TRABAJO FIN DE GRADO

INGENIERÍA DE TECNOLOGÍAS DE TELECOMUNICACIÓN

Implementación y evaluación de robustez de una técnica de watermarking para detección de deepfakes de voz

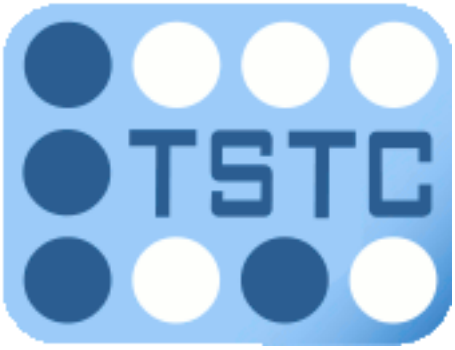
Autor
Pablo Hernández Manrique

Directores
Antonio Miguel Peinado Herreros
Ángel Manuel Gómez García

Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación

—

Granada, Junio de 2024



Implementación y evaluación de robustez de una técnica de watermarking para detección de deepfakes de voz

Autor
Pablo Hernández Manrique

Directores
Antonio Miguel Peinado Herreros
Ángel Manuel Gómez García

Implementación y evaluación de robustez de una técnica de watermarking para detección de deepfakes de voz

Pablo Hernández Manrique

Palabras clave: Marca de agua, Deepfake, Derechos de Autor, Propiedad Intelectual, Imperceptibilidad, Robustez, SNR, PESQ, BER, STOI, Accuracy, Error Medio Absoluto, Entropía Binaria Cruzada.

Resumen

En esta era, los datos, la autenticidad y la protección de los mismos han adquirido un protagonismo y una gran relevancia como consecuencia de la aparición de técnicas avanzadas de manipulación y edición de datos a través del uso inmoral de las tecnologías de inteligencia artifical emergentes, como los *deepfakes*. Estas técnicas, que son capaces de crear contenido hiperrealista y de mucha similitud con el original a partir del cual se generó, han puesto contra la espada y la pared a nuestro juicio para diferenciar entre lo verdadero y lo generado mediante inteligencia artificial.

Con este trabajo pretendemos hacer frente a esta situación haciendo uso de técnicas de marcado de agua basado en redes neuronales y aprendizaje profundo. Las marcas de agua nos permiten firmar el contenido de tal forma que podamos conocer la legitimidad de su procedencia en el receptor. Con este enfoque somos capaces de determinar qué contenido ha sido generado de manera fraudulenta. Se presentarán dos redes neuronales, cada una de ellas con una tarea conceptualmente opuesta, denominadas *Embedder* y *Detector*. El *Embedder*, basado en una arquitectura U-Net, se encargará de realizar una incrustación imperceptible de la marca de agua minimizando las diferencias entre las señales de voz originales y las señales de voz que contienen la marca de agua. El *Detector*, por su parte, trata de detectar la marca de agua sin errores. Dado que la tarea de cada red neuronal es opuesta, se propone una optimización conjunta para alcanzar el equilibrio entre los requisitos de cada una de ellas. Por último, con el fin de garantizar una reconstrucción de la señal de voz por parte del *Embedder* sin poner en compromiso la detección de la marca de agua por parte del *Detector*, se añade una función de coste complementaria al *Embedder* basada en el PESQ de las señales originales y con marca de agua.

Los resultados del estudio indican que el enfoque propuesto es efectivo, ofreciendo un nuevo mecanismo para proteger la integridad de las comunicaciones multimedia y combatir la propagación de informaciones falsas mediante deepfakes. Este trabajo contribuye a los esfuerzos de garantizar

la confiabilidad y seguridad de la información en la era digital, abordando directamente los desafíos emergentes asociados con las tecnologías de inteligencia artificial.

Project Title: Implementation and robustness evaluation of a watermarking technique for voice deepfake detection

Pablo Hernández Manrique

Keywords: Watermark, Deepfake, Copyright, Intellectual Property, Imperceptibility, Robustness, SNR, PESQ, BER, STOI, Accuracy, Mean Absolute Error, Binary Cross Entropy.

Abstract

In this era, data, authenticity, and their protection have taken on disproportionate prominence and critical relevance due to the emergence of advanced data manipulation and editing techniques through the unethical use of emerging artificial intelligence technologies, such as *deepfakes*. These techniques, capable of creating hyper-realistic content that closely resembles the original from which it was generated, have cornered our judgment in discerning between what is true and what is generated by artificial intelligence.

With this work, we aim to address this situation by using watermarking techniques based on neural networks and deep learning. Watermarking allows us to sign the content in such a way that we can know the legitimacy of its provenance at the recipient. With this approach we are able to determine which content has been fraudulently generated.Two neural networks will be presented, each with a conceptually opposite task, named *Embedder* and *Detector*. The *Embedder*, based on a U-Net architecture, will handle the imperceptible embedding of the watermark, minimizing the differences between the original voice signals and the signals that contain the watermark. The *Detector*, on the other hand, attempts to detect the watermark without errors. Since the task of each neural network is opposite, a joint optimization is proposed to balance the requirements of each. Finally, to ensure a reconstruction of the voice signal by the *Embedder* without compromising the watermark detection by the *Detector*, a complementary cost function based on the PESQ of the original and watermarked signals is added to the *Embedder*.

The results of the study indicate that the proposed approach is effective, offering a new mechanism to protect the integrity of multimedia communications and combat the spread of false information through deepfakes. This work contributes to efforts to ensure the reliability and security of information in the digital age, directly addressing the emerging challenges associated with artificial intelligence technologies.

D. **Antonio Miguel Peinado Herreros**, Profesor del Área de Teoría de la Señal del Departamento de Teoría de la Señal, Telemática y Comunicaciones de la Universidad de Granada.

D. **Ángel Manuel Gómez García**, Profesor del Área de Teoría de la Señal del Departamento Teoría de la Señal, Telemática y Comunicaciones de la Universidad de Granada.

Informan:

Que el presente trabajo, titulado ***Implementación y evaluación de robustez de una técnica de watermarking para detección de deepfakes de voz***, ha sido realizado bajo su supervisión por **Pablo Hernández Manrique**, y autorizamos la defensa de dicho trabajo ante el tribunal que corresponda.

Y para que conste, expiden y firman el presente informe en Granada a 24 de Junio de 2024 .

Los directores:

Antonio Miguel Peinado Herreros **Ángel Manuel Gómez García**

A mis padres y hermanas, por acompañarme en el camino.

A mis amigos, por la comprensión y la paciencia que tienen conmigo.

A mi abuela Antonia, por enseñarme lo que es el valor y la valentía.

... y a Elena.

Muchas gracias a todos por limarme las aristas.

Agradecimientos

Siempre me ha resultado muy complicado ponerme sentimental y expresar mi agradecimiento con palabras. Me considero una persona bastante cerrada en cuanto a los sentimientos se refiere y quizás se deba al temor y al pánico que me da sentirme juzgado por ello. Me arrepiento mucho de no decir *te quiero* más seguido ya que son unas palabras mágicas capaces de llenar el corazón de cualquiera. No sé porqué no las digo tan seguido si en realidad casi que no cuesta trabajo. Siempre hay una primera vez para todo y, supongo y solo supongo, que mi primera vez va ser en los agradecimientos de mi trabajo fin de grado.

En primer lugar, me gustaría dar mis agradecimientos más sinceros a mis tutores Antonio y Ángel por la cercanía científica y personal mostrada en sus reuniones. También me gustaría agradecerles por la oportunidad brindada para realizar este trabajo. Nada de esto hubiese sido posible sin ellos.

No me olvido tampoco de mis amigos y colegas de Guadix: Javi, Antonio, Josema, Álvaro, José y Enrique. En general a mis *Wendingos* y a los *Residentes* de la cochera número ocho. No podría estar más agradecido con los amigos que tengo. Por su oreja paciente y por los consejos que me dan. Mis amigos son los que me hacen entender que la vida quizás no es tan complicada como me imagino, que es mucho más simple. Parte de mi y de este trabajo también es vuestra. Porque os alegráis de mis logros como si vuestros fueran, fuistéis, sois y seréis mis amigos para siempre.

A mis padres, Montse y Cayetano, y a mis hermanas, María y Henar, por ser la familia que cualquier persona querría tener. Ya lo decía al principio, quizás no haya cosa de la que me arrepienta más en mi vida que de no deciros y repetiros lo mucho que os quiero. Por ser a quien acudo cuando me invade el miedo, por ser los brazos que, aunque cansados de trabajar, me levantan cuando no me siento fuerte, por ser todo lo que necesito cuando estoy mal. Os quiero.

Por último quería agradecerte a ti también, cariño. Aunque la vida y el destino no nos haya querido juntos solo espero de corazón que los *terraplanistas* no tengan razón y que, en realidad, la Tierra sí que sea redonda y los caminos que de partida fueron opuestos, aunque sea por solo un instante, vuelvan a cruzarse. Por ser como fuiste y eres conmigo, por ser la que irrumpe en mi cabeza cuando me invade el silencio. Dicen que el amor es la cura de todos los males, pero que también puede ser el origen de todos ellos. Que si algo es cierto del amor es que la persona con la cual eliges vivirlo será la razón por la cual ames u odies tu existencia en la Tierra. Tal vez he de confesar que muchas de las definiciones de lo que hoy entiendo por Amor las aprendí contigo. Hay partes de mí que solo existen contigo. He estado buscando sinónimos de olvidar que no conlleven abandonarte por completo. He buscado entre los mares del olvido otro significado que no sea perder tu recuerdo, pero olvidar es separarme de las raíces que me conectan contigo. Te cuelas en mis sueños pero me despierto con las manos vacías y con la boca llena de los *te quieros* que no te dije. Quizás, de nuevo, tendría que haberlo dicho con más frecuencia. Te doy las gracias por darme lugar en un mundo en el que yo pensaba que no iba a encajar nunca. Gracias por demostrarme que puedo y que se me puede querer. Mi corazón y este trabajo también son tuyos. Tu felicidad es la mía. Qué bien haber coincidido, a pesar de que no haya sido para siempre, agradezco el tiempo que nos tocó. Te quiero, te querré, te quise siempre, *Pioja*.

"(...) Bendita radiación de las antenas. Mientras sea tu voz la que me hable como me hablaste hace un minuto apenas. Te quiero, te querré, te quise siempre, desde antes de saber que te quería."

- Jorge Drexler - Telefonía (2018)

"(...) Porque una casa sin tí es una oficina, un teléfono ardiendo en la cabina, una palmera en el museo de cera, un éxodo de oscuras golondrinas. Y me envenenan los besos que voy dando y, sin embargo, cuando duermo sin tí contigo sueño(...) Y cuando vuelves hay fiesta en la cocina y bailes sin orquesta y ramos de rosas con espinas."

- Joaquín Sabina - Y sin embargo (1999)

Índice general

Índice de figuras

Índice de cuadros

Capítulo 1

Introducción

1.1. Descripción del problema

En la actualidad, las falsificaciones de imágenes, vídeos o voz suponen cada vez más un problema de gran magnitud tanto para los sistemas de biometría que hacen uso de ellas, como para la veracidad de la información o la integridad de las personas. Las falsificaciones de imágenes, vídeos o voz presentan un reto significativo para los sistemas biométricos del presente. Estos sistemas, que utilizan características únicas del individuo para verificar su identidad, han tenido una tendencia de uso creciente en los últimos años debido a la facilidad de integración en diferentes plataformas o dispositivos. A pesar de esto, la reciente aparición de las tecnologías de inteligencia artificial y aprendizaje profundo han permitido que la creación de imitaciones de voz altamente convincentes sea cada vez más accesible, planteando serios problemas de seguridad y poniendo en compromiso la integridad de los usuarios que frecuentemente utilizan estos sistemas biométricos.

Para hacer frente a esta problemática y preservar la propiedad intelectual e integridad de la voz, música, imágenes, vídeo o contenido multimedia en general, debemos de ser capaces de determinar si la procedencia del contenido es ilegítima. Para determinar la legitimidad de la procedencia del contenido podemos marcar al mismo antes de su transmisión con un mensaje imperceptible para los receptores en la medida de lo posible. Cuando nos encontramos con contenidos cuya procedencia podría considerarse como ilegítima, podemos hacer uso del mensaje introducido antes de la transmisión para conocer el origen del contenido. A esta técnica se le conoce como *Marca de Agua* o *Watermark*.

El *Marcado de Agua* o *Watermarking* es una técnica que modifica el pro-

pio audio para transmitir un mensaje. Esta técnica plantea numerosos retos, ya que estas modificaciones del audio deben de ser imperceptibles dentro de nuestras posibilidades para los usuarios, pero sí que deben de ser detectables para poder extraer el mensaje. Cabe mencionar que las marcas de agua son sensibles a los cambios violentos en el audio codificado. Si el audio codificado cambia en gran medida, la extracción de la marca de agua no podrá llevarse a cabo y, por tanto, no se logra comprobar la legitimidad de la procedencia de la información.

Actualmente, los sistemas de *watermarking* utilizan inteligencia artificial y aprendizaje máquina para incrustar las marcas de agua en las señales de voz. Con estos procedimientos es posible hacer que los sistemas de *watermarking* sean más robustos a diferentes tipos de ataques.

1.2. Motivación

Tal y como se deduce de la sección anterior, en un mundo cada vez más digitalizado y virtualizado, en el que la información y los datos fluyen libremente, la autenticidad y legitimidad de los contenidos multimedia que consumimos diariamente se ha convertido en un pilar fundamental que mantiene la integridad y veracidad de la información. Las imágenes, vídeos y señales de voz forman parte de la comunicación en la era digital siendo, por tanto, la protección de los mismos fundamental. Sin embargo, la facilidad de acceso y distribución viene de la mano de una amenaza de suma importancia: la aparición de falsificaciones convincentes generadas a través de inteligencia artificial, conocidos como *deepfakes*.

Ante esta situación, surge la necesidad de desarrollar mecanismos que garanticen la autenticidad y legitimidad de la procedencia de los contenidos multimedia. El *marcado de agua* o *watermarking* se presenta como una técnica innovadora y una solución prometedora frente a esta problemática. A su vez, el *marcado de agua* basado en redes neuronales es un campo que actualmente está en auge y que requiere de grandes esfuerzos investigadores para obtener soluciones óptimas y aplicables en un entorno real.

Con este trabajo se pretende abordar estos desafíos, explorando las diferentes posibilidades que ofrece el *watermaking* para asegurar la protección de la propiedad intelectual y la autenticidad de los contenidos multimedia en la era de la inteligencia artificial. Mediante técnicas de aprendizaje profundo, inteligencia artificial y *machine learning*, este trabajo busca contribuir al desarrollo de un sistema de *watermarking* eficaz, robusto y óptimo capaz de

resistir ataques más sofisticados. El desarrollo de este trabajo es fundamental, no solo para el avance de la investigación en el campo del *watermarking* digital basado en redes neuronales, sino que también es de suma importancia para la protección de la legitimidad de los contenidos multimedia y, en consecuencia, para la protección de la integridad del ser humano y de la sociedad.

1.3. Objetivos de este trabajo

El objetivo de este trabajo es desarrollar un sistema de *watermarking* robusto haciendo uso de técnicas de aprendizaje profundo. Por robustez entenderemos que la marca de agua pueda ser extraída del audio codificado tras haber sido transmitido por un canal en el que ha sido expuesto a diferentes ataques que tratan de suprimir la marca de agua introducida en el audio codificado sin degradar la calidad del mismo de manera significativa.

El reto principal es la construcción de un *Embedder* y un *Detector*. El *Embedder* se encargará de incrustar la marca de agua en el audio codificado y reconstruir dicho audio para transmitir el audio codificado junto con la marca de agua. El *Detector* se encarga de, una vez recibido el audio codificado junto con la marca de agua, extraer la marca de agua incrustada en el audio codificado. Estos dos sistemas son entrenados conjuntamente para lograr su cometido. Cada sistema, tanto *Embedder* como *Detector*, se basarán en redes neuronales convolucionales. A su vez, los ataques antes mencionados serán implementados como capas no entrenables entre el *Embedder* y el *Detector*

El sistema debe de obedecer a dos principios fundamentales: robustez e imperceptibilidad. Las señales de voz deben de ser reconstruidas por el *Embedder* y que auditivamente la marca de agua no sea percibida, es decir, que la distorsión introducida por la marca de agua debe de ser mínima e inapreciable, cumpliendo de esta forma con el principio de imperceptibilidad, y el *Detector* debe de ser capaz de extraer la marca de agua aunque las señales de voz hayan sido previamente atacadas y la calidad del audio codificado se reduzca, cumpliéndose así el principio de robustez. La figura 1.1 muestra un diagrama general del sistema implementado.

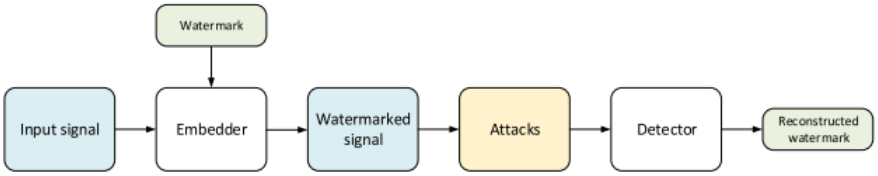


Figura 1.1: Arquitectura del Sistema.[1]

1.4. Delimitaciones

En el presente marco de estudio se asumirá durante todo el documento que las señales de voz codificadas no han sido modificadas antes de entrar al sistema que desarrollaremos en capítulos posteriores. Las señales de voz con las que entrenaremos nuestras redes neuronales provienen de la base de datos TIMIT, que se trata de una base de datos que contiene archivos de audio limpios.

El corpus TIMIT contiene transcripciones otrográficas, fonéticas y de palabras alineadas en el tiempo, así como un archivo de voz monocanal, codificada con 16 bits y muestreada con 16 kHz para cada señal de voz. Las transcripciones del corpus TIMIT han sido verificadas manualmente. Contiene conjuntos de prueba y de capacitación, equilibrados entre sí para para tener una buena cobertura fonética y dialectal. Los metadatos del hablante incluyen género, dialecto, fecha de nacimiento, altura, raza y nivel educativo. De los 630 hablantes, aproximadamente el 70 % son hombres y el 30 % mujeres [5].

1.5. Organización de la memoria

La memoria de este trabajo está estructurada en siete capítulos principales. A continuación se describe brevemente el contenido y el enfoque de cada uno de ellos.

- **Capítulo 1. Introducción:** En este capítulo se ha establecido el marco de trabajo, introduciendo al lector a la problemática central sobre la que se desarrolla este trabajo. También se explica la motivación a partir de la cual surge el proyecto. Por último, se delinean los objetivos específicos del proyecto.
- **Capítulo 2. Watermarking. Antecedentes, dominios y métodos:** En él se da una introducción a los sistemas de *watermarking*,

su origen, dominios en los que se pueden implementar los sistemas y ataques a los que estos se enfrentan.

