32 \times 50 \times 16$	-
Conv.	1×1 / 1×1	$32 \times 50 \times 32$	544
MFM	-	$32 \times 50 \times 16$	-
Batch norm.	-	$32 \times 50 \times 16$	-
Conv.	3×3 / 1×1	$32 \times 50 \times 32$	4640
MFM	-	$32 \times 50 \times 16$	-
MaxPool	2×2 / 2×2	$16 \times 25 \times 16$	-
FC	-	$2 \times q$	$12800 \times q$
MFM	-	q	-

TABLE 3. LC-GRNN architecture used in the experiments. MFM stands for Max Feature Map activation. FC stands for Fully Connected layer. "q" denotes the dimension of the embedding vectors extracted by the LC-GRNN.

RNN	Type	Filter / Stride	Output	# Parameters
Layer 1	Conv.	5×5 / 1×1	$256 \times 32 \times 16$	2496
	MFM	-	$256 \times 32 \times 8$	-
	MaxPool	2×1 / 2×1	$128 \times 32 \times 8$	-
	Batch Norm.	-	$128 \times 32 \times 8$	-
Layer 2	Conv.	1×1 / 1×1	$128 \times 32 \times 16$	864
	MFM	-	$128 \times 32 \times 8$	-
	Conv.	3×3 / 1×1	$128 \times 32 \times 32$	7008
	MFM	-	$128 \times 32 \times 16$	-
	MaxPool	2×1 / 2×1	$64 \times 32 \times 16$	-
	Batch Norm.	-	$64 \times 32 \times 16$	-
Layer 3	Conv.	1×1 / 1×1	$64 \times 32 \times 32$	3264
	MFM	-	$64 \times 32 \times 16$	-
	Conv.	3×3 / 1×1	$64 \times 32 \times 16$	6690
	MFM	-	$64 \times 32 \times 8$	-
	MaxPool	2×1 / 2×1	$32 \times 32 \times 8$	-
	Batch Norm.	-	$32 \times 32 \times 8$	-
-	FC	-	$2 \times q$	$16384 \times q$
	MFM	-	q	-

to increase the stability and convergence speed during the training process, and one fully connected layer with MFM activation where the embedding vectors are extracted.

3) Light Convolutional Gated Recurrent Neural Network

We also used the Light Convolutional Gated Recurrent Neural Network (LC-GRNN) that we proposed in our previous works [9], [47]. It was one of the ten top performing single systems of the ASVspoof 2019 challenge [8]. This archictecture, in contrast to the LCNN described above, is based on a RNN, thus, having the potential advantage that there is no need to truncate the utterance to extract the embeddings.

Table 3 shows a summary of the LC-GRNN architecture. It processes context windows of $W = 32$ frames with a shift of $\delta = 12$ frames. It consists of 3 recurrent layers, where each one has different light convolutional layers followed by a max-pooling operation which reduces the frequency dimension. Also, batch normalization is applied in order to increase the stability and convergence speed of the training process. Once all the frame-level context windows are processed by the convolutional and recurrent layers, 8 feature maps of size 32×32 are flattened to make up a feature vector of 8192 components. Then, this vector is fed to a fully connected layer with MFM activation to obtain the embedding vector of the utterance.

4) Training setup

The neural networks were trained using the Adam optimizer [48] with a learning rate of $3 \cdot 10^{-4}$. Also, early stopping was applied when no improvement of the loss on the validation set was obtained after five epochs. To prevent the problem of overfitting, a 60% dropout was applied in the fully connected layer of the two models. All the specified hyperparameters of the systems were optimized using the validation set of the data corpora. The Pytorch toolkit [49] was employed to implement the deep learning framework.

5) Final classifier

The embeddings extracted from the utterances were finally processed by a classifier, which produces a score per utterance, indicating whether the utterance is genuine or spoofed. Based on the results from our previous works [9], [47], we used a probabilistic linear discriminant analysis (PLDA). We also applied a posterior normalization of the scores. Provided the prior of the different classes is uniform, the normalized score of the embedding vector $\boldsymbol{e}$ is

$$p(\text{genuine}|\boldsymbol{e}) = \log \frac{\exp(p(\boldsymbol{e}|\text{genuine}))}{\sum_{j=1}^{N} \exp(p(\boldsymbol{e}|j))}, \qquad (26)$$

where $p(\boldsymbol{e}|j)$ is the log posterior predictive probability of the embedding vector $\boldsymbol{e}$ given class j ($j = 1, ..., N$).

C. LOSS FUNCTIONS

This section details the usage and hyper-parameters of the different loss functions employed to train the LCNN and LC-GRNN models. We used $N = 7$ and $N = 2$ training classes in the LA and PA scenarios, respectively. In the LA scenario, we used the 6 known spoofing attacks and the genuine class. In the PA scenario, we only used 2 classes: genuine and spoofed speech.

1) Cross entropy or softmax loss

This loss processes the embedding vectors with an additional fully connected layer with softmax activation of N neurons to discriminate between the genuine and the $N - 1$ spoofing classes of the training set. After that, it applies the NLL to build the cross-entropy or softmax loss.

2) Additive margin loss

In our preliminary experiments we evaluated the *cosface* [50], *arcface* [51] and *sphereface* [24] versions of the additive margin loss. The difference between them lies in the additional margin $m = 30°, 64°, 64°$ and the scaling factor $s = 0.4, 0.5, 1.35$, respectively. The best performance in the preliminary experiments was obtained with the *cosface* version, so that we evaluated it in the rest of the experiments as angular softmax loss.

3) Generalized end-to-end (GE2E) loss

The number of training classes (N) is equal to the number of spoofing attacks of the training set plus the genuine class. We evaluated two versions of the GE2E loss: (i) GE2E with only the softmax loss, and (ii) GE2E with the softmax and contrast losses together, as it is indicated in Eq. (8).

4) Siamese loss

We evaluated a siamese variant called siamese-classification hybrid architecture [52], which has been successfully applied for replay spoofing detection [14]. This siamese network was trained by outputting a softmax layer over the two targets: *similar* and *dissimilar* input pairs. Thus, the network was trained to identify genuine-genuine or spoof-spoof speech as *similar* input pairs, and genuine-spoof pairs as *dissimilar* inputs.

5) Triplet loss

We evaluated the triplet loss using all anchor-positive pairs of the mini-batch and selecting the *semi-hard* negative utterances which lie inside the margin $\alpha = 1.0$, as shown in Eq. (10).

6) KDE-based loss functions

We computed the log likelihood matrix $\mathbf{L}_{ji,k}$ for every mini-batch and evaluated three KDE-based loss functions: (i) KDE softmax from Eq. (16), (ii) combination of KDE softmax and contrast from Eq. (21), and (iii) KDE triplet from Eq. (19). We evaluated them using different types of kernel functions, as it is discussed in Section V-A1. In the KDE triplet loss, we used all anchor-positive pairs of the mini-batch and selected the *semi-hard* negative utterances which lie inside the margin $\alpha = 1.0$.

TABLE 4. Results on ASVspoof 2019 logical and physical access test sets in terms of EER (%) of the LCNN based anti-spoofing system trained using KDE based loss functions (embeddings size of 32 and batch size of 140) with different kernel functions and optimizable bandwidths.

Loss Function	EER (%) (Logical Access / Physical Access)			
	Uniform	Triangular	Epanechnikov	Gaussian
KDE - Softmax	5.38 / 2.71	5.17 / 2.46	5.09 / 2.26	5.04 / 2.32
KDE - Softmax + Contrast	5.19 / 2.16	5.05 / 2.04	4.93 / 1.85	4.85 / 1.73
KDE - Triplet	4.97 / 1.97	4.76 / 1.82	4.65 / 1.74	**4.56 / 1.65**

TABLE 5. Results on ASVspoof 2019 logical and physical access test sets in terms of EER (%) of the LCNN based anti-spoofing system trained using GKDE based loss functions (embeddings size of 32 and batch size of 140) with fixed and optimizable bandwidths.

Loss Function	EER (%) (Logical Access / Physical Access)			
	$\sigma_k^2 = 0.5$	$\sigma_k^2 = 1.0$	$\sigma_k^2 = 2.0$	Learnable σ_k^2
GKDE - Softmax	5.91 / 3.37	5.57 / 2.92	6.06 / 3.59	5.04 / 2.32
GKDE - Softmax + Contrast	5.86 / 2.81	5.65 / 2.42	5.91 / 2.97	4.85 / 1.73
GKDE - Triplet	5.49 / 2.62	5.24 / 2.28	5.57 / 2.70	**4.56 / 1.65**

D. PERFORMANCE METRICS

The evaluation of the anti-spoofing system is done in terms of the pooled equal error rate (EER) across all attacks, and the minimum normalized tandem detection cost function (min-tDCF) [53] for both the LA and PA scenarios, separately.

V. EXPERIMENTAL RESULTS

This section presents the results from the evaluation on the ASVspoof 2019 corpus. First, Section V-A evaluates the performance on the LA and PA evaluation sets of the anti-spoofing system based on a LCNN, which is trained using different embedding sizes, batch sizes and training techniques. Then, Section V-B is devoted to evaluate the performance of the anti-spoofing system based on a more complex neural network (LC-GRNN), which is trained with the proposed loss functions, and its performance is compared to other state-of-the-art systems.

A. LCNN RESULTS

1) Evaluation of the kernel function

The objective of this experiment is to analyze the performance of the proposed KDE loss functions when using different types of kernel functions. Table 4 reports the EERs obtained when training the LCNN with the proposed KDE based loss functions, with different kernels and using learnable bandwidths per training class (see next section for more details about the optimization of the bandwith). From our preliminary experiments, we chose an embedding size of 32 and a batch size of 140. As can be seen, the best performance is obtained with the Gaussian kernel, followed by the Epanechnikov [36], triangular [54] and uniform [54] kernels, respectively. The maximum difference of EER is 0.34 and 0.43%, which is achieved when comparing the uniform and Gaussian kernels in the KDE softmax and contrast loss function on the LA and PA scenarios, respectively. This means that there are no large differences of performance when employing different kernels. Since the Gaussian kernel obtains the best results, we will use it in the rest of the paper, and the resulting loss function will be referred to as Gaussian kernel density estimation (GKDE) based loss function.

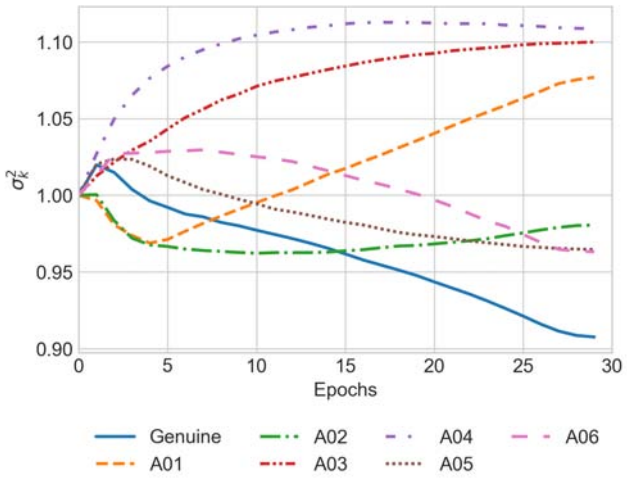

FIGURE 2. Class bandwidths optimization along the training process of the GKDE softmax loss function in the LA evaluation scenario.

2) Evaluation of the GKDE bandwidth

Next, we evaluate the performance achieved by the GKDE losses when using either fixed bandwiths σ_k^2 (0.5, 1.0 and 2.0) or learnable bandwidths, which are optimized along with the rest of parameters of the LCNN. As can be seen in Table 5, using a fixed bandwidth of $\sigma_k^2 = 1.0$ slightly achieves a better performance than using fixed bandwidths of $\sigma_k^2 = 0.5$ and $\sigma_k^2 = 2.0$. This can be due to the effect of under-smoothing and over-smoothing when using small and large bandwidths, respectively. However, the best performance is always obtained when the class bandwidths are optimized along with the rest of parameters of the neural network. For instance, optimizing the bandwidths with the rest of parameters overcomes the fixed bandwidth of $\sigma_k^2 = 1.0$ by an absolute EER of 0.68 and 0.63 % when evaluating the GKDE triplet loss function on the LA and PA scenarios, respectively.

