Universidad del País Vasco Euskal Herriko Unibertsitatea
BILBOKO INGENIARITZA ESKOLA
ESCUELA DE INGENIERÍA DE BILBAO

GRADO EN INGENIERÍA EN TECNOLOGÍA DE TELECOMUNICACIÓN

TRABAJO FIN DE GRADO

<INTEGRACIÓN DE BASES DE DATOS PARA LA DETECCIÓN DE ATAQUES MEDIANTE SPOOFING>

Alumno/Alumna: <Baranda, Barron, Jon Andoni>
Director: <Sánchez, De La Fuente, Jon>

Curso: <2019-2020>

Fecha: <Bilbao, 20, Julio, 2020>

Resumen

La seguridad es una demanda inherente a la condición humana sobre cualquiera de nuestros actos, pertenencias y nosotros mismos, en definitiva. La información, y su repercusión sobre nuestra propia integridad, tampoco está excluida de dicha demanda y los investigadores hemos de integrar el concepto de SEGURIDAD en el desarrollo de cada uno de los proyectos que abordamos.

En este campo podemos diferenciar dos ámbitos principales, la seguridad física y la seguridad de la información. La seguridad física es una estrategia para proteger las instalaciones, los activos, los recursos y las personas de los incidentes o acciones que pueden causar pérdidas o daños a estas entidades. La seguridad de la información, es una estrategia para proteger la integridad y privacidad del contenido con seguridad digital. A día de hoy la forma de identificación más común es el uso de contraseñas, llaves, tarjetas… Una pega de estos métodos es que pueden ser robados u olvidados.

Por otro lado, encontramos herramientas como la biometría, una práctica más nueva, que se está utilizando para implementar seguridad tanto física como de información. En comparación con los métodos tradicionales de contraseñas, llaves y similares, la biometría es una posesión que siempre se posee y ahí reside su principal ventaja. En la seguridad biométrica es común el uso de la huella dactilar, estructura facial, el iris o la voz. En lo que a esta última se refiere, la biometría de la voz, es la ciencia de utilizar la voz de una persona como una característica biológica de identificación única para autenticarla. También conocida como verificación de voz o reconocimiento de hablante, la biometría de voz permite un acceso rápido, no intrusivo y seguro para una variedad de casos de uso, desde *call centers*, aplicaciones móviles o aplicaciones en línea, hasta *chatbots*, dispositivos IoT (*Internet of Things*) y de acceso físico.

Si existe la necesidad de implementar sistemas de seguridad es por la existencia, a su vez, de un riesgo cierto; hay algo o alguien de quien protegerse. En el caso de la biometría de voz, son los denominados ataques *spoofing* o de suplantación de identidad los que constituyen una gran amenaza para la seguridad. De cara a hacer frente a estos ataques, diversos estudios e instituciones tienden a implementar módulos de detección de habla sintética (SSD). El funcionamiento de esta tecnología se basa en un clasificador que dispone de dos modelos diferentes, uno de habla humana y otro de habla sintética. Cuando un usuario trata de verificarse frente al sistema, la señal se compara con ambos modelos y, si la

diferencia de similitudes supera un umbral, se acepta como humana, en caso contrario se rechaza clasificándola como sintética.

Durante el desarrollo de esta tecnología, los sistemas deben ser entrenados y para ello se utiliza una gran cantidad de grabaciones de voz, que servirán para crear los modelos mencionados antes. A lo largo de este Trabajo de Fin de Grado se estudia la utilización de bases de datos por parte de estos sistemas para la detección de ataques mediante *spoofing*. Para llevar a cabo esta tarea se hace uso de un SSD basado tanto en parámetros espectrales MFCC como los parámetros de la fase armónica, RPS. Asimismo, se realizan pruebas con redes neuronales con el objetivo último de obtener resultados con menor probabilidad de error. Se hace uso de las denominadas redes neuronales DNN (*Deep Neural Networks*) para la mejora de la tarea de clasificación.

Palabras clave: Bases de datos, spoofing, redes neuronales, verificación de locutor.

eman ta zabal zazu
Universidad del País Vasco
Euskal Herriko Unibertsitatea
BILBOKO INGENIARITZA ESKOLA
ESCUELA DE INGENIERÍA DE BILBAO

Abstract

Security is an inherent demand of the human condition on any of our acts, belongings and ourselves in short. The information, and its repercussion on our own integrity, is not excluded from this demand either, and researchers must integrate the concept of SECURITY in the development of each of the projects that we tackle.

In this field we can differentiate two main areas, physical security and information security. Physical security is a strategy to protect facilities, assets, resources, and people from incidents or actions that can cause loss or damage to these entities. Information security is a strategy to protect the integrity and privacy of content with digital security. Today the most common form of identification is the use of passwords, keys, cards ... One drawback to these methods is that they can be stolen or forgotten.

On the other hand, we find tools such as biometrics, a newer practice, which is being used to implement both physical and information security. Compared to traditional methods of passwords, keys and similar, biometrics is a possession that is always possessed and that is its main advantage. In biometric security, the use of fingerprint, facial structure, iris or voice is common. As far as the latter is concerned, voice biometrics is the science of using a person's voice as a uniquely identifying biological feature to authenticate them. Also known as voice verification or speaker recognition, voice biometrics enables fast, non-intrusive and secure access for a variety of use cases, from call centers, mobile applications, or online applications, to chatbots, IoT (Internet of Things) and physical access.

If there is a need to implement security systems, it is due to the existence, in turn, of a certain risk; there is something or someone to protect yourself from. In the case of voice biometrics, it is the so-called spoofing or spoofing attacks that constitute a great security threat. In order to deal with these attacks, various studies and institutions tend to implement synthetic speech detection (SSD) modules. The operation of this technology is based on a classifier that has two different models, one of human speech and the other of synthetic speech. When a user tries to verify against the system, the signal is compared with both models and, if the difference in similarities exceeds a threshold, it is accepted as human; otherwise, it is rejected, classifying it as synthetic.

During the development of this technology, the systems must be trained and in order to accomplish this, a large number of voice recordings are used, which will serve to create the models mentioned above. Throughout this Final Degree Project, it is studied the use of databases by these systems to detect spoofing attacks. To

carry out this task, an SSD is used based on both MFCC spectral parameters and the harmonic phase parameters, RPS. Likewise, tests are carried out with neural networks with the ultimate objective of obtaining results with a lower probability of error. In this project, the so-called DNN neural networks (Deep Neural Networks) are used to improve the classification task.

Key words: Databases, spoofing, neural network, automatic speaker verification.

Laburpena

Segurtasuna giza izaerari datxekion eskaria da, gure egintza, ondasun eta, azken batean, geure buruaren gainekoa. Informazioa eta horrek gure osotasunean duen eragina ere ez daude eskari horretatik kanpo, eta ikertzaileok segurtasunaren kontzeptua txertatu behar dugu lantzen dugun proiektu bakoitzaren garapenean.

Eremu honetan, bi eremu nagusi bereiz ditzakegu: segurtasun fisikoa eta informazioaren segurtasuna. Segurtasun fisikoa estrategia bat da instalazioak, aktiboak, baliabideak eta pertsonak erakunde horiei galerak edo kalteak eragin diezazkieketen intzidenteetatik edo ekintzetatik babesteko. Informazioaren segurtasuna berriz, edukiaren osotasuna eta pribatutasuna segurtasun digitalarekin babesteko estrategia bat da. Gaur egun, identifikatzeko modurik ohikoena pasahitzak, giltzak eta txartelak erabiltzea da. Metodo horien alde txarra , lapurtu edo ahaztu egin daitezkeela da.

Bestalde, biometria bezalako tresnak aurkitzen ditugu, praktika berriago bat, segurtasun fisikoa zein informaziokoa ezartzeko erabiltzen dena. Pasahitz, giltza eta antzekoen metodo tradizionalen aldean, biometria beti edukitzen den edukitza da, eta hor datza bere abantaila nagusia. Segurtasun biometrikoan ohikoa da hatz-marka, aurpegi-egitura, irisa edo ahotsa erabiltzea. Azken horri dagokionez, ahotsaren biometria pertsona baten ahotsa identifikatzeko ezaugarri biologiko bakar gisa erabiltzeko zientzia da. Ahots-egiaztapen edo hiztun-aintzatespen gisa ere ezagutzen da, eta ahots-biometriak sarbide azkarra, ez intrusiboa eta segurua ahalbidetzen du erabilera-kasu anitzetarako: call center-ak, aplikazio mugikorrak edo lineako aplikazioak, chatbotak, IoT gailuak (*Internet of Things*) eta sarbide fisikokoak.

Segurtasun-sistemak inplementatzeko beharra, aldi berean, arrisku ziurra dagoela esan nahi du, hau da, bada babesteko zerbait edo norbait. Ahots-biometriaren kasuan, *spoofing* edo nortasuna ordezteko erasoak dira segurtasunerako mehatxu handiak. Eraso horiei aurre egiteko, hainbat azterlan eta erakundek hizkera sintetikoa hautemateko moduluak (SSD) inplementatzeko joera dute. Teknologia honen funtzionamendua bi eredu desberdin dituen sailkatzaile batean oinarritzen da, bata giza hizkerakoa eta bestea hizkuntza sintetikokoa. Erabiltzaile bat sistemaren aurrean bere burua egiaztatzen saiatzen denean, seinalea bi ereduekin alderatzen da eta, antzekotasun-aldeak atalase bat gainditzen badu, gizakitzat hartzen da; bestela, baztertu egiten da, sintetikotzat sailkatuz.

Teknologia hori garatzeko prozesuan, sistemak entrenatu egin behar dira, eta, horretarako, ahots-grabazio ugari erabiltzen dira, lehen aipatutako ereduak sortzeko. Gradu Amaierako Lan honetan, datu baseen erabilera aztertzen da sistema hauek *spoofing* bidezko erasoak detektatzeko atazan. Zeregin hau Aurrera eramateko, SSD bat erabiltzen da, MFCC parametro espektraletan eta fase harmonikoaren parametroetan (RPS) oinarrituta. Halaber, probak egiten dira sare neuronalekin, errore-probabilitate txikiagoko emaitzak lortzeko azken helburuarekin. DNN (*Deep Neural Networks*) sare neuronalak erabiltzen dira sailkapen-lana hobetzeko.

Hitz gakoak: Datu baseak, spoofing, sare neuronalak, esatari egiaztatzailea.

ÍNDICE

Lista de ilustraciones

Lista de tablas

Lista de acrónimos

- AAC: Advanced Audio Coding
- ANS: Artificial Neural Systems
- ASV: Automatic Speaker Verification
- BD: Base de Datos
- CPU: Central Processing Unit
- DCT: Discrete Cosine Transform
- DET: Detection Error Tradeoff
- DFT: Discrete Fourier Transform
- DNN: Deep Neural Networks
- EER: Equal Error Rate
- FAR: False Acceptance Rate
- FLAC: Free Lossless Audio Codec
- FRR: False Rejection Rate
- GMM: Gaussian Mixture Model
- GPU: Graphics Processing Unit
- HMM: Hidden Markov Model
- IoT: Internet of Things
- LSTM: Long-Short Term Memory
- MBE: MultiBand Excitation
- MFCC: Mel Frequency Cepstral Coefficients
- MGD: Modified Group Delay
- NIST: National Institute of Standards and Technology
- PA: Playback Attack
- RPS: Relative Phase Shift
- SSD: Synthethic Speech Detection

- TTS: Text To Speech
- VC: Voice Conversion
- WAV: Waveform Audio File Format

1.Introducción

En la era digital en la que vivimos, el flujo de información, entre particulares y organizaciones -sean empresas o Administraciones- o entre ambos, crece de forma exponencial, y con ello se multiplica asimismo el riesgo para la accesibilidad e integridad de la información, por lo que la Seguridad debe ser una prioridad; establecer quien puede y quien no puede acceder a un servicio. A pesar de que en los últimos años ha habido un gran avance en la seguridad informática, la forma de identificación dominante sigue siendo mediante contraseñas alfanuméricas.

Sin embargo, el hecho de hacer un gran uso de este tipo de contraseñas no significa que sea la mejor opción, es más, tiene grandes inconvenientes como son el olvido y el robo de las mismas. Por esta razón durante los últimos años se ha tratado de buscar una alternativa, una de las mejores es la seguridad mediante sistemas biométricos. Se entiende por biometría a cada una de las características particulares del individuo. Existen dos tipos, por un lado, se encuentra la biometría física (propiedades fisiológicas de un usuario) como podría ser la huella digital, el iris o la geometría facial y, por otro lado, la biometría de comportamiento (propiedad conductual de un usuario) como son el patrón de tecleado o el análisis de voz. A pesar de que la voz también puede utilizarse como un rasgo biométrico fisiológico, como es el caso de este trabajo.

En el caso de esta última, la voz, es una característica personal única, diferente para cada uno de nosotros. Simplemente escuchando una palabra somos capaces de identificar sin problemas la voz de familiares o amigos. La razón de esto es que al hablar hacemos vibrar las cuerdas vocales, generando así ondas de sonido, que durante su camino al exterior se van a ver modificadas por el tracto vocal (garganta, lengua, dientes, boca, etc.). Estas modificaciones dependen por lo tanto de características como la longitud del cuello o la posición de los dientes, haciéndola así única. Además, la voz puede obtenerse de forma muy natural para el usuario y pudiéndose obtener también a distancia, sin encontrarse presencialmente en el lugar que solicita la verificación, por teléfono, por ejemplo. No obstante, sí que pueden ser copiados o falsificados. Una de las grandes preocupaciones en la verificación por voz son los denominados ataques *spoofing*. El *spoofing* es el uso de técnicas de suplantación de identidad con fines maliciosos.

De cara a defenderse frente ataques de este tipo, una medida de protección popular es implementar junto con el sistema de verificación de locutor, un módulo SSD (*Synthetic Speech Detection*) para la detección de voz sintética. El funcionamiento de este módulo a rasgos generales es el siguiente: generar, por un

lado, modelos de voz con los que identificar a los usuarios aceptados y, por otro lado, generar modelos de los diferentes tipos de ataques que hay. La creación de los modelos se basa en el procesamiento de una gran cantidad de señales de voz almacenadas en bases de datos, por lo tanto, una parte fundamental para la seguridad es disponer de una base de datos amplia y bien gestionada.

Como durante los últimos años ha incrementado en gran medida el número de ataques *spoofing,* la lucha contra este tipo de ataques genera investigación internacional. Multitud de instituciones de todo el mundo trabajan actualmente en métodos de detección de suplantación, trabajando cada una con su base de datos. Las bases de datos utilizadas juegan un gran papel, por esta razón este trabajo de fin de grado se centra en la integración de bases de datos para la detección de ataques *spoofing*.

2. Contexto

Todo proyecto surge por un motivo, el cual en este caso es la necesidad de integrar múltiples bases de datos con el objetivo de mejorar la detección de ataques *spoofing*. En este apartado se va a presentar el contexto del tema, es decir, el conjunto de circunstancias por las que se ha considerado necesario este estudio.

2.1 Grupo de investigación AhoLab

Aholab *Signal Processing Laboratory* es el nombre del grupo de investigación del laboratorio de procesamiento de señales de la Universidad del País Vasco. El grupo fue creado en el año 1992 y desde entonces ha llevado a cabo proyectos como el primer conversor de texto a voz para euskera, inglés y castellano. El laboratorio se encuentra en Bilbao y centra sus investigaciones principalmente en los siguientes temas: conversión de texto a voz, reconocimiento de orador y de voz, es decir, en general, en el procesamiento de la voz. [1]

Ilustración 1: Logo del grupo de investigación Aholab

Una de las investigaciones que llevan a cabo actualmente y en la que se centra este trabajo es la verificación automática de orador para la detección de ataques *Spoofing*. Dentro de esta línea, el grupo Aholab ha participado en las diferentes ediciones del *Automatic Speaker Verification Spoofing and Countermeasures Challenge* [2] [3] [4] [5] [6], además de estar preparando su participación en el challenge ASVspoof 2021. Los desafíos de ediciones anteriores han sido en 2015 [4], 2017 [5] y 2019 [6]. El objetivo de la tecnología ASV (*Automatic Speaker Verification*) es ofrecer una solución flexible y de bajo coste para la autenticación biométrica de personas. Actualmente la fiabilidad de los sistemas ASV es considerada suficiente para la adopción por parte del mercado de masas, sin embargo, existe la preocupación de que esta tecnología sea vulnerable a ataques *spoofing*. Los ataques *spoofing* son aquellos en los que un "estafador" trata de manipular un sistema biométrico haciéndose pasar por otra persona. [3]

Este tipo de ataques pueden llevarse acabo de muchas formas, algunas relativamente sencillas, como podría ser el caso de imitar la voz de una persona. Otras por el contrario pueden llegar a ser muy avanzadas, llegando a utilizar diferentes tecnologías de voz. Los ataques más comunes en este último caso suelen ser: transformación de la voz de un hablante para que sea percibida como si fuera la de otro hablante, modificación de la voz de un sistema TTS (*Text to Speech*) mediante unas muestras de voz.