- **Capítulo 3. Arquitectura de las Redes Neuronales. Implementación del Sistema de Watermarking basado en U-Net:** En esta parte se da una explicación extensa de las arquitecturas y modelos que se van a implementar para desplegar el sistema de marcado de agua o *watermarking*. Se habla del fundamento y del funcionamiento de las mismas y de los resultados que se esperan de ellas. En primer lugar se dará una descripción de la arquitectura implementada para el *Embedder*, con una función de pérdida basada en el Mean Absolute Error. Posteriormente, se realizará lo mismo con la arquitectura del *Detector*, con una función de pérdida basada en la Binary Cross Entropy. Por último, en este capítulo se dará el marco teórico y la introducción al entrenamiento de la función de pérdidas basada en el PESQ como función de coste para las pérdidas del *Embedder* junto con el MAE.

- **Capítulo 4. Ataques implementados en el sistema:** En este capítulo se hace una revisión de los ataques implementados que afectan al funcionamiento de nuestro sistema. Se da una descripción de los mismos y una explicación sobre sus fundamentos matemáticos.

- **Capítulo 5. Marco y Resultados Experimentales:** Aquí se da una descripción detallada de la base de datos utilizada para entrenar los modelos y evaluar su funcionamiento. También se exponen los resultados experimentales obtenidos y se discuten los mismos, sus implicaciones y sus causas. Los resultados obtenidos serán diferenciados en tres etapas diferentes: la etapa de entrenamiento, donde se realiza el entrenamiento de las redes neuronales, la etapa de validación, donde probaremos el aprendizaje de las redes neuronales con un conjunto de datos diferentes y, por último, la etapa de testeo, donde mostraremos el rendimiento de nuestro sistema y su utilidad para un conjunto de datos más general.

- **Capítulo 6. Conclusión y trabajo futuro:** En este capítulo se da una conclusión del trabajo realizado y se habla de los puntos fuertes y, a su vez, de los defectos que el sistema implementado presenta. También se da una reflexión sobre el posible alcance que tiene el proyecto y se comentan algunas mejoras y los futuros pasos que podría tener el mismo, así como su perspectiva de futuro.

- **Capítulo 7. Presupuesto y Cronograma de Trabajo:** Para finalizar con la memoria se muestra un coste económico de la confección de este proyecto, contando con los recursos humanitarios y los materiales utilizados, así como un cronograma del desarrollo del mismo.

Capítulo 2

Watermarking. Antecendentes, dominios y métodos

2.1. Antecedentes

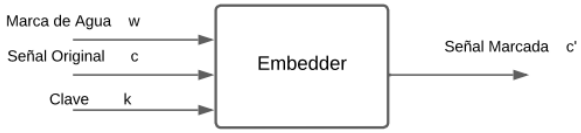
El marcado de agua es una técnica que data de la antigüedad. A lo largo de la historia, esta técnica ha evolucionado en gran medida, adaptándose a los avances tecnológicos y encontrando su aplicación en un gran número de campos de investigación. Dentro de este gran abanico de posibilidades, el *watermarking* digital emerge como una rama especializada, enfocada en la protección de la propiedad intelectual y la autenticación de la información en el mundo digital [2].

A lo largo de últimas décadas se han implementado numerosos métodos para la incrustación de marcas de agua en la información digital. Desde técnicas basadas en la incrustación de las marcas de agua mediante técnicas de procesado clásicas hasta las relativamente actuales redes neuronales.

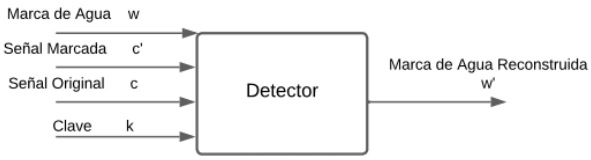
Las tecnologías de watermarking digital aparecieron en 1988, proporcionando confidencialidad, integridad y numerosas innovaciones a las comunicaciones digitales de la época. Para imitar el sistema de percepción humana, la entropía de la información juega un papel esencial en los sistemas de marcado de agua no basados en redes neuronales. Con el fin de alcanzar un compromiso entre la imperceptibilidad y la robustez, la entropía de la información puede obtenerse a través de un modelo JND (Just Noticeable Difference) como el propuesto en [6]. La entropía de la información puede definirse en términos de la distorsión provocada sobre la información original

y permite determinar la posición en la que el mensaje oculto fue incrustado.

Antes de la existencia de redes neuronales y las técnicas de aprendizaje profundo los sistemas de *watermarking* presentaban una arquitectura diferente a lo que actualmente estamos acostumbrados. En este tipo de escenarios, el mensaje que se debía de transmitir era incrustado en la información portadora a través de alguna técnica de *watermarking* junto a una clave o *key* que solo los dos extremos de la comunicación conocen. Esta señal portadora junto al mensaje pasa por un canal de transmisión en el que aparecen ataques como ruido aleatorio, compresión con pérdidas o supresión de muestras, lo que produce una distorsión en la señal portadora y en el mensaje que transporta. Estos ataques maximizan la entropía de la información, lo que, en consecuencia, conlleva un aumento de la ambigüedad del mensaje oculto que se desea transmitir. Por tanto, las técnicas de *watermarking* que no están basadas en redes neuronales solo pueden aplicarse en señales que tengan una entropía de información más alta que la introducida por los ataques en el canal [7]. En el proceso de extracción de la marca de agua, el *Detector*, junto a la clave que se conoce únicamente en el transmisor y en el receptor, extrae la marca de agua de la señal transmitida.



(a) Primera aproximación Embedder basada en [2].



(b) Primera aproximación Detector basada en [2].

Figura 2.1: Primera aproximación de un sistema de Watermarking [2].

A pesar de las notables diferencias de implementación en comparación con los sistemas de *watermarking* basados en redes neuronales, ambos obedecen a los mismos dos principios: imperceptibilidad y robustez. Por tanto, conceptualmente, el cometido final es el mismo. A pesar de las diferencias

de implementación, la influencia de los sistemas de marcado de agua clásicos sobre los sistemas de marcado de agua basados en redes neuronales es clara.

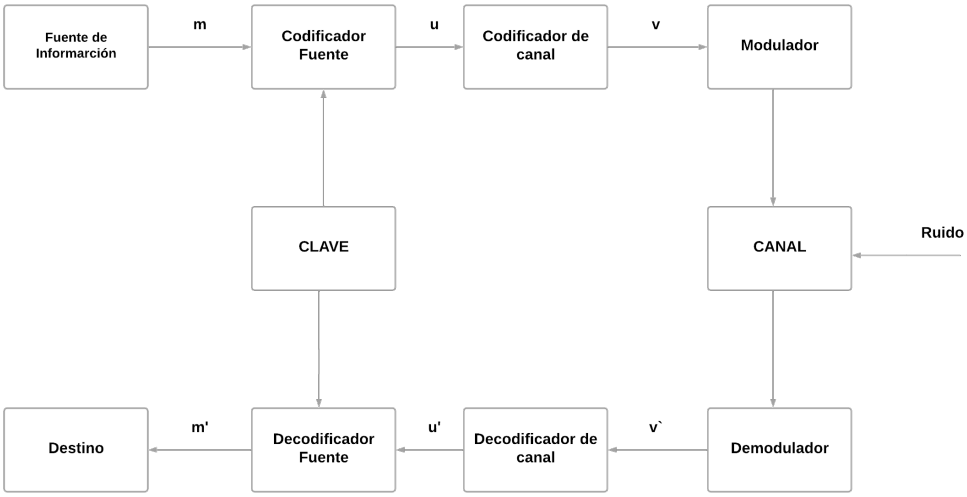


Figura 2.2: Sistema Completo de Watermarking sin DNNs [2]

2.2. Watermarking mediante Redes Neuronales

Como ya se mencionaba en la sección anterior, una marca de agua se trata de un mensaje que se incrusta en diversos contenidos multimedia como puede ser una imagen, un vídeo o una señal de voz. La marca de agua debe de ser incrustada de tal forma que la distorsión provocada por la misma sobre el contenido multimedia sea mínima e imperceptible.

El mensaje incrustado en el contenido multimedia que se desea transmitir, ya sean imágenes o señales de voz, en ocasiones llamados portadores o *carriers*, suele ser una cadena de texto o una señal binaria. La mayoría de los sistemas de *watermarking* constan de dos pasos principales:

1. ***Embedding*:** En primer lugar, se intenta ocultar el mensaje en el contenido portador o *carrier*. Este paso se realiza antes de que el contenido multimedia sea transmitido.
2. ***Detección*:** En este paso el receptor descodifica el contenido multimedia recibido en busca de una marca de agua que garantice la legitimidad de la procedencia de la transmisión.

En las primeras aportaciones en el campo del marcado de agua, la mayoría de los esfuerzos estaban orientados a imágenes y audio. Estos primeros

enfoques implementaban diferentes técnicas de codificación para garantizar la imperceptibilidad y robustez de la marca de agua como, por ejemplo, técnicas de espectro expandido (SS) [8], codificación LSB [9], codificación en fase [10], o, como la introducida a principios de este siglo, modulación del índice de cuantificación (QIM) [11].

De todo esto se desprende que la mayoría de métodos aplicados en sistemas de *watermaking* basan su clasificación en la técnica de codificación que utilizan para incrustar la marca de agua. No obstante, también se ha visto en la literatura una clasificación basada en el dominio en el que se incrusta la marca de agua. Estos dominios son el dominio del tiempo (DT) o el dominio transformado (DF) [1].

La fórmula genérica para una señal de voz con marca de agua en el dominio del tiempo (DT) puede ser expresada como en [1]:

$$y(n) = x(n) + \alpha w(n) \tag{2.1}$$

donde $y(n)$ representa la señal de voz con marca de agua, mientras que $x(n)$ y $w(n)$ representa la señal de voz original y la marca de agua respectivamente. El parámetro α determina la fuerza de la marca de agua incrustada, teniendo un gran impacto en la perceptibilidad y detectabilidad de la marca de agua. En el dominio transformado (DF) la fórmula genérica varía un poco y puede expresarse de la siguiente manera [1]:

$$Y(k) = X(k) + \alpha W(k) \tag{2.2}$$

donde $Y(k)$ representa a la señal con marca de agua en el dominio transformado, mientras que $X(k)$ y $W(k)$ representa la señal de voz original y la marca de agua en el mismo dominio transformado respectivamente. Existen varios dominios transformados como, por ejemplo, DFT (Discrete Fourier Transform), DWT (Discrete Wavelet Transform), FrFT (Fractional Fourier Transform) o DCT (Discrete Cosine Transform). En este trabajo, como en [1] y en [3], consideraremos la STFT (Short Time Fourier Transform) como extensión temporal de la DFT.

Ambos dominios son una solución válida como veremos en la sección 2.3 ya que la creciente demanda de métodos que aseguren la protección de la propiedad intelectual a través de redes neuronales e inteligencia artificial hacen que este tipo de soluciones sean las favoritas de los investigadores y desarrolladores en general. A pesar de las diferentes soluciones propuestas a lo largo de estas últimas décadas para esta problemática, su traslación directa al mundo de la inteligencia artificial y las redes neuronales produce

que las dificultades en el mundo del marcado de agua digital sean abordadas desde una perspectiva diferente. Este nuevo paradigma genera diferencias no solo en la forma en la que se miden la robustez, la imperceptibilidad y el rendimiento, sino también en el dominio de inserción como se mencionaba en líneas anteriores.

El punto fuerte de las redes neuronales es que poseen la habilidad de detectar patrones y características complejas automáticamente y de forma directa a través de los datos de entrada. La características extraídas por parte de las redes neuronales dependerán de la tarea encomendada a las mismas pero, en general, el proceso de aprendizaje de extracción de características se lleva a cabo a través de un entrenamiento con un conjunto de datos determinado. El proceso de aprendizaje es optimizado a través de la minimización de una función de pérdida determinada entre los datos originales y los generados por la red neuronal [12].

La idea principal detrás del *watermarking* digital basado en redes neuronales es aprovechar la redundancia de los parámetros aprendidos por el *Embedder* para incrustar información adicional afectando lo mínimo posible a la calidad de la señal reconstruida y, posteriormente, que el *Detector* sea capaz de extraer las características correspondientes a las marcas de agua previamente incrustadas.

2.2.1. STFT: Short Time Fourier Transform

El motivo que hay detrás de la elección de la representación de la señal de voz a través de la STFT es consecuencia de la naturaleza no estacionaria de la voz humana. La amplitud y el espectro de una señal pueden variar en el tiempo. Cuando hablamos, la frecuencia es distinta en los diferentes segmentos de la frase que pronunciamos y, en consecuencia, una completa caracterización de las señales no estacionarias en el dominio de la frecuencia debe de incluir un aspecto temporal [13]. Esto hace que el análisis en tiempo y frecuencia sea el más adecuado para procesar y, por ello, el uso de la STFT como alternativa para la representación de las señales de audio nos permitirá trabajar con las mismas más cómodamente. La STFT puede definirse a través de la siguiente expresión [1]:

$$STFT(n,k) = \sum_{m=-N/2}^{N/2-1} g(m)x(n+m)e^{-j2\pi mk/N} \tag{2.3}$$

donde $g(m)$ es una ventana que se desplaza sobre la señal en el dominio temporal. A través de esta ventana de longitud N se van extrayendo segmentos de la señal de voz de los que se calcula su transformada de Fourier.

Estos segmentos son combinados en una representación 2D del audio con el tiempo en un eje y la frecuencia en el otro eje. La representación compleja usualmente se expresa como amplitud y fase, donde la amplitud (magnitud) se conoce como espectrograma y se define como [1]:

$$SPEC(n,k) = |STFT(n,k)|^2 \tag{2.4}$$

donde k es el eje de las frecuencias y n es el eje temporal.

Como ya se mencionaba, la longitud de los segmentos viene dada por la longitud de la ventana o *window length*, mientras que la distancia entre ellos viene dada por la longitud de salto o *hop length* (veáse la figura 2.3). Comúnmente el *hop length* suele ser de una menor longitud que el tamaño de la ventana para que las ventanas se solapen entre sí.

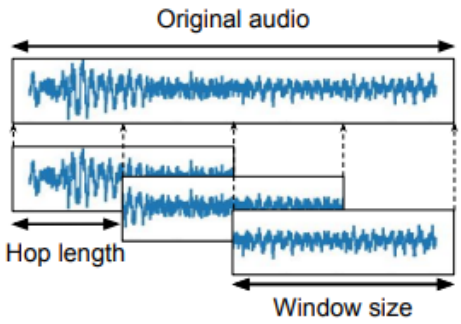


Figura 2.3: Ejemplo visual STFT [3]

Desde la STFT la forma de onda puede ser reconstruida sin pérdidas simple y llanamente haciendo uso de la operación inversa, es decir, la ISTFT (Inverse Short Fourier Transform). Esto puede verse de manera gráfica en la figura 2.4.

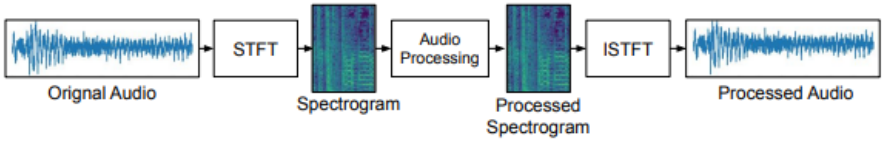


Figura 2.4: Obtención de la forma de onda a través de la ISTFT [3].

2.3. Elección del Dominio

Para abordar esta pregunta de manera tajante debemos de ser conocedores de las diferentes ventajas e incovenientes que nos proporcionan cada uno de estos dominios. En [4] se da una descripción detallada de las capacidades que nos aportan cada uno de estos dominios en el campo del *watermarking*.

Los sistemas de *watermaking* en el dominio del tiempo suelen estar dotados de una mayor sencillez que los sistemas de *watermarking* en el dominio transformado. Esta sencillez viene determinada por su *modus operandi*, ya que modifican directamente las muestras de audio, sin transformaciones previas. Generalmente son tratadas como soluciones *ad hoc* y representan una solución rápida para sistemas de marcado de agua.

Por otro lado, los sistemas de *watermarking* basados en el dominio transformado suelen ser los preferidos por investigadores y diseñadores. Para este tipo de sistema son necesarias dos operaciones más: la transformación y la transformación inversa antes y después de la incrustación de la marca de agua. Por este motivo, en los sistemas de este tipo es necesario garantizar que las muestras que componen nuestra señal en el dominio transformado puedan ser inversamente transformadas para recuperar la forma de onda en el dominio del tiempo [4].

Intuitivamente, podríamos considerar los sistemas basados en el dominio transformado como una mejor opción para alcanzar una imperceptibilidad avanzada debido a la disponibilidad de modelos psicoacústicos en el dominio transformado. A pesar de esto, un sistema basado en el dominio del tiempo con un mecanismo preciso de ajuste heurístico también puede alcanzar niveles altos de imperceptibilidad. En cuanto a la robustez y seguridad no se revelan datos significativos en la literatura que arroje un resultado definitivo sobre que dominio es mejor [4].

En este trabajo, como se mencionaba en la sección anterior, se opta por un sistema basado en el dominio transformado, siendo este dominio la STFT de la forma de onda de la señal de voz, siguiendo la línea de trabajo presentada en [1].

2.4. Tipos de ataques

La robustez en los sistemas de *watermarking*, junto a la imperceptibilidad, es una de las prioridades principales en el diseño de estos sistemas. Por

lo general, existe un equilibrio entre robustez e imperceptibilidad aunque, en ocasiones, la imperceptibilidad puede verse comprometida con el fin de alcanzar una mayor robustez, especialmente cuando el sistema hace frente a ataques más sofisticados.

Existen diferentes criterios para clasificar los ataques contra los sistemas de *watermarking*. Las señales de voz con la marca de agua incrustada pueden ser atacados intencionadamente o no, esta es la gran primera clasificación. Por ejemplo, una comprensión con pérdidas aparentemente es un ataque no intencionado mientras que la adición de ruido sí que podría ser un ataque intencionado. En [4] se da una explicación extensa de los diferentes ataques que existen, se discute su naturaleza y su ámbito de aplicación. En este estudio se hace una diferenciación entre ataques básicos y ataques avanzados. Cabe destacar que los ataques considerados en la mayoría de estudios de sistemas de *watermarking* son bastante limitados, lo que acrecienta el riesgo de que usuarios experimentados sean capaces de eliminar las marcas de agua con ataques no previstos en los estudios de los que hablamos. A su vez, en general, las formas de evaluar la robustez del sistema no están estandarizadas como en el caso de la imperceptabilidad. Por esta razón, es preferible entrenar al sistema contra el mayor número de ataques posible con el objetivo de garantizar su robustez y aplicabilidad en una gran cantidad de escenarios. Los ataques descritos en [4] vienen recogidos en la tabla 2.1 y su clasificación en 2.5.

Cuadro 2.1: Ataques descritos en [4].

Attacks	**Abbrev.**
Closed-loop	CLP
Requantization	RQZ
Filtering	FTR
Amplitude scaling	ASC
Lossy compression (MP3/AAC)	LCP
Adding white Gaussian noise	WGN
Adding echoes	ECH
Cropping	CRP
Jittering	JTR
Time shifting	TSH
Pitch-invariant time scaling	TSC
Time-invariant pitch scaling	PSC
Speed scaling	SPS
Mask attack	MSK
Replacement	RPM

Los ataques que implementaremos probarán la robustez de nuestro sistema y su eficacia para reconstruir la marca de agua en situaciones desfavorables. Los ataques implementados son: adición de ruido, filtrado paso-baja y supresión de muestras. El fundamento matemático de estos ataques se desarrollará con detalle en capítulos posteriores.