Fig. 2 shows the optimization process of the class bandwidths through the different epochs when training the LCNN for the LA scenario. Despite the values for the different classes are not very different, the bandwidth of the genuine class is the one which achieves the smallest value, followed

TABLE 6. Results on ASVspoof 2019 logical and physical access test sets in terms of EER (%) and min-tDCF of the LCNN based anti-spoofing system trained using different loss functions and embedding sizes, and a batch size of 280 utterances.

Loss Function	Logical Access Results (EER (%) / min-tDCF)			Physical Access Results (EER (%) / min-tDCF)		
	Emb. Size: 16	Emb. Size: 32	Emb. Size: 64	Emb. Size: 16	Emb. Size: 32	Emb. Size: 64
Softmax	5.84 / 0.1106	5.76 / 0.1078	6.10 / 0.1124	3.46 / 0.0975	3.22 / 0.0885	3.38 / 0.0984
Angular Softmax	6.04 / 0.1118	5.87 / 0.1084	6.21 / 0.1158	3.54 / 0.1003	3.16 / 0.0878	3.32 / 0.0917
Siamese	5.88 / 0.1062	5.64 / 0.1044	6.01 / 0.1087	4.02 / 0.1063	3.67 / 0.1021	3.51 / 0.0976
Triplet	5.38 / 0.1025	5.15 / 0.1001	5.42 / 0.1034	2.70 / 0.0822	2.54 / 0.0806	2.81 / 0.0828
GE2E - Softmax	6.74 / 0.1212	6.39 / 0.1138	6.89 / 0.1296	4.39 / 0.1151	4.26 / 0.1102	4.52 / 0.1164
GE2E - Softmax + Contrast	6.34 / 0.1178	6.14 / 0.1127	6.44 / 0.1202	3.94 / 0.1041	3.78 / 0.1008	4.11 / 0.1086
GKDE - Softmax	4.76 / 0.1009	4.42 / 0.0935	4.64 / 0.0994	2.19 / 0.0735	1.97 / 0.0711	2.28 / 0.0756
GKDE - Softmax + Contrast	4.51 / 0.0961	4.04 / 0.0905	**4.32 / 0.0903**	1.56 / 0.0517	1.40 / 0.0465	1.74 / **0.0548**
GKDE - Triplet	**4.28 / 0.0911**	**3.84 / 0.0857**	4.35 / 0.0943	**1.51 / 0.0486**	**1.34 / 0.0452**	**1.65** / 0.0557

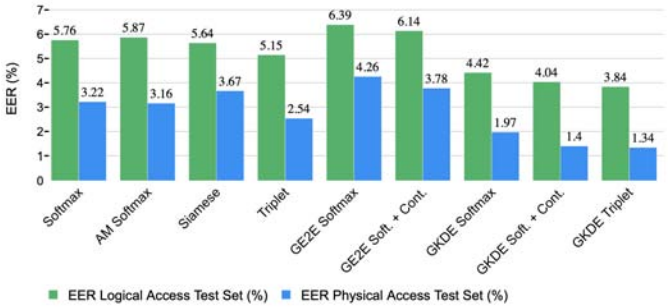

FIGURE 3. Bar plot of pooled EERs (%) evaluated in the logical and physical access test sets when the LCNN (embedding and batch size: 32 and 280, respectively) is trained with different techniques: (i) softmax; (ii) angular softmax; (iii) triplet loss; (iv) GE2E softmax; (v) GE2E softmax + contrast; (vi) GKDE softmax; (vii) GKDE softmax + contrast; (viii) GKDE triplet.

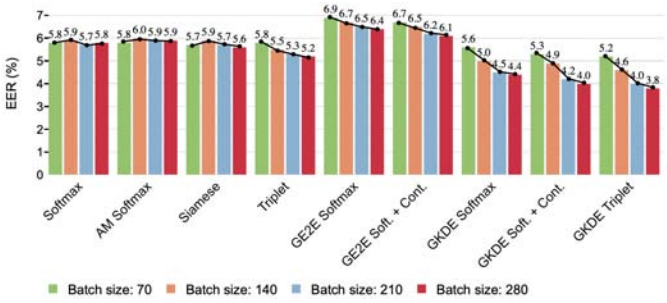

FIGURE 4. Bar plot of pooled EERs (%) evaluated in the logical access test set using an embedding size of 32 and training the LCNN with different batch sizes and techniques.

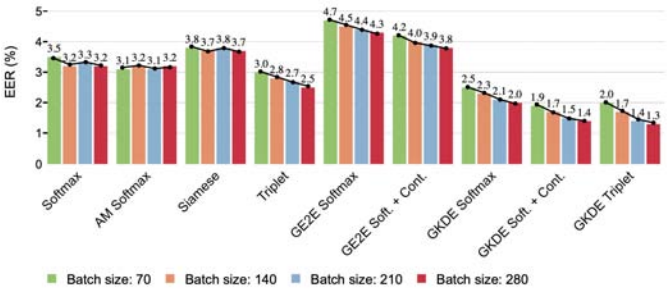

FIGURE 5. Bar plot of pooled EERs (%) evaluated in the physical access test set using an embedding size of 32 and training the LCNN with different batch sizes and techniques.

by the two types of VC attacks (A05 and A06). This result makes sense since genuine speech should be the most homogeneous class in the space of spoofing-aware embedding vectors. Furthermore, let us consider the three different types of speech data in the LA training set: (i) genuine speech, (ii) converted speech using two types of VC techniques (A05 and A06), and (iii) artificial speech using four types of TTS techniques (A01, A02, A03 and A04). As can be seen, the optimized bandwidths are similar within each group of speech nature, apart from the A02 attack which results to be more similar to VC attacks. This can be due to the fact that the waveform generator and acustic model employed to generate A02 attack are similar to the ones employed for generating the A05 attack [8].

According to the results of this study, we use learnable bandwidths in the rest of experiments of this work.

3) Evaluation of the embeddings size

Table 6 reports the EER and min-tDCF metrics achieved by the LCNN-based anti-spoofing system when trained using the maximum batch size which we can hold in our computational resources of 280 utterances ($N = 7$ classes and $M = 40$ utterances per class for the LA scenario, and $N = 2$ classes and $M = 140$ utterances per class for the PA scenario), different embedding sizes (16, 32 and 64) and the loss functions described in Sections II and III, namely: softmax, angular softmax, siamese, triplet, GE2E softmax, GE2E softmax and contrast, GKDE softmax, GKDE softmax and contrast, and GKDE triplet. It can be seen that the proposed GKDE based loss functions yield the best performance in terms of EER and min-tDCF, irrespective of the embedding size, on both the LA and PA evaluation scenarios. Regarding the loss functions described in Section II, the triplet loss achieves the best performance on both the LA and PA scenarios, followed by the softmax, angular softmax and siamese techniques. On the other hand, the GE2E based loss functions yield the worst performance. This could be due to the effect of smoothing caused by the use of a centroid for representing each class.

Moreover, the use of an embedding size of 32 is the

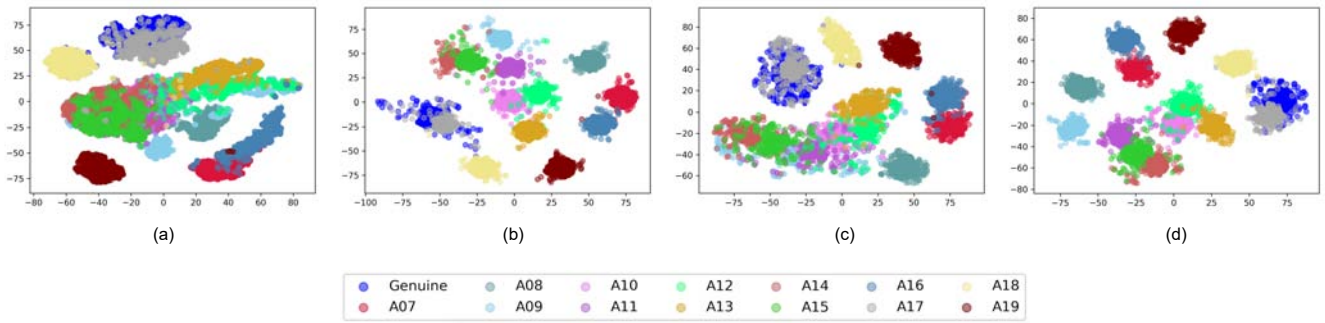

FIGURE 6. Representation of the logical access test embeddings using t-SNE: (a) softmax loss; (b) GKDE softmax loss; (c) triplet loss; (d) GKDE triplet loss.

best option for almost all the loss functions, and this size matches the embedding size selected in [45], which employs a similar LCNN based anti-spoofing system. To highlight the performance differences between the different techniques, Fig. 3 shows the pooled EERs achieved by each technique when using an embedding size of 32. As it can be seen, the proposed GKDE softmax loss function outperforms its counterpart softmax and GE2E softmax loss functions by an absolute pooled EER of 1.34 and 1.97% in the LA scenario, respectively, as well as by 1.25 and 2.29% in the PA scenario, respectively. Furthermore, when the softmax GE2E and GKDE loss functions are combined with a contrast loss, they yield a better performance due to the fact that the contrast loss helps to increase the inter-class variance. Related to this fact, the GKDE triplet loss, which is able to increase the inter-class variance while decreasing the intra-class variance at the same time, yields the best performance of all loss functions, outperforming its counterpart triplet loss by an absolute pooled EER of 1.31 and 1.20% in the LA and PA test sets, respectively.

4) Evaluation of the batch size

Fig. 4 and 5 shows the pooled EERs evaluated in the LA and PA test sets, respectively, obtained by training the LCNN with different loss functions and using different batch sizes (70, 140, 210 and 280). The objective is to study the effect of the batch size on the anti-spoofing results. The softmax, angular softmax and siamese loss functions are not affected by the selection of the batch size, since they almost obtain the same EER in the four cases of batch size. However, the performance of the rest of loss functions does depend on the batch size. For instance, the triplet loss employs an online selection of the positive and negative samples within the batch, and it is more likely to find hard samples in a larger mini-batch. Likewise, the GE2E and GKDE based loss functions attain better performance when increasing the batch size, since a better representation of every class is obtained. Moreover, this performance difference is more noticeable in the LA scenario than in the PA scenario, due to the fact that $M = 40$ utterances per class are employed in the LA scenario, while $M = 140$ utterances per class are used for training the LCNN in the PA scenario. It is also quite remarkable that the proposed GKDE based loss functions are the ones which quantitatively improve more their performance when using a larger batch size. This is due to the fact that KDE estimates the pdf of each class in a 32-dimensional space (embedding size) by placing a probability mass at every embedding sample within the mini-batch, so the more samples per mini-batch are used the more accurate is the representation of the pdf for the classes. In contrast, the GE2E based techniques represent each class with a centroid, being this representation less affected by the changes in the batch size in comparison with the KDE-based representation of every class in GKDE.

5) t-SNE embeddings representation

For illustrative purposes, we represent the LA test embeddings (10,000 embeddings per class) in a two-dimensional space using t-SNE [55], which preserves distances in a two-dimension space. Fig. 6 shows the embeddings obtained by the following loss functions: (a) softmax loss, (b) GKDE softmax loss, (c) triplet loss, and (d) GKDE triplet loss. As we can see, the clusters of the different LA attacks and genuine class are more separated in the GKDE based loss functions than in the classical softmax and triplet losses, which explains the better performance of the proposed GKDE based loss functions. According to the results of the ASVspoof 2019 challenge [8], the VC attack A17, which is generated using waveform filtering and employing a variational autoencoder as acoustic model, is the most difficult to detect. This fact can also be seen in the t-SNE embeddings representations, where the cluster of the A17 attack is the one that overlaps the most with the genuine class cluster in the four cases.

B. LC-GRNN RESULTS

To study the effect of employing a more complex neural network architecture, we also evaluated the effectiveness of the proposed GKDE losses on the LC-GRNN.

Table 7 compares the performance attained with the proposed GKDE based loss functions on the ASVspoof 2019 database using the LC-GRNN architecture and other other

TABLE 7. Comparison of single anti-spoofing systems performance on the ASVspoof 2019 logical and physical access test sets in terms of EER (%) and min-tDCF.