Como medida de protección de un sistema biométrico basado en voz frente ataques como los mencionados, el grupo Aholab decidió desarrollar un módulo SSD. Para ello se tomó la fase en la voz como elemento distintivo, ya que como el oído humano descarta la fase, muchas aplicaciones no realizan el esfuerzo de modelarla correctamente. Por ello puede llegar a permitir distinguir una voz natural de una voz procesada. Para lograr el objetivo deseado, el sistema SSD (*Sythetic Speech Detection*) debe cumplir los siguientes objetivos:

- Validar la capacidad de un sistema SSD basado en la parametrización RPS (*Relative Phase Shift*) [7] como base de un sistema de detección, para ello comparándolo con un sistema de referencia basado en parámetros MFCC (*Mel Frequency Cepstral Coefficients*) [8].

- Deberá ser un sistema independiente del locutor, por este motivo se crearán modelos con distintos locutores y con distinto número de locutores.

- Comprobar la idoneidad del sistema simulando ataques con las bases de datos proporcionadas por los *challenge*. En ellas se encuentran voces humanas, y versiones sintéticas y falsificadas mediante distintas técnicas.

2.2 Entrenamiento y simulación de ataques

Como se ha mencionado en los objetivos del módulo SSD del grupo Aholab, el sistema deberá ser independiente del locutor. Por este motivo, el sistema deberá ser tanto entrenado como testeado con diferentes modelos. Para realizar estas tareas es necesario disponer de una gran cantidad de grabaciones de voz.

Dichas grabaciones serán utilizadas para generar diferentes modelos, los cuales el sistema utilizará para comparar la señal de entrada con ellos, y en función de las similitudes tomar la decisión de aceptar o de rechazar la señal. La creación de modelos adecuados tiene una gran importancia, ya que los resultados variaran en función del número de grabaciones utilizadas para generar los modelos, entre otros parámetros.

Para llevar a cabo este procedimiento, el grupo de investigación ha decidido hacer uso de diversas bases de datos, las cuales deberán ser gestionadas de forma correcta, es decir, de forma eficaz para realizar las tareas necesarias y al mismo tiempo utilizar los recursos de los que dispone el grupo Aholab de la mejor manera posible. Es este punto en el que se centrara principalmente este Trabajo de Fin de Grado.

2.4 Uso de bases de datos

Durante los últimos años, Aholab ha participado en los llamados *challenge ASVspoof* [3]. La visión de estos desafíos es fomentar el desarrollo de contramedidas contra la falsificación y reunir una comunidad para diseñar, recopilar y distribuir bases de datos estándar con protocolos y métricas de evaluación estándar. El desafío tiene como objetivo simular el desarrollo de contramedidas generalizadas con potencial para detectar ataques de falsificación variados e imprevistos.

Los participantes son invitados a desarrollar algoritmos de detección de suplantación de identidad y presentar resultados para una base de datos estándar disponible gratuitamente. Por lo tanto, las bases de datos de las que se harán uso serán las proporcionadas por el desafío.

2.5 Uso de redes neuronales

Las redes neuronales artificiales ANS (*Artificial Neural Systems*) o DNN (*Deep Neural Networks*) son sistemas, hardware o software, de procesamiento, que copian esquemáticamente la estructura neuronal del cerebro para tratar de reproducir sus capacidades.

Un sistema de clasificación que tiene éxito es el basado en redes neuronales artificiales. Estas se utilizan para modelar un problema imitando el funcionamiento de las neuronas de los organismos vivos: un conjunto de elementos interconectados, sin una tarea fija para cada uno, pero que durante el entrenamiento van creando y reforzando ciertas conexiones para poder aprender.

3. Objetivos y alcance del trabajo

Este trabajo de fin de grado tiene como objetivo principal la integración de bases de datos para la detección de ataques *spoofing.* Se basa en diferentes puntos, que a su vez definen el alcance del mismo:

- **Validación del correcto funcionamiento con bases de datos independientes:** Llevar acabo un experimento de detección de voz sintética utilizando el proyecto ya realizado por el grupo de investigación. Para ello se utilizan las grabaciones que se encuentran en una única base de datos. De esta forma se obtienen unos resultados con los que valorar si el funcionamiento del proyecto de partida es correcto y está listo para llevar acabo el siguiente paso.

- **Integración de múltiples bases de datos:** Partiendo del experimento posterior, realizar las modificaciones necesarias con el objetivo de hacer uso de múltiples bases de datos en un único experimento. Durante el proceso hacer un uso eficiente de las bases de datos, lo que implica no guardar información redundante y acceder a las bases de datos únicamente cuando es estrictamente necesario. Durante las primeras pruebas se utilizan valores Gaussianos pequeños con el objetivo de agilizar el proceso, ya que implican un menor tiempo de procesamiento. Una vez comprobado el funcionamiento adecuado, se utilizan valores mayores. El objetivo es poder crear un mayor número de modelos.

- **Implementación de los sistemas anteriores mediante redes neuronales:** Como último objetivo y de cara a futuros proyectos, se realizan pruebas con redes neuronales. La principal ventaja de las DNN es que son capaces de aprender de la experiencia a partir de las señales o datos de manera muy eficiente. Por lo tanto, este es el último de los objetivos, realizar los experimentos anteriores con una tecnología más moderna.

4. Beneficios que aporta el trabajo

En este apartado se analizan los beneficios desde tres puntos de vista diferentes: social, económico y técnico.

4.1 Beneficios sociales

Como se ha mencionado en apartados anteriores, este trabajo forma parte de un proyecto mayor del grupo Aholab, que es la detección de voz sintética mediante la fase armónica, y que además tiene como objetivo participar en el *challenge ASVspoof* 2021. Es por este motivo que los beneficios sociales de este último proyecto podrían considerarse también beneficios del trabajo que se desarrolla en este documento.

La finalidad es poder hacer frente a ataques *spoofing* por lo que el principal beneficio social se podría decir que es dar una mayor seguridad y privacidad a los usuarios de sistemas de seguridad biométricos basados en voz. Ya que a través del proyecto se pretende dar una mayor fiabilidad y acierto a este tipo de sistemas. Hoy en día, en la sociedad de la información, establecer la privacidad de los datos y quien puede acceder o no a un determinado servicio es crucial. Es más, en algunos ámbitos el control de acceso es algo crítico, como podría ser el caso de las operaciones bancarias. Por esta razón se podría decir que este trabajo tiene un gran impacto social.

4.2 Beneficios económicos

En cuanto a los beneficios económicos se refiere, diferentes apartados de este trabajo de fin de grado ayudan a dar una mayor rentabilidad. En las bases de datos, se pretende almacenar únicamente los datos estrictamente necesarios para realizar la detección y las grabaciones necesarias para crear los modelos que usaran los *Scripts*. Durante el proceso se realizan múltiples mediciones que determinan cual es la configuración óptima de sistemas y modelos.

Teniendo en cuenta que el software de implementación y los modelos desarrollados ocupan mucho menos espacio que las bases de datos completas, se genera un gasto menor. Debido a que almacenamiento y el mantenimiento de dicho almacenamiento tiene un coste elevado. De esta forma no solo se ahorrará espacio,

sino que, gracias a hacer un uso adecuado de los equipos, estos podrían llegar a ser utilizados durante un periodo mayor.

Además, teniendo en cuenta los beneficios sociales mencionados en el punto anterior, si se obtiene un sistema de seguridad más robusto podría llegar a evitar tener que implementar otros sistemas de seguridad y en dicho caso, supondría también un ahorra de recursos y por tanto económicos.

4.3 Beneficios técnicos

Los beneficios técnicos son de los cuales se beneficiará el proyecto principal, es decir, el hecho de poder usar múltiples bases de datos al mismo tiempo. Gracias a la integración de múltiples bases de datos es posible generar un mayor número de modelos. Esto significa que es posible identificar y defenderse frente a más tipos de ataques *spoofing* ya que cada uno de estos modelos es generado utilizando diferentes técnicas. Además, puesto que es posible identificar más ataques, el porcentaje de acierto y de fiabilidad del sistema es mayor también.

Si a esto le sumamos también la última de las etapas de este trabajo, el uso de redes neuronales DNN, se obtiene un sistema más moderno y aún más preciso. Resumiendo, se obtiene un sistema más robusto frente a ataques *spoofing*.

5. Estado del Arte

En los últimos años la investigación orientada a la seguridad se ha centrado en los rasgos biométricos, ya que estos no pueden ser robados u olvidados. La voz es un gran ejemplo, dado que puede obtenerse de forma muy natural para el usuario y además podría realizarse a distancia, sin encontrarse presencialmente en el lugar que solicita la verificación, por teléfono, por ejemplo. Sin embargo, sí que pueden ser copiados o falsificados. Una de las grandes preocupaciones en la verificación por voz son los denominados ataques *spoofing*. Para hacerlos frente, hoy en día se desarrollan sistemas SSD que permiten detectar voces sintéticas que tratan de hacerse pasar por otros usuarios.

5.1 ASV (*Automatic Speaker Verification*)

Un Sistema ASV extrae las vocales características de un individuo para establecer la identidad, ya sea mediante restricciones de vocabulario fijo (dependiente del texto) o sin restricciones, de manera dinámica (independiente del texto). Podemos encontrar dos tipos de sistemas en función de si se declara la identidad del usuario o no, por un lado, el sistema de verificación de locutor y por otro lado el sistema de identificación de locutor. El funcionamiento es similar y es el siguiente:

Ambos sistemas reciben una señal de entrada, que no es más que la voz del usuario que trata de verificarse. De esta señal se obtienen sus características o parámetros. Más adelante se explica que opciones hay en lo que a la extracción de parámetros se refiere. La siguiente etapa es la clasificación, se trata de determinar si el locutor de la señal de entrada pertenece o no a uno de los locutores con permiso. Como se puede ver en la Ilustración 2 y en la Ilustración 3 este proceso es diferente en un sistema de verificación de locutor y en uno de identificación de locutor.

- **Sistema de verificación de locutor:** El usuario que trata de verificarse, además de proporcionar la señal de entrada dará su identidad. De esta forma el clasificador compara la señal de entrada con dos modelos: por un lado, con el modelo del locutor con el que trata de identificarse y, por otro lado, un modelo universal que recoge muestras de muchos locutores diferentes al de la identidad proporcionada.

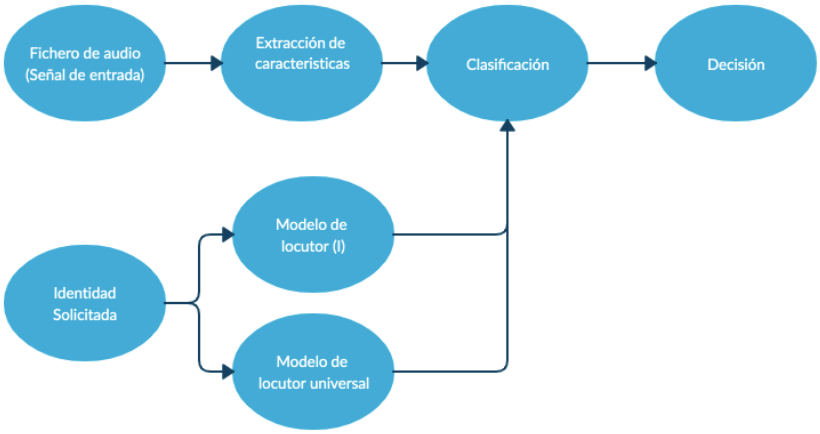

Ilustración 2: Esquema de un sistema de verificación de locutor

- **Sistema de identificación de locutor:** El usuario únicamente proporcionara la señal de entrada y será el propio clasificador el que compare dicha señal con cada uno de los modelos de los locutores de los que dispone.

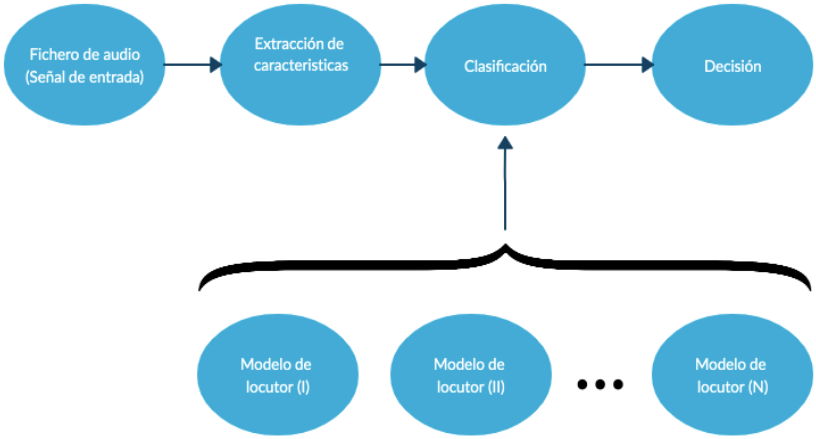

Ilustración 3: Esquema de un sistema de identificación de locutor

En base a los resultados del clasificador, se toma la decisión de aceptar o rechazar la señal.

5.2 Funcionamiento

El proceso principal de estos sistemas es la clasificación, que tiene una gran importancia, al fin y al cabo, en base a los resultados obtenido tras este proceso, se toma la decisión final. Se podría decir que esta tarea está formada por tres elementos: las bases de datos, los modelos y la tarea de clasificación en sí misma. Las bases de datos dan la posibilidad de almacenar muchas grabaciones, todas estas grabaciones serán utilizadas para crear los modelos mencionados en el apartado anterior y a su vez, estos modelos son utilizados para poder realizar la tare de clasificación.

El diseño de las bases de datos depende del tipo de ataque *spoofing* que se desea estudiar, por esta razón se deben utilizar múltiples bases de datos, protegiéndose así del mayor número posible de ataques. Las BDs más utilizadas para la verificación de locutor son las bases de datos estandarizadas NIST (*National Institute of Standards and Technology*), ya que estas están diseñadas con el objetivo de impulsar la investigación en reconocimiento de locutor. Sin embargo, durante este trabajo se hará uso de las bases de datos de los desafíos de años anteriores, más adelante en el apartado de las bases de datos se detalla más sobre las mismas. Estas se han utilizado en ocasiones para experimentos de *spoofing* como en [9]

Una vez teniendo las herramientas, en este caso la base de datos, es hora de crear los modelos. Los modelos imprescindibles en cualquier caso son los modelos de locutor. En caso de diseñar un sistema de verificación de locutor habrá que generar además un modelo universal. Las técnicas más utilizadas son modelos de Mezclas Gaussianas o GMM (para verificación de locutor independiente del texto) y HMM o *Hidden Markov Models* (para verificación independiente del texto).

Para realizar la tarea de clasificación se calcula el valor de verosimilitud, este valor indica la tasa de credibilidad que tiene la señal de entrada. Como se ha mencionado al comienzo de este apartado, la decisión final se toma en base a este valor de verosimilitud. Sin embargo, para ello primero se debe fijar un umbral y si el valor obtenido supera el umbral se acepta la señal y en caso contrario, se rechaza.

Por último, como se ha mencionado en apartados anteriores, existe una posibilidad alternativa en el proceso de la clasificación. Se trata de hacer uso de redes neuronales, en este caso las denominadas DNN (*Deep Neural Networks).* Una red neuronal profunda es una red artificial no profunda con varias capas ocultas

entre las capas de entrada y salida. El propósito principal es recibir un conjunto de entradas, realizar cálculos progresivamente complejos en ellas y dar salida para resolver los problemas de clasificación. [10]

5.3 Tipos de ataques

Para defenderse correctamente es imprescindible conocer al atacante, por ello en este apartado se analizan los tipos de ataques más comunes a los que se puede enfrentar un sistema ASV, y las técnicas más utilizadas para hacerles frente.

- Grabaciones: El atacante dispone de una grabación de alguno de los usuarios del sistema, este ataque se conoce como PA (*Playback Attack*). Soluciones: verificación dependiente del texto para que en cada acceso el texto varíe o localizar distorsiones generadas por micrófonos o altavoces utilizados en la grabación.

- Imitaciones: El emisor de la señal de entrada tratara de imitar la voz de u usuario del sistema. Soluciones: analizar el espectrograma en busca de cambios forzados en los formantes, principalmente, MFCC (*Mel Frequency Cepstral Coefficients*).

- Conversión de voces: Es muy similar al ataque de las imitaciones, pero en este caso requiere de conocimientos de tecnologías del habla. Son considerados una importante amenaza.

- Síntesis TTS (*Text-to-Speech*): Se basa en generar una voz a partir de un texto. Solución: módulo de voz sintética (SSD).

5.4 Medición de rendimiento

Puede ocurrir que el sistema sea capaz de detectar los ataques o no. Para medir el rendimiento en estas situaciones se utilizan se utilizan los siguientes valores:

- FAR (*False Acceptance Rate*): La falsa aceptación aumenta cuando se da por buena una identidad que no es correcta.

- FRR (*False Reject Rate*): El falso rechazo se da cuando se niega el acceso al locutor legítimo.