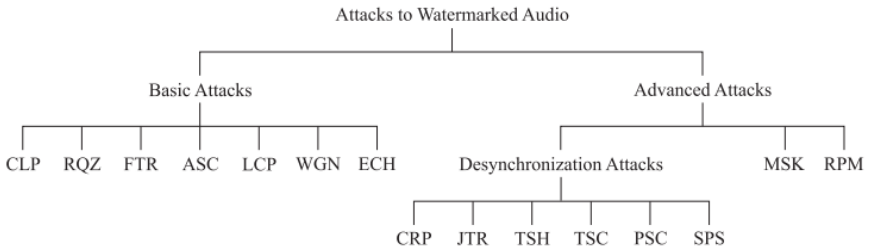


Figura 2.5: Clasificación de los ataques [4].

Capítulo 3

Arquitectura de las Redes Neuronales. Implementación del Sistema de Watermarking basado en U-Net.

3.1. Redes neuronales propuestas

Las arquitecturas de las redes neuronales presentadas en este trabajo están basadas en las expuestas en [1]. Como ya se mencionaba en el capítulo anterior, los sistemas de *watermarking* pueden dividirse en dos arquitecturas principales: el *Embedder* y *Detector*. La arquitectura que vamos a considerar es la típica que se considera en los sistemas de *watermarking*. Esta arquitectura viene mostrada en la figura 1.1 del capítulo 1. En [1], la arquitectura propuesta es capaz de cubrir dos modelos. Un primer modelo que utiliza la STFT como entrada y un segundo modelo que utiliza las señales de audio en crudo como entrada. En este trabajo, como venimos adelantando en capítulos anteriores, se opta por el modelo que utiliza la STFT como entrada debido a los inconvenientes que supone trabajar con señales de audio no estacionarias y con una elevada frecuencia de muestreo (16 kHz). A su vez, entrenar a las redes neuronales con los datos en crudo obliga a las mismas a aprender a extraer las características de los datos presentados en el bucle de entrenamiento, lo que entorpece el proceso de aprendizaje y produce peores resultados. Por otro lado, la extracción de características ya viene parcialmente proporcionada por el cálculo de la STFT, lo que hace que el primer modelo propuesto parta con una ventaja notable respecto al segundo.

Las arquitecturas propuestas tanto para el *Embedder* como para el de-

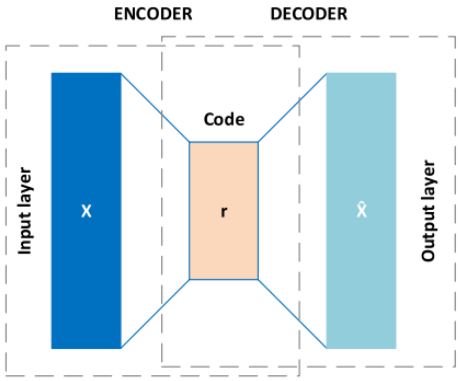
tector vienen descritas en las secciones 3.2 y 3.3 respectivamente. Dado que estas dos redes neuronales tienen dos tareas que pueden considerarse opuestas, cada una tiene su propia función de pérdida. Sin embargo, los gradientes producidos por el *Detector* se retropropagan hacia el *Embedder*. Como consecuencia de esto, las dos redes neuronales son dependientes la una de la otra y, por tanto, han de entrenarse conjuntamente [1].

3.2. Arquitectura del Embedder

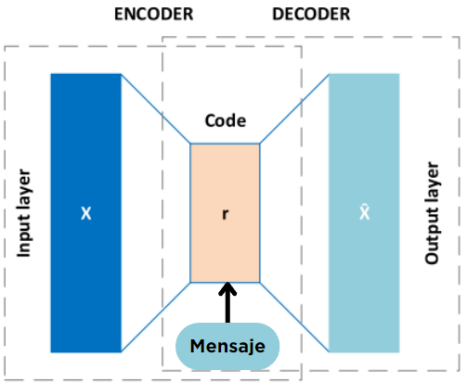
La arquitectura de la red que representa al *Embedder* está basada en el mismo concepto que una U-Net como las presentadas en [14] o en [15], conceptual y estructuralmente parecidas a las redes *Embedder*. Las redes neuronales *Embedder* toman una entrada, a la que llamaremos $\mathbf{x}$, que se trata del vector de características de la señal de voz de entrada, y, tras algunas operaciones, la transforman en una representación, que llamaremos $\mathbf{r}$, en un espacio latente. Posteriormente, realizan las operaciones inversas a $\mathbf{r}$ para obtener de nuevo a $\mathbf{x}$. A esta nueva representación de $\mathbf{x}$ la llamaremos $\hat{\mathbf{x}}$. El objetivo final es que $\hat{\mathbf{x}} - \mathbf{x} = \epsilon$ con ϵ lo más pequeño posible, esto es que $\hat{\mathbf{x}}$ y $\mathbf{x}$ sean muy similares entre sí (veáse la figura 3.1a).

En nuestro caso, utilizamos el *Embedder* para reducir la entrada a las dimensiones del espacio latente de tal forma que la marca de agua pueda ser escondida en la representación de la entrada en dicho espacio latente (veáse 3.1b). Es decir, la marca de agua es añadida a $\mathbf{r}$ pero el objetivo principal del *Embedder* sigue siendo el mismo. El *Embedder* debe de ser capaz de ignorar la marca de agua y eliminar cualquier diferencia que exista entre la entrada original, $\mathbf{x}$, y la reconstrucción de la entrada, $\hat{\mathbf{x}}$.

Aunque el *Embedder* tienda a eliminar la marca de agua para cumplir con la premisa de que $\hat{\mathbf{x}} - \mathbf{x} = \epsilon$, con ϵ lo más pequeño posible, el objetivo principal del sistema al completo es detectar la marca de agua. Este objetivo principal nos da a entender que el *Embedder* no puede ser entrenado de manera independiente sino que se debe entrenar junto al *Detector* para que la marca de agua no sea eliminada por completo en la reconstrucción de la entrada $\hat{\mathbf{x}}$ y pueda ser detectada en el *Detector* [1].



(a) Diagrama Embedder [1].



(b) Diagrama Embedder con mensaje [1].

Figura 3.1: Diagramas de las arquitecturas del Embedder

Este mensaje o marca de agua será una señal binaria generada aleatoriamente aunque, en algunas implementaciones de la literatura, como en [16] y [17], se utiliza este mensaje para transmitir información de valor. Por último, cabe destacar que la función de pérdida implementada en el *Embedder* es el error medio absoluto o *Mean Absolute Error* (MAE) entre la salida del *Embedder*, llamada $\hat{\mathbf{x}}$, y la señal original, llamada $\mathbf{x}$.

$$MAE(\mathbf{x}, \hat{\mathbf{x}}) = \frac{1}{N} \sum_{i=1}^{N} |\hat{\mathbf{x}} - \mathbf{x}| \tag{3.1}$$

donde N es la longitud de la señal o, lo que es lo mismo, el número de muestras, $\mathbf{x}$ es un vector que representa a la señal original y $\hat{\mathbf{x}}$ es un vector que representa a la señal reconstruida por el *Embedder*. Como podemos ver, la sumatoria que se encarga de calcular el MAE va hasta el número de muestras porque calculamos el error muestra a muestra para determinar el error total cometido en toda la señal y finalmente normalizarlo por el número

total de muestras. De esta forma conseguimos el error promedio cometido por el *Embedder* en la reconstrucción de la señal completa.

3.2.1. Modelo propuesto para el Embedder

Como ya se mencionó en secciones anteriores, la entrada del modelo propuesto para implementar el *Embedder* se trata de la STFT de las señales de voz que forman nuestra base de datos. Para el cálculo de la STFT utilizaremos una ventana de Hanning de 1024 muestras de longitud, resultando en una STFT con el mismo número de puntos en frecuencia. A su vez, la STFT está calculada con ventanas parcialmente superpuestas en la que los instantes de tiempo están separados 512 muestras. El resultado del cálculo de la STFT con estos parámetros es una matriz de dimensiones $512 \times 64 \times 2$, donde 512 representa el número de puntos de frecuencia, 64 el número de tramas temporales y 2 el número de canales, en este caso la parte real e imaginaria de la STFT. El motivo que hay detrás de la elección de estas dimensiones para la representación de la STFT de las señales de voz se debe a que queremos trabajar con señales de longitud fija para facilitar la tarea del *Embedder*. A su vez, escogemos 512 puntos de frecuencia y 64 tramas temporales porque, al tratarse de números que son potencia de dos, nos permiten conservar la simetría de la arquitectura U-Net.

La parte de la arquitectura que realiza el downsampling consta de 5 bloques que realizan la convolución bidimensional de la señal de entrada, reduciendo las dimensiones espaciales de la misma y aumentando el número de filtros. Tras la convolución bidimensional, la señal pasa por una capa de normalización (*batch normalization*) y por una función de activación Leaky ReLU cuyo parámetro α está fijado a 0.2. Tras los 5 bloques que componen la parte de downsampling de la arquitectura la representación de la señal de entrada en el espacio latente adopta unas dimensiones de $16 \times 2 \times 256$, donde, siguiendo la misma estructura que antes, 16 representa el número de puntos de frecuencia, 2 el número de tramas temporales y 256 el número de canales. En este punto, la marca de agua es incrustada en la señal mediante la concatenación al final de la dimensión de los canales. La marca de agua se trata de una señal binaria generada a partir de un conjunto de mensajes escogidos aleatoriamente, dando lugar a un vector unidimensional de 512 bits. Dado que la marca de agua es un vector de una dimensión, esta es repetida 16×2 veces antes de ser incrustada con el fin de incrementar la probabilidad de que sea preservada. El tensor resultante tras la concatenación de la marca de agua con la representación de la señal de entrada en el espacio latente rompe la simetría de la arquitectura de la U-Net, lo que afecta a la parte de upsampling y reconstrucción del *Embedder*. Este problema queda resuelto en [1] añadiendo otra capa de convolución bidimensional que reduce

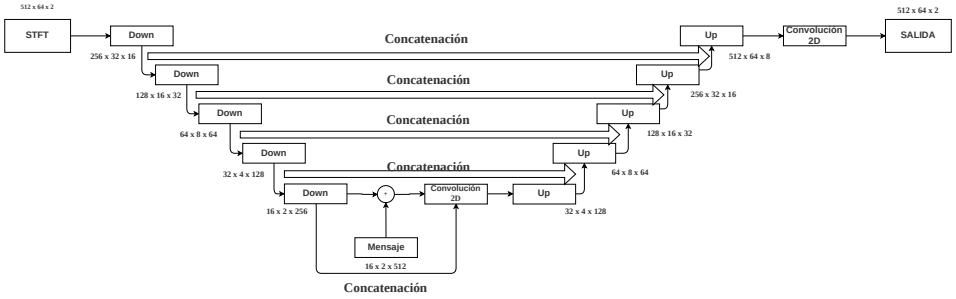


Figura 3.2: Arquitectura del Embedder basada en U-Net [1].

el número de canales y restaura la simetría.

La segunda parte de la arquitectura del *Embedder* trata de reconstruir la señal de voz inicialmente introducida. Esta parte también consta de 5 bloques, pero en lugar de realizar la convolución bidimensional como hacen los bloques de downsampling, estos realizan la convolución bidimensional transpuesta. Tras la convolución transpuesta, y de la misma forma que la parte de downsampling, la señal pasa por una capa de normalización (*batch normalization*), posteriormente pasa una capa de *dropout* con una probabilidad del 50 % y, por último, por una capa de activación ReLU. El objetivo de la capa de normalización es el mismo que en la parte de downsampling, normaliza los datos para su procesado. La introducción de la capa de *dropout* tiene como objetivo dotar a la red de robustez. Si eliminamos el 50 % de la salida de cada capa estamos obligando a la red a adaptarse a los ataques. Los últimos dos bloques de la parte del *Embedder* que realiza el upsampling no tienen capas de *dropout*.

Otro aspecto a destacar es que la salida de los bloques de downsampling que realizan la convolución bidimensional son concatenadas antes del bloque correspondiente al upsampling que realiza la convolución bidimensional transpuesta (veáse la figura 3.2). Esto tiene como propósito facilitar al *Embedder* la tarea de la reconstrucción de la señal de voz. Tras los 5 bloques de upsampling obtenemos un tensor de dimensiones $512 \times 64 \times 8$ mientras que el introducido inicialmente tenía dimensiones $512 \times 64 \times 2$. Para conservar esta simetría entre los tensores de entrada y salida, justo después del último bloque del upsampling, se añade otro bloque que realiza la convolución bidimensional, reduciendo el número de canales de 8 a 2.

Cuadro 3.1: Especificaciones del Embedder. Dimensiones de la señal a lo largo de la Red.

Tipo	Filtros	Tamaño/Paso	Salida
Convolucional 2D	16	5 x 5 / 2	256 x 32
Convolucional 2D	32	5 x 5 / 2	128 x 16
Convolucional 2D	64	5 x 5 / 2	64 x 8
Convolucional 2D	128	5 x 5 / 2	32 x 4
Convolucional 2D	256	5 x 5 / 2	16 x 2
-Incrustación de la Marca de Agua- 512 x 1			
Convolucional 2D	256	5 x 5 / 2	16 x 2
Convolucional 2D transpuesta	128	5 x 5 / 2	32 x 4
Convolucional 2D transpuesta	64	5 x 5 / 2	64 x 8
Convolucional 2D transpuesta	32	5 x 5 / 2	128 x 16
Convolucional 2D transpuesta	16	5 x 5 / 2	256 x 32
Convolucional 2D transpuesta	8	5 x 5 / 2	512 x 64
Convolucional 2D	2	5 x 5 / 1	512 x 64

3.3. Arquitectura del Detector

La señal reconstruida, que viene dada por la salida del *Embedder*, es la entrada del *Detector*. La arquitectura propuesta para implementar el detector es similar a la propuesta en [18], realizando funciones similares a un clasificador de imágenes. Estrictamente hablando, nuestro detector no es un clasificador de imágenes pero, desde el punto de vista de las redes neuronales, podemos considerarlo como tal. El objetivo principal del *Detector* que vamos a implementar es detectar todos los bits que conforman la marca de agua introducida anteriormente en el *Embedder*. Lo comparamos con un clasificador de imágenes porque, al tratarse la marca de agua de una señal binaria, el detector se encarga de clasificar cada uno de los bits de la marca de agua como 0 o 1.

En el caso del detector, la función de pérdida implementada es la Binary Cross Entropy (BCE). Esta función de pérdida mide el etiquetado incorrecto de la clase de datos por parte de un modelo, es decir, nos da el número de bits que el detector clasifica incorrectamente. La expresión que rige es función de pérdida es la siguiente:

$$BCE(\mathbf{y}, \hat{\mathbf{y}}) = -\frac{1}{n}\sum_{i=1}^{n} y_i log(p(\hat{y}_i)) + (1 - y_i) log(p(1 - \hat{y}_i)) \tag{3.2}$$

donde y_i es el bit objetivo $p(\hat{y}_i)$ es la predicción realizada por la red neuronal,

n es la longitud de la marca de agua, $\mathbf{y}$ es un vector binario que representa a la marca de agua original incrustada en el *Embedder* e $\hat{\mathbf{y}}$ es un vector binario que representa a la marca de agua reconstruida por el *Detector*.

3.3.1. Modelo propuesto para el Detector

El modelo propuesto para implementar el detector consta de 6 bloques que realizan un downsampling a la señal de entrada del detector (salida del *Embedder*). Estos bloques que realizan el downsampling de la señal de entrada son bastante similares a los de la red del *Embedder*. Los bloques del *Detector* realizan la convolución bidimensional, reduciendo las dimensiones del tensor de entrada hasta llegar al final. Para reconstruir la salida se usa una capa *fully conected* tras aplanar el tensor en el último bloque (veáse la figura 3.3). En este punto nos queda un tensor de 512×1 al que llamaremos marca de agua reconstruida. Este tensor, de la misma forma que el incrustado en el *Embedder*, será repetido 16×2 veces con el fin de compararlo con el inicialmente generado e incrustado.

Al igual que en el *Embedder*, tras cada bloque de convolución bidimensional, los tensores pasarán por una capa de normalización (*batch normalization*) con el fin de normalizar los datos. Posteriormente, se utilizará una función de activación Leaky ReLU con un parámetro α de 0.2.

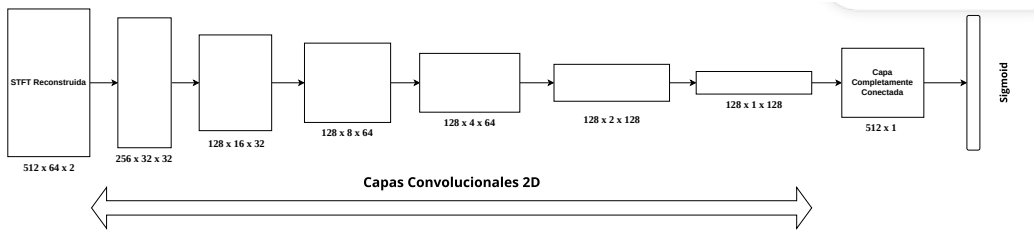


Figura 3.3: Diagrama de bloques del Detector [1].

Cuadro 3.2: Especificaciones del Detector. Dimensiones de la señal a lo largo de la Red.

Tipo	Filtros	Tamaño/Paso	Salida
Convolucional 2D	32	5 x 5 / 2	256 x 32
Convolucional 2D	32	5 x 5 / 2	128 x 16
Convolucional 2D	64	5 x 5 / 1 x 2	128 x 8
Convolucional 2D	64	5 x 5 / 1 x 2	128 x 4
Convolucional 2D	128	5 x 5 / 1 x 2	128 x 2
Convolucional 2D	128	5 x 5 / 1 x 2	128 x 1
Fully Conected		16384 x 512	512

3.4. Entrenamiento de los modelos

Haciendo eco de lo previamente dicho en secciones anteriores, el proceso de entrenamiento de ambas redes neuronales ha de realizarse de manera conjunta porque el *Embedder* y el *Detector* tienen tareas que pueden considerarse opuestas [1]. Como ya se dijo antes, el objetivo principal de la red que hace de *Embedder* es llevar a cabo la reconstrucción de la señal de voz de entrada, reduciendo al mínimo las distorsiones y diferencias entre la señal reconstruida y la señal original. En este proceso de reconstrucción, el *Embedder* tiende a eliminar la marca de agua con el fin de que la señal reconstruida sea lo más parecida posible a la original. Por tanto, si entrenamos el *Embedder* de manera independiente, la señal reconstruida por el mismo no contendrá la marca de agua.

Por esta razón se debe de condicionar al *Embedder* durante el entrenamiento de las redes neuronales, ya que, si el *Detector* espera una señal de audio que contenga la marca de agua pero la señal reconstruida por el *Embedder* no contiene dicha marca de agua o algún residuo de la misma, el *Detector* no puede aprender nada y, por tanto, el entrenamiento no resulta satisfactorio. Por otra parte, si restringimos demasiado al *Embedder* con el fin de que el *Detector* cumpla con su tarea, la tarea del *Embedder* pasa a un segundo plano y, en consecuencia, las señales de voz originales no se reconstruyen correctamente.