System	EER (%) / min-tDCF	
	Logical Access	Physical Access
Baseline: CQCC + GMM [8]	9.57 / 0.2366	11.04 / 0.2454
Baseline: LFCC + GMM [8]	8.09 / 0.2116	13.54 / 0.3016
STFT + LCNN + AM [26]	4.53 / 0.1028	-
CQT + LCNN + AM [26]	-	1.23 / 0.0295
TDNN + Softmax [56]	8.44 / 0.2251	-
SincNet + Softmax [57]	20.11 / 0.3563	2.11 / 0.0527
ResNet + Softmax [12]	7.69 / 0.2166	4.43 / 0.1070
LC-GRNN + Softmax [9]	6.28 / 0.1523	2.23 / 0.0614
GRCNN + Softmax [47]	3.85 / 0.0952	1.09 / 0.0234
LC-GRNN + GKDE - Softmax	3.77 / 0.0842	1.06 / 0.0222
LC-GRNN + GKDE - Soft. + Cont.	3.39 / 0.0805	0.97 / 0.0210
LC-GRNN + GKDE - Triplet	**3.03 / 0.0776**	**0.92 / 0.0198**

state-of-the-art single anti-spoofing systems from the literature. As can be seen, our proposed systems outperform the baseline anti-spoofing systems released with this database (CQCC + GMM and LFCC + GMM), as well as the other top performing single systems (presented to the ASVspoof 2019 Challenge [8]) and our previous GRCNN [47], in both the LA and PA scenarios. Specifically, the LC-GRNN trained with the GKDE based triplet loss yields a 5.06 % and 10.12 % lower pooled EER than the best baseline systems of the LA and PA scenarios, respectively. In addition, it achieves a 3.25 % and 1.31 % better pooled EER than the same system trained with the classical softmax loss proposed in our previous work [9] for both the LA and PA scenarios, respectively.

According to this evaluation, we can conclude that the proposed GKDE based loss functions are effective for different types of neural network architectures such as CNNs, RNNs and their combination. Moreover, the proposed single anti-spoofing systems are among the best state-of-the-art systems at detecting the recent attacks based on the latest technologies [8].

VI. CONCLUSION

In this paper we proposed various loss functions, based on kernel density estimation (KDE) techniques, which estimate the probability density function (pdf) of every training class in each mini-batch, and compute a log likelihood matrix by using the embedding vectors and pdfs of all training classes within the mini-batch. These loss functions address three main problems that have been detected in conventional loss functions: (i) the training samples which belong to the same class are kept close to each other and the dissimilar instances are kept far away on the embedding space by using hard negative mining, (ii) the training classes are fully represented by all the samples within the mini-batch, by estimating with KDE a pdf per class which places a probability mass at every embedding sample, and (iii) the concept of distance measure between embedding vectors is replaced by the concept of the probability that an embedding vector belongs to a certain class, which has the advantage of avoiding the selection of an appropiate distance measure and embedding normalization technique. Experimental results on the ASVspoof 2019 database have shown that the proposed losses outperform other conventional loss functions that have been used so far for training DNN-based antispoofing systems. Furthermore, it is shown that the performance gains are not restricted to a sole neural network architecture, but the proposed loss functions are effective for training different types of neural networks such as CNNs, RNNs and their combination.

We hope that this new concept of loss functions can be rather considered a general approach since it can be applied to any DNN-based embedding extraction system which comprises fully connected layers. As future work, we will evaluate the proposed loss functions in other speech related tasks such as ASV and integration of ASV and PAD systems.

ACKNOWLEDGMENT

This work has been supported by the Spanish MINECO/FEDER Project TEC2016-80141-P. 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] R. Naika, "An overview of automatic speaker verification system," Advances in Intelligent Systems and Computing, vol. 673, 2018.
[3] 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.
[4] 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.
[5] "Presentation attack detection." [Online]. Available: https://www.iso.org/standard/67381.html
[6] 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, 2015.
[7] 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, 2017.
[8] 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, 2019.
[9] 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, 2019.
[10] X. Wu, R. He, Z. Sun, and T. Tan, "A light CNN for deep face representation with noisy labels," IEEE Transactions on Information Forensics and Security, vol. 13, no. 11, pp. 2884–2896, 2015.
[11] A. Gomez-Alanis, A. M. Peinado, J. A. Gonzalez, and A. M. Gomez, "A deep identity representation for noise robust spoofing detection," in Proc. Interspeech, 2018.
[12] M. Alzantot, Z. Wang, and M. B. Srivastava, "Deep residual neural networks for audio spoofing detection," in Proc. Interspeech, 2019.

[13] 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, 2018.

[14] K. Sriskandaraja, V. Sethu, and E. Ambikairajah, "Deep siamese architecture based replay detection for secure voice biometric," in Proc. Interspeech, 2018.

[15] L. Wan, Q. Wang, A. Papir, and I. Lopez-Moreno, "Generalized end-to-end loss for speaker verification," in Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2019.

[16] H.-S. Heo, J. weon Jung, I.-H. Yang, S.-H. Yoon, H. jin Shim, and H.-J. Yu, "End-to-end losses based on speaker basis vectors and all-speaker hard negative mining for speaker verification," in Proc. Interspeech, 2019.

[17] 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.

[18] E. Hoffer and N. Ailon, "Deep metric learning using triplet network," in Proc. SIMBAD, 2015.

[19] D. Yi, Z. Lei, S. Liao, and S. Z. Li, "Deep metric learning for person re-identification," in Proc. International Conference on Pattern Recognition, 2014, pp. 34–39.

[20] K. Sohn, "Improved deep metric learning with multi-class n-pair loss objective," in Proc. Neural Information Processing Systems (NIPS), 2016.

[21] 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), 2015.

[22] E. Parzen, "On estimation of a probability density function and mode," The Annals of Mathematical Statistics, vol. 33, pp. 1065–1076, 1962.

[23] J. Koloda, A. M. Peinado, and V. Sanchez, "Kernel-based MMSE multimedia signal reconstruction and its application to spatial error concealment," IEEE Transactions on Multimedia, vol. 16, no. 6, pp. 1729–1738, 2014.

[24] L. Weiyang, W. Yandong, Y. Zhiding, L. Ming, R. Bhiksha, and S. Le, "Sphereface: Deep hypersphere embedding for face recognition," in Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[25] W. Feng, C. Jian, L. Weiyang, and L. Haijun, "Additive margin softmax for face verification," IEEE Signal Processing Letters, vol. 25, no. 7, pp. 926–930, 2018.

[26] G. Lavrentyeva, S. Novoselov, A. Tseren, M. Volkova, A. Gorlanov, and A. Kozlov, "STC anti-spoofing systems for the asvspoof2019 challenge," in Proc. Interspeech, 2019.

[27] Y. Yu, L. Fan, and W. Li, "Ensemble additive margin softmax for speaker verification," in Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2019.

[28] L. Yutian, G. Feng, O. Zhijian, and S. Jiasong, "Angular softmax loss for end-to-end speaker verification," in Proc. International Symposium on Chinese Spoken Language Processing, 2018.

[29] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "Deepface: Closing the gap to human-level performance in face verification," in Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014, pp. 1701–1708.

[30] E. Ahmed, M. Jones, and T. K. Marks, "An improved deep learning architecture for person re-identification," in Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 3908–3916.

[31] K. Chen and A. Salman, "Extracting speaker-specific information with a regularized siamese deep network," in Proc. Neural Information Processing Systems (NIPS), 2011.

[32] C. Zhang, K. Koishida, and J. H. L. Hansen, "Text-independent speaker verification based on triplet convolutional neural network embeddings," IEEE/ACM Transactions on Audio, Speech and Language Processing, vol. 26, no. 9, pp. 1633–1644, 2018.

[33] C. Zhang and K. Koishida, "End-to-end text-independent speaker verification with triplet loss on short utterances," in Proc. Interspeech, 2017.

[34] P. Green, A. H. Seheult, and B. Silverman, "Density estimation for statistics and data analysis," Monographs on Statistics and Applied Probability, vol. 26, 1988.

[35] B. Turlach, "Bandwidth selection in kernel density estimation: A review," CORE and Institut de Statistique, vol. 19, pp. 1–33, 1993.

[36] V. A. Epanechnikov, "Non-parametric estimation of a multivariate probability density," Theory of Probability and its Applications, vol. 14, no. 1, pp. 153–158, 1969.

[37] K. Weinberger, J. Blitzer, and L. Saul, "Distance metric learning for large margin nearest nearest neighbor classification," in Proc. Neural Information Processing Systems (NIPS), 2006.

[38] D. Matrouf, J.-F. Bonastre, and C. Fredouille, "Effect of speech transformation on impostor acceptance," in Proc. IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP), 2006, pp. 933–936.

[39] M. Morise, F. Yokomori, and K. Ozawa, "WORLD: A vocoder-based high-quality speech synthesis system for real-time applications," IECE Transactions on Information and Systems, vol. 99, no. 7, pp. 1877–1884, 2016.

[40] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. W. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," ArXiv, vol. abs/1609.03499, 2016.

[41] D. W. Griffin and J. S. Lim, "Signal estimation from modified short-time fourier transform," in Proc. IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP), 1983.

[42] K. Tanaka, T. Kaneko, N. Hojo, and H. Kameoka, "Synthetic-to-natural speech waveform conversion using cycle-consistent adversarial networks," in Proc. IEEE Spoken Language Technology Workshop (SLT), 2018, pp. 632–639.

[43] X. Wang, S. Takaki, and J. Yamagishi, "Neural source-filter-based waveform model for statistical parametric speech synthesis," in Proc. IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP), 2019, pp. 5916–5920.

[44] K. Kobayashi, T. Toda, G. Neubig, S. Sakti, and S. Nakamura, "Statistical singing voice conversion with direct waveform modification based on the spectrum differential," in Proc. Interspeech, 2014.

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

[46] I. J. Goodfellow, D. Warde-Farley, M. Mirza, A. C. Courville, and Y. Bengio, "Maxout networks," in Proc. International Conference on Machine Learning (ICML), 2013.

[47] 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.

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

[49] 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), 2017.

[50] H. J. Wang, Y. Wang, Z.-F. Zhou, X. Ji, Z. Li, D. Gong, J. Zhou, and W. Liu, "Cosface: Large margin cosine loss for deep face recognition," in Proc. IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 5265–5274.

[51] J. Deng, J. Guo, and S. Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," in Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 4685–4694.

[52] N. N. Vo and J. Hays, "Localizing and orienting street views using overhead imagery," in Proc. European Conference on Computer Vision (ECCV), 2016.

[53] 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, 2018.

[54] Y. Soh, Y. Hae, A. Mehmood, R. H. Ashraf, and I. Kim, "Performance evaluation of various functions for kernel density estimation," Open Journal of Applied Sciencies, vol. 3, no. 1, pp. 58–64, 2013.

[55] L. van der Maaten and G. E. Hinton, "Visualizing data using t-SNE," Journal of Machine Learning Research, vol. 9, pp. 2579–2605, 2008.

[56] C. Su-Yu, W. Kai-Cheng, and C. Chia-Ping, "Transfer-representation learning for detecting spoofing attacks with converted and synthesized speech in automatic speaker verification system," in Proc. Interspeech, 2019.

[57] H. Zeinali, T. Stafylakis, G. Athanasopoulou, J. Rohdin, I. Gkinis, L. Burget, and J. H. Cernocký, "Detecting spoofing attacks using VGG and SincNet: BUT-Omilia submission to ASVspoof 2019 challenge," in Proc. Interspeech, 2019.

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.

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.

2.2.2 Loss Function for Biometric Systems

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

- Alejandro Gomez-Alanis, Jose A. Gonzalez-Lopez, S. Pavankumar Dubagunta, A. M. Peinado and Mathew Maigmai.-Doss, "On Joint Optimization of Automatic Speaker Verification and Anti-spoofing in the Embedding Space", *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 1579-1593, November 2020.
    - Status: Published.
    - Impact Factor (JCR 2020): 7.178
    - Subject Category: Engineering, Electrical & Electronic. Ranking 21/273 (D1).
    - Subject Category: Computer Science, Theory & Methods. Ranking 8/110 (D1).