- DET (*Detection Error Tradeoff*): Permite visualizar fácilmente todos los puntos de operación de un sistema., ya que es la representación de FRR como función de FAR.

- EER (*Equal Error Rate*): Se utiliza como punto de referencia en DET, es el punto de equilibrio entre FAR y FRR.

La utilización de estos valores es clave, porque en base a los criterios establecidos se obtendrán unos resultados u otros. Un buen sistema no debe ser tan permisivo como para dar todas las señales de entrada por buenas. Por el contrario, tampoco puede ser muy estricto, ya que, en ese caso todas las señales serian rechazadas. Se debe buscar un punto de equilibrio en el que el sistema sea seguro y funcional al mismo tiempo.

5.5 SSD (*Synthetic Speech Detection*)

Como se ha mencionado en apartados anteriores, para la detección de voz sintética se desarrollan sistemas SSD. También se ha mencionado el uso de los parámetros RPS para el desarrollo de estos, sin embargo, la fase harmónica no es la única posibilidad. En la actualidad las principales opciones son tres.

Detección de voz sintética basada en:

- **Parámetros espectrales:** MFCC o *Mel Cepstum*, junto con sus derivadas y segundas derivadas son los parámetros espectrales principalmente utilizados para buscar diferencias entre las componentes de orden elevado de señales naturales y sintéticas. También se utilizan otros parámetros como los LFCC, por ejemplo.

- **Prosodia:** Una señal sintética frente a una natural tiene más posibilidades de formar patrones repetitivos, por este motivo los patrones prosódicos utilizados son la frecuencia fundamental o la duración de los sonidos.

- **Fase armónica:** Las técnicas más frecuentes son dos: la primera es utilizando parámetros RPS y la segunda es utilizando parámetros MGD (*Modified Group Delay*).

Existe la posibilidad también de crear sistemas de fusión basados en los resultados de distintos tipos.

A lo largo de este trabajo de fin de grado se utiliza la detección de voz sintética basada en la fase armónica, y se comparan los resultados obtenidos con la detección de voz sintética mediante parámetros espectrales.

5.6 Redes Neuronales

Antes de entender que es una red neuronal es fundamental comprender como es una neurona. Existen diferentes tipos de neuronas, sin embargo, para adentrarse en el campo de las redes neuronales por el momento es suficiente con conocer un modelo típico. Sus elementos principales son:

- **Soma:** Es el cuerpo central que contiene el núcleo, de él parten diferentes ramificaciones llamadas dendritas, siendo la principal de ellas el siguiente elemento a analizar.
- **Axón:** Es una fibra tubular, que se ramifica para conectarse con otras neuronas.
- **Sinapsis:** Son las ramificaciones del axón, es decir, zonas de conexión entre diferentes neuronas.

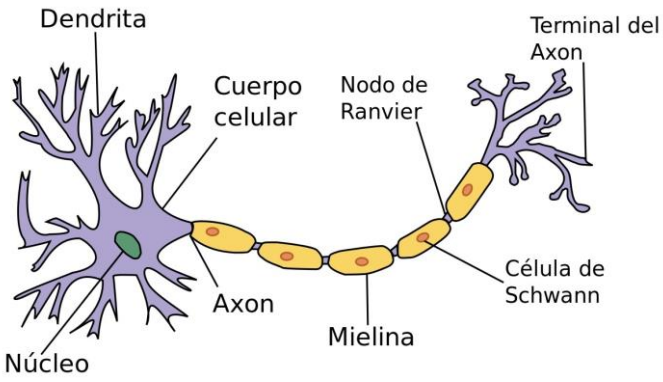

Ilustración 4: Estructura de una neurona biológica común

Las RNA imitan la estructura hardware del sistema nervioso de cara a construir un sistema de procesamiento de la información paralelo, distribuido y adaptativo, que puede presentar un comportamiento que se podría clasificar como "inteligente".

En la Tabla 1 [19] se puede observar que las neuronas comparadas con una CPU son simples, lentas y menos fiables. A pesar de ello se trata de imitar las neuronas, ya que existen problemas que para un ordenador convencional son muy

complejos y, sin embargo, el cerebro humano los resuelve de forma eficaz. Algunos de estos problemas o tareas son, por ejemplo; visión de objetos inmersos en el ambiente natural, respuesta ante estímulos del entorno o el reconocimiento del habla. Este último ejemplo es el motivo por el que el uso de redes neuronales es muy interesante para este trabajo de fin de grado.

	Cerebro	**Ordenador**
Velocidad de proceso	$10^{-2}\ seg\ (100Hz)$	$10^{-9}\ seg\ (1000\ Mhz)$
Estilo de procesamiento	Paralelo	Secuencial
Número de procesadores	$10^{11} - 10^{14}$	pocos
Conexiones	10.000 por procesador	pocas
Almacenamiento del conocimiento	Distribuido	Direcciones físicas
Tolerancia a fallos	Amplia	Nula
Tipo de control del proceso	Auto-organizado	Centralizado

Tabla 1: Cerebro frente a ordenador

5.6.1 Estructura general de la red neuronal

No cabe duda de que el elemento principal son las neuronas. La distribución de estas dentro de la red se hace mediante niveles o también llamados capas, cada una con un determinado número de neuronas. Se pueden diferenciar tres tipos de capas. Por un lado, están las de entrada, estas son las que reciben la información del exterior, por ejemplo, grabaciones de voz. Por otro lado, están las capas de salida, se encargan de enviar la información al exterior. Por último, nos encontramos las capas ocultas que no son más que las encargadas de procesar la información y comunicarse con otras capas.

Una vez determinado que las neuronas se agrupan en capas, falta saber cómo se conectan las neuronas de diferentes capas. Existen diferentes tipos de uniones: todos con todos, lineal y predeterminado. Sin embargo, en el caso de la red neuronal utilizada a lo largo del trabajo, se utiliza la primera de todas, todos con todos. Consiste básicamente en unir todas las neuronas de una capa con todas las neuronas de la siguiente capa. En la Ilustración 5 se puede observar todo lo explicado hasta

ahora.

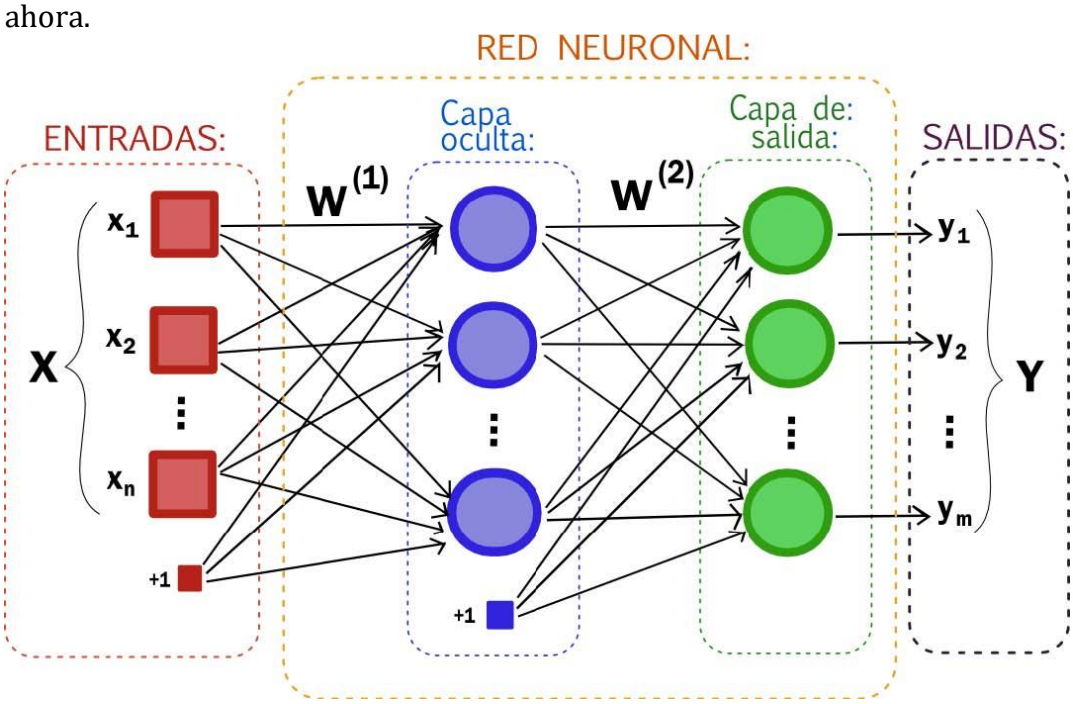

Ilustración 5: Representación de una red neuronal artificial

5.6.2 Estructura de una neurona artificial

La red neuronal es un conjunto de muchas neuronas y para ello es fundamental que cada una realice su tarea y que se comuniquen entre todas. Para ello, una neurona estándar consiste en diferentes partes que se pueden observar también en la Ilustración 6.

- Un conjunto de entradas que llamaremos $x_j(t)$.
- Unos pesos sinápticos w_{ij}. Representan la intensidad de interacción entre cada neurona presináptica j y postsináptica i.
- Una regla de propagación $h_i(t) = \sigma\left(w_{ij}, x_j(t)\right); h_i(t) = \sum w_{ij} x_j(t)$. Proporciona el valor del potencial postsináptico de la neurona i, en función de sus pesos y entradas.
- Una función de activación $y_i(t) = f_i(h_i(t))$. Representa simultáneamente la salida de la neurona y su estado de activación.

- Un umbral o *bias* θ_i, que se resta del potencial sináptico y le da un grado de libertad adicional a la neurona.

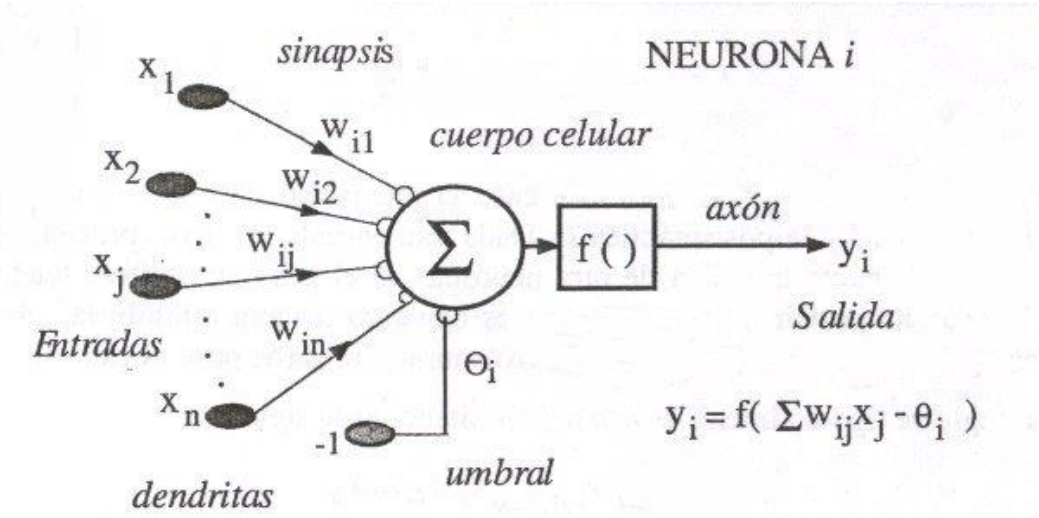

Ilustración 6: Representación de una neurona artificial

6. Análisis de alternativas

Para la elaboración y desarrollo de este trabajo de fin de grado se hace uso diversos recursos, en este apartado se presentan y analizan las diferentes alternativas y se argumenta la decisión tomada. Los puntos que se van a analizar son: lenguaje de programación, parametrización escogida para el tratamiento de las señales y finalmente, el método de trabajo.

6.1 Lenguaje de programación

Más adelante se explica cómo es el funcionamiento el sistema que se utiliza a lo largo del proyecto. Se explica de forma tanto teórica como matemática el funcionamiento de cada uno de los procesos, sin embargo, el encargando de hacer estas operaciones es un ordenador. Por esta razón, existe la necesidad de crear unos archivos denominados scripts, que no son más que unos documentos que contienen instrucciones, escritas en un código de programación. De cara a dar uniformidad al trabajo se desea establecer un mismo lenguaje de programación para todos los archivos, pero lenguajes de programación hay muchos y las posibilidades infinitas. Estas son algunas de las posibles elecciones:

- **Phyton**: Es un lenguaje de programación versátil multiplataforma, cuya característica principal es su código legible y limpio. Una de las razones de su éxito es que cuenta con una licencia de código abierto, permitiendo así su utilización en cualquier escenario.

- **R**: Es uno de los lenguajes más extendidos en el campo de las investigaciones científicas, muy común en proyectos de *machine learning*, minería de datos, bioinformática… Una función muy útil es la posibilidad de cargar diferentes bibliotecas o paquetes con funcionalidades de cálculo y graficación.

- **Matlab**: Es un programa computacional que ejecuta una gran variedad de operaciones y tareas matemáticas. Tiene una cualidad muy similar a las bibliotecas de R, aquí se llaman *toolbox* y ofrecen funcionalidades desarrolladas profesionalmente, probadas rigurosamente y totalmente documentadas.

En la próxima tabla, Tabla 2, se muestra una comparación entre algunas de las principales características de los lenguajes mencionados.

	MATLAB	Python	R
Redes neuronales	x	x	x
Random Forest	x	x	x
SVM	x	x	-
Librerías	Toolbox	Scikit-learn	Github

Tabla 2: Comparación lenguajes de programación

Algo a tener en cuenta a la hora de escoger el lenguaje de programación es que este no sea el que limite el proyecto, es decir que ofrezca las herramientas necesarias para poder llevar a cabo tareas tanto en el presente, como en el futuro. Las características mostradas en la Tabla 2, son algunas de las más utilizadas en tareas que implican tomas de decisiones. El grupo de investigación Aholab ya está trabajando actualmente con redes neuronales, por ejemplo. Se descarta R por la falta de SVM (un algoritmo de aprendizaje empleado para la clasificación binaria o regresión), ya que puede llegar a ser muy útil en este proyecto. Entre Phython y Matlab se toma la decisión de utilizar Matlab principalmente por que los *toolbox* son muy potentes y el grupo de investigación ya dispone de algunos que podrían ser implementados. Además, en el laboratorio de la universidad se han llevado a cabo proyectos del mismo campo con la herramienta Matlab, por lo tanto, el aprendizaje puede llegar a ser más sencillo.

6.2 Parametrización

En el estado del arte, apartado 5, se ha explicado que el sistema SSD puede estar basado en tres tipos: parámetros espectrales, prosodia y fase armónica. De cara a decidir qué método utilizar se van a comparar resultados obtenidos con cada una de las diferentes técnicas a lo largo de los años en diferentes proyectos. De esta forma, se busca determinar que parametrización puede ser válida y cual no.

- **Parámetros Espectrales:**

    1. Un sistema SSD junto con un verificador de locutor, trabajando en paralelo y basado en la base de datos japonesa ATR [11], resultó en una tasa de falsa aceptación del 0,69%.

    2. Con parámetros Mel Cepstrum y con sus derivadas, buscando las componentes de orden más elevado de señales naturales y sintéticas en [12] consigue tasas de EER menores al 2%.

    3. Una vez más mediante parámetros Mel Cepstrum en [13], esta vez creando i-vectors. Se fusionó verificador de locutor con SSD, obteniendo tasas de EER menores al 2%.

- **Prosodia:**

    4. En [14] se detectan señales sintéticas paramétricas basadas en HMM. Consiguiendo así valores de EER menores al 10%.

    5. En [15], trabajando con la detección de patrones de frecuencia fundamental, la búsqueda de patrones pitch consigue resultados EER menores al 10%.

- **Fase Armónica:**

    6. En [16], con la base de datos WSJ y con ataques basados en señales transcodificadas mediante STRAIGHT, se obtiene un EER por debajo del 3%.

    7. En [17] con MGDF-phase *(Modified Group Delay Function*), se llega a un rendimiento cercano al 4% de EER.

Analizando los resultados de proyectos pasados se puede llegar a comprender porque es la parametrización mediante parámetros espectrales la más utilizada. Sin embargo, la parametrización mediante RPS ha dado buenos resultados y se ha probado con éxito. Teniendo en cuenta esto y que este trabajo forma parte de otro mayor del grupo de investigación Aholab en el que se utiliza la fase armónica, a lo largo del trabajo se hace uso de esta parametrización, aprovechando también la

parametrización de parámetros espectrales y sus buenos resultados para hacer una comparación.

6.3 Método de trabajo

Especialmente debido a la situación en la que se encuentra el mundo en la fecha de realización de este trabajo, es decir, el confinamiento por el virus Covid-19, es necesario establecer formas de trabajo alternativas. El método escogido afecta directamente al trabajo en diferentes aspectos: tiempo, rendimiento, accesibilidad… Afortunadamente, la situación no ha impedido realizar el proyecto y se han podido analizar diferentes posibilidades:

- **Almacenamiento en línea y procesado local:** En esta situación se contempla el hecho de utilizar los servidores de la universidad para almacenar todos los archivos del proyecto. El procesado de los mismos sin embargo haría de forma local, utilizando el ordenador propio, descargando previamente los ficheros necesarios.