Para cumplir con estas condiciones, en [1] asignan diferentes pesos a las redes neuronales en el proceso de entrenamiento con el fin de balancear el proceso de aprendizaje del *Embedder* y del *Detector* y asegurar la convergencia de ambas redes neuronales. Los pesos de cada red neuronal varían con cada época del proceso de entrenamiento. Podemos definirlos a través

de las siguientes funciones definidas por partes:

$$w_e(i) = \begin{cases} \gamma_{w_e} & \text{si } i \leq 1 \\ \gamma_{w_e} + (i-1)\Delta_{w_e} & \text{si } 1 < i \leq M \end{cases} \tag{3.3}$$

$$w_d(i) = \begin{cases} \gamma_{w_d} & \text{si } i \leq 1 \\ \gamma_{w_d} - (i-1)\Delta_{w_d} & \text{si } 1 < i \leq M \end{cases} \tag{3.4}$$

En [1], las funciones definidas por partes son diferentes. Nosotros las hemos adaptado a nuestra conveniencia. Igualmente, los parámetros fijados en [1] son diferentes a los que nosotros hemos utilizado. A pesar de estas diferencias, conceptualmente el funcionamiento y el objetivo son los mismos. En las ecuaciones 3.3 y 3.4 los parámetros γ_{w_e}, γ_{w_d}, Δ_{w_e} y Δ_{w_d} son parámetros de diseño y M es el número total de épocas. En la siguiente tabla vienen recogidos los valores numéricos de los parámetros utilizados:

Cuadro 3.3: Valores de los parámetros utilizados para los pesos del modelo

Parámetro	**Valor**
γ_{w_e}	1
γ_{w_d}	2
M	200
Δ_{w_e}	0.2
Δ_{w_d}	0.2

Los valores fijados para los parámetros de diseño son escogidos siguiendo un razonamiento simple. En las primeros épocas del entrenamiento, el *Embedder* prevalece sobre el detector, en consecuencia, asignamos pesos mayores a la red del *Detector* en las primeros épocas del entrenamiento para balancear esta situación [1]. Podemos visualizar gráficamente la evolución de los pesos durante el entrenamiento en la figura 3.4.

Figura 3.4: Pesos del Embedder y del Detector

La función de pérdida total, fruto de la combinación de las funciones de pérdida del *Embedder* y del *Detector*, se rige a través de la siguiente expresión:

$$L_{Total} = W_{Embedder} \cdot L_{Embedder} + W_{Detector} \cdot L_{Detector} \tag{3.5}$$

donde $W_{Embedder}$ es el peso asignado a la pérdida del *Embedder*, $L_{Embedder}$ es la pérdida asociciada al *Embedder*, definida como el MAE, $W_{Detector}$ es el peso asignado al *Detector* y $L_{Detector}$ es la pérdida asociada al *Detector*, definida como el BCE.

A su vez, los pesos iniciales seleccionados para las dos redes neuronales son de suma importancia. Si introducimos valores iniciales demasiado altos podemos introducir inestabilidad y que, a raíz de ellos, las redes neuronales diverjan. Por ello, como puede verse en 3.4, reducimos el peso del *Detector* paulatinamente mientras que aumentamos el peso del *Embedder*, asegurando de esta forma estabilidad en la convergencia.

3.5. Función de pérdida basada en el PESQ

En este apartado, además de las funciones de pérdida que poseen cada uno de los sistemas (MAE para el *Embedder* y BCE para el *Detector*), se propone una métrica perceptual para la evaluación de la calidad del habla, que también es adecuada como función de pérdida para el entrenamiento de sistemas basados en redes neuronales y aprendizaje profundo [19].

En [19], esta métrica es derivada del algoritmo de evaluación perceptual de la calidad del habla [20] y, a su vez, es calculada a partir de los espectros de potencia de las señales de habla procesadas y de referencia. De esta forma, dos términos de perturbación que tienen en cuenta la distorsión una vez se han considerado los efectos de enmascaramiento auditivo y umbral, son capaces de corregir las funciones de pérdida del *Embedder* y del *Detector* añadiendo una serie de criterios perceptuales basados en las propiedades psicoacústicas del sistema auditivo humano.

Para la consecución de la implementación de esta función de pérdida basada en las cualidades psicoacústicas del oído humano se realiza una adaptación del algoritmo *percpetual evaluation of speech quiality* [21] (PESQ), que se conoce como una métrica de uso común para la evaluación de la calidad de las señales de voz, como una función de pérdida para el entrenamiento de las redes neuronales. El PESQ es una métrica muy utilizada en sistemas de comunicación para medir la calidad del habla en los mismos. Es una métrica objetiva que compara una señal de referencia, supuesta inalterada y sin distorsión, con una señal que ha sido degradada, midiendo el impacto que las distorsiones tienen en la calidad percibida.

Con el fin de tener en cuenta las características perceptuales mencionadas en las líneas anteriores, modificamos la función de pérdida del *Embedder* añadiendo un término de perturbación basado en el algoritmo PESQ: una perturbarción simétrica y otra asimétrica, ambas calculadas en una base de *frame-by-frame*. La perturbación simétrica, a la que llamaremos $\mathbf{D_t^{(s)}}$, considera la diferencia absoluta entre los espectros de sonoridad procesados y limpios cuando se tienen en cuenta los efectos del enmascaramiento auditivo. Por otra parte, la perturbación asimétrica, a la que llamaremos $\mathbf{D_t^{(a)}}$, es calculada a partir de la perturbación simétrica pero ponderando de manera diferente las diferencias positivas y negativas de sonoridad. Este planteamiento se debe a que las diferencias negativas (componentes espectrales omitidas o atenuadas) son percibidas de manera diferente a las positivas (como el ruido aditivo) como consecuencia de los efectos provocados por el enmascaramiento.

3.5.1. Transformación al Dominio Perceptual

Las perturbaciones simétricas y asimétricas son calculadas en el dominio espectral de sonoridad, el cual es perceptualmente más próximo a la audición humana. En [19], los espectros de potencia se transforman en vectores a una escala Bark de frecuencia mediante una matriz de transformación a escala de

frecuencias Bark, previamente calculada y definida como $\mathbf{H}$, de la siguiente forma:

$$\mathbf{b_t} = \mathbf{H} \cdot \mathbf{x_t} \tag{3.6}$$

donde $\mathbf{b_t} = [\mathbf{B_{t,0}}, \mathbf{B_{t,1}}, ..., \mathbf{B_{t,Q-1}}]^\mathbf{T}$ es el espectro Bark con un total de Q bandas Bark y $\mathbf{x_t} = [|\mathbf{X_{t,0}}|^\mathbf{2}, |\mathbf{X_{t,1}}|^\mathbf{2}, ..., |\mathbf{X_{t,F-1}}|^\mathbf{2}]^\mathbf{T}$ siendo $\mathbf{X_{t,i}}$ la transformada en el dominio de la frecuencia de las señales. Tras esto, aplicamos la ley de Ziwcker para transformar cada banda del espectro Bark a una escala de sonoridad en sonios de la siguiente manera:

$$S_{t,q} = s_t \cdot \left(\frac{P_0(q)}{0.5}\right)^\gamma \cdot \left[\left(0.5 + 0.5 \cdot \frac{B_{t,q}}{P_0(q)}\right)^\gamma - 1\right] \tag{3.7}$$

donde s_t es un factor de escala de sonoridad, $P_0(q)$ es el umbral absoluto de audición para la banda Bark q-ésima y γ es fijado a 0.23, es decir, audición normal. Finalmente, aquellas bandas que están por debajo de $P_0(q)$ en cuanto a su nivel de sonoridad son fijadas a cero ya que no pueden ser percibidas por los humanos.

Estas transformaciones vectoriales son aplicadas tanto a los espectros de potencia objetivo como a los mejorados, $\mathbf{x_t}$ y $\hat{\mathbf{x_t}}$, obteniendo de esta forma un espectro de sonoridad objetivo y otro mejorado, $\mathbf{s_t} = [\mathbf{S_{t,0}}, \mathbf{S_{t,1}}, ..., \mathbf{S_{t,Q-1}}]^\mathbf{T}$ y $\hat{\mathbf{s_t}} = [\hat{\mathbf{S}}_{\mathbf{t,0}}, \hat{\mathbf{S}}_{\mathbf{t,1}}, ..., \hat{\mathbf{S}}_{\mathbf{t,Q-1}}]^\mathbf{T}$ respectivamente.

3.5.2. Cálculo de las Perturbaciones Simétrica y Asimétrica

Aquí se muestra la simplificación presentada en [19] del cálculo del vector simétrico de perturbaciones propuesto en el algoritmo que calcula el PESQ a través de la aplicación de un recorte central sobre la diferencia absoluta entre los espectros de sonoridad, descrito de la siguiente manera:

$$d_t^{(s)} = max(|\hat{\mathbf{s_t}} - \mathbf{s_t}| - \mathbf{m_t}, 0) \tag{3.8}$$

donde $\mathbf{m_t}$ es el factor de recorte, descrito a través de la siguiente expresión:

$$\mathbf{m_t} = 0.25 \cdot min(\hat{\mathbf{s_t}}, \mathbf{s_t}) \tag{3.9}$$

Podemos obtener el vector de perturbación asimétrica simplemente haciendo $d_t^{(a)} = d_t^{(s)} \odot \mathbf{r_t}$, donde $\odot$ corresponde a una multiplicación elemento a elemento y $\mathbf{r_t}$ es un vector de razones de asimetría cuyos componentes son calculados a partir del espectro Bark de la siguiente forma:

$$R_{t,q} = \left(\frac{\hat{B}_{t,q} + \epsilon}{B_{t,q} + \epsilon}\right)^\lambda \tag{3.10}$$

La razón de asimetría tiene en cuenta las diferencias positivas y negativas entre los espectros mejorado y objetivo en el dominio transformado Bark, aplicando un aumento o una atenuación a la perturbación simétrica, respectivamente. Las constantes ϵ y λ son fijadas a 50 y 1.2 respectivamente (el porqué de fijar dichos parámetros a estos valores se explicará en la subsección posterior).

Finalmente, podemos obtener los términos de perturbación simétrica y asimétrica de forma vectorizada para cada frame de la siguiente forma:

$$\mathbf{D_t^{(s)}} = ||\mathbf{w}||_1^{1/2} \cdot ||\mathbf{w} \odot d_t^{(s)}||_2 \tag{3.11}$$

$$\mathbf{D_t^{(a)}} = ||\mathbf{w} \odot d_t^{(a)}||_1 = \mathbf{w^T} \cdot d_t^{(a)} \tag{3.12}$$

donde $\mathbf{w}$ es un vector rellenado con pesos proporcionales a las bandas Bark, obtenidos de [20].

3.5.3. Preprocesado Espectral y Ecualización

Con el objetivo de reutilizar las constantes perceptuales y valores prefijados en el estándar del PESQ ([20]), es necesario aplicar un paso de preprocesado sobre $\mathbf{x_t}$ y $\hat{\mathbf{x}}_\mathbf{t}$ con el fin de obtener un espectro PESQ equivalente. Así, en el algoritmo PESQ, antes de realizar la transformación al dominio Bark, el nivel sonoro de ambas señales se iguala a un nivel de audición estándar. Para esto, los niveles de ganancia son calculados a partir de los valores RMS estimados de las señales de voz filtradas, en la banda de 350 hasta 3250 Hz. En [19], dado que las señales en el dominio del tiempo no están disponibles, la normalización de la ganancia se realiza en el dominio espectral de la siguiente manera:

$$\bar{\mathbf{x}}_\mathbf{t} = \mathbf{x_t} \cdot \frac{P_c}{\frac{1}{T}\sum_t (\mathbf{g^T} \cdot \mathbf{x_t})} \tag{3.13}$$

donde $\mathbf{g}$ es una máscara de ponderación espectral que imita a un filtrado paso-banda, mientras que P_c es un factor de corrección de potencia que tiene en cuenta la longitud del frame, la superposición y el uso de ventanas aplicado durante el cálculo espectral.

Adicionalmente, el algoritmo presenta una ecualización frecuencial para paliar los pequeños efectos de filtrado constante, y una ecualización de ganancia para corregir las variaciones de ganancia a corto plazo. Este factor de ganancia está acotado por el rango $[3 \cdot 10^{-4}, 5]$ y suavizado a lo largo del tiempo. En la propuesta descrita en [19], la ecualización en la frecuencia se realiza sobre el espectro degradado, $\hat{\mathbf{s}}_\mathbf{t}$, en lugar de realizarlo sobre

el espectro de referencia, $\mathbf{s_t}$, para evitar modificaciones sobre los espectros de referencia que puedan provocar un mal funcionamiento de la función de pérdida durante el entrenamiento. A su vez, el factor de ganancia solo se limita al intervalo expuesto, pero no se suaviza a lo largo del tiempo, ya que los términos de perturbación calculados en [19] están diseñados para calcularlos *frame-by-frame*.

3.5.4. Implementación de la Función de Pérdida Basada en el PESQ al Entrenamiento del Sistema

Para garantizar la convergencia de las redes neuronales en nuestro sistema, emplearemos una estrategia análoga a la descrita en [1]. En dicho trabajo, se asignan pesos específicos a las funciones de pérdida del *Embedder* y del *Detector*, lo cual es esencial para asegurar una adecuada optimización de ambas redes. En nuestro caso, adoptaremos una aproximación similar al ajustar la función de pérdida del *Embedder*. Integraremos una combinación ponderada que incluirá el Mean Absolute Error (MAE) y una función de pérdida basada en la calidad perceptual, conocida como PESQ loss. Este esquema de ponderación es crucial para mantener la convergencia del sistema y optimizar el rendimiento en términos de fidelidad y percepción auditiva. Una primera aproximación de la pérdida asociada a la reconstrucción de la señal de voz sería:

$$L_{Embedder} = L_{MAE}(output, input) + W_{PESQ} \cdot L_{PESQ}(output, input) \quad (3.14)$$

donde W_{PESQ} es el peso asignado a la pérdida calculada con la función basada en la calidad perceptual descrita en [19], L_{MAE} es la pérdida calculada a través del MAE y L_{PESQ} es la pérdida basada en el PESQ calculada a través de la función de pérdida descrita en [19].

El peso asignado a la función de pérdida basada en el PESQ a lo largo del entrenamiento quedan reflejados en la figura 3.5.

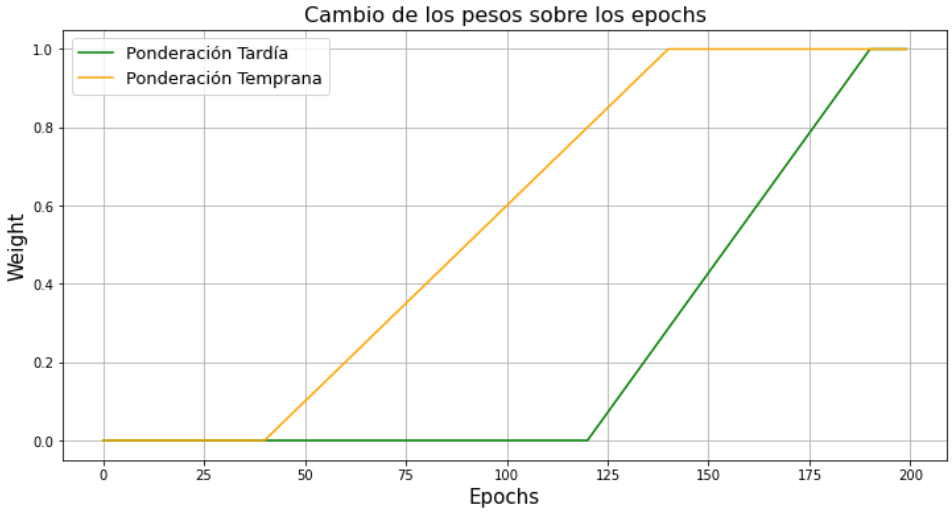


Figura 3.5: Evolución de los pesos W_{PESQ} a lo largo del entrenamiento.

Siguiendo un enfoque similar al propuesto en [1], garantizamos la convergencia de los componentes *Embedder* y *Detector* de nuestro modelo. Durante las fases iniciales del entrenamiento, el sistema respeta el diseño original al no incorporar la contribución de la función de pérdida basada en la calidad perceptual (PESQ). En esta etapa, la pérdida del *Embedder* se concentra exclusivamente en el Mean Absolute Error (MAE), que mide la diferencia media absoluta entre los valores reales y los predichos, proporcionando una cuantificación objetiva de la distorsión inducida por la inserción de la marca de agua en la señal reconstruida. A medida que el *Detector* alcanza la capacidad de detectar la marca de agua de manera efectiva, la función de pérdida basada en PESQ empieza a desempeñar un papel más significativo. Este cambio permite una reconstrucción más precisa de la señal de voz, lo que resulta en una mejora notable en métricas clave como la Relación Señal-Ruido (SNR), la Relación Señal-Distorsión (SDR), la calidad percibida (PESQ) y la Inteligibilidad Objetiva a Corto Plazo (STOI), superando así las prestaciones del sistema original descrito en [1].

Como puede notarse en la figura 3.5 tenemos dos escenarios posibles de ponderación en la adición de la pérdida basada en el PESQ: una ponderación temprana y una ponderación tardía. La diferencia esencial entre estos escenarios es la época en la que la pérdida basada en el PESQ comienza a tomar relevancia en el entrenamiento. En el primer escenario, al que hemos llamado "ponderación tardía", la pérdida basada en el PESQ comienza a tomar protagonismo hacia el final del entrenamiento, garantizando así una mejora en la reconstrucción de la señal. En el segundo escenario, al que hemos llamado "ponderación temprana", la pérdida basada en el PESQ comienza a

aparecer en una fase temprana del entrenamiento, siendo su presencia mayor durante toda esta fase. Los resultados obtenidos respecto al BER y la calidad de la señal reconstruida se estudiarán en el capítulo 5.

En la figura 3.6 podemos ver la evolución de cada una de las pérdidas a lo largo del entrenamiento considerando la aproximación dada por la ecuación 3.14 para el esquema de ponderación tardía. Podemos observar como tanto la pérdida MAE como la PESQ van decreciendo con el pasar de las épocas de entrenamiento.