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

Alejandro Gomez-Alanis, Jose A. Gonzalez-Lopez, S. Pavankumar Dubagunta, 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).**

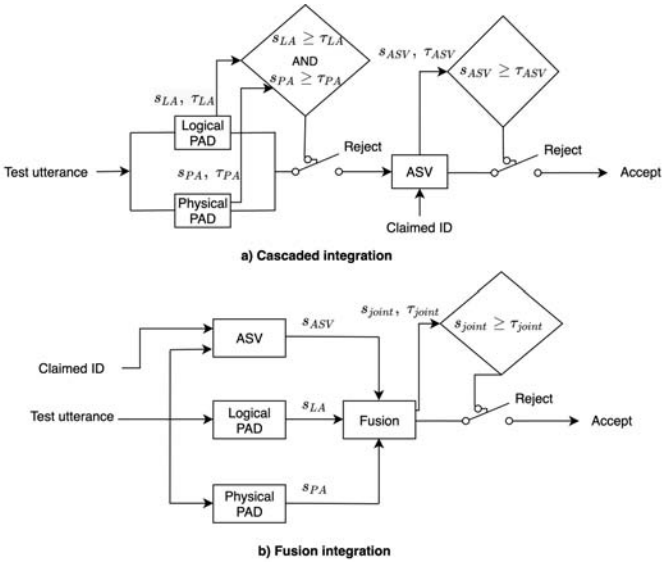

Fig. 1. Block diagram of two score-level integration systems: (a) cascaded (PAD preceeding ASV) integration; (b) fusion integration. s_{LA}, s_{PA}, s_{ASV} and s_{joint} denote the scores of the LA, PA, ASV and joint integration systems, respectively. Likewise, τ_{LA}, τ_{PA}, τ_{ASV} and τ_{joint} 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,

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

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

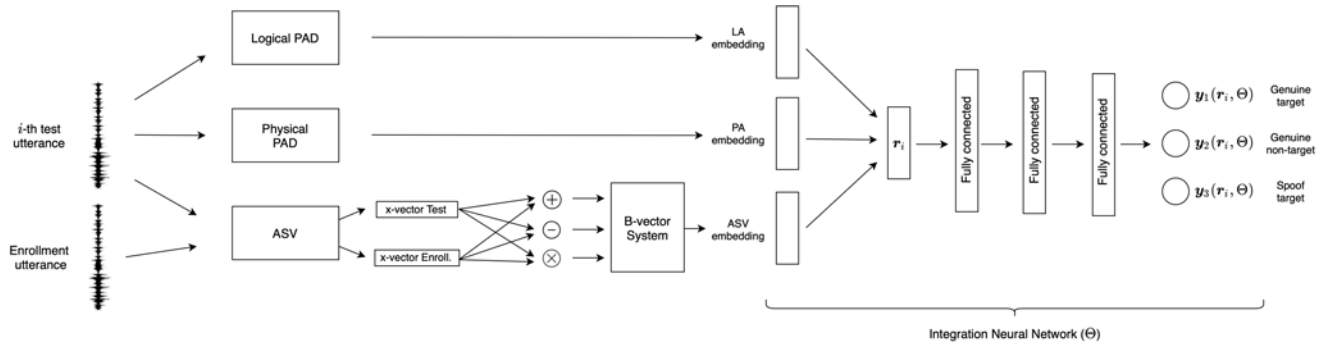

Fig. 2. Proposed integration neural network framework. System overview for classifying a pair of enrollment and test utterances into one of the three integration classes: C_1 (target genuine), C_2 (genuine non-target) and C_3 (spoof target). The LA and PA spoofing embeddings are extracted from the STFT 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,

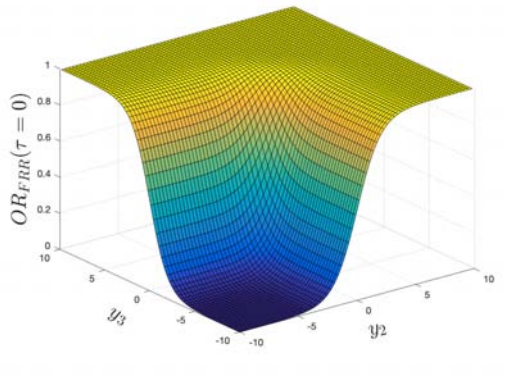

Fig. 3. Logical OR_{FRR} function, $\text{OR}_{\text{FRR}}(\tau = 0) = {}^{(\sigma(\boldsymbol{y}_2)+\sigma(\boldsymbol{y}_3))}/_{(1+\sigma(\boldsymbol{y}_2)\sigma(\boldsymbol{y}_3))}$, which is softly activated when any of the two negative scores ($\boldsymbol{y}_2$ or $\boldsymbol{y}_3$) is greater 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

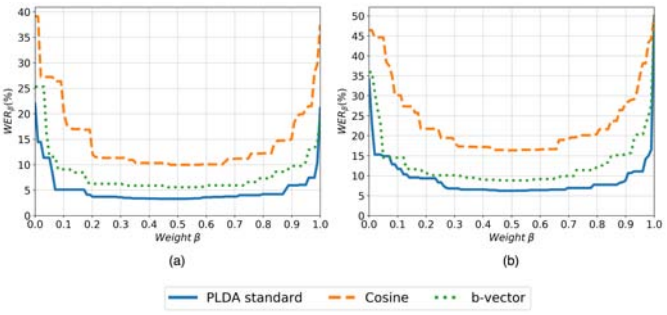

Fig. 4. WER_β for the following ASV scoring systems: cosine, b-vector and PLDA standard. The EPC is evaluated on the development licit scenarios ($\omega = 0$) of the ASVspoof 2019 sets. (a) Logical Access. (b) Physical Access.

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.

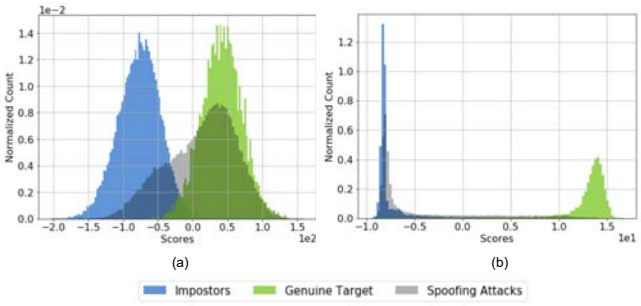

Fig. 5. Scores distribution of genuine target accesses, genuine non-target or impostors accesses, and *spoofing* attacks, evaluated in the logical access dataset. (a) ASV system (PLDA standard). (b) Integration Neural Network (AUE).

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,

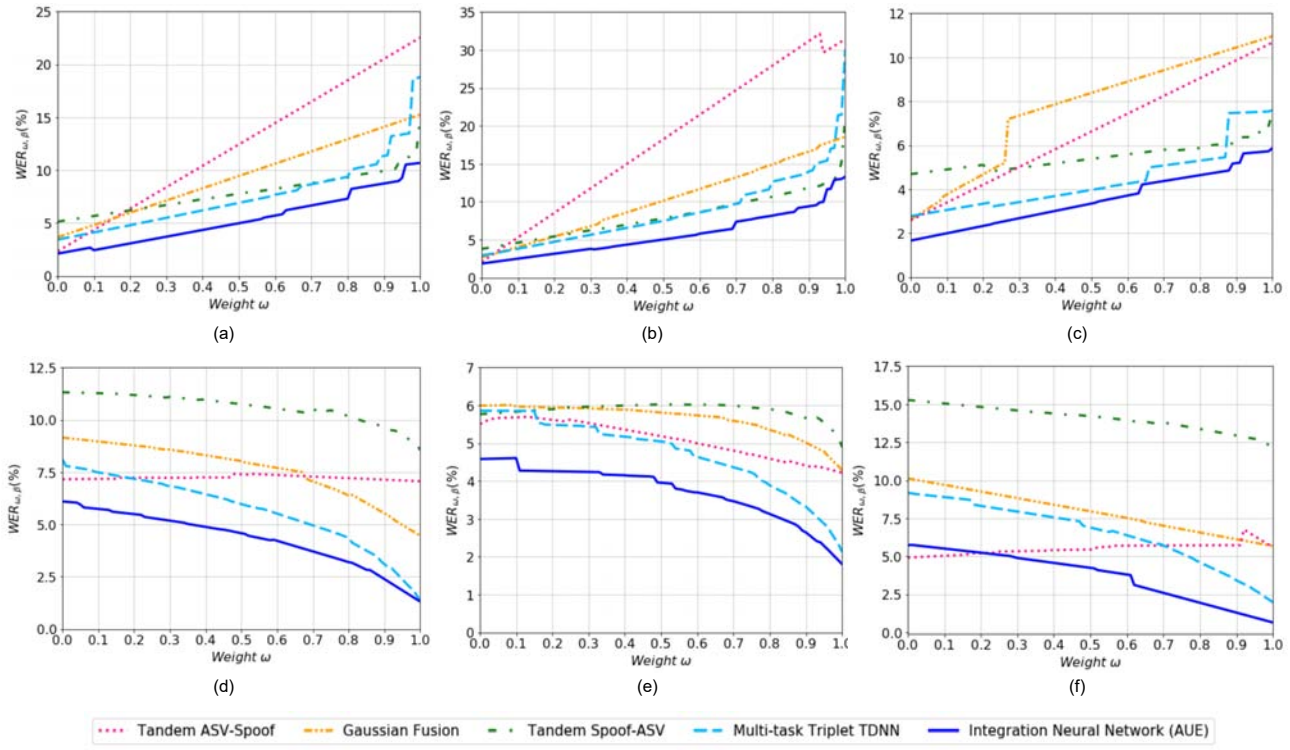

Fig. 6. Expected Performance and Spoofability Curves (EPSC) of different ASV and integration systems evaluated at different operating points and datasets. (a) Logical Access ($\beta = 0.5$). (b) Logical Access ($\beta = 0.8$). (c) Logical Access ($\beta = 0.2$). (d) Physical Access ($\beta = 0.5$). (e)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

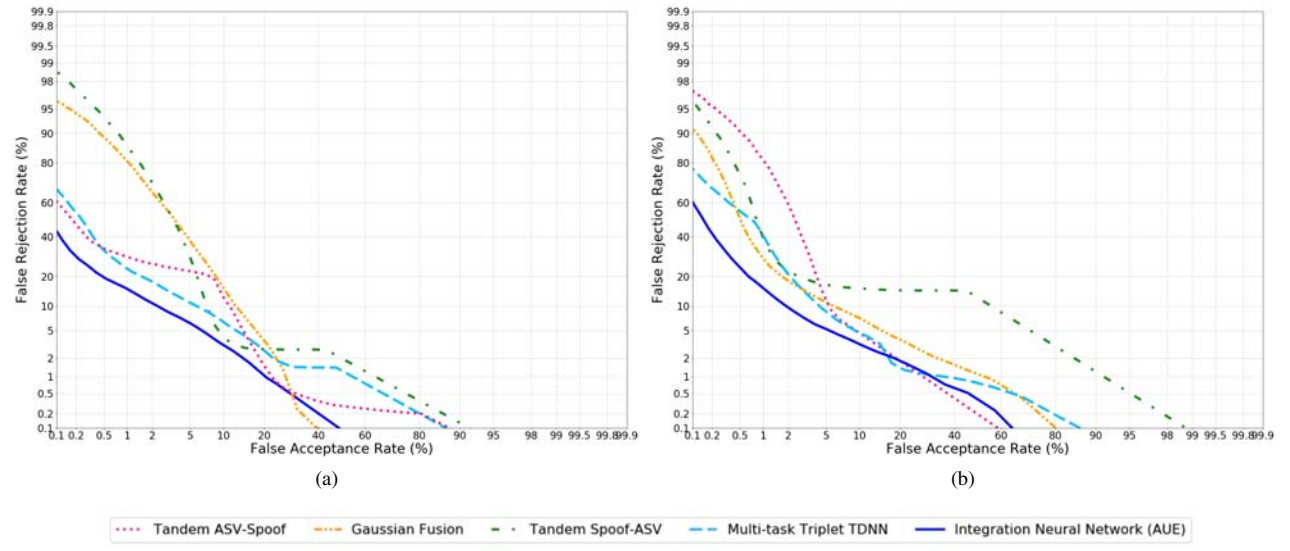

Fig. 7. Detection Error Tradeoff (DET) curves of different integration systems evaluated in the evaluation datasets of the ASVspoof 2019 database: (a) Logical Access; and (b) Physical Access.