- **Almacenamiento local y procesado remoto:** A pesar de ser una posibilidad es implantarle, ya que la cantidad de almacenamiento necesaria es muy alta y no es común disponer de dicho almacenamiento en ámbitos no laborales. El procesado remoto se haría utilizando las CPUs y GPUs del laboratorio de la universidad.

- **Almacenamiento y procesado local:** Al igual que en la opción anterior el almacenamiento local no es viable. El procesado local sin embargo sería posible si se dispone de una gráfica relativamente potente. La ventaja principal de esto sería no depender de un dispositivo remoto que al fin y al cabo puede llegar a dar problemas, ya sea por red o por otra razón.

- **Almacenamiento y procesado remoto:** No cabe duda que en situaciones como la descrita es una de las mejores opciones, ya que te permite utilizar dispositivos de forma remota. No es necesario disponer del hardware necesario. Especialmente útil en este caso siendo hardware de nivel profesional.

Analizando las diferentes posibilidades la decisión final ha sido utilizar el último de los métodos, es decir, almacenamiento y procesado remoto. La razón principal es la sencillez del método, ya que incluso desde un dispositivo tan común como un móvil es posible conectarse a los servidores de forma remota. Únicamente es necesario disponer de conexión a internet, y ni si quiera tiene que ser muy veloz ya que para la gran mayoría de las tareas se comparten comandos o archivos de texto.

Sin embargo, no se descarta la posibilidad de utilizar almacenamiento y procesado local en caso de que los servidores o GPUs se encuentren en mantenimiento o no estén disponibles, ya que es posible y se debe contemplar. De esta forma se podría seguir trabajando y progresando sin ningún tipo de dependencia e incluso sin conexión a internet.

7. Análisis de riesgos

Ser consciente de los posibles riesgos que puede tener un proyecto, así como saber su origen y posible solución es de vital importancia. En caso de darse la desafortunada situación en la que ocurre uno de estos riesgos, se podrían causar retrasos en el tiempo. Sin embargo, si se establece de antemano la respuesta que se le va a dar a cada uno, creando así un plan de contingencia, los efectos de los riesgos podrían verse reducidos. Por ello a lo largo de este apartado primero se detectan los riesgos potenciales. Segundo, se analizan las posibilidades de que ocurran y su posible impacto en el proyecto. Tercero y último, se prepara un plan de actuación.

7.1 Posibles riesgos

Estos son los posibles riesgos que pueden darse:

- **Errores en los servidores:** No cabe duda de que estas herramientas son fundamentales para llevar acabo el trabajo y puede ocurrir que se encuentren fuera de servicio por algún error de hardware o software. En estos casos no sería posible hacer uso de ellas y por lo tanto habría retrasos. Además, habría que sumar el hecho de que en ocasiones es necesario ir a la universidad para solucionar estos problemas, y si la universidad se encuentra cerrada como ha sido el caso durante el confinamiento del virus Covid-19, el problema incrementa.

- **Base de datos corrupta:** Como en cualquier otro ordenador, puede ocurrir que los ficheros se corrompan, por ejemplo, por un corte de luz repentino mientras se realizaba alguna tarea. Este no es el único caso que puede provocar la corrupción de datos, de modo que es importante tenerlo en cuenta.

- **GPUs ocupadas:** En ocasiones, debido al alto número de participantes en el laboratorio puede ocurrir que las GPUs disponibles estén ocupadas y por lo tanto no se puedan realizar las tareas, causando así retrasos. Lo mismo ocurre si los servidores o GPUs se encuentran realizando labores de mantenimiento.

7.2 Impacto de riesgos

Cada uno de los posibles riesgos mencionados en el punto anterior tiene un impacto diferente. En los siguientes puntos se analiza cada uno detalladamente:

- **Errores en los servidores:** En este tipo de dispositivos se realizan chequeos periódicamente, de cara a prevenir este tipo de problemas. Y en cuanto al caso particular mencionado, una pandemia mundial, está claro que es un caso aislado y la probabilidad de que ocurra es mínima. Sin embargo, en caso de ocurrir el impacto sería alto y de gravedad, ya que estos equipos contienen toda la información de los proyectos realizados por el grupo de investigación. No obstante, se realizan *backups* diarios para evitar pérdidas de información.

- **Base de datos corrupta:** La probabilidad de ocurrir esto es pequeña y su impacto se podría considerar medio. El hecho de que una de las bases de datos esté corrupta significa dos cosas: o se debe obtener de nuevo o si esto no es posible, se debe encontrar un sustituto. En cualquiera de los dos casos será necesario un tiempo extra para solucionarlo.

- **GPUs ocupadas:** En el laboratorio se dispone de 4 de estos equipos, por lo que el problema puede surgir fácilmente. En caso de ocurrir en repetidas ocasiones puede ser un problema grave, pero por lo general, es un contratiempo mínimo.

Teniendo en cuenta el análisis, en la Tabla 3 se representa el impacto de los riesgos posibles:

		IMPACTO				
		5%	**10%**	**20%**	**40%**	**80%**
PROBABILIDAD	**Rara 10%**			Base de datos corrupta (0,02)		Errores en los servidores (0,08)
	Difícil 30%					
	Posible 50%					
	Puede ocurrir 70%	GPUs ocupadas (0,035)				
	Casi seguro 90%					

Tabla 3: Relación probabilidad impacto de los riesgos

7.3 Plan de contingencia

Finalmente, como se ha mencionado antes, una vez analizadas las posibilidades y el impacto de cada uno de los riesgos es hora de establecer un plan de contingencia. La idea es establecer unas pautas a seguir en el caso de que ocurriera alguno de los mencionados. Para llevar a cabo esta tarea se han fijado tres criterios: aceptar el riesgo, controlar el riesgo y reducir el riesgo. En el caso de este proyecto:

- **Aceptar el riesgo:** Dada esta situación, se decidiría como hacerle frente sin cambiar el plan inicial del proyecto. En este grupo entra el caso de las GPUs ocupadas, debido principalmente a su bajo impacto. Si esto ocurriera se buscaría otra tarea a realizar mientras se espera a que alguna de las GPUs del laboratorio sea liberada. Podría ser un buen momento para documentar los resultados obtenidos hasta el momento, por ejemplo.

- **Controlar el riesgo:** Al igual que antes se acepta el riesgo, sin embargo, se establece un método para controlarlo. En este caso entra el riesgo de errores en los servidores. Cuando uno de estos errores sea detectado se debe avisar de inmediato para que el responsable pueda solucionarlo lo antes posible. En cuanto a la continuación del trabajo, se deberá disponer previamente de los scripts con los que se trabaja de forma local, ya que estos archivos no son pesados y es posible disponer de ellos sin los servidores. De esta forma es posible trabajar en el código mientras los errores son solucionados.

- **Reducir el riesgo:** En ciertos casos es posible hacer algo para reducir el impacto de los riesgos. Este es el caso de las bases de datos corruptas. La idea aquí es activar un software de mantenimiento que se encargue de comprobar el estado de las bases de datos y la integridad de los datos de forma periódica. De esta forma el estado de los equipos y contenido es monitoreado continuamente.

8. Descripción del sistema.

8.1 Arquitectura

Para poder entender a la perfección los diferentes pasos realizados y explicados a continuación, previamente, es indispensable obtener unos conceptos básicos y generales del funcionamiento del sistema SSD planteado por el grupo de investigación Aholab. En este aparatado, se verá cómo se ha realizado a lo largo del trabajo la parametrización, la obtención de la verosimilitud y la decisión final.

El funcionamiento del sistema utilizado se centra en la búsqueda de señales de entrada sintéticas. Se busca también que el sistema sea independiente del locutor, sin necesidad de conocer la información de los locutores registrados en el sistema.

Para llevar a cabo esta tarea, la clasificación está basada en GMMs (*Gaussian Mixture Model*), donde los parámetros utilizados son RPS. Sin embargo, para autenticar el correcto funcionamiento del sistema y la posibilidad de utilizar la fase armónica para la detección de voces sintéticas, se comparan los resultados obtenidos con un sistema de clasificación basado también en GMMs, pero donde la parametrización utiliza parámetros espectrales, MFCC.

El sistema está compuesto por diferentes etapas y elementos que se pueden observar en la Ilustración 7 [18]. En los próximos puntos se analiza detalladamente cada una de las etapas. A modo de resumen, el sistema recibe una señal de entrada, con ella se realiza la parametrización para obtener los valores necesarios para el análisis. Se utilizan modelos sintéticos y naturales para determinar la credibilidad de la señal de entrada. A partir de los resultados obtenido se toma la decisión de aceptar la señal como legitima o no.

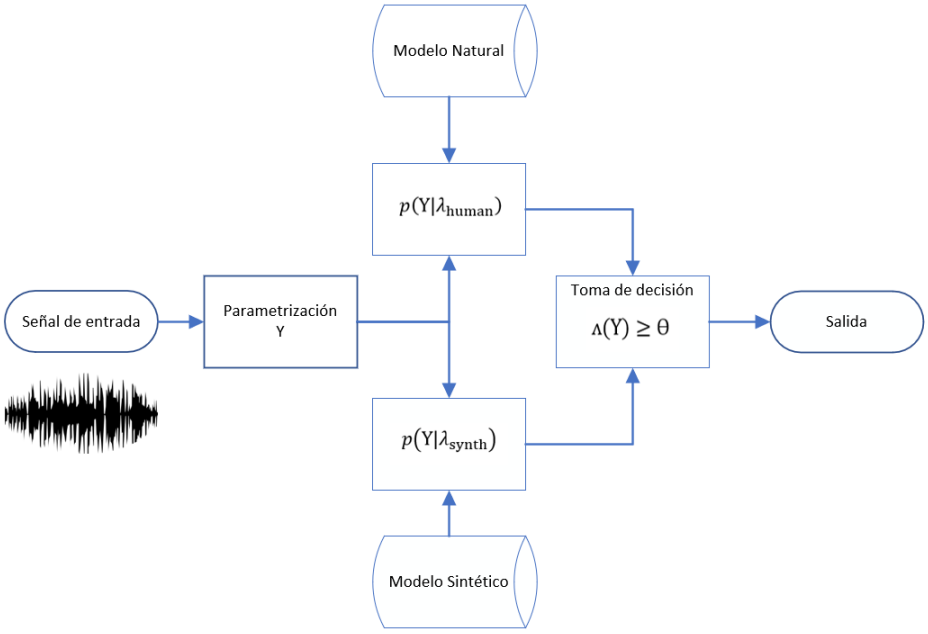

Ilustración 7: Esquema sistema SSD

8.1.1 Señal de entrada

Como es obvio, para comenzar el análisis el sistema recibirá una señal de audio, esta señal será la voz del locutor que desea ser verificado. El formato de archivo escogido en este caso ha sido WAV (*Waveform Audio File Format*), un formato de audio digital propiedad de la compañía Microsoft. El principal motivo de escoger este tipo de fichero frente a otros como podrían ser mp3, AAC o FLAC es que se trata de un formato no comprimido, por lo tanto, permite la máxima calidad de audio posible. Además, es un archivo relativamente simple, por lo que son más fáciles de procesar y editar. Por el contrario, sí que tienen una desventaja, su tamaño, son archivos pesados y largos debido principalmente a la calidad de audio que contienen.

No cabe duda de que las ventajas de este formato de archivo son muy útiles para un trabajo de estas características, por lo que a pesar de su desventaja se ha decidido trabajar con ellos. No obstante, el hecho de que ocupen más espacio es algo a tener en cuenta, en especial a la hora de gestionar la base de datos.

Cuando el archivo de audio entre en el sistema, deberá ser analizado para poder realizar la siguiente etapa, la parametrización. En la Ilustración 8 se muestra cual es el procedimiento que se sigue para la preparación tanto para parametrización MFCC como RPS.

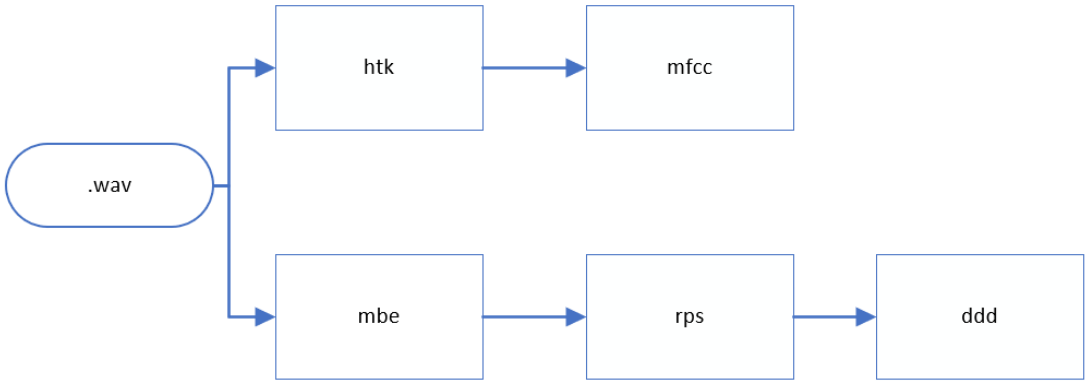

Ilustración 8: Esquema de extracción de parámetros

Previo a la parametrización, las señales son reducidas a 8kHz para limitar la carga de cálculo y su componente continua (DC) se filtra. Además, la polaridad de las señales se homogeneiza, ya que la parametrización RPS es altamente sensible a los cambios de polaridad. La energía de las señales también se normaliza. Este proceso es necesario porque la energía no identifica al hablante, es decir: distintos hablantes pueden hablar con la misma energía y, sobre todo, un único hablante hablar con energías diferentes en cada instancia.

8.1.2 Parametrización

Parametrización se entiende como describir o estudiar algo mediante parámetros, en este caso RPS y MFCC. A lo largo del trabajo se mantiene la información de la fase como parámetro principal. Sin embargo, de cara a validar la idoneidad de la utilización de la fase, los resultados obtenidos con parámetros RPS son comparados con los obtenidos mediante parametrización MFCC.

8.1.2.1 RPS

RPS es una representación de la información de la fase armónica. El análisis armónico modela cada trama de una señal como una suma de sinusoides armónicamente relacionadas con el tono o frecuencia fundamental. Su representación matemática es la siguiente:

$$h(t) = \sum_{k=1}^{N} A_k \cos\left(\varphi_k(t)\right) \quad (1)$$

$$\varphi_k(t) = 2\pi k f_o t + \theta_k \quad (2)$$

La representación RPS consiste en calcular el cambio de fase entre cada armónico y el componente fundamental (k=1) en un punto específico del período fundamental, es decir, el punto donde $\varphi_0 = 0$.

La ecuación (3) define la transformación RPS que permite calcular los parámetros RPS (ψ_k) de las fases instantáneas en cualquier punto (t_a) de la señal. Los valores RPS se enrollan (*wrapping*) en el intervalo [-π,π] y nos dan una idea clara de la estructura de la fase.

$$\psi_k(t_a) = \varphi_k(t_a) - k\varphi_1(t_a) \quad (3)$$

Sin embargo, la cantidad de parámetros RPS es variable en función del número de armónicos y debido a su alta dimensionalidad y discontinuidad, puede ocurrir el llamado desenrollado (*unwrapping*). Para hacer frente a esto y llegar a una parametrización útil para el modelado, es necesario un procesado adicional denominado, parametrización DCT-Mel-RPS. El proceso es el siguiente:

1) Se calculan las diferencias de los valores RPS desenrollados mediante 48 filtros basados en la escala Mel. Y se guarda el valor promedio. La escala Mel es una escala psicoacústica construida a partir de dos tonos que se percibían a una misma distancia de otros.

2) A la diferencia se le aplica una DCT (*Discrete Cosine Transform*). Se recorta o limita a 20 valores y se le añade el valor promedio del paso anterior.

De esta forma se obtienen los parámetros DCT-Mel-RPS, que contienen información relevante.

Finalmente, un elemento destacable y que no se debe olvidar es la implementación de un detector de polaridad. Ya que los RPS están directamente relacionados con la

forma de onda de las señales y en ocasiones ocurre que la polaridad esta invertida, en estos casos la polaridad se vuelve a invertir.

8.1.2.2 MFCC

Una vez realizada la parametrización RPS, se realizará la parametrización MFCC para poder realizar posteriormente la comparación de los resultados. Al igual que en el apartado anterior primero se debe comprender que son estos parámetros. Los MFCC son coeficientes para la representación del habla basados en la percepción auditiva humana. Muestran las características locales de la señal de voz asociadas al tracto vocal. Se obtienen a partir del espectro de la señal a la que se le ha aplicado previamente un filtrado perceptual basado en la escala Mel [18].

Los coeficientes cepstrales se derivan de la transformada discreta de Fourier o DFT (*Discrete Fourier Transform).* Su particularidad es que en MFCC las bandas de frecuencia están situadas logarítmicamente, según la escala Mel, en la que el punto de referencia se define equiparando un tono de 1000 Hz.