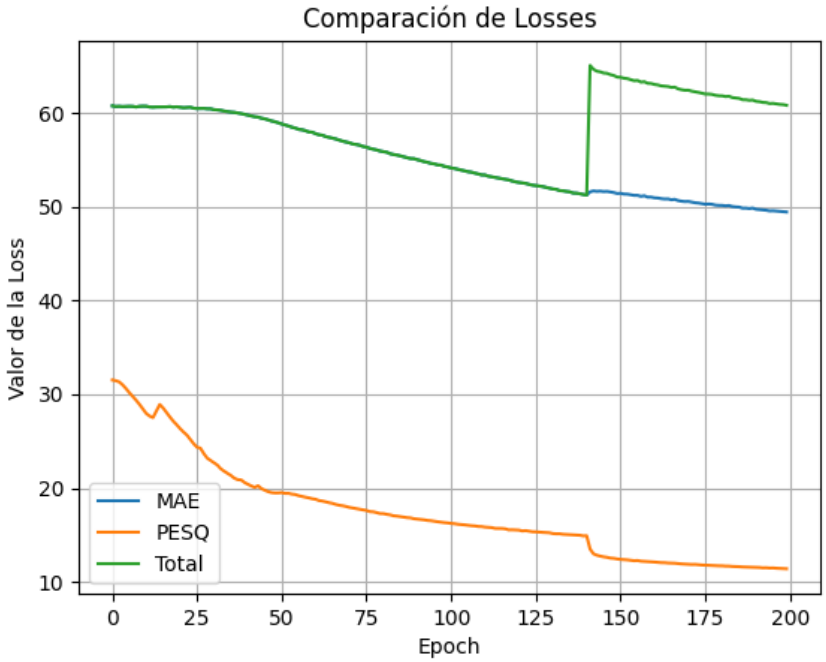


Figura 3.6: Comparativa de las pérdidas MAE, PESQ y Total

Tanto el MAE como la pérdida basada en el PESQ presentan una tónica decreciente a medida que aumentan las épocas del entrenamiento, lo que nos sugiere que el *Embedder* está cometiendo menos errores a la hora de reconstruir la señal. A su vez, podemos observar como la curva de la *Loss Total* coincide con la curva del MAE hasta la centésima vigésima época, en la que comienza a aparecer la pérdida basada en el PESQ con su respectivo peso según el esquema de ponderación tardía. Como vemos, esta aparición supone un aumento del valor numérico de la pérdida total generada en la reconstrucción. Este aumento no implica una peor reconstrucción ya que, como podemos ver, tanto MAE como la pérdida basada en el PESQ siguen decreciendo de manera independiente conforme el entrenamiento avanza. El

aumento en el valor de la pérdida total es consecuencia única y exclusivamente de la adición de un nuevo término como sumando basado en la pérdida PESQ junto al MAE.

Por último, con el objetivo de no desestabilizar la pérdida del *Embedder* y que esta afecte a la percepción de la misma por parte del *Detector* se procede a la modificación de los pesos relativos entre la pérdida MAE y la pérdida PESQ para que el conjunto de ambos no modifique su peso respecto al *Detector*. Para ello realizamos una normalización de la siguiente forma:

$$L_{Embedder} = \left(\frac{L_{MAE}}{L_{MAEmax}} + W_{PESQ} \frac{L_{PESQ}}{L_{PESQmax}} \right) \cdot \frac{L_{MAEmax}}{2} \tag{3.15}$$

donde L_{MAE} es la pérdida cometida por el MAE, L_{MAEmax} es la pérdida máxima cometida por el MAE, L_{PESQ} es la pérdida cometida por la función de pérdida basada en el PESQ, $L_{PESQmax}$ es la pérdida máxima cometida por la función de pérdida basada en el PESQ y W_{PESQ} es el peso asignado. Con este planteamiento conseguimos una convergencia equilibrada sin que el *Detector* altere significativamente sus resultados como consecuencia de la adición de un nuevo término a la pérdida cometida en la reconstrucción de la señal de voz.

Capítulo 4

Ataques implementados en el sistema.

En este capítulo nos centraremos en una de las facetas fundamentales de los sistemas de *watermarking*: la robustez. Nuestro sistema de *watermarking* no solo sirve para incrustar el mensaje en la señal portadora con propósitos de autenticación y protección de los derechos de autor del contenido multimedia, sino que debe de ser capaz de resistir y recuperarse de ataques malitencionados que tienen como propósito poner en entredicho el funcionamiento de nuestro sistema y eliminar la marca de agua. La integridad del sistema de *watermarking* se evalúa a través de la robustez del mismo ante estos ataques. Por este motivo, el entrenamiento de las redes neuronales incluyendo estos ataques es primordial para garantizar la solidez del sistema propuesto [1].

Durante el desarrollo de este capítulo, presentaremos y describiremos los ataques específicos con los que entrenaremos a nuestras redes neuronales *Embedder* y *Detector* para evaluar la fortaleza de nuestro sistema. Los ataques implementados son el filtrado paso-baja de la señal, la adición de ruido aleatorio a la señal y la supresión aleatoria de muestras de la señal. Estos ataques son descritos en las secciones 4.1, 4.2 y 4.3 respectivamente y representan una variedad de escenarios potenciales en los que la marca de agua podría verse seriamente comprometida.

El análisis de la respuesta del sistema ante estos ataques no solo es vital para confirmar lo viable que resulta nuestra arquitectura sino que también puede resultar útil para identificar áreas de mejora.

4.1. Ataque de filtrado paso-baja

Para la implementación de este ataque, los filtros de Butterworth son usados como un ejemplo de filtros paso-baja. Este tipo de filtros son comúnmente utilizados como ataques en sistemas de *watermarking* porque pueden borrar la marca de agua y provocar que el sistema no logre su detección de manera satisfactoria. En la mayoría de los casos y dadas las características fisiológicas del sistema auditivo humano, las marcas de agua en las señales de voz son incrustadas en las altas frecuencias debido a que los cambios en estas frecuencias son aproximadamente imperceptibles para el oído humano. Sin embargo, cuando atacamos la señal que contiene la marca de agua incrustada en las altas frecuencias con un filtro paso-baja, la marca de agua puede quedar completamente eliminada. Por esta razón, es importante diseñar nuestro sistema de *watermarking* de tal forma que sea resistente a este tipo de filtrados malintencionados.

Los filtros paso-baja de Butterworth son un tipo de filtros que son utilizados para minimizar el rizado en la respuesta en frecuencia del filtro. También se les conoce como filtros de respuesta máximamente plana dado que tienen el factor de *roll-off* más angosto posible sin causar rizado en otras partes del espectro de la respuesta en frecuencia. La amplitud de la respuesta en frecuencia puede definirse a través de la siguiente expresión [1]:

$$|H(j\omega)| = \frac{1}{\sqrt{1 + (\frac{\omega}{\omega_c})^{2N}}} \tag{4.1}$$

donde ω_c es la frecuencia de corte del filtro y N es el orden del mismo. Para este estudio y como en [1] se diseña un filtro paso-baja de decimosexto orden con una frecuencia de corte de 4 kHz. Esta frecuencia de corte es escogida con el objetivo de no provocar alteraciones notables en las señales de voz. Podemos ver su respuesta en frecuencia en la figura 4.1.

Para implementar este filtro previamente necesitamos calcular su respuesta al impulso. La respuesta al impulso de un filtro de Butterworth puede calcularse a través de la siguiente ecuación:

$$h(t) = \sum_{k=1}^{N} r_k e^{s_k t} u(t) \tag{4.2}$$

donde s_k son los polos de la respuesta en frecuencia, $u(t)$ es la función escalón unitario y r_k son los coeficientes obtenidos tras la descomposición parcial de la función de transferencia $H(s)$ (siendo $s = j\omega$).

Tras calcular la respuesta al impulso, podemos aplicar el filtro a la señal de voz con la marca de agua incrustada simplemente realizando la convolución de las dos funciones.

$$y(t) = x(t) * h(t) \tag{4.3}$$

que en el dominio discreto del tiempo la convolución puede expresarse como:

$$y(n) = \sum_{m=-\infty}^{\infty} x(m)h(n-m) \tag{4.4}$$

Cabe destacar que la aplicación del filtro puede ser implementada como una capa convolucional no-entrenable de las redes neuronales propuestas para el *Embedder* y el *Detector*. A su vez, las derivadas parciales respecto a la entrada que son necesarias para retropropagar el error a través de estas capas se calculan como:

$$\frac{\partial y(n)}{\partial x(m)} = h(n-m) \tag{4.5}$$

y la derivada de la función de pérdida E respecto a la señal de entrada, aplicando la regla de la cadena queda como:

$$\frac{\partial E}{\partial x(m)} = \sum_{n=0}^{L} \frac{\partial E}{\partial y(n)} \frac{\partial y(n)}{\partial x(m)} = \sum_{n=0}^{L} \frac{\partial E}{\partial y(n)} h(n-m) \tag{4.6}$$

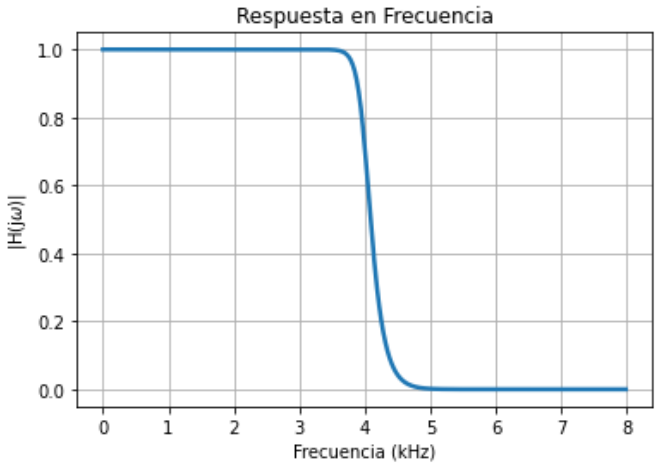


Figura 4.1: Respuesta en frencuencia del filtro de Butterworth. Orden 16 y $f_c = 4kHz$

Figura 4.2: Respuesta al impulso del filtro de Butterworth. Orden 16 y $f_c = 4kHz$

El motivo que hay detrás de escoger un filtro de Butterworth con una frecuencia de corte de 4 kHz se debe a que una frecuencia de corte más baja puede causar alteraciones audibles en las señales de voz. A su vez, un filtro paso-baja de Butterworth con una frecuencia de corte de 4 kHz es suficiente para poner en compromiso la detección de la marca de agua por parte del *Detector* [1].

4.2. Ataque de adición de ruido aleatorio

El ataque más común en los sistemas de marcado de agua es la adición de ruido a la señal que contiene la marca de agua. Cabe destacar que este ataque cubre todo los tipos de ataques que están basados en técnicas de compresión con pérdidas. La adición de ruido trata de destruir la marca de agua incrustada en la señal de voz portadora, haciendo que la tarea de detección se vuelva bastante más complicada o, incluso, imposible. Para la implementación del ataque de adición de ruido se ha escogido un modelo de ruido procedente de [22]:

$$y(n) = x(n) + \alpha\epsilon(n) \tag{4.7}$$

donde $x(n)$ representa a la señal de entrada, $\epsilon(n)$ a la señal del ruido e $y(n)$ la señal de salida tras haber sido atacada con el ruido aleatorio. El parámetro α nos permite controlar la potencia del ruido añadido a la señal $x(n)$. En [1], el valor seleccionado para este parámetro es de $\alpha = 0.009$. Con este valor, el ruido no es perceptualmente audible por los usuarios pero si tiene la suficiente potencia como para hacer que la tarea de detección de la marca

de agua se complique. En [1] eligen este valor porque sus señales poseen un rango dinámico de ±1. Dado que el rango dinámico de nuestra señal es mayor, aproximadamente ±13000, el nivel de degradación introducido ha de ser mayor para alcanzar un ruido de 30 dB. Podemos calcularlo a través de la siguiente expresión:

$$\alpha = \sqrt{\frac{\sigma}{10^{30/10}}} \tag{4.8}$$

donde σ es la potencia promedio de las señales utilizadas. Con esta expresión se obtiene que el nivel de degradación α introducido es de 16.

De la misma forma que el ataque de filtrado paso-baja, el ataque de adición de ruido aleatorio puede implementarse como una capa no-entrenable de las redes neuronales. Esto implica que solo el gradiente respecto a la señal de entrada $x(m)$ debe de ser retropropagado a las capas predecesoras durante el entrenamiento de las redes neuronales. Las derivadas parciales de la señal de salida $y(n)$ respecto a la señal de entrada $x(m)$ pueden expresarse como:

$$\frac{\partial y(n)}{\partial x(m)} = \begin{cases} 1 & \text{si } n = m \\ 0 & \text{si } n \neq m \end{cases} \tag{4.9}$$

y, por la regla de la cadena, podemos calcular la porción total de error E que se propaga hacia la entrada $x(m)$ como:

$$\frac{\partial E}{\partial x(m)} = \sum_{n=0}^{L} \frac{\partial E}{\partial y(n)} \frac{\partial y(n)}{\partial x(m)} = \frac{\partial E}{\partial y(m)} \tag{4.10}$$

donde L es la longitud total de la señal de entrada.

4.3. Ataque de supresión aleatoria de muestras

Con el ataque de supresión aleatoria de muestras lo que hacemos es poner a un conjunto de muestras seleccionadas aleatoriamente a 0. A pesar de que este ataque tiene un modelo sumamente sencillo, demuestra ser un tanto problemático para los sistemas de *watermarking* existentes. Para implementar este ataque haremos uso de una máscara de unos y ceros generada de manera aleatoria. Podemos describir este ataque a través de la siguiente expresión:

$$y(n) = mask(n) \cdot x(n) \tag{4.11}$$

Esta máscara generada aleatoriamente determina cuales son las muestras que van a ser suprimidas. Las derivadas parciales respecto a la entrada x

pueden expresarse como:

$$\frac{\partial y(n)}{\partial x(m)} = \begin{cases} mask(n) & \text{si } n = m \\ 0 & \text{si } n \neq m \end{cases} \tag{4.12}$$

y, aplicando la regla de la cadena, obtenemos el error E:

$$\frac{\partial E}{\partial x(m)} = \sum_{n=0}^{L} \frac{\partial E}{\partial y(n)} \frac{\partial y(n)}{\partial x(m)} = \frac{\partial E}{\partial y(m)} \cdot mask(n) \tag{4.13}$$

Con el fin de preservar la calidad de la señal reconstruida por el *Embedder* y al mismo tiempo hacer que el ataque sea efectivo y dificulte la tarea del *Detector*, el número de muestras suprimidas aleatoriamente será de 1000 para cada señal de 32768 muestras (que muestreadas a 16 kHz, suponen señales de aproximadamente 2 segundos de duración) lo que representa aproximadamente el 3 % de la señal de entrada.

Capítulo 5

Marco y Resultados Experimentales

5.1. Dataset

En el ámbito de la evaluación del funcionamiento de las redes neuronales no existe un *dataset* estandarizado que nos asegure buenos resultados. Por ejemplo, en [23] se utilizan archivos del *dataset* SQAM, mientras que en [1] se hace uso de un conjunto de archivos de voz del Parlamento de Montenegro. Esta base de datos contiene 6199 archivos de audio de una duración de 10 minutos cada uno. En nuestro caso, el *dataset* utilizado es la base de datos $TIMIT$. Sin embargo, en este proyecto se hace uso de una porción de la base de datos $TIMIT$ para realizar el entrenamiento de nuestras redes neuronales, como realizan en [21].

La base de datos $TIMIT$ (Acoustic-Phonetic Continuous Speech Corpus) es una base de datos muy influyente en el campo del procesamiento del habla. Esta base de datos fue desarrollada conjuntamente por el *Instituto de Tecnología de Massachusetts*, el SRI internacional y *Texas Instruments* en el año 1986. Aunque nosotros estamos utilizando esta base de datos con propósitos diferentes, en su origen esta fue diseñada para proporcionar datos acústico-fonéticos para el desarrollo y evaluación de algoritmos de reconocimiento automático del habla [5].

Esta base de datos, aunque nosotros no hagamos uso de la misma al completo, contiene grabaciones de voz de 630 hablantes de las ocho grandes regiones dialectales de los Estados Unidos. Cada oyente leyó 10 oraciones, lo que nos da un total de 6300 oraciones contenidas cada una de ellas en un archivo de audio con una duración aproximada de 7 a 10 segundos. La

selección de oraciones está diseñada para para cubrir todos los fonemas del inglés estadounidense en diferentes contextos coarticulatorios. Aunque el ámbito de uso de esta base de datos en sus inicios era las investigaciones en el reconocimiento del habla, actualmente es utilizada en otras áreas del procesamiento del habla, inlcuyendo la síntesis del habla y el estudio de la fonética, así como el aprendizaje automático, el *machine learning* y la inteligencia artifical.

La base de datos $TIMIT$ ha sido esencial para el avance del campo del reconocimiento automático del habla. Ha servido como un estándar de facto para la evaluación de los sistemas relacionados con el campo de las tecnologías del habla.

Como ya se mencionaba anteriormente, para el entrenamiento y test de nuestro sistema no hacemos uso de la base de datos al completo. Los archivos de audio contenidos en esta base de datos están muestreados con una frecuencia de muestreo de 16 kHz. Esta frecuencia de muestreo es escogida en función de las características del espectro de frecuencia audible por el humano. Las frecuencias audibles por el humano van desde los 20 Hz hasta los 20 kHz pero la frecuencia de la voz humana como máximo alcanza un valor de 8 kHz en algunas personas. Por esta razón, podemos muestrear las señales de audio y, aún así, preservar la calidad de las mismas. Según el teorema de Nyquist-Shannon debemos de escoger una frecuencia de muestreo al menos dos veces superior al máximo de la frecuencia. Por tanto, y como se mencionaba anteriormente, la frecuencia de muestreo de los archivos de audio es de 16 kHz.

En los procesos de entrenamiento, las señales de audio se cortan en segmentos de 32768 muestras. Con una frecuencia de muestreo de 16 kHz esto equivale a segmentos de aproximadamente 2.044 segundos. Este tamaño de segmento se elige principalmente por la longitud de la marca de agua. Si la marca de agua es muy corta, el sistema de incrustación (*Embedder*) podría aprender a eliminarla mientras intenta reconstruir la señal de audio con la menor distorsión posible. Por otro lado, si la marca de agua es demasiado larga, comparable en tamaño al segmento, podría introducir demasiado ruido y distorsión en la señal de voz, impidiendo que el *Embedder* funcione correctamente y reconstruya la señal de manera eficiente. Además, es importante que la longitud de la marca de agua sea similar a la longitud de la ventana que usamos para calcular la transformada de Fourier de tiempo corto (STFT) de las señales de voz, para asegurar que ambos procesos estén alineados en términos de escala [1].

5.2. Métricas de Evaluación

En la evaluación de los sistemas de marcado de agua es de suma importancia medir la calidad e inteligibilidad de las señales de voz reconstruidas así como el número de bits erróneos obtenidos en la marca de agua reconstruida. Para ello, utilizamos una serie de métricas que nos permiten cuantificar los diferentes aspectos del rendimiento del sistema. En esta sección se procede a la descripción de las métricas de evaluación utilizadas en el análisis del rendimiento del sistema:

- **SNR:** Mide la relación entre la potencia de la señal deseada y la potencia del ruido de fondo. Una SNR alta indica una señal más clara con menos ruido de fondo.

$$SNR[dB] = 10 \cdot log\left(\frac{P_{señal}}{P_{señal} - P_{reconstruida}}\right)$$

  donde $P_{señal}$ es la potencia de la señal original y $P_{reconstruida}$ es la potencia de la señal reconstruida por el *Embedder*.

- **SDR:** Cuantifica la relación entre la señal deseada y la distorsión introducida por el procesamiento [24]. Viene definida por la siguiente formula:

$$SDR[dB] = 10 \cdot log\left(\frac{||\mathbf{s_{target}}||^2}{||\mathbf{e_{res}}||^2}\right)$$

  con:

  - $\mathbf{s_{target}} = \alpha \cdot \mathbf{s}$
  - $\alpha = \frac{\hat{\mathbf{s}}^\mathbf{T}\mathbf{s}}{||\mathbf{s}||^2}$
  - $\mathbf{e_{res}} = \hat{\mathbf{s}} - \mathbf{s_{target}}$

  siendo $\mathbf{s}$ el vector que representa a la señal original, $\hat{\mathbf{s}}$ el vector que representa a la señal estimada por el *Embedder*, $\mathbf{s_{target}}$ el vector que representa la proyección estimada en la línea definida por la señal original y $\mathbf{e_{res}}$ el error residual ortogonal a la señal objetivo.