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

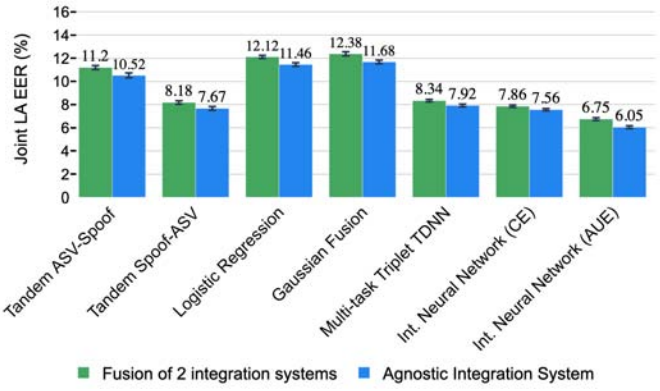

Fig. 8. Averaged joint EERs (%) evaluated in the LA test scenario of the agnostic and fusion of 2 integration systems. Mean intervals are presented at 95% of confidence.

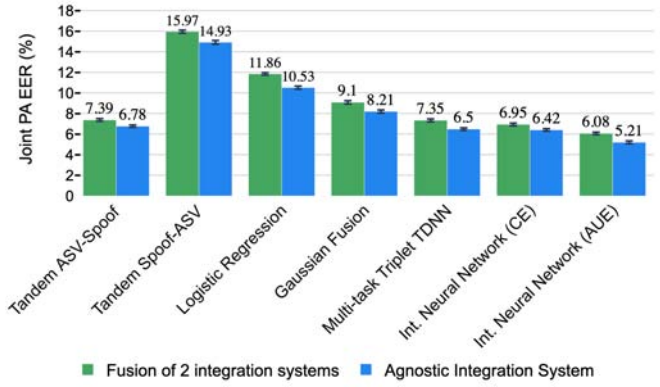

Fig. 9. Averaged joint EERs (%) evaluated in the PA test scenario of the agnostic and fusion of 2 integration systems. Mean intervals are presented at 95% of confidence.

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 on Automatic Speech Assessment and Recognition (ASR) using raw signals of speech. He has about four years of industry experience in ASR 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 R&D Institute India. Prior to that, he received his Master of Science by Research in Electrical Engineering for his work on Feature Normalisation for Robust ASR, 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.

2.3 Adversarial Examples for Biometric Systems

2.3.1 Adversarial Transformation of Spoofing Attacks for Voice Biometrics

- Alejandro Gomez-Alanis, Jose A. Gonzalez-Lopez and A. M. Peinado, "Adversarial Transformation of Spoofing Attacks for Voice Biometrics", *Proc. IberSPEECH*, pp. 255-259, Valladolid, Spain, March 2021.
    - Status: Published.
    - Awards: Best paper of the conference *Iberspeech 2021*.

Adversarial Transformation of Spoofing Attacks for Voice Biometrics

Alejandro Gomez-Alanis, Jose A. Gonzalez-Lopez, Antonio M. Peinado

University of Granada

agomezalanis@ugr.es, joseangl@ugr.es, amp@ugr.es

Abstract

Voice biometric systems based on automatic speaker verification (ASV) are exposed to *spoofing* attacks which may compromise their security. To increase the robustness against such attacks, anti-spoofing or presentation attack detection (PAD) systems have been proposed for the detection of replay, synthesis and voice conversion based attacks. Recently, the scientific community has shown that PAD systems are also vulnerable to adversarial attacks. However, to the best of our knowledge, no previous work have studied the robustness of full voice biometrics systems (ASV + PAD) to these new types of adversarial *spoofing* attacks. In this work, we develop a new adversarial biometrics transformation network (ABTN) which jointly processes the loss of the PAD and ASV systems in order to generate white-box and black-box adversarial *spoofing* attacks. The core idea of this system is to generate adversarial *spoofing* attacks which are able to fool the PAD system without being detected by the ASV system. The experiments were carried out on the ASVspoof 2019 corpus, including both logical access (LA) and physical access (PA) scenarios. The experimental results show that the proposed ABTN clearly outperforms some well-known adversarial techniques in both white-box and black-box attack scenarios.

Index Terms: Adversarial attacks, automatic speaker verification (ASV), presentation attack detection (PAD), voice biometrics.

1. Introduction

Voice biometrics aims to authenticate the identity claimed by a given individual based on the speech samples measured from his/her voice. Automatic speaker verification (ASV) [1] is the conventional way to put voice biometrics into practical usage. However, in recent years, ASV technology has been shown to be at risk of security threats performed by impostors who want to gain fraudulent access by presenting speech resembling the voice of a legitimate user [2, 3]. Impostors could use either logical access (LA) attacks [4], such as text-to-speech synthesis (TTS) and voice conversion (VC) based attacks, or physical access (PA) attacks such as replay based attacks [5].

To protect voice biometrics systems [6], it is common to develop anti-spoofing or presentation attack detection (PAD) [7] techniques which allow for differentiating between *bonafide* and *spoofing* speech [8, 9, 10]. Typically, the resulting biometrics system is a score-level cascaded integration of PAD and ASV subsystems, as depicted in Fig. 1. This is the same integration as the one used in the last two ASVspoof challenges [5, 11].

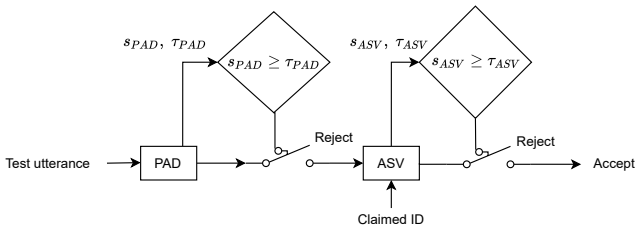

Figure 1: *Block diagram of a score-level cascaded integration biometrics system. s_{PAD}, τ_{PAD} and s_{ASV}, τ_{ASV} denote the scores and thresholds of the PAD and ASV systems, respectively.*

To make things more complex, different investigations [12, 13] have recently shown that PAD systems are also vulnerable to adversarial attacks [14]. These attacks can easily fool deep neural network (DNN) models by perturbing benign samples in a way normally imperceptible to humans [15]. Adversarial attacks can be divided into two main categories: white-box and black-box attacks. In this work, we refer to white-box attacks as those where the attacker can access all the information of the victim model (i.e., model architecture and its weights). Likewise, we will use the term black-box for those attacks where the attacker does not know any information about the victim model but it can be queried multiple times in order to estimate a surrogate model (student) of the victim model (teacher), using the binary responses (acceptance/rejection) of the victim model as ground-truth labels.

The main contributions of this work are:

- Investigate the robustness of full voice biometrics systems (ASV + PAD) under the presence of adversarial *spoofing* attacks.
- Propose an adversarial biometrics transformation network (ABTN) which is able to generate adversarial *spoofing* attacks in order to fool the PAD system without being detected by the ASV system.
- To the best of our knowledge, adversarial *spoofing* attacks have only been studied on logical access scenarios (TTS and VC based attacks). In this work, we also include physical access scenarios (replay based attacks).

The rest of this paper is organized as follows. Section 2 outlines some well-known adversarial attacks employed as baselines in this work. Then, in Section 3, we describe the proposed ABTN for white-box and black-box scenarios. After that, Section 4 outlines the speech corpora, systems details, and metrics employed in the experiments. Section 5 discusses the experimental results. Finally, we summarize the conclusions derived from this research in Section 6.

This work has been supported by the Spanish Ministry of Science and Innovation Project No. PID2019-104206GB-I00/AEI/10.13039/501100011033. 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.

2. Background

Adversarial *spoofing* examples can be generated by adding a minimally perceptible perturbation to the input *spoofing* utterance in order to do a refinement of the *spoofing* attack. In this work, we focus on targeted attacks, which aim to fool the PAD system by maximizing the probability of a targeted class (*bonafide*) different from the correct class (*spoof*). Specifically, to generate adversarial *spoofing* attacks, we fix the parameters $\boldsymbol{\theta}$ of a well-trained DNN-based PAD model and perform gradient descent to update the *spoofing* spectra of the input utterance so that the PAD model classifies it as a *bonafide* utterance. Mathematically, our goal is to find a sufficiently small perturbation $\boldsymbol{\delta}$ which satisfies:

$$\begin{aligned} \tilde{\boldsymbol{X}} &= \boldsymbol{X} + \boldsymbol{\delta}, \\ f_{\boldsymbol{\theta}}(\boldsymbol{X}) &= y, \\ f_{\boldsymbol{\theta}}(\tilde{\boldsymbol{X}}) &= \tilde{y}, \end{aligned} \tag{1}$$

where f is a well-trained DNN-based PAD model parameterized by $\boldsymbol{\theta}$, $\boldsymbol{X}$ denotes the sequence of speech feature vectors extracted from the input *spoofing* utterance (short time Fourier transform (STFT), typically), y is the true label corresponding to $\boldsymbol{X}$, $\tilde{y}$ is the targeted label class of the attack (*bonafide* class), $\tilde{\boldsymbol{X}}$ denotes the perturbed input features, and $\boldsymbol{\delta}$ is the additive perturbation. Typically, $\boldsymbol{\Delta}$ is the feasible set of the allowed perturbation $\boldsymbol{\delta}$ ($\boldsymbol{\delta} \in \boldsymbol{\Delta}$), which formalizes the manipulative power of the adversarial attack. Normally, $\boldsymbol{\Delta}$ is a small l_∞-norm ball, that is, $\boldsymbol{\Delta} = \{\boldsymbol{\delta} \mid \|\boldsymbol{\delta}\|_\infty \leq \epsilon\}$, $\epsilon \geq 0 \in \mathbb{R}$.

There are multiple ways to generate the perturbation $\boldsymbol{\delta}$, where the fast gradient sign method (FGSM) [16] and the projected gradient descent (PGD) [17] methods are the most popular adversarial attack procedures. The FGSM attack consists of taking a single step along the direction of the gradient, i.e.,

$$\boldsymbol{\delta} = \epsilon \cdot \text{sign}(\nabla_{\boldsymbol{X}} Loss(\boldsymbol{\theta}, \boldsymbol{X}, y)), \tag{2}$$

where $Loss$ denotes the loss function of the neural network ($\boldsymbol{\theta}$), and the sign method simply takes the sign of its gradient. Unlike the FGSM, which is a single-step method, the PGD is an iterative method. Starting from the original input utterance $\boldsymbol{X}_0 = \boldsymbol{X}$, the input utterance is iteratively updated as follows:

$$\begin{aligned} &\boldsymbol{X}_{n+1} = \text{clip}(\boldsymbol{X}_n + \alpha \cdot \text{sign}(\nabla_{\boldsymbol{X}} Loss(\boldsymbol{\theta}, \boldsymbol{X}, y)), \\ &\text{for } n = 0, ..., N-1, \end{aligned} \tag{3}$$

where $n = 0, ..., N-1$ is the iteration index, N is the number of iterations, $\alpha = \epsilon/N$, and the clip() function applies element-wise clipping such that $\|\boldsymbol{X}_n - \boldsymbol{X}\|_\infty \leq \epsilon$, $\epsilon \geq 0 \in \mathbb{R}$.

3. Proposed method

The performance of the FGSM and PGD methods is limited by the possibility of sticking at local optima of the loss function. Moreover, both methods have a limited search space ($\boldsymbol{\Delta}$) so that the perturbed *spoofing* speech $\tilde{\boldsymbol{X}}$ is perceptually indistinguishable from the original *spoofing* speech $\boldsymbol{X}$.

In this work, we propose the Adversarial Biometrics Transformation Network (ABTN), which is a neural network that transforms a *spoofing* speech signal into an adversarial *spoofing* speech signal against a target biometrics system. Formally, an ABTN can be defined as a neural network $g_{f,h} : \boldsymbol{X} \rightarrow \tilde{\boldsymbol{X}}$, where $f(\boldsymbol{X})$ and $h(\boldsymbol{X})$ are the PAD and ASV models of the target biometrics system, respectively. The PAD and ASV models can provide either a probability distribution across class labels (white-box scenario) or just a binary decision (black-box scenario). In both scenarios, the objective of the ABTN is to generate adversarial *spoofing* attacks from *spoofing* speech in order to fool the PAD system while not being detected by the ASV system, i.e., while not modifying the speaker information.