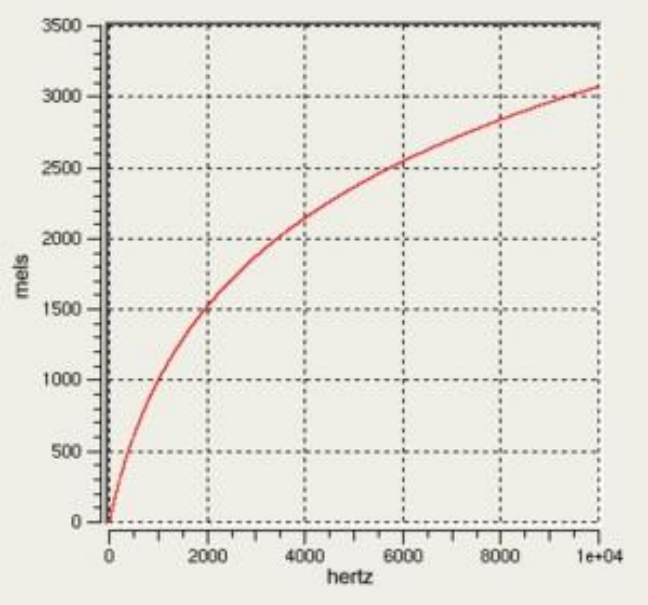

Ilustración 9: Gráfica Mel – Hz

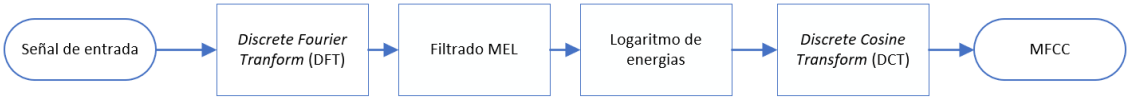

Ilustración 10: Esquema del proceso de extracción de parámetros MFCC

El proceso de obtención de estos parámetros es el siguiente:

1) Se aplica DFT a la trama de la señal entrante.
2) Se aplica un filtrado Mel.
3) Se calcula el logaritmo de la energía de cada una de las frecuencias Mel.
4) Se aplica DCT a las log-energías del paso anterior, obteniendo así los parámetros MFCC.

8.1.3 Toma de decisiones

Dada una entrada Y y un hablante hipotético S se trata de determinar si Y fue dicho por S. El sistema debe probar la secuencia de vectores de parámetros Y de longitud N, comparándola con los modelos natural y sintético. De esta forma se calcula la verosimilitud correspondiente a cada uno, es decir, p(Y | λ_{human}) y p(Y | λ_{synt}). Partiendo de estos valores se calcula el ratio de verosimilitud Λ mediante la siguiente ecuación:

$$\Lambda(Y) = log\ p(Y|\lambda_{human}) - \log p(Y|\lambda_{synt}) \quad (4)$$

Donde:

$$\log p(Y|\lambda) = \frac{1}{N}\sum_{n=1}^{N} \log p(y_n|\lambda) \quad (5)$$

Se tomará como human la señal de entrada si supera un umbral previamente fijado, θ. Este umbral tiene una gran importancia ya que se fija su valor en función del resultado deseado. Es decir, para un sistema con seguridad muy alta se utiliza un umbral elevado y para uno con menor seguridad un umbral pequeño. ¿Pero cuando se desea un sistema con menos seguridad? No es que se quiera diseñar un sistema menos seguro, si no que a un sistema con un umbral muy alto le costara más aceptar como legitimo un locutor, se dan menos falsas aceptaciones, pero, por el contrario, se aumentan también los falsos rechazos.

Durante este trabajo para fijar un umbral apropiado se han utilizado los ya mencionados valores FAR y FRR. Para calcular estos valores se ha utilizado las siguientes ecuaciones:

$$FAR\ (\theta) = \frac{Número\ de\ candidatos\ impostores\ \Lambda\ > \theta}{Número\ total\ de\ candidatos} \quad (6)$$

$$FRR\ (\theta) = \frac{Número\ de\ candidatos\ legítimos\ \Lambda \leq \theta}{Número\ total\ de\ candidatos} \quad (7)$$

Una vez obtenidos los valores, se representa la curva también mencionada anteriormente DET. Esta curva permite visualizar fácilmente todos los puntos de operación de un sistema., ya que es la representación de FRR como función de FAR. La curva se representa utilizando una escala basada en la desviación normal, de esta forma una distribución gaussiana se ve como una recta en esta escala. Finalmente, se utiliza el EER como punto de referencia en DET, ya que es el punto de equilibrio entre FAR y FRR.

Al ser el EER el punto de operación donde los errores de aceptación y rechazo se igualan, se ha utilizado dicho valor como umbral.

Una vez generados los ficheros que caracterizan las señales de entrada, se deben elaborar los modelos, los cuales son modelos de mezcla Gaussiana, GMM. En este trabajo se han creado dos modelos, uno para voces de habla natural o humana y otro para voces artificiales o sintéticas. A lo largo del proyecto se les ha llamado *human* y *sint*, respectivamente.

El algoritmo GMM es un modelo paramétrico utilizado clásicamente en muchas técnicas de reconocimiento de habla, trata de estimar la función de densidad de probabilidad de los parámetros pertenecientes a cada clase c mediante una suma ponderada de M distribuciones gaussianas. Su representación matemática es la siguiente:

$$P_c(x) = \sum_{i=1}^{M} w_k^c N(x, u_k^c, \Sigma_k^c) \quad (6)$$

Donde:

1) $N(x, u_k^c, \Sigma_k^c)$ representa la distribución nominal p-dimensional, siendo μ el vector medio y Σ la matriz covarianza definida como:

$$N(x, u_k^c, \Sigma_k^c) = \frac{1}{(2\pi)^{p/2}} |\Sigma|^{-1/2} \exp\left[-\frac{1}{2}(x-u)^T \Sigma^{-1}(x-u)\right] \quad (7)$$

2) Los términos w_k^c representan los pesos de la componente k:

$$\sum_{i=1}^{M} w_k^c = 1 \quad y \quad w_k^c > 0 \quad (8)$$

A partir de los modelos, mediante un clasificador bayesiano, se estima la probabilidad de que la señal de entrada sea de una de las clases. Explicado de otra forma, se calculan las probabilidades p(c|x) de que una muestra x pertenezca a cada una de las clases c, y se selecciona la más probable. Aplicando para ello la regla de Bayes, donde p(c) es la probabilidad a priori de la clase c, p(x|c) es la probabilidad de que ocurra x en la hipótesis de la clase c y p(c|x) son las probabilidades a posteriori.

$$\check{c} = \arg max_c\, P(c\,|x) = \arg max_c\, \frac{P(x\,|c)P(c)}{P(x)} = \arg max_c\, P(x|c)P(c) \quad (9)$$

Al ser un modelo de mezcla gaussiana un factor muy importante a tener en cuenta es el número de gaussianas utilizada. Ya que utilizar un número de componentes muy bajo resulta en un modelo que no puede aproximar la distribución con suficiente precisión, es decir, estará subentrenado y por lo tanto la probabilidad de error del sistema aumenta. Por el contrario, si se utiliza un número de componentes extremadamente alto, el sistema estará sobreentrenado, habiendo aprendido demasiado detalle y perdiendo la capacidad de generalizar ante muestras desconocidas. A lo largo del trabajo se debe decidir el número de gaussianas utilizar. Durante la etapa de validación del correcto funcionamiento se han utilizado 128 gaussianas ya que utilizar un número menor implica un tiempo de procesado inferior. Los resultados tienen un error mayor, sin embargo, el objetivo de esta etapa no es conseguir el mejor resultado posible. Utilizando un numero de gaussianas mayor, como, por ejemplo, 512 se le podría dar más precisión.

Finalmente se ha fijado el número máximo de iteraciones con las que se generan los modelos, 10. La razón principal de escoger este valor es que las diferencias significativas entre iteraciones sucesivas se producen en las 10 primeras iteraciones. Valores superiores a 10 mantienen unas diferencias muy bajas y solamente aumentaría el tiempo de cálculo de los modelos. De cara a ahorrar cálculo de tiempo también se ha fijado que, si entre dos iteraciones no se supera un umbral de 0,0001 de diferencia, el proceso se detiene a pesar de no haber llegado a las 10 iteraciones.

8.1.4 Modelado

Como se ha mencionado en apartados anteriores, se han utilizado las bases de datos de los ASVspoofing Challenge de años pasados. En algunos casos se han utilizado vocoders para generar las señales de voz artificiales necesarias para crear los modelos de voz sintética. Gracias a esto se simplifica el proceso, ya que se utilizan vocoders en lugar de crear un sistema de conversión de voz o una voz adaptada para cada uno de los locutores del sistema.

8.1.5 Sistema independiente del locutor

Hay dos tipos de reconocimiento de voz: independiente del locutor y dependiente del locutor. Por un lado, el software dependiente del locutor opera aprendiendo las características únicas e individuales de la voz de un hablante. Los nuevos usuarios primero deben “entrenar” el software mediante grabaciones. Esto significa que los usuarios tienen que leer algún tipo de texto antes de poder usar el software.

Por otro lado, un sistema independiente del locutor es aquel en el que el sistema no necesita ser entrenado específicamente para reconocer el acento y pronunciación de un individuo. Cuando se busca un sistema de detección de señal sintética lo más generalista posible, es necesario diseñar el sistema para que sea independiente del locutor sin que la eficiencia se vea afectada en comparación con su equivalente dependiente del locutor.

Para conseguir un sistema de este último tipo se diseñan dos modelos, humano y sintético. Estos dos modelos son entrenados utilizando grabaciones de múltiples locutores. El objetivo es que al crear los modelos GMM sean capaces de modelar las características comunes a todos ellos (locutores cuyas grabaciones no han participado deberían tener también estas características). Por esta razón el

número de locutores utilizados para construir los modelos humanos y sintéticos tiene una gran importancia.

8.2 Base de datos

Para poder utilizar los modelos mencionados en el apartado anterior, primero deben ser generados, tanto los modelos específicos de locutor como los universales. Es necesario por tanto pasar por una fase de entrenamiento, para la cual es indispensable disponer de grabaciones de los locutores. En este apartado se verá la estructura de la base de datos del grupo de investigación, las diferentes bases de datos utilizadas y finalmente, como se ha hecho uso de ellas.

En cuanto a la evaluación del rendimiento del sistema, un factor importante a tener en cuenta es el tipo de ataques a los que se tendrá que enfrentar. Ya que en este trabajo se está analizando el uso de las bases de datos frente ataques *spoofing*, las BDs utilizadas tendrán que poder ser utilizadas para simular el mayor número posible de ataques. Además, como se ha mencionado en el contexto, los resultados obtenidos a lo largo de este trabajo serán utilizados para el *challenge* ASVspoof 2021, por esta razón las bases de datos utilizadas tienen que cumplir con los requisitos del desafío. Por lo tanto, se utilizarán las BDs proporcionadas por los desafíos a lo largo de los últimos años.

8.2.1 Base de datos ASVspoof 2015

El objetivo de esta base de datos es estimular el desarrollo de nuevas contramedidas de suplantación generalizadas que sean capaces de detectar ataques de suplantación variables, implementados con algoritmos múltiples y diferentes. La tarea a realizar con la base de datos es distinguir el habla genuina del habla falso [4].

Está formada por discursos genuinos y falsos. Los discursos genuinos han sido recolectados de 106 hablantes (45 hombres y 61 mujeres), sin efecto significativo de canal o ruido de fondo. La voz falsificada se genera a partir de los datos genuinos utilizando varios algoritmos de falsificación. La base de datos está dividida en tres subconjuntos, la primera para el entrenamiento del sistema, la segunda para el desarrollo y la tercera para la evaluación. No hay superposición de hablantes en los tres subconjuntos con respecto a los hablantes utilizados en la conversión de voz o la adaptación TTS. En la Tabla 4 se recogen los elementos de la base de datos.

	Hablantes		Enunciados	
Subconjunto	**Hombre**	**Mujer**	**Genuino**	**Falso**
Entrenamiento	10	15	3750	12625
Desarrollo	15	20	3497	49875
Evaluación	20	26	9404	200000

Tabla 4: Estructura de la BDs ASVspoof 2015

En lo que, a los enunciados de entrenamiento se refiere, cada enunciado falsificado se genera de acuerdo con uno de los tres algoritmos de conversión de voz y dos síntesis de voz. Los sistemas de conversión de voz incluyen aquellos basados en la selección de cuadros, el cambio de pendiente espectral y un juego de herramientas de conversión de voz disponible públicamente dentro del sistema Festvox. Ambos sistemas de síntesis de voz se implementan con el sistema de sintetizador basados en modelos ocultos de Markov. Los enunciados de desarrollo son generados a partir de uno de los 5 algoritmos usados para generar los enunciados de entrenamiento. Finalmente, los enunciados de evaluación se generan utilizando diferentes algoritmos, tanto los 5 mencionados anteriormente como otros denominados *unknown* o desconocidos.

8.2.2 Base de datos ASVspoof 2017

El objetivo principal de esta base de datos es evaluar la precisión de detección de ataques de suplantación de identidad, en particular para detectar la repetición. Hace un uso considerable del corpus dependiente de texto RedDots, asi como una versión reproducida del mismo. El primero sirve como fuente de grabaciones genuinas y el segundo como fuente de grabaciones falsas de repetición. Este último se recopiló reproduciendo un subconjunto de los enunciados originales del corpus RedDots a través de varias configuraciones de repetición, que consisten en dispositivos variados, altavoces y dispositivos de grabación, en entornos de cuatro países europeos diferentes. [5]

Al igual que en la anterior BDs, está dividida en tres subconjuntos, la primera para el entrenamiento del sistema, la segunda para el desarrollo y la tercera para la evaluación. En la Tabla 5 se detalla el contenido de la BDs.

Subconjunto	Hablantes	Enunciados	
		Genuino	FALSO
Entrenamiento	10	1508	1508
Desarrollo	8	760	950

Tabla 5: Estructura de la BDs ASVspoof 2017

Los datos (enunciados) de evaluación incluyen una mezcla similar de enunciados falsos y genuinos de repetición. Algunas de las condiciones de repetición son exactamente las mismas que en las partes de entrenamiento y desarrollo. La mayoría de los ataques de repetición se originan a partir de configuraciones diferentes de las partes de entrenamiento y desarrollo de forma intencionada. De esta forma se podrá evaluar en condiciones no testeadas durante las etapas anteriores.

8.2.3 Base de datos ASVspoof 2019

Esta base de datos está diseñada centrándose en las contramedidas para los tres tipos principales de ataques; los derivados de TTS (*Text-to-speech*), VC (*Voice Conversion*) y los ataques de repetición. Su objetivo es determinar si los avances en la tecnología TTS y VC representan una amenaza mayor para la verificación automática de locutor y la confianza en las contramedidas de falsificación. [6]

Una vez más la BDs está dividida en tres subconjuntos, la primera para el entrenamiento del sistema, la segunda para el desarrollo y la tercera para la evaluación. En la Tabla 6 se detalla el contenido de la BDs.

	Hablantes		Enunciados			
Subconjunto	Hombre	Mujer	Acceso Lógico		Acceso Físico	
			Bona Fide	Spoof	Bona fide	Spoof
Entrenamiento	8	12	2580	22800	5400	48600
Desarrollo	8	12	2548	22296	5400	24300

Tabla 6: Estructura de la BDs ASVspoof 2019

Durante las grabaciones de esta base de datos se han contemplado dos escenarios denominados acceso lógico y acceso físico. La calidad del habla sintética bien entrenada y la voz convertida producida con las tecnologías actuales ahora indistinguible del habla de buena fe o "bona fide". Ya que las tecnologías de la

actualidad pueden ser utilizadas para crear escenarios convincentes, se denomina acceso lógico. En el escenario de acceso físico se considera los ataques de suplantación de identidad que se realizan a nivel del sensor. Esto implica que las señales de buena fe y falsas se propagan a través de un espacio físico antes de la adquisición. Se asume que los ataques son de tipo repetición, donde se captura una grabación de una señal de buena fe.

8.3 Redes neuronales

El último de los objetivos de este trabajo de fin de grado es explorar la adaptación a tecnologías de clasificación más recientes, siendo una de las más populares las redes neuronales DNN.

Las redes neuronales profundas DNN (*Depp Neural Network)* tratan de aumentar sus capacidades replicando la estructura neuronal del cerebro, aprendiendo así de la experiencia. Estas son sistemas, hardware o software, de procesamiento. Por lo tanto, si tratan de replicar un cerebro, es de gran interés analizar una neurona real.

8.3.1 Capas de la red neuronal utilizada

Para este trabajo se han utilizado 5 capas o *layers*, cada una tiene un objetivo y unas cualidades diferentes, son las siguientes: *Sequence Input Layer, LSTM Layer, Softmax Layer, Fully Connected Layer* y *Classification Layer* [20].