- **PESQ:** Se trata de un estándar internacional de la ITU ([20]) para evaluar la calidad perceptual del habla en sistemas de comunicación. Esta métrica compara la señal original con la procesada y proporciona una puntuación que refleja cómo percibirá un usuario la calidad de la señal. Las puntuaciones fluctúan entre -0.5 en el peor de los casos y 4.5 en el mejor.

- **STOI:** Mide la inteligibilidad de la señal, es decir, lo fácil que resulta entender el contenido de la señal. Proporciona un valor entre 0 y 1, donde 1 indica perfecta inteligibilidad.
- **BER:** Mide la proporción de bits que han sido alterados respecto a una señal binaria original. Un BER bajo nos indica una transmisión fiable.

5.3. Rendimiento del sistema en la fase de entrenamiento

En esta sección, presentaremos y analizaremos detalladamente los resultados obtenidos durante el entrenamiento del sistema compuesto por las redes neuronales *Embedder* y *Detector*. La efectividad del sistema se evaluará a través de su desempeño en diversas fases de entrenamiento, incluyendo las curvas de aprendizaje, que comprenden tanto las métricas de pérdida como la Tasa de Error de Bit (BER) como la pérdida obtenida en la reconstrucción de la señal de voz definida como el Mean Absolute Error (MAE). Además, examinaremos los patrones de convergencia de estas métricas en distintos contextos experimentales, utilizando variados niveles de probabilidades de ataque (0 %,10 %, 20 % y 30 %). Con probabilidad de ataque nos referimos a la probabilidad que existe de que la señal reconstruida por el *Embedder*, en la que va incluida la marca de agua, una vez sale del *Embedder* y pasa por un canal, sea asaltada por una cierta cantidad de ataques con el fin de eliminar la marca de agua. La cantidad de ataques es generada aleatoriamente, por lo que un mayor porcentaje de probabilidad de ataques implica una mayor cantidad de los mismos. Se eligen unos valores de 0 %, 10 %, 20 % y 30 % para el hiperparámetro de probabilidad de ataque con el objetivo de evaluar el rendimiento del sistema presentado bajo diferentes niveles de condiciones de degradación [1]. El análisis de los resultados en este apartado nos permite afirmar la validez del enfoque implementado para el conjunto de datos de entrenamiento.

5.3.1. Fase de Entrenamiento del Sistema Original

En este apartado se procede a la exposición de los resultados obtenidos durante el proceso de entrenamiento del sistema propuesto en [1].

Tanto la curva del Bit Error Rate por parte del *Detector* como la curva de pérdidas del *Embedder* nos permitirán visualizar de manera gráfica el proceso del entrenamiento del sistema y la convergencia de cada una de las redes neuronales.

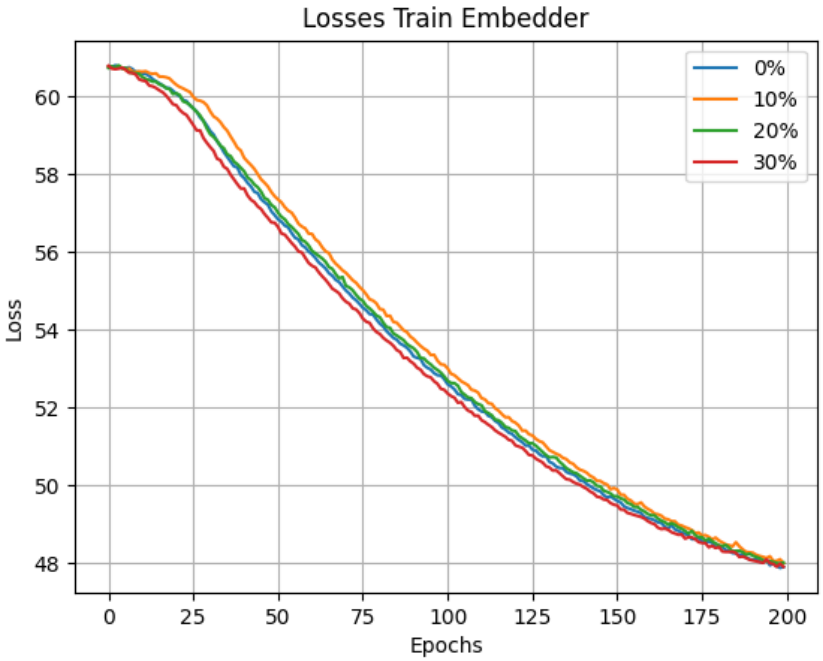


Figura 5.1: Curva de las pérdidas del Embedder en el proceso de entrenamiento.

En la curva mostrada en la figura 5.1, calculada a partir del MAE entre la entrada y la salida del *Embedder* podemos observar una tendencia a la baja a medida que avanzamos en las épocas del entrenamiento y sobre el conjunto de entrenamiento. Esta tendencia a la baja significa que el MAE está decreciendo, es decir, que el *Embedder* está cometiendo menos errores de cara a reconstruir la señal de entrada y que, en consecuencia, está aprendiendo a reconstruirla a través de la retropropagación de los gradientes.

Como vemos en la figura 5.1, en todos los escenarios de nivel de degradación posible (0 %, 10 %, 20 % y 30 %), la señal reconstruida aparenta estar igual de bien predicha en todos los escenarios, sin embargo, si nos fijamos bien, en el escenario de 30 % de degradación la señal se reconstruye ligeramente mejor respecto al resto de niveles de degradación. Resulta algo paradójico que este sea el resultado obtenido, pero es lo que se espera de este escenario. Cuando el nivel de degradación es más elevado la convergencia del *Detector* se complica en cierta medida. Esto es aprovechado por parte del *Embedder* para garantizar su convergencia, arrojando resultados ligeramente mejores que el resto de niveles de degradación.

En cuanto a la convergencia del *Detector*, podemos ver en la figura 5.2 el BER medido sobre el conjunto de entrenamiento respecto a la marca de agua originalmente incrustada en la señal de voz portadora en el *Embedder*. Puede notarse que la convergencia es relativamente rápida en todos los casos de degradación posible, retrasándose un poco la caída del BER a medida que aumenta el nivel de degradación. A su vez, observando la figura 5.2 podemos notar que los que mejores resultados en cuanto al BER vienen dados por los escenarios del 0 % de ataques (es lo normal, pues no hay degradación) y el escenario de 30 % de ataques. De la misma forma que para el caso de la convergencia del *Embedder*, resulta algo contraintuitivo que la mayor robustez frente a la degradación de la señal venga dada por parte del sistema que más degradación presenta. Este fenómeno se debe a la naturaleza estocástica de la generación de ataques durante el entrenamiento. Los ataques son generados de manera aleatoria, lo que implica que, dada la naturaleza estocástica de la generación de los ataques, cabe la posibilidad de que el modelo que mejor resultados presente sea el modelo entrenado con un 30 % de ataques junto al que no presenta ningún ataque, es decir, el modelo entrenado con un 0 % de ataques.

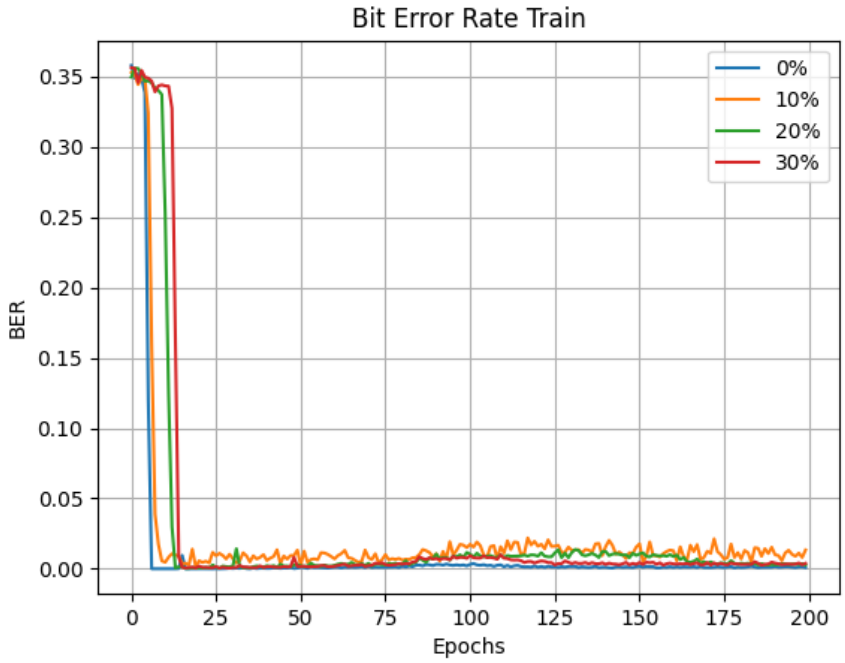


Figura 5.2: Bit Error Rate en tanto por uno del Detector en el proceso de entrenamiento.

5.3.2. Fase de Entrenamieto del Sistema Propio con Ponderación Tardía

En este enfoque, al que hemos llamado "ponderación tardía", se plantea la aparición de la pérdida basada en el PESQ hacia el final del entrenamiento, exactamente comienza a aparecer en la centésima vigésima época, aumentando su peso paulatinamente hasta llegar a la unidad en la centésima nonagésima época. Es posible analizar el avance del entrenamiento a través de dos resultados clave. El primero es la evolución de la función de pérdida del *Embedder*, que proporciona una representación directa del error de reconstrucción a lo largo del proceso de optimización. El segundo es el BER del *Detector*, que refleja la eficacia con la que el sistema detecta las marcas de agua incrustadas en las señales de voz que pasan por el *Embedder*. Ambos parámetros, visualizados en forma de curvas de pérdida, ofrecen una buena representación del rendimiento del modelo a lo largo del entrenamiento.

La curva de pérdidas del *Embedder* presenta una forma algo diferente a la presentada en el sistema original propuesto (figura 5.1). En este caso tenemos una primera decreciente en la que solo participa el MAE hasta la centésima vigésima época, en la que la pérdida basada en el PESQ comienza a aparecer, es por esta razón la subida en la curva de la pérdida de la señal de audio(figura 5.3).

Observando esta curva, vemos como se muestra una tónica decreciente de la pérdida del *Embedder* hasta la centésima vigésima época, lo que nos indica una mejora en la capacidad del *Embedder* para minimizar la MAE en el proceso de recosntrucción de la señal.

A partir de la centésima vigésima época podemos observar un marcado crecimiento como consecuencia directa de la aparición de la pérdida basada en el PESQ. Este incremento en la pérdida no implica necesariamente que la reconstrucción de la señal se esté deteriorando. En lugar de eso, esto sugiere que la incorporación de una métrica adicional como suplemento a el MAE con unas características distintas altera la naturaleza de la optimización. Dada la adición de un nuevo término a la pérdida de la reconstrucción de la señal, es normal observar un aumento en el valor numérico de la pérdida observada. Es normal que este aumento en el valor de la pérdida se dé ya que estamos sumando a el MAE la pérdida basada en el PESQ con un peso correspondiente al esquema con ponderación tardía (veáse figura 3.5).

Realizar una comparativa entre las curvas antes y después de la centésima vigésima epoch no es válida debido a que la aparición de la pérdida basada

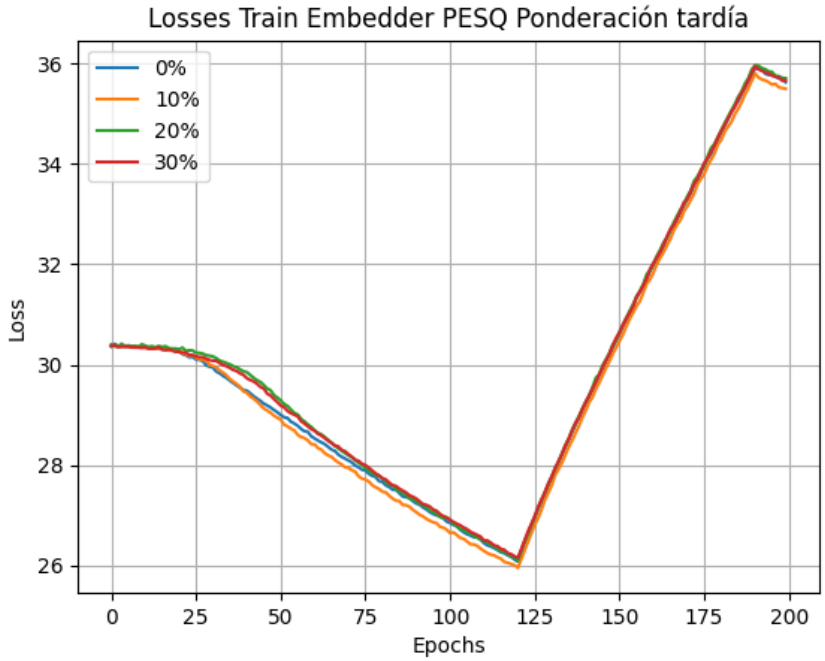


Figura 5.3: Curva de las pérdidas del Embedder en el proceso de entrenamiento. Ponderación Tardía.

en el PESQ con un peso que crece con las épocas produce un aumento del valor numérico de la pérdida pero no empeora la reconstrucción de la señal porque, como pudo verse en la figura 3.6 del capítulo 3, tanto el MAE como la pérdida basada en el PESQ decrecen de manera independiente conforme avanzan las épocas del entrenamiento.

Por último, cabe destacar que todas las curvas presentan un comportamiento similar a la hora de reconstruir la señal, no destacando ninguna por encima de la otra.

En relación con la Tasa de Error de Bits (BER) durante las etapas iniciales del entrenamiento, se observa que todas las curvas presentan valores elevados de BER. Sin embargo, estas rápidamente convergen hacia valores significativamente más bajos. Aunque las configuraciones con un 10 % y un 30 % de ataques con ponderación tardía de la pérdida PESQ muestran una convergencia más lenta, todos los modelos logran reducir el BER a niveles similares entre sí. Esto indica que, a pesar de las diferencias en la estabilización del rendimiento, todos los modelos alcanzan un grado similar de precisión en la detección de la marca de agua hacia el final del proceso de

entrenamiento, exceptuando al modelo entrenado con un 0 % de ataques, que mantiene el BER por debajo del 1 %. Es lo esperado de este modelo ya que no se presentan ataques durante su entrenamiento (figura 5.4).

En la fase final del entrenamiento, el modelo entrenado con un 20 % de ataques con un esquema ponderación tardía de la pérdida PESQ exhibe los resultados más desfavorables. Este fenómeno de nuevo es atribuido a la naturaleza estocástica del proceso de entrenamiento en la generación de ataques. Debido a la naturaleza estocástica del sistema, es posible que el número de ataques generados sea superior en un escenario caracterizado por un menor porcentaje de ataques. Esta situación podría comprometer significativamente los resultados obtenidos en términos del BER. Como consecuencia, se observaría una degradación en el desempeño del sistema, resultando en una disminución de la eficacia en la tarea de detección de la marca de agua.

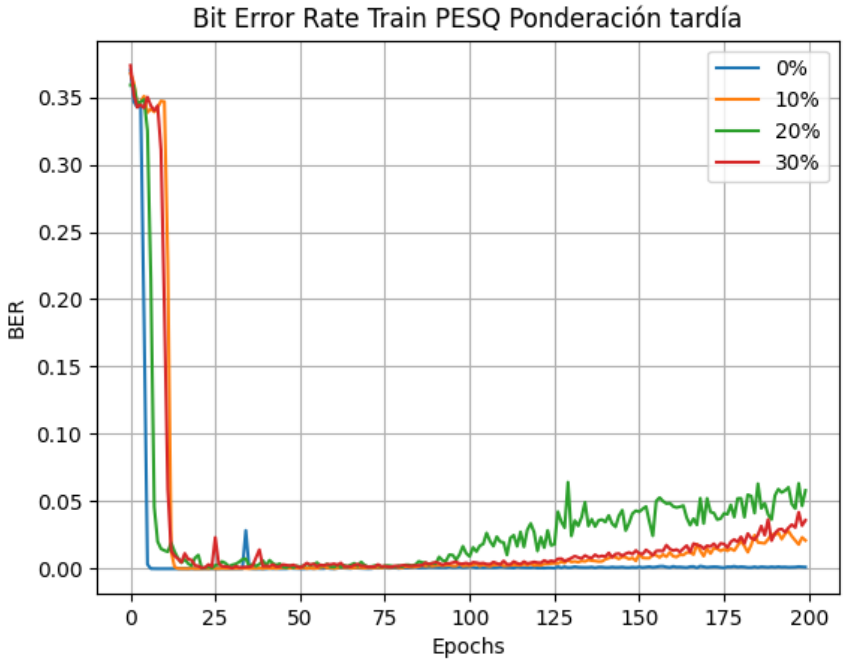


Figura 5.4: Bit Error Rate en tanto por uno del Detector en la Fase de Entrenamiento. Ponderación Tardía.

5.3.3. Fase de Entrenamiento del Sistema Propio con Ponderación Temprana

Similarmente a lo discutido previamente, esta sección detalla los resultados alcanzados durante la fase de entrenamiento de un modelo en el cual la pérdida basada en el PESQ se incorpora desde una etapa inicial del proceso. En este contexto, se analiza la integración temprana de la métrica PESQ como parte de la función de pérdida del *Embedder*, evaluando su influencia en la convergencia y rendimiento general del modelo a lo largo del entrenamiento.

La curva de las pérdidas del *Embedder* presenta un comportamiento bastante similar al presentado en el apartado anterior, con la salvedad de que, en este caso, el crecimiento en la pérdida debido a la pérdida PESQ comienza en la cuadragésima época (figura 5.5).

De nuevo, en este caso tenemos una parte decreciente que coincide con el decaimiento del MAE hasta la cuadragésima época en la que aparece la pérdida basada en el PESQ con su respectivo peso. El incremento sufrido en la pérdida del *Embedder* no significa que la reconstrucción de la señal se esté degradando, sino que nuevamente, la adición de este nuevo término a la pérdida total del *Embedder* produce un aumento en el valor numérico en la pérdida que refleja la aparición de la pérdida basada en el PESQ como sumando junto con el MAE. Después del pico, la pérdida disminuye nuevamente hasta el final del entrenamiento, estabilizándose en valores próximos al final del entrenamiento para todos los niveles de degradación. Este patrón podría indicar que el modelo ha logrado una adaptación eficaz a la combinación de ambas métricas de distinta naturaleza hacia el final del proceso, consolidando las mejoras en las métricas de evaluación de la reconstrucción de la señal (SNR, SDR, PESQ y STOI) sin sacrificar en gran medida la minimización de la pérdida original. Sin embargo, no se logra la convergencia total de los modelos debido a las limitaciones de cómputo que la plataforma Google Colab presenta.