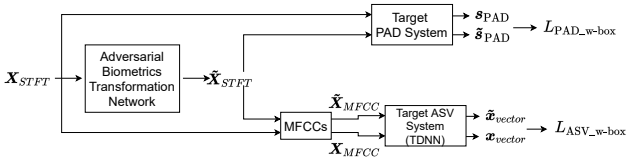

Figure 2: *Proposed adversarial biometrics transformation system for white-box scenarios.*

3.1. White-box scenario

The architecture of the proposed ABTN system for the white-box scenario is depicted in Fig. 2. The output of the ABTN is fed into the target biometrics system which is composed of a PAD and an ASV system based on a time-delay neural network (TDNN) [18] for x-vector extraction (in fact, this is the only component of the ASV system that we need). The objective of this system is to train the ABTN so that it can generate adversarial attacks from *spoofing* speech which are able to fool the PAD system while, at the same time, it does not cause any changes to the ASV output (i.e., it does not change the speaker representation given by the corresponding x-vector). To train the ABTN, the PAD and ASV network parameters are frozen but the gradients are computed along them in order to back-propagate them to the ABTN parameters. To find the optimal parameters of the ABTN in the white-box (w-box) scenario, we minimize the following loss function:

$$L_{\text{w-box}} = L_{\text{PAD\_w-box}}(\boldsymbol{s}_{\text{PAD}}, \tilde{\boldsymbol{s}}_{\text{PAD}}) + \beta \cdot L_{\text{ASV\_w-box}}(\boldsymbol{x}_{\text{vector}}, \tilde{\boldsymbol{x}}_{\text{vector}}), \tag{4}$$

where,

$$L_{\text{PAD\_w-box}}(\boldsymbol{s}_{\text{PAD}}, \tilde{\boldsymbol{s}}_{\text{PAD}}) = \|r_\alpha(\boldsymbol{s}_{\text{PAD}}) - \tilde{\boldsymbol{s}}_{\text{PAD}}\|_2, \tag{5}$$

$$L_{\text{ASV\_w-box}}(\boldsymbol{x}_{\text{vector}}, \tilde{\boldsymbol{x}}_{\text{vector}}) = \|\boldsymbol{x}_{\text{vector}} - \tilde{\boldsymbol{x}}_{\text{vector}}\|_2. \tag{6}$$

$L_{\text{PAD\_w-box}}$ and $L_{\text{ASV\_w-box}}$ are the loss components associated to the PAD and ASV systems, respectively, and β is a hyper-parameter to weight the importance of the two losses. $\boldsymbol{s}_{\text{PAD}}$ and $\tilde{\boldsymbol{s}}_{\text{PAD}}$ are the probability output vectors from the PAD system of the original and adversarial *spoofing* utterances, respectively. Likewise, $\boldsymbol{x}_{\text{vector}}$ and $\tilde{\boldsymbol{x}}_{\text{vector}}$ denote the x-vectors of the original and adversarial *spoofing* utterances, respectively, and r_α is a reranking function which can be formulated as

$$r_\alpha(\boldsymbol{s}_{\text{PAD}}) = norm\left(\begin{cases} \alpha \cdot \max(\boldsymbol{s}_{\text{PAD}}) & k = 0 \\ \boldsymbol{s}_{\text{PAD}}(k) & k \neq 0 \end{cases}\right), \tag{7}$$

where k is the index class variable of the $\boldsymbol{s}_{\text{PAD}}$ probability vector, $\alpha > 1$ is an additional hyper-parameter which defines how large $\boldsymbol{s}_{\text{PAD}}(k = 0)$, i.e., the probability of the *bonafide* class, is with respect to the current maximum probability class, and $norm$ is a normalizing function which rescales its input to be a valid probability distribution.

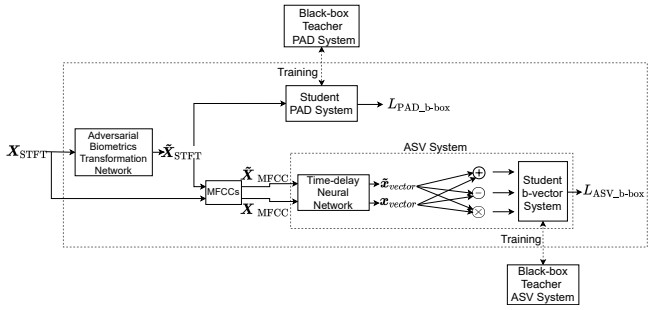

Figure 3: *Proposed adversarial biometrics transformation system for black-box scenarios.*

3.2. Black-box scenario

The architecture of the proposed ABTN system for the black-box scenario is depicted in Fig. 3. Similarly to the white-box scenario, the objective of this system is to generate adversarial attacks from *spoofing* speech which are able to fool the target (teacher) PAD system and, at the same time, bypass the target (teacher) ASV system by not modifying the speaker information represented by the corresponding x-vector. However, the limitation of the black-box scenario is that we do not have access to the parameters of the target biometrics system. Thus, we train a student PAD and a b-vector [19] based ASV systems by making requests to the target black-box biometrics system which only responds with a binary decision of acceptance or rejection, using these binary decisions as ground-truth labels. Therefore, the student PAD and b-vector systems are trained as binary classifiers in order to mimick the performance of the teacher PAD and ASV systems, respectively. Specifically, the student b-vector system computes the probability that the two input x-vectors belong to the same speaker, i.e., that $P(b(\boldsymbol{x}_{\text{vector}}, \tilde{\boldsymbol{x}}_{\text{vector}}) = 1)$, where b denotes the b-vector model.

To train the ABTN in the black-box scenario, the student PAD and ASV network parameters are also frozen but the gradients are computed along them in order to back-propagate them to the ABTN parameters. To find the optimal parameters of the ABTN in the black-box (b-box) scenario, we minimize the following loss function:

$$L_{\text{b-box}} = L_{\text{PAD\_b-box}}(\tilde{\boldsymbol{s}}_{\text{PAD}}) + \beta \cdot L_{\text{ASV\_b-box}}(\boldsymbol{x}_{\text{vector}}, \tilde{\boldsymbol{x}}_{\text{vector}}), \quad (8)$$

where,

$$L_{\text{PAD\_b-box}}(\tilde{\boldsymbol{s}}_{\text{PAD}}) = \|onehot(k=0) - \tilde{\boldsymbol{s}}_{\text{PAD}}\|_2\,, \quad (9)$$

$$L_{\text{ASV\_b-box}}(\boldsymbol{x}_{\text{vector}}, \tilde{\boldsymbol{x}}_{\text{vector}}) = 1 - P(b(\boldsymbol{x}_{\text{vector}}, \tilde{\boldsymbol{x}}_{\text{vector}}) = 1). \quad (10)$$

$L_{\text{PAD\_b-box}}$ and $L_{\text{ASV\_b-box}}$ are the loss components associated to the PAD and ASV systems, respectively. Moreover, the function $onehot$ denotes the one-hot function and $k = 0$ is the index of the *bonafide* class, so that the PAD system is fooled by firing the input *spoofing* utterance as a *bonafide* utterance.

4. Experimental Setup

This section briefly describes the speech corpora and metrics employed in our experiments, as well as the details of the proposed system.

4.1. Speech corpora

We conducted experiments on the ASVspoof 2019 database [20] which is split into two partitions for the assessment of LA and PA scenarios. This database also includes protocols for evaluating the performance of PAD, ASV and integration (biometrics) systems. Thus, we used this corpus for training the standalone PAD systems in the LA and PA scenarios, separately. Then, we generated adversarial *spoofing* attacks using only the *spoofing* utterances, so that they can bypass the biometrics system. We did not generate any adversarial examples from *bonafide* utterances, since we argue that they would not be *bonafide* anymore.

On the other hand, we also employed the Voxceleb1 [21] to train a TDNN [18] as an x-vector extractor for the ASV system. Also, following [6], a b-vector [19] ASV scoring system was trained in the black-box scenario using the *bonafide* utterances from the ASVspoof 2019 and Voxceleb1 development datasets.

4.2. Spectral analysis

Speech signals were analyzed using a Hanning analysis windows of 25 ms length with 10 ms of frame shift. Log-power magnitude spectrum features (STFT) with 256 frequency bins were obtained to feed all the PAD systems. The ASV systems were fed with Mel-frequency cepstral coefficients (MFCCs) obtained with the Kaldi recipe [22]. Only the first 600 frames of each utterance were used to extract acoustic features.

4.3. Implementation details

Two state-of-the-art PAD systems were adapted from different works, i.e., a light convolutional neural network (LCNN) [2] and a Squeeze-Excitation network (SENet50) [23]. The PAD scores were directly obtained from the *bonafide* class of the softmax output. For ASV, a TDNN x-vector model [18] was trained as an embedding extractor. Then, a probabilistic linear discriminant analysis (PLDA) [24] and a b-vector system [19] were trained as ASV scoring systems.

The proposed ABTN is formed by five convolutional layers with 16, 32, 48, 48 and 3 channels, respectively, and a kernel size of 3×3, followed by leaky ReLU activations. It was trained using the Adam optimizer [25] with a learning rate of $3 \cdot 10^{-4}$. Also, early stopping was applied to stop the training process when no improvement of the loss across the validation set was obtained. The values of α and β were empirically set to 10 and 0.001, respectively, using a grid search on the validation set.

4.4. Evaluation setup

The PAD systems were evaluated using the pooled equal error rate ($\text{EER}_{\text{spoof}}$) across all attacks. Likewise, the ASV systems were also evaluated using the EER_{ASV}, employing both *bonafide* utterances (target and non-target) and *spoofing* utterances. Any utterance rejected by either the PAD or ASV subsystems was assigned arbitrarily a $-\infty$ score for computing the integration performance. Then, the integration (biometrics) systems were evaluated using the joint EER ($\text{EER}_{\text{joint}}$) and the minimum normalized detection cost function (min-tDCF) [26] with the same configuration as the one employed in the ASVspoof 2019 challenge [11]. All the PAD, ASV and biometrics systems were evaluated using the ASVspoof 2019 test datasets.

System	Logical Access Attacks				Physical Access Attacks			
	$\text{EER}_{\text{spoof}}(\%)$	$\text{EER}_{\text{ASV}}(\%)$	$\text{EER}_{\text{joint}}(\%)$	min-tDCF	$\text{EER}_{\text{spoof}}(\%)$	$\text{EER}_{\text{ASV}}(\%)$	$\text{EER}_{\text{joint}}(\%)$	min-tDCF
No Attack	5.91	31.10*	20.13	0.1252	4.77	18.62*	13.37	0.1238
FGSM ($\epsilon = 0.1$)	5.98	31.14*	20.32	0.1279	7.50	18.65*	15.47	0.2157
PGD ($\epsilon = 0.1$)	5.95	31.13*	20.25	0.1267	6.08	18.63*	14.38	0.1717
FGSM ($\epsilon = 1.0$)	8.15	31.53*	25.44	0.1287	35.64	18.71*	26.54	0.9335
PGD ($\epsilon = 1.0$)	7.02	31.46*	25.37	0.1266	44.42	18.83*	26.77	0.9665
FGSM ($\epsilon = 2.0$)	2.01	30.11*	14.13	0.0623	1.02	17.61*	11.82	0.0380
PGD ($\epsilon = 2.0$)	4.97	31.38*	22.62	0.1078	29.29	18.44*	25.28	0.8677
FGSM ($\epsilon = 5.0$)	0.00	19.46*	2.45	0.0000	0.00	11.37*	11.79	0.0000
PGD ($\epsilon = 5.0$)	0.16	19.09*	2.56	0.0058	0.00	9.48*	11.79	0.0000
Proposed ABTN	**35.19**	**31.52***	**39.15**	**0.5829**	**95.17**	**18.87***	**36.63**	**1.0000**

Table 1: *Results of the black-box adversarial attacks on the ASVspoof 2019 logical access (LA) and physical access (PA) test sets in terms of $\text{EER}_{spoof}(\%)$, $\text{EER}_{ASV}(\%)$, $\text{EER}_{joint}(\%)$ and min-tDCF. The target PAD system is based on a LCNN, while the student PAD system is based on a SENet50. The target ASV system is based on a TDNN + PLDA, while the student ASV system is based on a TDNN + b-vector. (*) The ASV evaluation includes both bonafide and spoofing utterances.*

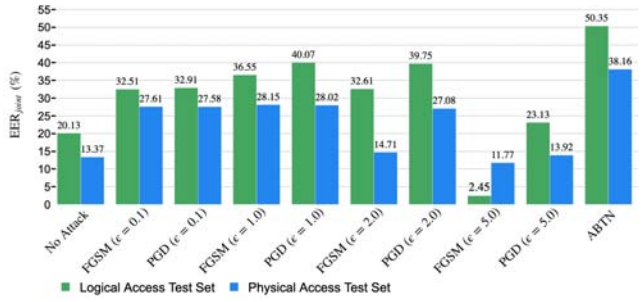

Figure 4: *$\text{EER}_{joint}(\%)$ of the white-box adversarial attacks on the ASVspoof 2019 logical and physical access test sets.*

5. Experimental Results

The performance of the baseline biometrics system is shown in Table 1 as 'No Attack'. The LA and PA PAD systems are among the best single systems evaluated in the ASVspoof 2019 challenge [11]. The ASV system yields an EER of 4.75 and 7.25% in the LA and PA datasets when evaluating only the target and non-target *bonafide* utterances. However, its performance is degraded to 31.10 and 18.62% in the LA and PA test datasets when the *spoofing* utterances are also evaluated, as shown in Table 1.