Ilustración 11: Arquitectura red LSTM

8.3.1.1 Sequence Input Layer

Primero la red debe recibir la información que va a procesar, a estos datos se les llama a partir de ahora secuencias. Una secuencia es una serie de datos, imágenes, palabras, notas musicales, sonidos… que siguen un orden especifico y

tienen únicamente significado cuando se analizan en conjunto y no de manera individual. Por supuesto en este trabajo las secuencias son grabaciones de voz.

Este tipo de redes neuronales son capaces de procesar tanto a la entrada como a la salida secuencias, sin importar su tamaño y además teniendo en cuenta la correlación existente entre los diferentes elementos de dichas secuencias. A la entrada de la DNN le llamamos x_t y a la salida y_t, siendo t el instante de tiempo. El elemento que le permite al sistema tener esa memoria que se ha mencionado antes es el hecho de que, en cada instante de tiempo, a partir de la entrada se genera una activación, a_t. Esta activación es la información que compartes las neuronas entre sí, por ejemplo, la activación del instante t se genera a partir de la activación del instante t-1. Sin embargo, esto solo permitiría tener memoria a corto plazo.

La función de esta capa es recibir los valores de entrada y transmitir la primera activación a la capa LSTM.

8.3.1.2 LSTM Layer

En el caso de nuestra red entran en juego la llamada celda de estado, un flujo de información que atraviesa la red de principio a fin. Esta determina la información a almacenarse en una celda. Para controlar lo que ocurre en la celda de estado se hace uso de unas estructuras llamadas puertas. Estas puertas están compuestas por una función de activación, unos productos de entrada y el contenido de la celda. La función de activación en este proyecto ha sido una sigmoide, esta produce unos valores de entre 0 y 1. En función del dicho valor (representado en porcentaje) se determina si el valor de entrada pasa la puerta o no. Para controlar lo que ocurre con la memoria disponemos de 3 tipos de puertas:

- **Forget:** Permite olvidar o eliminar elementos de la memoria. Su ecuación matemática es la siguiente:

$$f_t = \sigma\left(W_f * [a_{t-1}, x_t] + b_f\right) \quad (12)$$

- **Update:** Permite añadir nuevos elementos a la memoria. La ecuación 13 determina que valores van a ser actualizados. Y la ecuación 14 determina nuevos candidatos a ser añadidos al estado.

$$i_t = \sigma\left(W_i * [a_{t-1}, x_t] + b_i\right) \quad (13)$$

$$\check{C}_t = tanh\,(W_c * [a_{t-1}, x_t] + b_c) \quad (14)$$

- **Output:** Permite crear el estado oculto actualizado, definido por la ecuación 15 y finalmente, se obtiene la salida o activación del estado actual con la ecuación 16.

$$O_t = \sigma\,(W_o * [a_{t-1}, x_t] + b_o) \quad (15)$$

$$a_t = O_t * tanh(C_t) \quad (16)$$

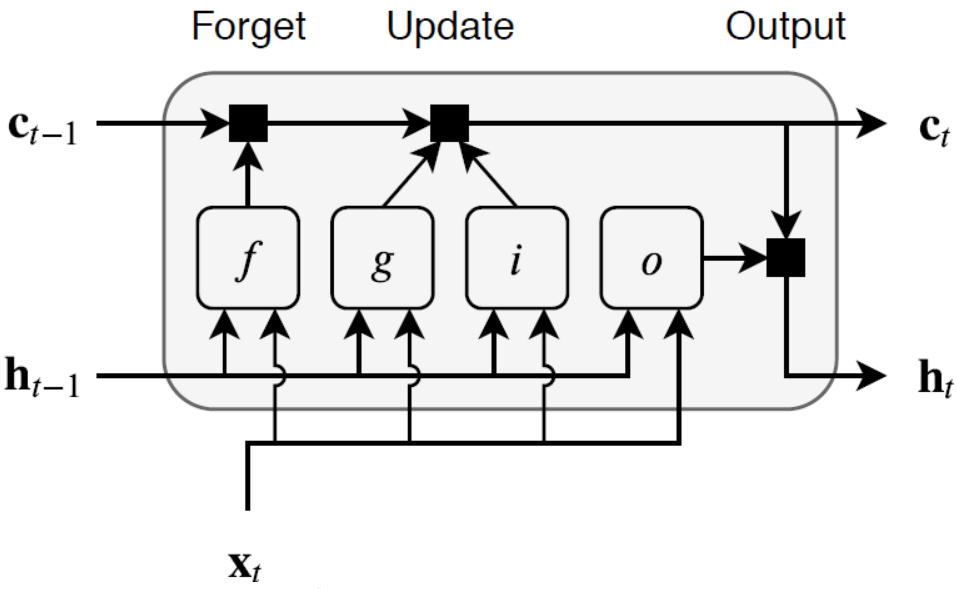

Ilustración 12: Puertas Forget, Update y Output

8.3.1.3 Softmax Layer

En el caso de la puerta de salida o puerta output, se utiliza la función tanh para obtener el siguiente estado de activación. Sin embargo, utilizar esta función tiene un inconveniente y es que requiere un alto coste de computación para su cálculo. Por esta razón es necesario implementar esta capa.

Esta capa se plantea como un clasificador de múltiples clases, en el que la entrada pertenece a una única clase. Esta capa devuelve una lista de probabilidades de pertenencia a cada una de las posibles clases, sumando un total del 100%.

8.3.1.4 Fully Connected Layer

El objetivo de esta capa es combinar toda la información local aprendida por las capas anteriores sobre las grabaciones de audio de la entrada, para identificar patrones mayores.

8.3.1.5 Classification Layer

Finalmente, la capa de clasificación calcula la perdida de entropía cruzada en los problemas de clasificación con varias clases y en las que las clases son mutuamente excluyentes.

9. Descripción de la solución propuesta. Diseño

A lo largo de este apartado se describen los diferentes pasos realizados para la elaboración del trabajo de fin de grado. Para ello se sigue el orden cronológico de las diferentes tareas llevadas a cabo, de esta manera es más sencillo de entender y de realizar el seguimiento de los resultados.

Ya que los objetivos principales de este trabajo de fin de grado son tres, este apartado se va a dividir en tres apartados. Primero se realizan las pruebas del sistema con las bases de datos de forma independiente.

Posteriormente, se realiza la misma prueba, pero en este caso con las bases de datos, pero de forma cruzada. Es decir, en un mismo análisis se utilizan múltiples bases de datos y se analiza el efecto que esto provoca.

Finalmente, y como última gran fase del proyecto, se utilizan redes neuronales para mejorar el funcionamiento del sistema.

9.1 Análisis con bases de datos independientes

Como primera fase del proyecto se encuentra la tarea de analizar cuál es la situación actual del proyecto. Al comienzo del trabajo las bases de datos eran usadas de forma independiente, por lo que en esta etapa se realizan tres análisis, uno por cada una de las bases de datos descritas en el apartado 7. Como se ha mencionado en apartados anteriores en realidad en cada uno de los análisis habrá que realizar dos, uno con parámetros RPS y otro con parámetros MFCC. Por lo que son un total de 6 análisis los que se realizan en este apartado.

Para poder hacer estos análisis se utiliza la herramienta Matlab, con ella se escriben y se ejecutan los *scripts* necesarios (el formato de archivos Matlab es .m). Estos ficheros contienen las características del sistema explicadas en el apartado 6, pero en forma de código computacional.

9.1.1 Análisis MFCC

En el caso del análisis con parámetros MFCC se utilizan un total de 7 *scripts.* En los siguientes aparatados se explica cual es nombre de cada uno de los archivos y sus funciones. El nombre del fichero es de gran importancia ya que nos indica a simple vista la función de dicho *script* y en los casos necesarios otra información relevante como en el número de gaussianas o el uso (sintético o humano). Por esta razón la estructura del nombre de los ficheros se ha respetado a lo largo de todo el trabajo.

9.1.1.1 Parametrización

La función del primer *script* es realizar la parametrización. En cada una de las fases habrá dos de estos archivos, uno será de tipo *train* y otro de tipo *test*, ya que se deben obtener los parámetros de los ficheros que se utilizan tanto para el entrenamiento como para el test. En este caso obtener los parámetros MFCC a partir de unos archivos htk. La razón de utilizar los parámetros htk en lugar de los .wav como fuente, es que las bases de datos de los *challenge* que se han utilizado, proporcionaban los datos de dicha forma.

En esta etapa los pasos realizados son los siguientes:

1. Leer cada uno de los ficheros de entrada y convertidos a formato raw.

2. Analizar los archivos raw en busca de las partes denominadas *voiced*. En las grabaciones de voz puede que haya partes que no nos interese analizar porque son silencios, a estos tramos se le ha llamado *unvoiced.* Por esta razón buscamos solo las partes con información relevante.

3. Obtener los parámetros MFCC solo de las partes *voiced* y crear un directorio para guardar estos de forma ordenada.

9.1.1.2 Entrenamiento

En el caso del entrenamiento se utilizan dos ficheros en función de los datos que se utilicen, la estructura del nombre es la siguiente: “train_MFCC_tipo_M.m”. Los dos tipos que nos encontramos son las humanas legítimas y las sintéticas o generadas como ataques, a las que denominaremos human y sint respectivamente.

El objetivo principal de este script es la creación de los modelos, para ello se utilizan los parámetros MFCC obtenidos en el apartado anterior. Se guardan los modelos con el formato de Matlab, ya que se van a utilizar más adelante.

9.1.1.3 Test

Una vez más se utilizan dos ficheros ya que se debe realizar el test con los dos modelos generados en el paso anterior, uno para modelar señales legítimas humans, y otro que modele los ataques sintéticos.

En este apartado se realizan un total de 4 test, ya que se utilizan las 4 combinaciones posibles de los modelos:

1) En primer lugar, las señales legítimas humanas se enfrentan al modelo creado con señales legítimas humanas
2) Se repite el proceso, pero las señales legitimas humanas en este caso se enfrentarán al modelo creado con señales sintéticas.
3) Una vez más se repite el proceso, en esta ocasión las señales sintéticas son las que se enfrentan al modelo creado con señales humanas.
4) Finalmente, las señales sintéticas se enfrentan al modelo creado con señales sintéticas también.

Los archivos de salida obtenidos se guardan para ser utilizados en el último paso.

9.1.1.4 Resultados

Finalmente, utilizando los 4 archivos de test generados en el apartado anterior se obtienen los resultados finales. La medida de resultado que se utiliza será el *Equal Error Rate*, EER. En este caso simplemente guardaremos el resultado de la ejecución como un fichero para que así podamos ver los resultados. Los valores que nos da son los siguientes:

1) **PTT:** Este valor hace referencia al punto de trabajo. La finalidad de este valor es encontrar el punto donde las dos distribuciones (human y sint) son simétricas.

2) **M:** Hace alusión a valor promedio del *score.* La necesidad de este valor surge de que se calcula en realidad un score de human y otro de sintético, por lo tanto, un valor promedio de los dos puede llegar a ser muy representativo de los resultados.

3) **Error human:** Como el nombre indica da el error de las muestras humanas únicamente.

4) **EER:** Ya se ha explicado previamente que se trata del *Equal Error Rate*, el cual es el punto de equilibrio entre FAR y FRR.

9.1.2 Análisis RPS

Al igual que en el análisis MFCC, en el caso del análisis con parámetros RPS también se utilizan un total de 7 scripts. En los siguientes aparatados se explica las funciones de estos archivos, sin embargo, gran parte es muy similar al análisis de parámetros MFCC.

9.1.2.1 Parametrización

Se generarán los parámetros RPS en base a los archivos .wav siguiendo los siguientes pasos:

1) Obtener los parámetros MBE (*MultiBand Excitation*) y comprobar la polaridad. En caso de ser necesario será invertida.

2) Obtener los parámetros RPSRPS estáticos a partir de los MBE, pero solo de los *voiced*, de manera análoga a lo que se ha explicado con los parámetros MFCC. Esta vez, sin embargo, no solo se tiene en cuenta la actividad vocal, sino que también se excluyen sonidos sordos como la s o la f.

3) A partir de los parámetros RPS-RPS estáticos obtener los valores RPS dinámicos o ddd, es decir la segunda y tercera derivada.

Cada uno de los valores, MBE, RPSRPS y ddd son guardados de forma ordenada para su posterior utilización.

9.1.2.2 Entrenamiento

De manera idéntica a cómo se generaban los modelos en el experimento basado en parámetros MFCC descrito en el apartado 0, se entrenan dos modelos, correspondientes a las características de las señales legítimas humanas, y a las sintéticas utilizadas como ataque de *spoofing*.

9.1.2.3 Test

Este aparatado es exactamente igual al test con parámetros MFCCMFCC, la única diferencia es que en este caso los parámetros de entrada que se utilizan son los ddd obtenidos para el test.

9.1.2.4 Resultados

Por último, este paso es exactamente igual con parámetros MFCCMFCC y con parámetros RPSRPS. Es decir, a partir de los test realizados obtener los valores finales. Los valores que se obtienen en esta última fase son los mismos que los obtenidos con parámetros MFCCMFCC, es decir: PTT, M, Error human y EER.

9.1.3 Conclusión

Una vez terminados los análisis llega la hora de ver los resultados y sacar una conclusión. Los datos obtenidos en esta primera etapa son fundamentales para la siguiente, ya que posteriormente se valora si utilizar múltiples bases de datos es beneficioso o no. Por lo tanto, se podría decir que estos resultados son la base del proyecto.

A continuación, se muestran los datos finales estructurados en tablas. La Tabla 7 representa los resultados en función del valor EER. En la columna de la izquierda BD representa las diferentes bases de datos: ASVspoof 2015, ASVspoof 2017 y ASVspoof 2019.

EER		
BD	**MFCC**	**RPS**
2015	10,50%	0,89%
2017	16,18%	34,34%
2019	12,91%	28,65%

Tabla 7: Error del sistema, calculado como EER, para las diferentes bases de datos

Los valores resaltados en la Tabla 7 son los resultados medidos en porcentaje. Como se puede apreciar por los resultados, la eficiencia es superior con parámetros MFCC que con parámetros RPS excepto en el caso de la base de datos de 2015, que es un caso particular. Por este motivo, merece la pena analizar uno a uno los resultados.

1) **ASVspoof 2015:** Los porcentajes de *Equal Error Rate* son muy bajos en ambos casos, es decir, se obtienen buenos resultados, siendo los obtenidos mediante la información de la fase los mejores. Un posible motivo de esto es que, al ser la base de datos más antigua de las tres, durante el desarrollo de las señales sintéticas se haya dejado de lado el tratamiento de la fase. El oído humano capta la respuesta frecuencial de los sonidos en magnitud, pero descarta la fase, por este motivo en muchas aplicaciones de las tecnologías del habla no se realiza un gran esfuerzo en modelar correctamente la fase. Gracias a esto la fase puede utilizarse como elemento distintivo que permite distinguir voz natural de voz sintética o procesada. Los resultados con parámetros MFCC son también buenos, a pesar de no obtener los mismos resultados que con parametrización RPS.

2) **ASVspoof 2017:** En este caso los resultados no son excelentes con ninguno de los dos tipos de parametrizaciones, sin embargo, en el caso de MFCC son resultados aceptables, a diferencia de los resultados con RPS. La tecnología avanza con los años y las tecnologías del habla comienzan a darle una mayor importancia a la fase, este puede ser la razón principal de los malos resultados con esta BDs.

3) **ASVspoof 2019:** Una situación muy similar a la anterior, resultados mejores en ambos casos, pero en lo que a la utilización de la fase se refiere los resultados siguen sin ser buenos.

9.2 Análisis con múltiples bases de datos

Una vez realizadas las pruebas con las bases de datos de forma independiente y sabiendo cual es el rendimiento del sistema, es hora de trabajar con varias bases de datos al mismo tiempo. La finalidad de esta etapa es ver si los resultados mejoran cuando se le proporciona más información al sistema. Es decir, al realizar modelos con información de más fuentes (las tres bases de datos), el sistema dispone de más datos con los que realizar la verificación.

A grandes rasgos el procedimiento a seguir es el mismo, pero en esta etapa se añaden más fuentes en los scripts, es decir, se fusionan datos. Sin embargo, un factor muy importante a tener en cuenta es que en esta etapa se trabaja con más información y por lo tanto esto significa, un mayor tiempo de procesamiento en los servidores del laboratorio y más información que manejar y almacenar. Por esta razón previamente se determina como se van a organizar los ficheros y los datos.

De cara a verificar correctamente los resultados, se hacen diferentes pruebas, cada una cruzando diferentes datos. Primero se realizan tres pruebas uniendo las bases de datos en parejas: 2015-2017, 2017-2019, 2019-2015. Una vez terminadas se hace la prueba final. Es esta en la que mayor información tiene el sistema, ya que la fusión de datos es 2015-2017-2019, las tres bases de datos en una única prueba.

A la hora de evaluar los resultados, un factor muy importante a tener en cuenta es que la información de cada una de estas BDs tiene una finalidad, ya que están pensadas para ataques de diferentes tipos, por lo que de por si unir las bases de datos debería hacer el sistema más seguro. La mejora de precisión sin embargo es algo que habrá que valorar una vez vistos los resultados. Por último, se valora también la relación tiempo/resultado, ya que, si el tiempo de procesado aumenta en exceso y la mejora es mínima, tal vez no sea de interés dotar de tanta información al sistema.