A su vez, observando todas las curvas podemos notar que el pico es mucho menor en el modelo entrenado con una degradación del 30 %, lo que nos indica que la reconstrucción de la señal será mejor en este modelo. Esto es debido a la tardía convergencia del *Detector*, que es aprovechada por el *Embedder* para reconstruir mejor la señal.

Respecto al BER y la curva de aprendizaje del *Detector*, al principio del entrenamiento todos los modelos con sus respectivos niveles de degradación

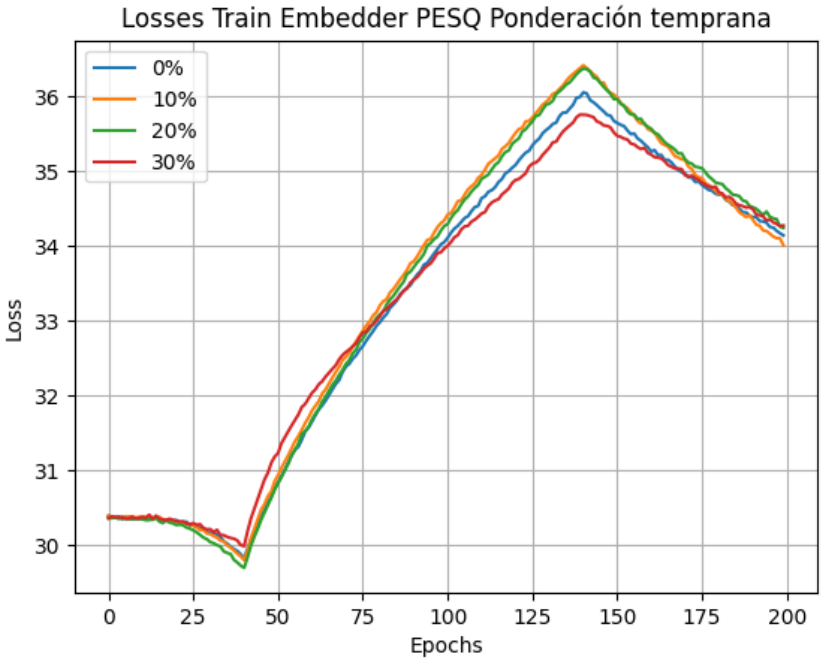


Figura 5.5: Curva de las pérdidas del Embedder en el proceso de entrenamiento. Ponderación temprana.

presentan altos niveles de BER, destacando algunos picos del modelo entrenado con 30 % de ataques. A partir de cierta época, todos los modelos convergen hacia valores de BER muy bajos. Este rápido descenso en el valor del BER nos indica que el modelo se adapta rápidamente y mejora notoriamente en la tarea de detectar los bits de la marca de agua. Una vez se ha realizado este descenso en el BER todos los modelos se estabilizan, presentando peores resultados el modelo entrenado con un 10 % de ataques, como consecuencia de la naturaleza estocástica en la generación de los ataques en el proceso de entrenamiento. Aunque resulta contraintuitivo que un modelo entrenado con menor probabilidad de ataques presente peores resultados en el BER, la naturaleza aleatoria en la generación de los ataques lo convierte en un escenario probable.

En cuanto a los modelos entrenados con un 20 % y un 30 % de probabilidad de ataques, ambos exhiben un BER bastante similar a lo largo del entrenamiento. Sin embargo, el modelo con una probabilidad del 30 % de ataques se puede considerar más robusto. Esto se debe a que ha sido expuesto a una mayor cantidad de ataques durante su fase de entrenamiento, lo que potencialmente le da una mejor capacidad de adaptación y resistencia

frente a perturbaciones similares en un escenario real. Dicha robustez adicional es de suma importancia en aquellos escenarios en los que la cantidad e intensidad de ataques puedan variar a lo largo del tiempo, haciendo que el modelo entrenado con un 30 % de ataques tenga mayor estabilidad respecto a los demás, dotándolo de una fiabilidad superior en la reconstrucción de la marca de agua (figura 5.6).

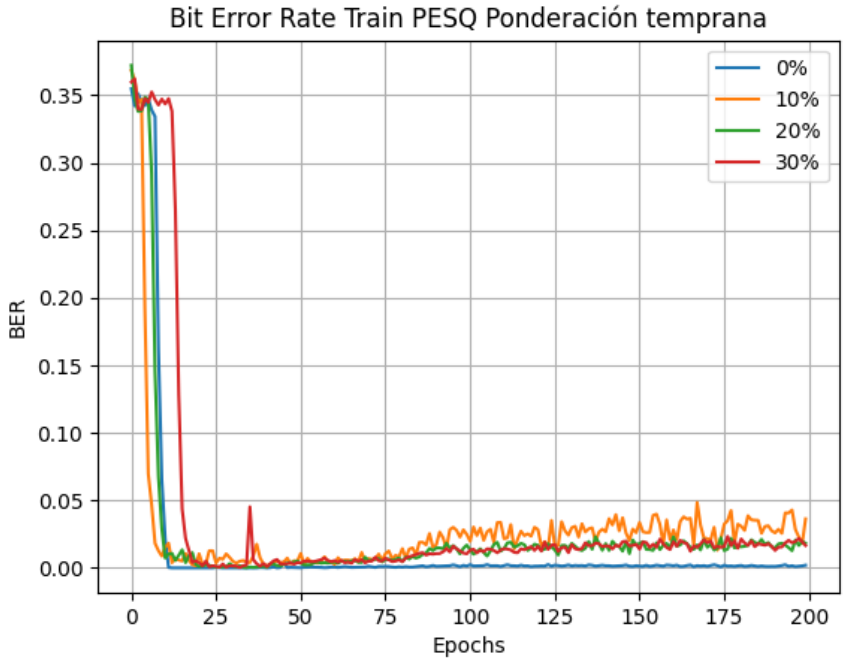


Figura 5.6: Bit Error Rate en tanto por uno del Detector en la fase de entrenamiento. Ponderación Temprana.

5.4. Rendimiento del sistema en la fase de validación

Antes de evaluar el rendimiento y la aplicabilidad de nuestro sistema de watermarking, es esencial estudiar y verificar su funcionamiento adecuado durante el proceso de validación. Esta etapa es esencial en el desarrollo de modelos basados en redes neuronales y aprendizaje automático, sirviendo como un enlace entre el entrenamiento y las pruebas finales de evaluación. Durante esta fase nos aseguramos de que el modelo no solo aprenda de los datos de entrenamiento, sino que también muestre un buen desempeño con un conjunto de datos nuevo.

En esta sección mostraremos los resultados obtenidos en el proceso de validación. El análisis de estos resultados nos permitirá confirmar que el modelo, además de adaptarse a los datos con los que fue entrenado, tiene la capacidad de generalizar su funcionamiento a nuevos conjuntos de datos, demostrando de esta forma sus aptitudes para pasar a la fase final de evaluación.

5.4.1. Fase de Validación del Sistema Original

En esta sección se exponen los resultados obtenidos en el proceso de validación del sistema original propuesto en [1]. El análisis de los resultados obtenidos nos permitirá discutir si existe una correspondencia entre el proceso de entrenamiento y el proceso de validación, así como la determinación de la capacidad de generalización del sistema.

Las pérdidas mostradas por el *Embedder* en el proceso de validación (figura 5.7) muestran el mismo patrón de decaimiento que el que se mostraba en el proceso de entrenamiento (figura 5.1). Todas las curvas exhiben un patrón de disminución de pérdidas a medida que avanzan las épocas del proceso de validación. Durante todo el proceso de validación podemos observar como el modelo entrenado con un 30 % de ataques presenta valores de pérdida por debajo del resto. Esto, de la misma forma que la curva de pérdidas mostrada en el entrenamiento (veáse figura 5.1), se debe a la tardía convergencia del *Detector* del modelo entrenado con un 30 % de ataques respecto al resto de modelos. Esta convergencia tardía por parte del *Detector* es aprovechada por el *Embedder* para garantizar su convergencia, dando lugar a resultados ligeramente mejores en la reconstrucción de la señal.

Respecto al BER obtenido en el proceso de validación, el comportamiento del mismo sigue también el mismo patrón de decaeimiento que el obtenido en el entrenamiento para los distintos escenarios de degradación. Observando la figura 5.8 podemos notar que en las primeras épocas todas las curvas presentan una rápida caída del BER, lo cual nos indica que el *Detector* está aprendiendo eficientemente las características necesarias para minimizar en gran medida los errores cometidos en los bits de salida que componen la marca de agua. A su vez, podemos observar diferencias marcadas entre los distintos niveles de ataque, especialmente al inicio del proceso. La curva de 0 % de ataques presenta una bajada más suave y estable, mientras que las curvas correspondientes a 10 %, 20 % y 30 % presentan unas fluctuaciones más pronunciadas, especialmente la de 30 % que presenta unos picos más altos de BER durante las primeras épocas. También podemos notar que,

aunque acaba por estabilizarse, la curva del 10 % muestra fluctuaciones más marcadas durante las primeras épocas. Esto se debe nuevamente a la estocasticidad en la generación de ataques. Aunque existan escenarios con un porcentaje elevado de ataques, es posible que la cantidad de ataques generados sea mayor a pesar de que su probabilidad de generación de ataques sea menor. Tal es el caso del escenario con un 10 % de ataques, el cual presenta resultados peores en la tarea de la detección de la marca de agua a pesar de tener un porcentaje menor de ataques.

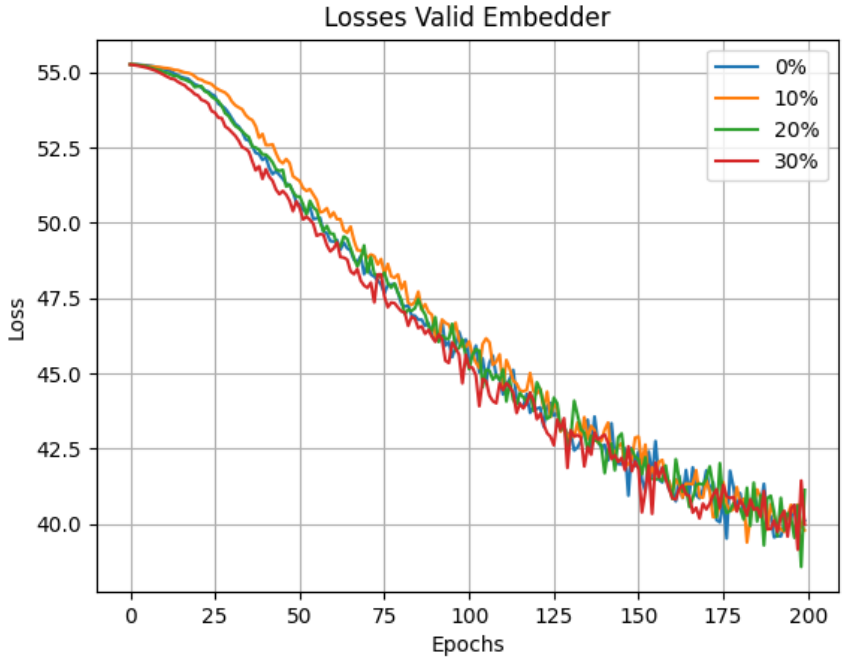


Figura 5.7: Curva de las pérdidas del Embedder en el proceso de validación.

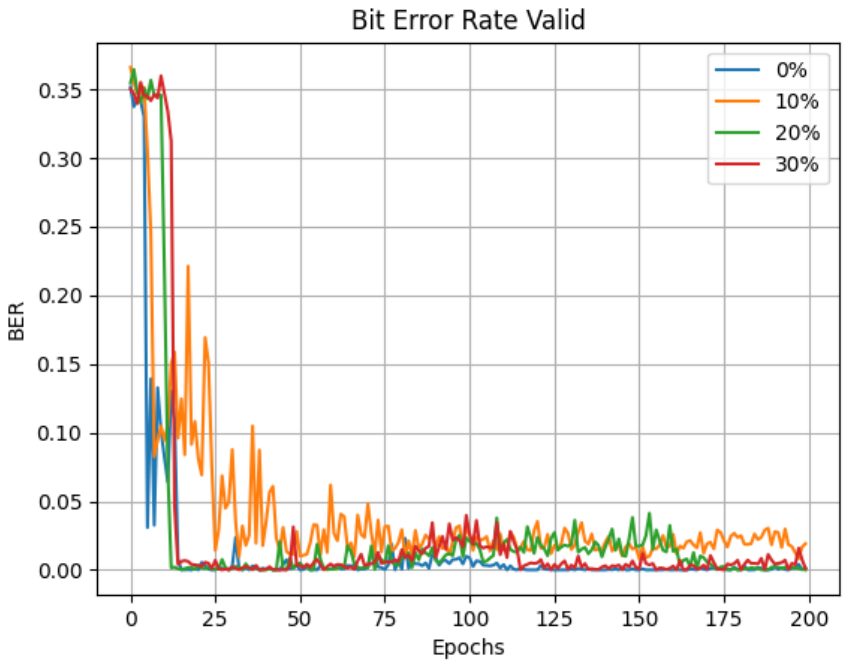


Figura 5.8: Bit Error Rate en tanto sobre uno en el proceso de validación

Si hacemos una comparativa entre la figura que presenta las pérdidas del *Embedder* (figura 5.7) y el BER cometido por el *Detector* 5.8, podemos observar que ambas se estabilizan al final de la validación, a pesar de que no se ha alcanzado la convergencia total por limitaciones de cómputo. La consecución de estos valores nos permite deducir que las redes, tanto *Embedder* como *Detector*, han alcanzado una buena capacidad de generalización para datos diferentes a los utilizados en el entrenamiento, incluso bajo la presencia de las diferentes probabilidades de ataque utilizadas en el proceso.

5.4.2. Fase de Validación del Sistema Propio con Ponderación Tardía

En esta sección se presentan los resultados alcanzados durante la fase de validación del sistema con ponderación tardía de la pérdida basada en el PESQ. Se analiza la curva de convergencia del *Embedder* y el BER del *Detector*, observando que ambos parámetros muestran patrones de comportamiento que son análogos a los obtenidos durante el entrenamiento. Este análisis detallado permite una comparación directa entre los procesos de aprendizaje durante las fases de entrenamiento y validación, destacando las similitudes y las diferencias entre las curvas de aprendizaje obtenidas en

ambos procesos bajo las diferentes condiciones de operación impuestas con el fin de comprobar si existe una correspondencia entre los procesos de entrenamiento y validación y una capacidad de generalización del sistema.

La curva de pérdidas del *Embedder* presenta el mismo patrón de convergencia que el mostrado en el entrenamiento (figura 5.5). En ambas figuras podemos ver un patrón de decaimiento al principio hasta la centésima vigésima época, en la que comienza a aparecer la pérdida basada en el PESQ. De nuevo, este incremento en la pérdida no nos indica que el modelo esté empeorando de cara a la reconstrucción de la señal. Como se indicó en el capítulo 3 de este trabajo, este aumento es consecuencia de la adición del término basado en la pérdida PESQ con su peso correspondiente. Esta adición de un nuevo término en la función de pérdida del *Embedder* supone un aumento en el valor numérico de la pérdida, pero, como veíamos en la figura 3.6, tanto MAE como la pérdida PESQ disminuyen a lo largo del entrenamiento y de la validación, lo que implica que la reconstrucción de la señal sí que está mejorando y que el aumento es causa de la adición del nuevo término (figura 5.9).

Se puede apreciar que los modelos, independientemente de sus distintos niveles de degradación, muestran pérdidas muy similares a lo largo de la validación, sugiriendo una capacidad uniforme en términos de reconstrucción de la señal, lo que tiene como consecuencia unos valores homogéneos en las métricas de evaluación de la reconstrucción de la señal (SNR, SDR, PESQ y STOI) en todos los modelos. No obstante, mediante un análisis más detallado, se observa que el modelo entrenado con un 30 % de ataques muestra un rendimiento ligeramente superior en las últimas épocas. Este comportamiento indica que la tardía convergencia del *Detector* en el modelo entrenado con un 30 % de ataques es aprovechada por el *Embedder* para garantizar su convergencia, lo que se traduce en una sutil mejora en la reconstrucción de la señal.

En referecia a la curva del BER del *Detector*, todos los modelos presentan un alto valor en el BER en las primeras épocas del proceso de validación. El comportamiento es similar al mostrado en el entrenamiento del modelo. Al principio el BER presenta valores elevados pero a medida que avanzamos en las épocas este desciende drásticamente para todas las probabilidades de ataque implementadas. Este decaimiento rápido del BER nos indican que los modelos se adaptan eficientemente y mejoran su capacidad de detección en condiciones adversas. Tras la disminución inicial, el BER se estabiliza pero muestra fluctuaciones evidentes en las épocas siguientes. Estas fluctuaciones son especialmente notorias en los modelos entrenados con un 20 % y un 30 % de ataques, lo que nos indica una mayor sensibilidad a la presencia de

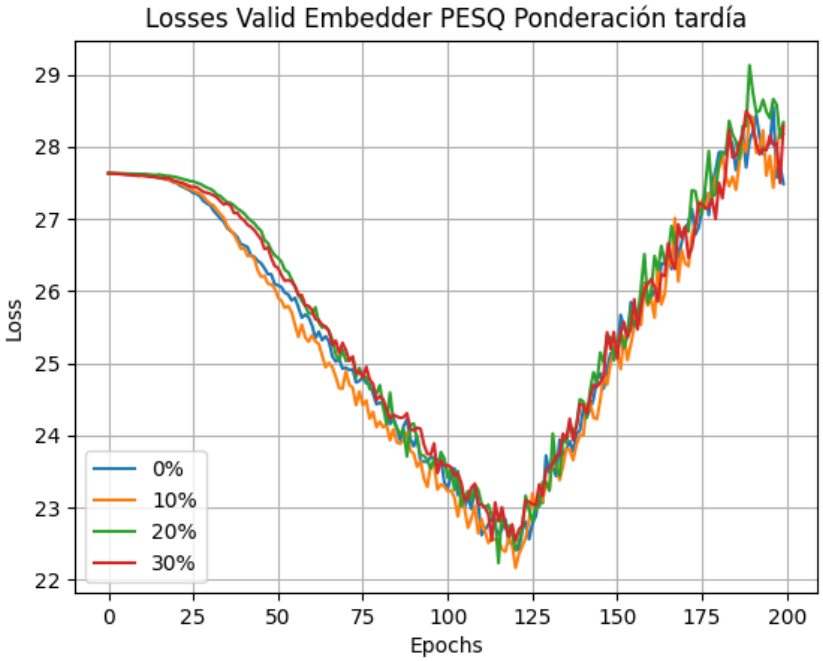


Figura 5.9: Curva de las pérdidas del Embedder en el proceso de validación. Ponderación Tardía.

ataques. Sin embargo, el modelo entrenado con un 30 % de ataques es capaz de sobreponerse, reducir y mantener valores aceptables, lo cual sugiere que el modelo presenta una mejor capacidad de adaptación y recuperación, presentando valores más bajos en el BER que el modelo entrenado con un 20 % de ataques (figura 5.10). De la misma forma que para el sistema original, resulta algo sorprendente que el modelo que presenta una mayor probabilidad de ataques dé lugar a un mejor rendimiento en la detección de la marca de agua. Esto es consecuencia de la aleatoridad con la que se generan los ataques. A pesar de que es contraintuitivo que un modelo con mayor probabilidad de ataques sea el más robusto, es un escenario probable y que, como podemos observar, ocurre.