5.1. White-box scenario

Fig. 4 shows the $\text{EER}_{\text{joint}}$ of the white-box adversarial attacks evaluated in the ASVspoof 2019 LA and PA test sets. The PAD and ASV systems are the state-of-the-art LCNN and TDNN + PLDA, respectively. As it was expected, PGD achieves slightly better results than FGSM due to its iterative procedure for generating the adversarial attacks. Moreover, the proposed ABTN outperforms the rest of adversarial attacks, obtaining 10.28% and 10.14% higher $\text{EER}_{\text{joint}}$ with respect to the best PGD configuration ($\epsilon = 1.0$) in the LA and PA test sets, respectively. It is worth noticing that when the hyper-parameter ϵ of the FGSM and PGD methods is equal or higher than 2.0, the biometrics system is able to detect the perturbation noise added by these adversarial attacks. In these cases, the performance of the *spoofing* attacks is even worse than when not using any adversarial attack (denoted by 'No Attack').

5.2. Black-box scenario

Table 1 shows the performance metrics for the black-box scenario. The target biometrics system consists of the same state-of-the-art LCNN (PAD) and TDNN + PLDA (ASV) systems evaluated in the previous section. The student PAD and ASV systems are the SENet50 and the TDNN + b-vector systems, respectively.

The proposed ABTN attacks outperform the best FGSM and PGD configurations by 27.04 and 50.75% of $\text{EER}_{\text{spoof}}$, and by 13.71 and 9.86% of $\text{EER}_{\text{joint}}$, respectively. Also, the min-tDCF metric, which shows the performance of the biometrics system on a different operating point with respect to the $\text{EER}_{\text{joint}}$ [26], is significatively higher for the proposed ABTN adversarial attacks. As in the white-box scenario, it is worth noticing that the best adversarial attacks do not affect the performance of the ASV system with respect to the baseline system, since the perturbation noise of these attacks is not detected by the ASV system. However, when the hyper-parameter $\epsilon \geq 2.0$, both the PAD and ASV systems are able to detect the perturbations added by the FGSM and PGD methods, and hence, the biometrics system performs even better than the baseline system (denoted by 'No Attack'). However, the proposed ABTN method does not suffer from this issue since it is trained so that the added perturbation noise does not modify the speaker information from the *spoofing* utterance.

6. Conclusion

In this work, we studied the robustness of state-of-the-art voice biometrics systems (ASV + PAD) under the presence of adversarial *spoofing* attacks. Moreover, we proposed an adversarial biometrics transformation network (ABTN) for both white-box and black-box scenarios which is able to generate adversarial *spoofing* attacks in order to fool the PAD system without being detected by the ASV system. Experimental results have shown that biometric systems are highly sensitive to adversarial *spoofing* attacks in both logical and physical access scenarios. Moreover, the proposed ABTN system clearly outperforms other popular adversarial attacks such as the FGSM and PGD methods in both white-box and black-box scenarios. In the future, we would like to use the generated adversarial attacks for adversarial training in order to make the biometrics system more robust against these attacks.

7. References

[1] R. Naika, "An overview of automatic speaker verification system," in *Advances in Intelligent Systems and Computing*. New York, NY, USA: Springer, 2018, vol. 673.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] A. Gomez-Alanis, J. A. Gonzalez-Lopez, S. P. Dubagunta, A. M. Peinado, and M. Magimai.-Doss, "On joint optimization of automatic speaker verification and anti-spoofing in the embedding space," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 1579–1593, 2021.

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

[8] A. Gomez-Alanis, A. M. Peinado, J. A. Gonzalez, and A. M. Gomez, "A deep identity representation for noise robust spoofing detection," in *Proc. Interspeech*, Hyderabad, India, 2018, pp. 676–680.

[9] 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.

[10] 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.

[11] 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.

[12] S. Liu, H. Wu, H. yi Lee, and H. Meng, "Adversarial attacks on spoofing countermeasures of automatic speaker verification," in *Proc. IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2019, pp. 312–319.

[13] Y. Zhang, Z. Jiang, J. Villalba, and N. Dehak, "Black-box attacks on spoofing countermeasures using transferability of adversarial examples," in *Proc. Interspeech*, 2020, pp. 4238–4242.

[14] K. Ren, T. Zheng, Z. Qin, and X. Liu, "Adversarial attacks and defenses in deep learning," *Engineering*, vol. 6, no. 3, pp. 346 – 360, 2020.

[15] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," in *Proc. International Conference on Learning Representations (ICLR)*, Banf, Alberta, Canada, 2014.

[16] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *Proc. International Conference on Learning Representations (ICLR)*, San Diego, CA, USA, 2015.

[17] Y. Deng and L. J. Karam, "Universal adversarial attack via enhanced projected gradient descent," in *Proc. IEEE International Conference on Image Processing (ICIP)*, 2020, pp. 1241–1245.

[18] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust DNN embeddings for speaker recognition," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Calgary, Alberta, Canada, 2018, pp. 5329–5333.

[19] 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.

[20] 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.

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

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

[23] C.-I. Lai, N. Chen, J. Villalba, and N. Dehak, "ASSERT: Anti-Spoofing with Squeeze-Excitation and Residual Networks," in *Proc. Interspeech*, 2019, pp. 1013–1017.

[24] 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.

[25] 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.

[26] 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.

Chapter 3

Conclusions and Future Work

3.1 Conclusions

This thesis aimed at designing anti-spoofing and voice biometrics systems that outperform other state-of-the-art techniques for automatic speaker verification. In order to do so, we have adopted different approaches. In this section, we briefly summarize the four research topics tackled in this thesis.

- First, we have have dealt with the problem of *spoofing* attack detection for voice biometric systems. The main problem here is the lack of robustness and generalization across different databases. We addressed this issue by proposing a novel neural network architecture which can be used for detecting both logical and physical access *spoofing* attacks. The proposed convolutional RNN-based architecture is able to process the whole input utterance without cropping it or applying any post-processing combination of chunks. Moreover, since noisy acoustic scenarios can significantly degrade the performance of anti-spoofing systems, we have also proposed two noise-aware techniques based on the usage of masks which help to effectively reduce the performance degradation. Our best performing technique involves the computation and use of signal-to-noise masks that inform the DNN-based *spoofing* embedding extractor of the noise probability for each time-frequency bin in the input speech spectrogram.

- Secondly, we also proposed new loss functions which can be effectively used by anti-spoofing and integration of ASV and anti-spoofing systems. We have proposed a new probabilistic loss function for supervised metric learning, where every training class is represented with a probability density function using all the samples of the mini-batch and is estimated through kernel density estimation. We can argue that each class is more accurately represented than in other popular loss functions. Moreover, the proposed loss function replaces the concept of distance between embeddings in negative hard-mining techniques by the concept that an embedding belongs to a class with a given probability. This has the advantage of avoiding the selection of an appropiate distance measure and tuning extra hyper-parameters such as distance margins. Furthermore, we also propose a new loss function for integration systems based on the expected performance and spoofability curve (EPSC) [69] which allows to optimize the voice biometric system in the operating range, instead of only one operating point, in which it is expected to work during evaluation. These proposals allow to improve significantly the performance of both anti-spoofing and complete voice biometric systems.

- Third, we have studied the integration of ASV and anti-spoofing systems at the score-level and at the embedding-level. To avoid the integration of ASV and anti-spoofing systems at the score-level using scores computed separately, we proposed a new neural network architecture for integrating the systems at the embedding-level which exploits the fact that ASV and anti-spoofing systems share the *bonafide* speech subspace. Thus, the proposed integration system is able to model the three main biometric speech subspaces: *bonafide* speech, *zero-effort* attacks and *spoofing* attacks. Experimental results on the ASVspoof 2019 corpus show that the joint processing of the ASV and PAD embeddings with the proposed integration neural network clearly outperforms other state-of-the-art techniques trained and evaluated on the same conditions.

- Finally, we have studied the robustness of the state-of-the-art voice biometric systems under the presence of adversarial *spoofing* attacks. Furthermore, we also proposed a new DNN-based generator network for this type of attacks which is trained using existing *spoofing* attacks and it can be used for finetuning the biometric system in order to make it more robust to adversarial *spoofing* attacks. Experimental results show that voice biometric systems are highly sensitive to adversarial *spoofing* attacks in both logical and physical access scenarios. Moreover, the proposed ABTN generator clearly outperforms other classical adversarial attacks techniques such as the fast gradient signed method (FGSM) [86] and the projected gradient descent (PGD) [87].

Conclusiones

Esta tesis se ha centrado en diseñar sistemas biométricos de voz y anti-spoofing que superen a otras técnicas del estado del arte para la verificación automática de locutores. Para ello, hemos adoptado diferentes enfoques. En este apartado, resumimos brevemente los cuatro temas de investigación abordados en esta tesis.

- En primer lugar, nos hemos centrado en el problema de la detección de ataques de suplantación de identidad para sistemas biométricos de voz. El principal problema aquí es la falta de solidez y generalización en diferentes bases de datos. Abordamos este problema proponiendo una arquitectura de red neuronal novedosa que se puede utilizar para detectar ataques de acceso tanto lógicos como físicos. La arquitectura convolucional basada en RNN propuesta es capaz de procesar toda la locución de entrada sin recortarla ni aplicar ninguna combinación de fragmentos de posprocesamiento. Además, dado que los escenarios acústicos ruidosos pueden degradar significativamente el rendimiento de los sistemas de anti-spoofing, también hemos propuesto dos técnicas de detección de ruido basadas en el uso de máscaras que ayudan a reducir eficazmente la degradación del rendimiento. Nuestra técnica de mejor rendimiento consiste en el uso de máscaras de señal a ruido que informan al extractor de características de suplantación basado en DNNs de la probabilidad de ruido para cada intervalo de tiempo-frecuencia en el espectrograma de la voz de entrada.

- En segundo lugar, también hemos propuesto nuevas funciones de coste que pueden ser utilizadas eficazmente para anti-spoofing y sistemas de integración biométricos. Hemos propuesto una nueva función de coste probabilística para el aprendizaje métrico supervisado, donde cada clase de entrenamiento se representa con una función de densidad de probabilidad utilizando todas las muestras del lote de entrenamiento y que se obtiene mediante

técnicas de estimación de densidad del kernel. Podemos argumentar que cada clase está representada con mayor precisión que en otras funciones de coste populares. Además, la función de coste propuesta reemplaza el concepto de distancia entre *embeddings* en técnicas de minería negativa por el concepto de que de que un *embedding* pertenece a una clase con una probabilidad determinada. Esto tiene la ventaja de evitar la selección de una medida de distancia adecuada y ajustar hiperparámetros adicionales como los márgenes de distancia. Además, también hemos propuesto una nueva función de coste para sistemas de integración que permite optimizar el sistema biométrico de voz en el rango de operación, en lugar de en un solo punto de operación, en el que se espera que trabaje el sistema en producción. Estas propuestas permiten mejorar significativamente el rendimiento de los sistemas biométricos de voz.