9.2.1 Organización

No cabe duda de que a la hora de realizar cualquier trabajo tener una buena organización es imprescindible, y más aún en este caso que se trabaja con grandes cantidades de información. Se han tomado diferentes medidas en lo que a la organización se refiere, gracias a ellas el proceso ha resultado más sencillo y eficiente.

Por un lado, en los servidores del grupo de investigación se ha dado acceso a la información de las bases de datos de las que se hace uso a lo largo del trabajo, a

los scripts que se han utilizado, a la herramienta Matlab que se encarga de compilar y ejecutar el código y finalmente, a otros datos utilizados u obtenidos por el grupo Aholab a lo largo de los años que pueden llegar a ser útiles. Por supuesto todos los datos dentro de este usuario están organizados para futuras ocasiones

Por otro lado, una de las medidas más importantes es hacer un uso adecuado del almacenamiento de los servidores. En múltiples ocasiones se puede utilizar la misma información que en casos pasados, por esta razón a la hora de organizar la base de datos gran parte se ha realizado mediante los llamados *symbolic link* o accesos directos. Gracias a estos, se hace referencia a datos almacenados en diferentes partes del servidor evitando duplicaciones. Este simple método es muy efectivo para reducir los costes de almacenamiento.

Por último, el procesado de las grabaciones y los archivos puede llegar a ser una tarea compleja para los equipos del laboratorio. En muchas ocasiones realizar un entrenamiento puede llegar a tomar días. De cara a evitar problemas a la hora de llevar a cabo estos procesos, las tareas más exigentes se dividen en partes, a pesar de que esta medida implique más tiempo.

9.2.2 Desarrollo

El desarrollo de esta etapa en sí mismo es igual que en el caso de bases de datos independientes. Sin embargo, lo que si cambia son los datos y la preparación de estos. Se deben cruzar los datos de las bases de datos y esto implica varias etapas:

Primero se deben reorganizar los datos para que los scripts puedan acceder a ellos. Todo ello se hará con los mencionados *symbolic links*. De esta forma se ahorra espacio. Ya que como los parámetros MFCC y RPS son los mismos tanto con bases de datos simples como múltiples, ha sido posible reutilizar los valores ya obtenidos en la fase anterior.

Segundo se debe indicar a los scripts que los datos que debe utilizar son de diferentes fuentes (bases de datos). Para ello se hace uso de ficheros índice, en los que hay un listado con las rutas de cada uno de los ficheros necesarios. Por supuesto se utilizan distintos ficheros índices para los modelos sintéticos o humanos, y para cada combinación de base de datos.

La tercera y última etapa se trata básicamente de esperar los resultados, pero se debe tener en cuenta que el tiempo de procesamiento de cada una de las etapas

será mucho mayor debido a la gran cantidad de datos a utilizar. Cuantos más datos se utilicen, mayor tiempo requiere.

9.2.3 Conclusión

Teniendo en cuanta los resultados mostrados en la Tabla 8 (en el caso del uso de las bases de datos de forma independiente) como datos de partida, se analizan en este punto los resultados de las diferentes combinaciones de BDs. Se tiene en cuanta también el hecho de que en estos casos son necesarios más recursos computacionales y más tiempo. Este apartado se va a dividir en dos puntos, por un lado, los resultados obtenidos con parejas de bases de datos y, por otro lado, los resultados con la combinación de las tres bases de datos.

En esta primera tabla, Tabla 8, se muestran los resultados del proyecto mediante parejas de BDs en función del valor EER. Indicando en la columna de la izquierda la combinación de bases de datos correspondiente y separando los resultados por el tipo de parametrización utilizada.

EER		
BD	**MFCC**	**RPS**
2015-2017	1,32%	2,51%
2017-2019	15,63%	29,68%
2019-2015	8,58%	11,81%

Tabla 8: Error del sistema, calculado como EER, para las diferentes combinaciones de bases de datos

A simple vista se pueden observar mejoras con respecto a la implementación de las bases de datos de forma independiente. A pesar de ello hay resultados variados. Primero, en el caso de la combinación 2015-2017 se obtienen muy buenos resultados con ambas parametrizaciones. Son valores de EER bajos y que dan como resultado un sistema funcional de cara a la detección de ataques mediante *spoofing*. Segundo, con la combinación 2017-2019, se obtienen unos valores no tan buenos. Es decir, en ambos casos son mejores que los obtenidos en la etapa anterior. Utilizando parámetros MFCC se logra un EER de 15,63%, no es un mal resultado y se podría decir que es relativamente preciso. Por contra, el valor EER obtenido mediante la información de la fase no es tan bueno, 29,68%, no es un resultado bueno. Un sistema así podría llegas a dar por buenas a señales sintéticas o por malas señales buenas en repetidas ocasiones. Tercero y último, la combinación 2019-2015 da lugar a un sistema bastante fiable con buenos resultados. Como se previa, los resultados son mejores en todos los casos con parametrización MFCC. De cara a

comparar esta etapa con la anterior de forma más cómoda, en la Tabla 9 se muestran todos los valores en conjunto.

EER					
BD	**MFCC**	**RPS**	**BD**	**MFCC**	**RPS**
2015-2017	1,32%	2,51%	**2015**	10,50%	0,89%
2017-2019	15,63%	29,68%	**2017**	16,18%	34,34%
2019-2015	8,58%	11,81%	**2019**	12,91%	28,65%

Tabla 9: Comparación de error del sistema, calculado como EER

Teniendo en cuenta que con la unión de las bases de datos en parejas se han mejorado los resultados en todos los casos, la lógica dice que si unimos las tres en conjunto obtendremos un único sistema preciso y fiable. El procedimiento que se ha seguido es muy similar a las anteriores dos etapas, por lo tanto, por no repetir lo mismo pasamos directamente a los resultados. Se pueden observar en la Tabla 10.

EER		
BD	**MFCC**	**RPS**
2015-2017-2019	8,74%	12,78%

Tabla 10: Error del sistema, calculado como EER, para el conjunto entero de bases de datos

Para empezar, no cabe duda de que los resultados son buenos. Independientemente de la parametrización este proceso ha dado lugar a un sistema más robusto, preciso y fiable que los anteriores. Ya que no solo es el hecho de obtener unos valores EER bajos, sino que, además, es capaz de detectar un mayor número de ataques (los ataques diseñados para cada una de las bases de datos). Por lo tanto, esta parte del proyecto se podría considerar un éxito, se ha obtenido un sistema que podría llegar a utilizarse a protegerse frente a ataques mediante *spoofing*.

9.3 Integración de redes neuronales

La última de las etapas de este trabajo es, como ya se ha mencionado en el apartado de los objetivos, realizar los experimentos anteriores con una tecnología más moderna. Hablamos de realizar la detección de señales sintéticas mediante ataques *spoofing* utilizando redes neuronales DNN.

A lo largo del trabajo se ha utilizado una red neuronal LSTM (*Long-Short Term Memory*), la ventaja de este tipo de redes es que son capaces de recordar un dato relevante en la secuencia y preservarlo durante varios instantes de tiempo. Por lo tanto, puede tener memoria a corto plazo (al igual que las redes recurrentes básicas), pero también memoria de largo plazo.

Utilizando la red neuronal estructurada en las 5 capas mencionadas anteriormente, se comienza el proyecto. Se realiza la misma tarea 3 veces, una por cada una de las bases de datos de las que se dispone. Aprovechando que la tarea de parametrización de dichas bases de datos ya se ha realizado en las etapas anteriores, se han utilizado esos parámetros MFCC como parámetros de entrada x_i.

Al igual que antes, se debe realizar la tarea de entrenamiento y posteriormente la de testeo, la diferencia es que en este caso se realiza todo al mismo tiempo. Previamente se han los siguientes parámetros de cara a conseguir los mejores resultados posibles:

- **Execution Enviroment:** Hace referencia al hardware que se utiliza para llevar a cabo todo el proceso, en este caso se ha optado por utilizar la GPU.

- **MaxEpoch:** Época o *epoch* en inglés, hace alusión al número de iteraciones con las que se realiza el entrenamiento utilizando los parámetros de entrada. Para este trabajo se ha fijado un valor de 100 épocas, ya que se considera suficiente. Además, se debe tener en cuenta que cuantas más épocas, mayor es el tiempo de procesamiento.

- **MBS (MiniBatch size):** Define el número de muestras que se propagan a través de la red DNN. Se ha optado por 250.

- **Shuffle:** Se ha utilizado para definir que la entrada de las señales, es decir los parámetros de entrada tanto de entrenamiento como de test se ordenen de forma aleatoria antes de comenzar.

Un dato importante a tener en cuenta es que a pesar de que los datos y los scripts utilizados sean siempre los mismos, en los test no dan siempre los mismos resultados. La razón de esto es que en este proyecto se ha trabajado con la característica *shuffle,* que hace que los datos se ordenen siempre de manera diferente, de forma aleatoria. Por ello es normal que haya pequeñas diferencias.

En la Ilustración 13 se puede observar un gráfico que representa la precisión del sistema en función de las iteraciones realizadas.

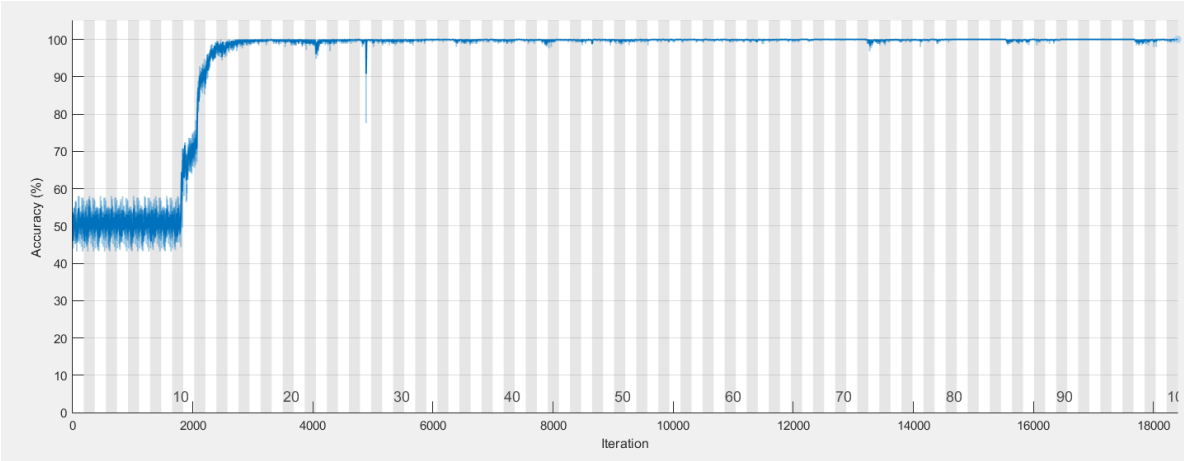

Ilustración 13: Gráfico de precisión con la base de datos ASVspoof 2019

De la misma forma también se obtiene un gráfico que nos muestra las perdidas en función de las iteraciones. Este tipo de gráficos son muy útiles, porque son muy representativos de los resultados de un entrenamiento. Por ejemplo, en el caso de la Ilustración 14, nos damos cuentas gracias a la gráfica que a partir de la iteración 10.000 no ha habido mejora.

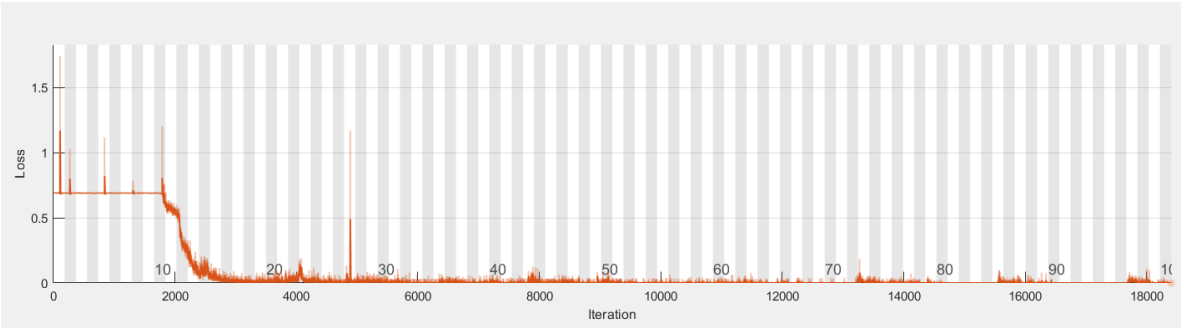

Ilustración 14: Gráfico de perdidas con la base de datos ASVspoof 2019

Una vez realizada la ejecución se obtendrá un valor acc, es decir el valor *accuracy* o de precisión del sistema. Sin embargo, como se pretende comparar los resultados obtenidos con los de las anteriores fases, se debe trasladar ese resultado a valores EER.

9.3.1 Conclusión

Una vez realizado el entrenamiento y test del sistema con las configuraciones mencionadas antes, es hora de analizar los resultados finales, se pueden observar en la Tabla 11.

BD	EER
2015	10,25%
2017	12,52%
2019	10,44%

Tabla 11: Error del sistema, calculado como EER, para las diferentes bases de datos

Los resultados han sido muy positivos, superando en todos los casos los resultados obtenidos en las fases anteriores independientemente del tipo de parametrización. A excepción del caso de la base de datos de 2015 con parámetros RPS, que obtenía unos resultados particularmente buenos debido a las características de la base de datos como ya se ha explicado. En la Tabla 12 se puede observar una comparación de los resultados finales obtenidos durante las diferentes pruebas a lo largo del trabajo.

EER			
BD	**GMM(MFCC)**	**GMM (RPS)**	**DNN**
2015	10,50%	0,89%	10,25%
2017	16,18%	34,34%	12,52%
2019	12,91%	28,65%	10,44%

Tabla 12: Error del sistema, calculado como EER, para las diferentes bases de datos

10. Planificación del proyecto

Este Trabajo de Fin de Grado se ha dividido en diferentes partes, en cada una de ellas se han llevado a cabo diferentes tareas. No cabe duda de que algunas ocupaciones requieren de un mayor tiempo que otras, por esta razón se indica también el tiempo consumido por cada una de las labores.

Se debe mencionar por supuesto a las personas que han participado en este proyecto. Ya que al fin y al cabo si la ayuda de ellos hubiera sido muy complicado llevar a cabo este trabajo de fin de grado.

- Jon Sánchez: Director del TFG, ingeniero en telecomunicaciones y profesor de la Escuela de Ingeniería de Bilbao. Su función principal ha sido la de elegir el trabajo y guiar y ayudar con los pasos realizados en el trabajo.

Participante	Abreviatura	Función
Jon Andoni Baranda	JA	Ingeniero Junior
Jon Sánchez	JS	Ingeniero Senior

Tabla 13: Participantes en el proyecto

En los próximos apartados se va a describir el trabajo realizado dividido en paquetes y explicando las tareas de cada uno de ellos. Se especifica la fecha de inicio y fin, duración total en semanas y en horas y los participantes en cada una de las tareas.

10.1 PT1: Definición del proyecto

Al igual que en cualquier otra investigación, una etapa fundamental es conocer el campo de estudio y la situación del mismo. El objetivo de esta parte por lo tanto es una toma de contacto tanto con la detección de señales sintéticas como con los *challenge* ASVspoof. Las pautas a seguir han sido las siguientes:

1) Comprender el funcionamiento de un sistema SSD, las diferentes partes del proceso.

2) Entender como se ha llevado a cabo la detección de señales sintéticas durante los últimos años en el caso del grupo Aholab, ya que existen diferentes métodos. Y familiarizarse con las herramientas del laboratorio, principalmente el servidor y las bases de datos.

3) Conocer los procedimientos de los desafíos, la finalidad de estos durante años anteriores, conocer las bases de datos proporcionadas y las normativas y formas de evaluación.

PT1: Definición del proyecto	
Fecha inicio	20 de febrero de 2020
Fecha final	5 de marzo de 2020
Duración (días)	2 semanas
Duración (horas)	-
Participantes	JA

Tabla 14: Estimación de duración de Tarea 1

10.2 PT2: Implementación de bases de datos independientes

Una vez adquiridos los conocimientos base, a pesar de que a lo largo del trabajo es donde se vaya a obtener la mayor parte del conocimiento, es hora de empezar el primer proyecto. En esta fase, se plantea la integración de las bases de datos de los desafíos ya mencionados, pero en proyectos independientes. Los objetivos de esta fase son: comprender como se realiza la integración de una base de datos en un sistema SSD, analizar los resultados del proyecto ya que serán utilizados como punto de partida y como comparación con proyectos futuros.

Esta parte se ha dividido en tres tareas, una tarea o una implementación por cada una de las bases de datos. En la Tabla 15 se hace referencia a estas tareas con los nombres de: A2015, A2017 y A2019. Los nombres hacen referencia a la base de datos de cada una de las tareas.

- **A2015:** Implementación ASVspoof 2015.
- **A2017:** Implementación ASVspoof 2017.
- **A2019:** Implementación ASVspoof 2019.

PT2: Implementación de bases de datos independientes	
Fecha inicio	5 de marzo de 2020
Fecha final	12 de junio de 2020
Duración (días)	12 semanas
Duración (horas)	80 horas
Participantes	JA, JS
Tareas a realizar	A2015, A2017, A2019

Tabla 15: Estimación de duración de Tarea 2

10.3 PT3: Implementación de múltiples bases de datos

En esta tercera parte se utilizan las mismas bases de datos y los mismos proyectos que en la parte 2, sin embargo, se harán combinaciones diferentes de las bases de datos. Por esta razón las tareas a realizar son: modificación de los scripts de la parte dos para que sean funcionales en esta parte, organización y administración de las bases de datos y finalmente, análisis de los resultados.

Una vez más se ha dividido la parte por tareas, una tarea o una implementación por cada una de las posibles combinaciones de bases de datos. En la Tabla 16 se hace referencia a estas tareas con los nombres de: B1517, B1719, B1915 y B151719. Los nombres hacen referencia a las combinaciones realizadas.

- **B1517:** Implementación conjunto ASVspoof 2015 y ASVspoof 2017.
- **B1719:** Implementación conjunto ASVspoof 2017 y ASVspoof 2019.
- **B1915:** Implementación conjunto ASVspoof 2019 y ASVspoof 2015.
- **B151719:** Implementación conjunto ASVspoof 2015, ASVspoof 2017 y ASVspoof 2019.

PT3: Implementación de múltiples bases de datos	
Fecha inicio	12 de junio de 2020
Fecha final	3 de julio de 2020
Duración (días)	3 semanas
Duración (horas)	60 horas
Participantes	JA, JS
Tareas a realizar	B1517, B1719, B1915, B151719

Tabla 16: Estimación de duración de Tarea 3

10.4 PT4: Implementación de sistema basado en redes neuronales

Por último, se comienza a realizar pruebas con redes neuronales. La idea es integrar las bases de datos junto con las redes neuronales para hacer así un sistema con mayor acierto y con una progresión mayor.

Este apartado es de gran importancia ya que es muy interesante de cara a proyectos futuros. La implementación de redes neuronales supone un gran avance en el aprendizaje de ataques en tecnologías de esta clase. Se ha realizado una única tarea, implementando la base de datos de 2019 con redes neuronales, esta tarea se le ha llamado: C2019.

- **C2015:** Implementación red neuronal ASVspoof 2015.
- **C2017:** Implementación red neuronal ASVspoof 2017.
- **C2019:** Implementación red neuronal ASVspoof 2019.

PT4: Implementación de redes neuronales	
Fecha inicio	3 de julio de 2020
Fecha final	17 de julio de 2020
Duración (días)	2 semanas
Duración (horas)	40 horas
Participantes	JA, JS
Tareas a realizar	C2015, C2017, C2019

Tabla 17: Estimación de duración de Tarea 4

10.5 PT5: Gestión del proyecto

Por último, en este paquete se realiza la documentación del proyecto. El objetivo es elaborar una memoria que detalle todo lo realizado en este trabajo de fin de grado. Las tareas realizadas en este paquete son dos principalmente: D1 (Escritura del documento) y D2 (Reuniones de seguimiento).

PT5: Gestión del proyecto	
Fecha inicio	20 de febrero de 2020
Fecha final	18 de julio de 2020
Duración (días)	20 semanas
Duración (horas)	50 horas
Participantes	JA, JS
Tareas a realizar	D1, D2

Tabla 18: Estimación de duración de Tarea 5

10.6 Diagrama de GANTT

El resumen de las tareas llevadas a cabo a lo largo el proyecto se resume en el siguiente diagrama de GANTT, donde se diferencian los diferentes paquetes de trabajo, indicando la fecha de inicio y finalización de los mismos, y las relaciones entras las diferentes tareas realizadas. Ilustración 15.

Id	Modo de tarea	Nombre de tarea	Comienzo	Fin
1		**PT1**	**jue 20/02/20**	**jue 05/03/20**
2		Definición del proyecto	jue 20/02/20	jue 05/03/20
3		**PT2**	**jue 05/03/20**	**vie 12/06/20**
4		Implementación ASVspoof 2015	jue 05/03/20	jue 04/06/20
5		Implementación ASVspoof 2017	jue 04/06/20	lun 08/06/20
6		Implementación ASVspoof 2019	lun 08/06/20	vie 12/06/20
7		**PT3**	**vie 12/06/20**	**vie 03/07/20**
8		Implementación conjunto 15-17	vie 12/06/20	mié 17/06/20
9		Implementación conjunto 17-19	jue 18/06/20	dom 21/06/20
10		Implementación conjunto 19-15	dom 21/06/20	jue 25/06/20
11		Implementación conjunto 15-17-19	jue 25/06/20	vie 03/07/20
12		**PT4**	**vie 03/07/20**	**vie 17/07/20**
13		Red neuronal 2015	vie 03/07/20	jue 09/07/20
14		Red neuronal 2017	jue 09/07/20	lun 13/07/20
15		Red neuronal 2019	lun 13/07/20	vie 17/07/20
16		**PT5**	**jue 20/02/20**	**sáb 18/07/20**
17		Documentación	vie 21/02/20	sáb 18/07/20
18		**Reunión de seguimiento**	**jue 20/02/20**	**jue 16/07/20**

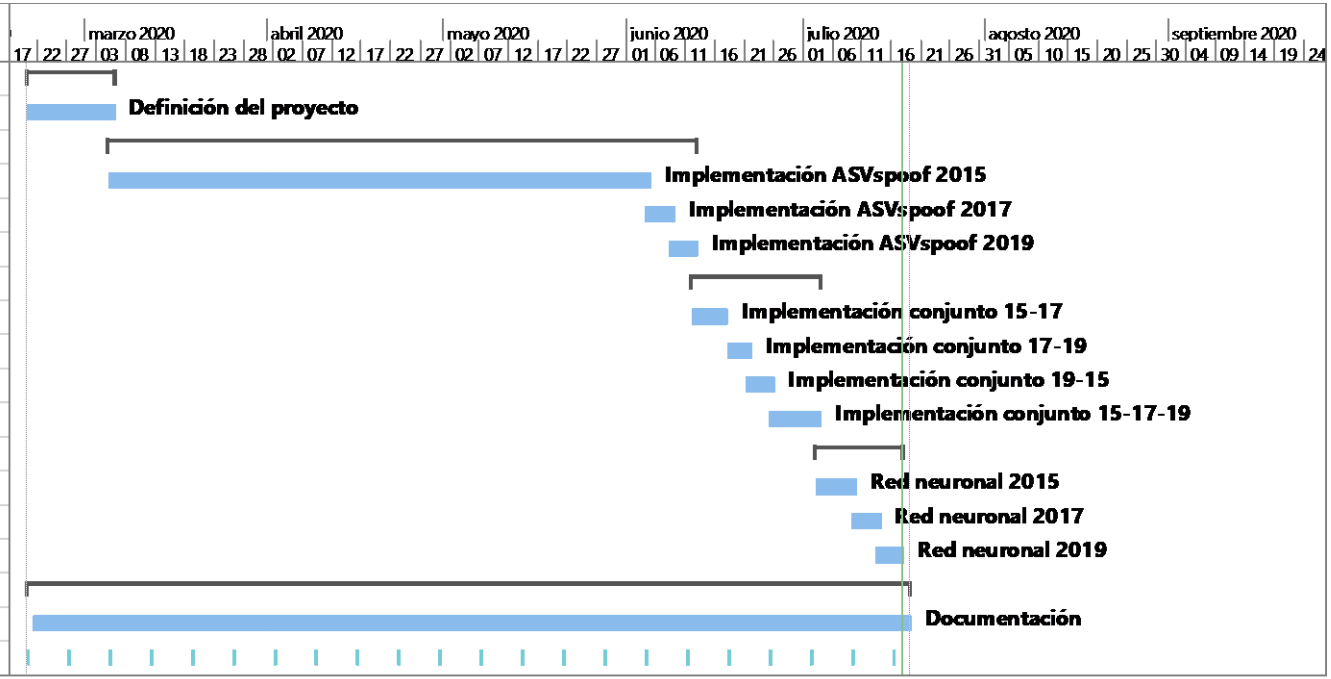

Ilustración 15: Diagrama de GANTT

11. Presupuesto

Una parte fundamental de todo proyecto es su aspecto económico y su respectivo presupuesto. En este apartado se hace un análisis económico teniendo en cuenta las personas involucradas (ingeniero senior y junior), los recursos materiales utilizados como son ordenadores, software para la realización del análisis de las señales, servidores para el almacenaje de los datos… y finalmente, otros costes de uso y mantenimiento.

De cara a que sea más claro y sencillo el análisis de este aparatado, se organizan los diferentes cálculos en tablas. Se empieza por lo tanto con la tabla de recursos humanos, Tabla 19, que incluye cargos, tasas horarias y número de horas de los empleados.

Código	Nombre	Tasa horaria	Número de horas	Total
E1	Ingeniero senior	120€/h	75h	9.000€
E2	Ingeniero junior	60€/h	300h	18.000 €
			Total	27.000 €

Tabla 19: Estimación de costes de recursos humanos

En cuanto al coste de los recursos materiales se refiere, se debe calcular también la amortización de cada uno de los equipos, software o instrumentos utilizados a lo largo del proyecto. Datos fundamentales para estos cálculos son la vida útil de dichos equipos y la cantidad de semanas que han sido utilizados. Entre estos gastos encontramos la licencia del software Matlab, licencia de Microsoft Office para la documentación, el pc empleado para el proyecto y los servidores utilizados de almacenaje. Se resumen sus cálculos en la Tabla 20.

Código	Nombre	Coste	Vida útil	Número de semanas	Total
O	Ordenador	1.400 €	6 Años	20 semanas	89,74 €
LM	Licencia Matlab	2.000 €	10 Años	20 semanas	76,92 €
SL	Servidores Laboratorio	19.751€	15 Años	20 semanas	506,43€
MO	Microsoft Office	100€	1 Año	20 semanas	38,46€
				Total	711,55 €

Tabla 20: Estimación de costes de recursos materiales

Finalmente, hay que tener en cuenta que tener estos equipos también conlleva unos costes de gestión, se resumen todos en un precio de 200€. Además de los empleados y recursos materiales, también hay que tener en consideración el coste del uso y mantenimiento de los mismos. Algunos de estos costes pueden ser: consumo eléctrico de los equipos, coste de alojamiento, coste de conexión de red… Los costes directos están directamente relacionados con estos últimos gastos, por lo tanto, se asume un 10 % de los costes directos como costes no directos. Resumiendo, los costes mencionados posteriormente y los de este último punto, es decir, los gastos totales del proyecto, se representan en la Tabla 21.

Concepto	Total
Horas internas	27.000 €
Amortización	711,55 €
Costes logística y gestión	200 €
Costes directos	27.911,55 €
Costes no directos	2.791,15€
Total	**30.702,7 €**

Tabla 21: Resumen de gastos totales del proyecto

12. Conclusiones y trabajos futuros.

Analizando los resultados obtenidos durante la implementación de las bases de datos ASVspoof 2015, 2017 y 2019 en los trabajos existentes, se han obtenido buenos resultados. Tanto con el uso de la información de la fase, parámetros RPS, como con parámetros MFCC, los resultados sí que han mejorado. En el caso de la unión de las bases de datos en grupos de dos, la fiabilidad ha aumentado hasta en un 16,84%. En el caso de la unión de todo el conjunto, de las tres bases de datos juntas, los resultados han sido buenos, con un EER del 12,78%. Y no solo la tasa EER es menor, si no que de esta forma el sistema está diseñado y preparado para hacer frente a un mayor número de tipos de ataques. Teniendo esto en cuenta se puede decir que sí que ha tenido un impacto positivo la integración de múltiples bases de datos.

Un trabajo muy interesante que podría realizarse en un futuro es hacer experimentos de *cross-db*. La razón es que esto puede dar más información, ya que se cruzarían los datos de las diferentes bases de datos cruzando los modelos.

En lo que a la implementación de las redes neuronales se refiere, los resultados han sido muy positivos, superando en todos los casos los obtenidos con los métodos tradicionales de los trabajos existentes. Además, este proceso ha sido menos demandante tanto de tiempo como de recursos. Por lo tanto, no cabe duda de que las redes neuronales es el camino a seguir.

Finalmente, se ha de determinar cómo se continuará a partir de ahora. Teniendo en cuenta los resultados de este trabajo, se desea continuar implementando las redes neuronales. Por lo tanto, uno de los objetivos de cara al futuro es seguir depurando las redes DNN para incrementar así su rendimiento. Una tarea que se podría llevar a cabo y que resulta muy interesante es realizar las mismas pruebas realizadas en este trabajo, es decir la implementación de múltiples bases de datos, pero utilizando las redes neuronales.

Referencias bibliográficas

[1] Aholab Taldea, «Sitio Web Aholab,» 2020. [En línea]. Available: https://aholab.ehu.eus.

[2] J. Sanchez, I. Saratxaga, I. Hernáez, E. Navas y D. Erro, «The AHOLAB RPS SSD spoofing challenge 2015 submission.,» de *Interspeech 2015*, Dresde, Germany, 2015.

[3] ASVspoof, [En línea]. Available: https://www.asvspoof.org.

[4] Z. Wu, T. Kinnunen, N. Evans, J. Yamagishi, C. Hanilçi, M. Sahidullah y A. Sizov, «ASVspoof 2015: the First Automatic Speaker VerificationSpoofing and Countermeasures Challenge,» de *Interspeech*, Dresde, Germany, 2015.

[5] T. Kinnunen, M. Sahidullah, H. Delgado, M. Todisco, N. Evans, J. Yamagishi y K. A. Lee, «The ASVspoof 2017 Challenge:Assessing the Limits of Replay Spoofing Attack Detection,» de *Interspeech*, Estocolmo, Suecia, 2017.

[6] M. Todisco, X. Wang, V. Vestman, M. Sahidullah, H. Delgado, A. Nautsch, J. Yamagishi, N. Evans, T. Kinnunen y K. A. Lee, «ASVspoof 2019: Future Horizons in Spoofed and Fake Audio Detection,» de *Interspeech*, Gratz, Austria, 2019.

[7] I. Saratxaga, I. Hernáez, D. Erro, E. Navas y J. Sanchez, «Simple representation of signal phase for harmonic speech models,» *Electronic Letters*, nº 45, 2009.

[8] S. Imai, «Cepstral analysis synthesis on the mel frequency scale,» de *ICASSP '83. IEEE International Conference on Acoustics, Speech, and Signal Processing*, Boston, EEUU, 1983.

[9] IEEE, 2015. [En línea]. Available: https://ieeexplore.ieee.org/abstract/document/7029029.

[10] F. Sancho Caparrini, «Redes Neuronales: una visión superficial,» Sevilla, 2019.

[11] A. Kurematsu, K. Takeda, Y. Sagisaka, S. Katagiri, H. Kuwabara y K. Shikano, «ATR Japanese speech database as a tool of speech recognition and synthesis,» *Speech Communication*, vol. 9, nº 4, pp. 357-363, 1990.

[12] L.-W. Chen, W. Guo y L.-R. Dai, «Speaker verification against synthetic speech,» de *7th International Symposium on Chinese Spoken Language Processing* , Tainan, Taiwan, 2010.

[13] E. Khoury, K. Terhi, A. Sizov, Z. Wu y S. Marcel, «Introducing I-vectors for joint anti-spoofing and speaker verification,» de *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, Singapur, 2014.

[14] A. Ogihara, H. Unno y A. Shiozaki, «Discrimination Method of Synthetic Speech Using Pitch Frequency against Synthetic Speech Falsification,» *IEICE*

Transactions on Fundamentals of Electronics, Communications and Computer Sciences, Vols. %1 de %2E88-A, nº 1, 2005.

[15] P. L. de Leon, B. Stewart y J. Yamagishi, «Synthetic speech discrimination using pitch pattern statistics derived from image analysis,» de *Interspeech*, Oregon, EEUU, 2012.

[16] P. L. de Leon, I. Hernáez, I. Saratxaga, M. Pucher y J. Yamagishi, «Detection of synthetic speech for the problem of imposture,» de *Interspeech*, Florencia, Italia, 2011.

[17] Z. Wu, E. S. Chng y H. Li, «Detecting Converted Speech and Natural Speech for anti-Spoofing Attack in Speaker Recognition,» de *Interspeech*, Portland, EEUU, 2012.

[18] J. Sanchez, Utilización de la fase armónica en la detección de voz sintética, Bilbo: Ehu / Upv, 2016.

[19] S. E. R. Montiel, Reconocimiento de voz para un control de acceso mediante red neuronal de retropropagación, México, D.F, 2009.

[20] MathWorks, [En línea]. Available: https://es.mathworks.com/help/deeplearning/ug/long-short-term-memory-networks.html.