- En tercer lugar, hemos estudiado la integración de ASV y sistemas de anti-spoofing a nivel de puntuación y a nivel de *embeddings*. Para evitar la integración de ASV y sistemas de anti-spoofing a nivel de puntuación utilizando puntuaciones calculadas por separado, hemos propuesto una nueva arquitectura de red neuronal para integrar los sistemas a nivel de *embeddings* que explota el hecho de que el sistema de ASV y los sistemas de anti-spoofing comparten el subespacio de voz genuino. Por lo tanto, el sistema de integración propuesto es capaz de modelar los tres subespacios biométricos de voz principales: voz genuina, ataques de esfuerzo cero y ataques de suplantación de identidad. Los resultados experimentales en la base de datos ASVspoof 2019 muestran que el procesamiento conjunto de lao *embeddings* de ASV y PAD con la red neuronal de integración propuesta supera claramente a otras técnicas del estado del arte entrenadas y evaluadas en las mismas condiciones.

- Por último, hemos estudiado la robustez de los sistemas biométricos de voz de última generación bajo la presencia de ataques de suplantación de voz adversarios. Además, también hemos propuesto una nueva red generadora basada en DNNs para este tipo de ataques que se entrena usando ataques de suplantación ya existentes y que se puede usar para ajustar el sistema biométrico con el fin de hacerlo más robusto frente a ataques de suplantación de identidad adversarios. Los resultados experimentales muestran que los sistemas biométricos de voz son muy sensibles a los ataques de suplantación adversarios en escenarios de acceso tanto lógico como físico. Además, el generador ABTN propuesto supera claramente a otras técnicas clásicas de ataques adversarios, como el método rápido con signo del gradiente (FGSM) [86] y el gradiente del descenso proyectado (PGD) [87].

3.2 Future work

Having developed the techniques presented in this thesis, various research topics have arisen that deserve further reseach in the future.

KDE-based loss functions can be applied to other classification applications. The new proposed concept of loss functions based on KDE techniques can be rather considered a general approach since it can be applied to any DNN-based embedding extraction system which comprises fully connected layers. Thus, the proposed loss functions can be applied to any classification problem which is solved by using neural networks.

A thorough study on the robustness and generalization of anti-spoofing systems. Although the proposed anti-spoofing techniques have been shown to work well on single databases

such as ASVspoof 2015, 2017 and 2019, we have not studied the performance results across different databases [7], i.e., training with the development dataset of one database and evaluating with either the development or evaluation dataset of a different database. One of the main problems of voice anti-spoofing systems is their ability to perform well across different types of *spoofing* attacks, i.e., evaluating with attacks which are different to the attacks employed during training. Thus, it is worth exploring this type of generalization study and propose new robust techniques which can perform well across different datasets.

Employ the generated adversarial *spoofing* attacks in order to finetune the voice biometric system through adversarial training [88] and make it more robust against adversarial *spoofing* attacks. Although the proposed generator (ABTN) network is able to generate very precise adversarial *spoofing* attacks, we still need to propose a defense strategy such as adversarial training in order to improve the robustness of the biometric system. In fact, this is still ongoing work.

It has been shown that the techniques proposed in this thesis can be successfully applied to voice biometric systems. We also envision that the proposed techniques can be effectively applied in other biometric applications, taking into account that its hyperparameters should be adapted according to the new biometric system.

Bibliography

[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. Jain, A. Ross, and S. Prabhakar, "An introduction to biometric recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 1, pp. 4–20, 2004.

[3] A. K. Jain, S. Pankanti, S. Prabhakar, L. Hong, and A. Ross, "Biometrics: A grand challenge," in *Proceedings of the International Conference on Pattern Recognition (ICPR)*, vol. 2, 2004, pp. 935–942.

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

[5] 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.

[6] *Presentation attack detection*. [Online]. Available: `https://www.iso.org/standard/67381.html`.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] J. Yamagishi, X. Wang, M. Todisco, M. Sahidullah, J. Patino, A. Nautsch, X. Liu, K. A. Lee, T. Kinnunen, N. Evans, and H. Delgado, "ASVspoof 2021: accelerating progress in spoofed and deepfake speech detection," in *Proc. 2021 Edition of the Automatic Speaker Verification and Spoofing Countermeasures Challenge*, 2021, pp. 47–54.

[13] 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.

[14] 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.

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

[16] 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.

[17] 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.

[18] 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.

[19] 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.

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

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

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

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

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

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

[26] S. Yadav and A. Rai, "Learning discriminative features for speaker identification and verification," in *Proc. Interspeech*, 2018.

[27] J. Yang, C. You, and Q. He, "Feature with complementary of statistics and principal information for spoofing detection," in *Proc. Interspeech*, 2018.

[28] 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.

[29] 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. DOI: 10.1109/MSP.2015.2462851.

[30] 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.

[31] J. Huang, J. Li, and Y. Gong, "An analysis of convolutional neural networks for speech recognition," in *Proc. International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015.

[32] W. Cai, D. Cai, W. Liu, and M. Li, "Countermeasures for automatic speaker verification replay spoofing attack: On data augmentation, feature representation, classification and fusion," in *Proc. Interspeech*, 2017.

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

[34] K. Okabe, T. Koshinaka, and K. Shinoda, "Attentive statistics pooling for deep speaker embedding," in *Proc. Interspeech*, 2018.

[35] C. Zhang, C. Yu, and J. 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.

[36] S. Scardapane, L. Stoffl, F. Rohrbein, and A. Uncini, "On the use of deep recurrent neural networks for detecting audio spoofing attacks," in *Proc. International Joint Conference on Neural Networks (IJCNN)*, 2017.

[37] K. Sriskandaraja, V. Sethu, and E. Ambikairajah, "Deep siamese architecture based replay detection for secure voice biometric," in *Proc. Interspeech*, 2018.

[38] Z. Chen, W. Zhang, Z. Xie, X. Su, and D. Chen, "Recurrent neural networks for automatic replay spoofing attack detection," in *Proc. International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018.

[39] L. Huang and C. Pun, "Audio replay spoof attack detection using segment-based hybrid feature and densenet-lstm network," in *Proc. International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019.

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

[41] 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.

[42] Z. Yu, C. Zhao, Z. Wang, Y. Qin, Z. Su, X. Li, F. Zhou, and G. Zhao, "Searching central difference convolutional networks for face anti-spoofing," in *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[43] L. Li, Z. Xia, A. Hadid, X. Jiang, H. Zhang, and X. Feng, "Replayed video attack detection based on motion blur analysis," *IEEE Transactions on Information Forensics and Security*, vol. 14, pp. 2246–2261, 2019.

[44] L. Weiyang, W. Yandong, Y. Zhiding, L. Ming, R. Bhiksha, and S. Le, "Sphereface: Deep hypersphere embedding for face recognition," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[45] W. Feng, C. Jian, L. Weiyang, and L. Haijun, "Additive margin softmax for face verification," *IEEE Signal Processing Letters*, vol. 25, no. 7, pp. 926–930, 2018.

[46] G. Lavrentyeva, S. Novoselov, A. Tseren, M. Volkova, A. Gorlanov, and A. Kozlov, "STC anti-spoofing systems for the ASVspoof2019 challenge," in *Proc. Interspeech*, Graz, Austria, 2019, pp. 1033–1037.

[47] Y. Yu, L. Fan, and W. Li, "Ensemble additive margin softmax for speaker verification," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019.

[48] L. Yutian, G. Feng, O. Zhijian, and S. Jiasong, "Angular softmax loss for end-to-end speaker verification," in *Proc. International Symposium on Chinese Spoken Language Processing*, 2018.

[49] L. Wan, Q. Wang, A. Papir, and I. Lopez-Moreno, "Generalized end-to-end loss for speaker verification," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019.

[50] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "Deepface: Closing the gap to human-level performance in face verification," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 1701–1708.

[51] E. Ahmed, M. Jones, and T. K. Marks, "An improved deep learning architecture for person re-identification," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 3908–3916.

[52] K. Chen and A. Salman, "Extracting speaker-specific information with a regularized siamese deep network," in *Proc. Neural Information Processing Systems (NIPS)*, 2011.

[53] E. Hoffer and N. Ailon, "Deep metric learning using triplet network," in *Proc. SIMBAD*, 2015.

[54] 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.

[55] C. Zhang, K. Koishida, and J. H. L. Hansen, "Text-independent speaker verification based on triplet convolutional neural network embeddings," *IEEE/ACM Transactions on Audio, Speech and Language Processing*, vol. 26, no. 9, pp. 1633–1644, 2018.

[56] C. Zhang and K. Koishida, "End-to-end text-independent speaker verification with triplet loss on short utterances," in *Proc. Interspeech*, 2017.

[57] 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.

[58] 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.

[59] 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.

[60] 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.

[61] 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.

[62] 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.

[63] 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.

[64] 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.

[65] 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.

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

[67] 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.

[68] 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.

[69] 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.

[70] S. Liu, H. Wu, H.-y. Lee, and H. Meng, "Adversarial attacks on spoofing countermeasures of automatic speaker verification," in *Proc. IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2019, pp. 312–319.

[71] Y. Zhang, Z. Jiang, J. Villalba, and N. Dehak, "Black-box attacks on spoofing countermeasures using transferability of adversarial examples," in *Proc. Interspeech*, 2020, pp. 4238–4242.

[72] K. Ren, T. Zheng, Z. Qin, and X. Liu, "Adversarial attacks and defenses in deep learning," *Engineering*, vol. 6, no. 3, pp. 346–360, 2020.

[73] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," in *Proc. International Conference on Learning Representations (ICLR)*, Banf, Alberta, Canada, 2014.

[74] 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.

[75] H.-S. Heo, J.-w. Jung, I.-H. Yang, S.-H. Yoon, H.-j. Shim, and H.-J. Yu, "End-to-end losses based on speaker basis vectors and all-speaker hard negative mining for speaker verification," in *Proc. Interspeech*, 2019.

[76] D. Yi, Z. Lei, S. Liao, and S. Z. Li, "Deep metric learning for person re-identification," in *Proc. International Conference on Pattern Recognition*, 2014, pp. 34–39.

[77] K. Sohn, "Improved deep metric learning with multi-class n-pair loss objective," in *Proc. Neural Information Processing Systems (NIPS)*, 2016.

[78] E. Parzen, "On estimation of a probability density function and mode," *The Annals of Mathematical Statistics*, vol. 33, pp. 1065–1076, 1962.

[79] W. Hardle, A. Werwatz, M. Muller, and S. Sperlich, "Nonparametric and semiparametric models," *Springer Series in Statistics*, 2004.

[80] A. M. Peinado, J. Koloda, A. M. Gomez, and V. E. Sanchez, "A statistical analysis of the kernel-based mmse estimator with application to image reconstruction," *Signal Processing: Image Communication*, vol. 55, pp. 41–54, 2017.

[81] X. Wu, R. He, Z. Sun, and T. Tan, "A light cnn for deep face representation with noisy labels," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 11, pp. 2884–2896, 2018.

[82] D. Wang, U. Kjems, M. Perdersen, J. Boldt, and T. Lunner, "Speech intelligibility in background noise with ideal binary time-frequency masking," *Journal of Acoustical Society of America*, vol. 125, pp. 2336–2347, 2009.

[83] D. Wang and J. Cheng, "Supervised speech separation based on deep learning: An overview," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 10, pp. 1702–1726, 2018.

[84] L. Yutian, G. Feng, O. Zhijian, and S. Jiasong, "Angular softmax loss for end-to-end speaker verification," in *Proc. International Symposium on Chinese Spoken Language Processing*, 2018.

[85] J. Deng, J. Guo, and S. Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 4685–4694.

[86] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *Proc. International Conference on Learning Representations (ICLR)*, San Diego, CA, USA, 2015.

[87] Y. Deng and L. J. Karam, "Universal adversarial attack via enhanced projected gradient descent," in *Proc. IEEE International Conference on Image Processing (ICIP)*, 2020, pp. 1241–1245.

[88] H. Wu, S. Liu, H. Meng, and H.-y. Lee, "Defense against adversarial attacks on spoofing countermeasures of asv," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 6564–6568.