A su vez, cabe destacar que, a la vista de los resultados obtenidos, los modelos con mayor nivel de degradación, especialmente el de 30 %, aunque presentan unas fluctuaciones más pronunciadas, podrían estar mejor preparados para hacer frente a las situaciones adversas, lo que resulta una ventaja clara en una aplicación real del sistema.

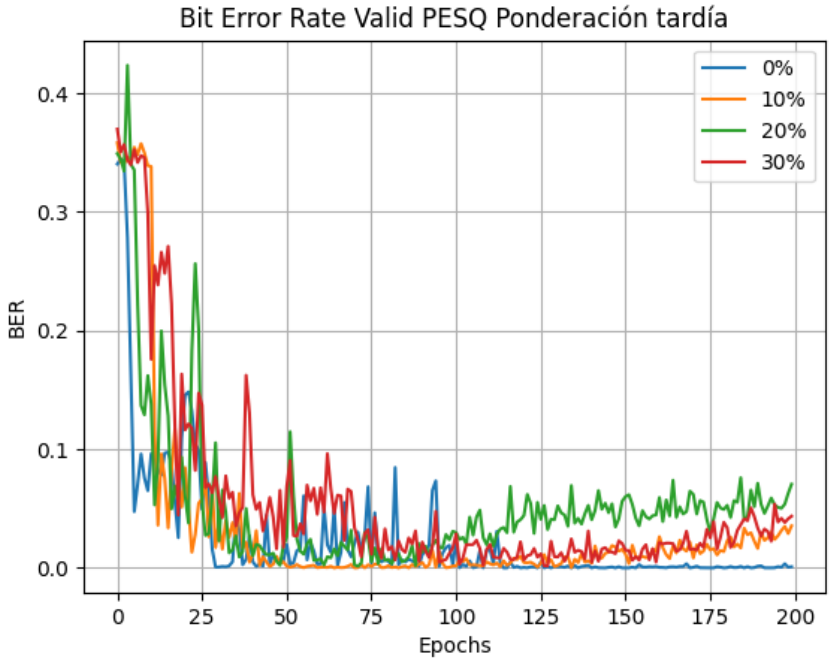


Figura 5.10: Bit Error Rate en tanto sobre uno en el proceso de validación. Ponderación Tardía.

5.4.3. Fase de Validación del Sistema Propio con Ponderación Temprana

En esta sección presentamos los resultados obtenidos durante la fase de validación del sistema con ponderación temprana de la pérdida basada en el PESQ. Realizaremos un análisis detallado de la curva de convergencia del *Embedder* y del la curva del BER del *Detector*. A través del estudio de estas curvas podremos dejar en claro que el comportamiento de las mismas es análogo al de las curvas obtenidas durante el entrenamiento. Este estudio permite comparar los resultados obtenidos en el entrenamiento con los obtenidos en validación, con el fin de comprobar si hay una correspondencia entre ellos y el sistema es capaz de generalizar su rendimiento para un conjunto de datos diferente.

La curva de pérdidas del *Embedder* presenta el mismo patrón de convergencia que la mostrada en el entrenamiento (figura 5.5). Vemos un decaimiento del MAE por parte de todos los modelos hasta la cuadragésima época, en la que comienza a tomar protagonismo la función de pérdida basada en el PESQ. Como ya venimos mencionando con anterioridad, el incremento

en el valor de la pérdida con la aparición de la pérdida basada en el PESQ a partir de la cuadragésima época no se debe a que la reconstrucción de la señal se esté empeorando, sino que este aumento se debe a la aparición de la pérdida basada en el PESQ con su respectivo peso como sumando a la pérdida cometida en la reconstrucción de la señal de voz junto con el MAE. Esta aparición de un nuevo sumando a la pérdida hace que el valor numérico de la misma aumente pero, como venimos diciendo en los apartados anteriores, estas dos partes de la curva no son comparables.

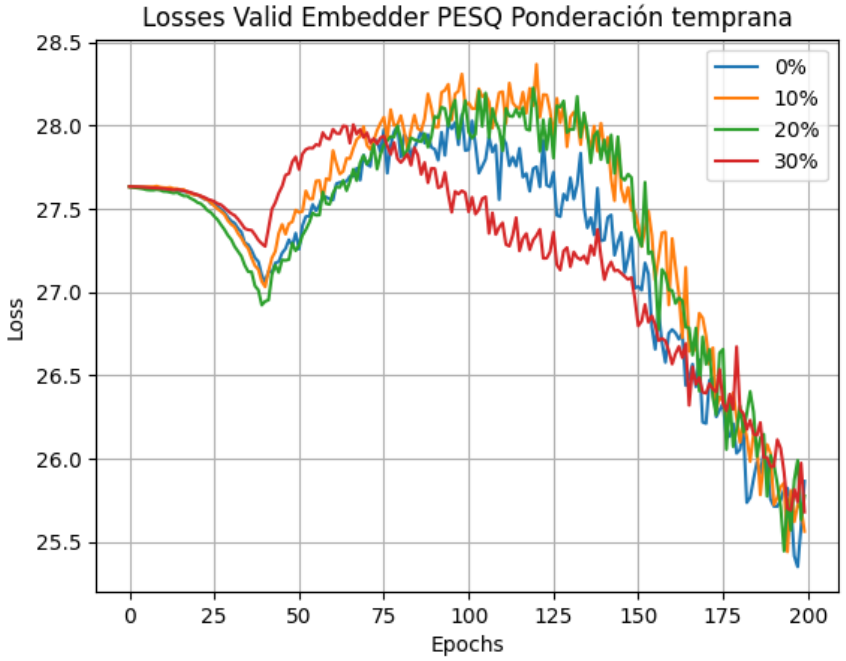


Figura 5.11: Curva de las pérdidas del Embedder en el proceso de validación. Ponderación Temprana.

En el análisis de la figura 5.11 , se observa que, aunque todos los modelos convergen hacia valores similares al final del proceso de validación, el modelo sometido a un 30 % de ataques muestra valores notablemente inferiores de pérdida a lo largo de la mayor parte del proceso de validación. Esta disminución notable en la pérdida sugiere una capacidad superior de dicho modelo para la reconstrucción de la señal. Esta ventaja en el desempeño puede ser atribuida la convergencia tardía del *Detector*, que es aprovechada por el *Embedder* para converger más rápido y arrojar unos resultados mejores en cuanto a la reconstrucción de la señal se refiere.

En relación con la curva del BER proporcionada por el *Detector* (figura

5.12) podemos observar el mismo comportamiento que en el entrenamiento y en el resto de casos presentados en la memoria. En las primeras épocas podemos ver un alto valor de BER en todos los modelos que posteriormente desciende de manera brusca, lo que es indicativo de una buena adaptación del modelo de cara a detectar los bits que conforman la marca de agua previamente incrustada en el *Embedder*. Tras esta abrupta caída del BER, este se estabiliza pero con notables fluctuaciones en los modelos de 10 % y 20 % de degradación. Tanto el modelo entrenado sin degradaciones y el modelo entrenado con un 30 % de ataques presentan un comportamiento más uniforme y adecuado de cara a una buena recosntrucción de la marca de agua. A partir de la centésima época todos muestran un comportamiento uniforme con ligeras fluctuaciones pero los que mejores resultados presentan son los de 0 % de ataques y el de 30 % de ataques. Dado que el modelo de 30 % ha sido entrenado con más ataques y arroja los mejores resultados en el BER, después del modelo entrenado sin degradaciones, se considera el más robusto de todos.

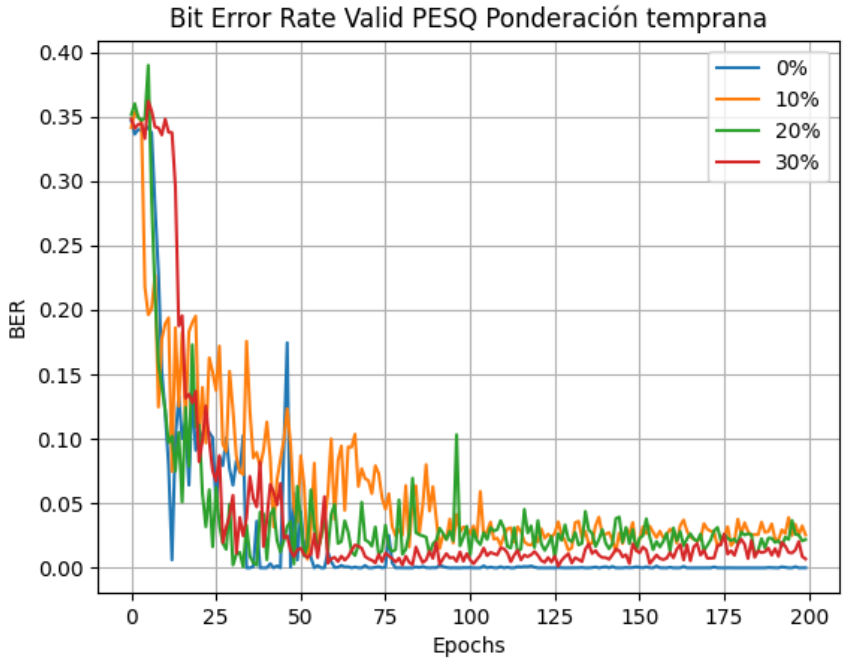


Figura 5.12: Bit Error Rate en tanto sobre uno en la fase de validación. Ponderación Temprana.

5.5. Rendimiento del sistema en la fase de Evaluación

Tras completar las fases de entrenamiento y validación de nuestro sistema de watermarking, procedemos a la fase de evaluación. Esta etapa es tan importante como las anteriores, ya que nos permite evaluar la efectividad real y la robustez del sistema. Además, nos ayuda a confirmar si los resultados obtenidos durante el entrenamiento y la validación son aplicables a un conjunto de datos diferente más cercanos a un entorno real. Los resultados obtenidos en esta sección demostrarán la capacidad del sistema para operar eficazmente en condiciones cotidianas y su habilidad para manejar datos que simulan escenarios realistas.

En esta sección presentamos los resultados experimentales de la etapa de evaluación. A través del análisis de los resultados obtenidos podremos confirmar la validez del enfoque implementado para el sistema *watermarking*, así como su capacidad de adaptación a las adversidades que se le opongan.

5.5.1. Fase de Evaluación del Sistema Original

En esta sección se exponen los resultados correspondientes a diversos escenarios, diferenciados por distintos niveles de degradación en el sistema original. Cuando mencionamos al sistema original nos referimos al desarrollado en [1], al que no se le añade la función de pérdida basada en el PESQ. El objetivo es evaluar cuál de estos escenarios demuestra mayor robustez en la reconstrucción de la marca de agua, así como en la imperceptibilidad de las distorsiones provocadas por dicha marca en la señal de voz.

Para cada modelo entrenado con su nivel de degradación correspondiente (0 %, 10 %, 20 % y 30 %) se realizarán pruebas cruzadas para los distintos niveles de degradación. Con este enfoque conseguiremos determinar cuál de todos los modelos es el mejor en la tarea de detección de la marca de agua y reconstrucción en la señal de voz.

Resultados de evaluación para la SNR

Cuadro 5.1: SNR (dB) para diferentes niveles de degradación en modelos entrenados

Modelos entrenados	**Probabilidad de Ataque**			
	0 %	**10 %**	**20 %**	**30 %**
0 % Ataques	7.12	7.12	7.12	7.12
10 % Ataques	7.14	7.14	7.14	7.14
20 % Ataques	7.14	7.14	7.14	7.14
30 % Ataques	7.24	7.24	7.24	7.24

Resultados de evaluación para la SDR

Cuadro 5.2: SDR (dB) para diferentes niveles de degradación en modelos entrenados

Modelos entrenados	**Probabilidad de Ataque**			
	0 %	**10 %**	**20 %**	**30 %**
0 % Ataques	6.47	6.47	6.47	6.47
10 % Ataques	6.54	6.54	6.54	6.54
20 % Ataques	6.52	6.52	6.52	6.52
30 % Ataques	6.63	6.63	6.63	6.63

Resultados de evaluación para el PESQ

Cuadro 5.3: PESQ para diferentes niveles de degradación en modelos entrenados

Modelos entrenados	**Probabilidad de Ataque**			
	0 %	**10 %**	**20 %**	**30 %**
0 % Ataques	2.25	2.25	2.25	2.25
10 % Ataques	2.22	2.22	2.22	2.22
20 % Ataques	2.27	2.27	2.27	2.27
30 % Ataques	2.23	2.23	2.23	2.23

Resultados de evaluación para el STOI

Cuadro 5.4: STOI para diferentes niveles de degradación en modelos entrenados

Modelos entrenados	**Probabilidad de Ataque**			
	0 %	**10 %**	**20 %**	**30 %**
0 % Ataques	0.93	0.93	0.93	0.93
10 % Ataques	0.93	0.93	0.93	0.93
20 % Ataques	0.935	0.935	0.935	0.935
30 % Ataques	0.9335	0.9335	0.9335	0.9335

Resultados de evaluación para el BER

Cuadro 5.5: BER (%) para diferentes niveles de degradación en modelos entrenados

Modelos entrenados	**Probabilidad de Ataque**			
	0 %	**10 %**	**20 %**	**30 %**
0 % Ataques	0	1.5	2.6	4.2
10 % Ataques	0.7	2.7	4.2	7
20 % Ataques	0	0.08595	0.88	0.15
30 % Ataques	0	0	0	0.2

Observando las tablas podemos concluir que el modelo que mejor reconstruye la señal de voz y detecta la marca de agua en todos los escenarios posibles es el modelo entrenado con un 30 % de nivel de degradación, proporcionando un BER de 0.05 % en promedio, una SNR de 7.24 dB en promedio, una SDR de 6.63 dB en promedio, un PESQ de 2.23 en promedio y un STOI de 0.9335 en promedio. Aunque el PESQ obtenido es ligeramente mejor para el modelo entrenado con un 20 % de ataques, la diferencia en la robustez que existe entre estos dos sistemas hace que el modelo entrenado con un 30 % de ataques sea mejor opción.

Es importante señalar que en lo que respecta a la reconstrucción de la señal, las métricas de evaluación de los distintos modelos muestran resultados bastante homogéneos, sin que ninguno sobresalga notablemente sobre los demás. Sin embargo, la diferencia clave radica en el BER observado en la reconstrucción de la marca de agua. En este aspecto, el modelo entrenado con un 30 % de ataques muestra un desempeño superior, distinguiéndose claramente del resto. Este factor convierte a este sistema en el más efectivo de todos en términos generales.

Podemos plasmar los resultados. En la figura 5.13a se observa como la

forma de onda de la señal reconstruida es bastante similar a la de la señal original, especialmente en aquellas zonas donde la amplitud es elevada. Esto implica que en las zonas de alta energía de la señal la distorsión provocada por la marca de agua es menos evidente que en las regiones de la señal menos energéticas. Esto tiene como consecuencia que, desde el punto de vista perceptual, las partes de las frases que se correspondan con silencios aparecerá un ruido de fondo casi imperceptible, que para ser escuchado se debe de prestar especial atención.

5.5.2. Fase de Evaluación del Sistema Propio con Ponderación Tardía

A continuación se comentan los resultados correspondientes a los diversos escenarios propuestos con distintos niveles de degradación para el sistema con la pérdida PESQ con ponderación tardía. De la misma forma que para el apartado anterior el cometido de esta sección es evaluar la robustez de todos los modelos para los distintos niveles de degradación, así como la imperceptibilidad de la marca de agua incrustada en la señal.

Para cada modelo se realizarán pruebas cruzadas con los distintos niveles de degradación posibles. Con este análisis conseguiremos determinar cuál de todos los sistemas es el más robusto frente a las degradaciones y reconstruye mejor la señal de voz.

Resultados de evaluación para la SNR

Cuadro 5.6: SNR (dB) para diferentes niveles de degradación en modelos entrenados

Modelos entrenados	**Probabilidad de Ataque**			
	0 %	**10 %**	**20 %**	**30 %**
0 % Ataques	7.84	7.84	7.84	7.84
10 % Ataques	7.92	7.92	7.92	7.92
20 % Ataques	7.89	7.89	7.89	7.89
30 % Ataques	7.96	7.96	7.96	7.96

Resultados de evaluación para la SDR

Cuadro 5.7: SDR (dB) para diferentes niveles de degradación en modelos entrenados

Modelos entrenados	**Probabilidad de Ataque**			
	0 %	**10 %**	**20 %**	**30 %**
0 % Ataques	7.26	7.26	7.26	7.26
10 % Ataques	7.35	7.35	7.35	7.35
20 % Ataques	7.32	7.32	7.32	7.32
30 % Ataques	7.40	7.40	7.40	7.40

Resultados de evaluación para el PESQ

Cuadro 5.8: PESQ para diferentes niveles de degradación en modelos entrenados

Modelos entrenados	**Probabilidad de Ataque**			
	0 %	**10 %**	**20 %**	**30 %**
0 % Ataques	2.46	2.46	2.46	2.46
10 % Ataques	2.46	2.46	2.46	2.46
20 % Ataques	2.36	2.36	2.36	2.36
30 % Ataques	2.44	2.44	2.44	2.44

Resultados de evaluación para el STOI

Cuadro 5.9: STOI para diferentes niveles de degradación en modelos entrenados

Modelos entrenados	**Probabilidad de Ataque**			
	0 %	**10 %**	**20 %**	**30 %**
0 % Ataques	0.945	0.945	0.945	0.945
10 % Ataques	0.95	0.95	0.95	0.95
20 % Ataques	0.945	0.945	0.945	0.945
30 % Ataques	0.9476	0.9476	0.9476	0.9476

Resultados de evaluación para el BER

Cuadro 5.10: BER (%) para diferentes niveles de degradación en modelos entrenados

Modelos entrenados	**Probabilidad de Ataque**			
	0 %	**10 %**	**20 %**	**30 %**
0 % Ataques	0	10.7	15.7	21.4
10 % Ataques	0	4.47	5.85	10.35
20 % Ataques	0	2.9	6.67	9.27
30 % Ataques	0	0.9	3.33	5.6

Observando las tablas que exponen los resultados para las diferentes métricas de evaluación podemos dejar en claro que la situación es bastante similar a la mostrada en el apartado anterior. En general, todos los modelos reconstruyen la señal igual de bien, con una ligera superioridad en el modelo entrenado con un 10 % de ataques en métricas como el PESQ y el STOI. En el caso de la SNR y la SDR, el modelo entrenado con un 30 % de ataques es sutilmente superior.

Sin embargo, el factor determinante es el BER cometido por los modelos entrenados con diferentes niveles de degradación. De nuevo, en este caso, la robustez es superior en el modelo entrenado con un 30 % de ataques, presentando un BER de 2.45 % en promedio. Esto hace que este sistema sea el más idóneo de los presentes para su aplicación en una situación real ya que, en términos globales, es el mejor de los entrenados con ponderación tardía.

Podemos ver los resultados del *Embedder* en la reconstrucción de la señal en la figura 5.14. En este caso, observando la figura podemos notar que ocurre algo similar que en el caso anterior (figura 5.13a). Dado que tanto el sistema original y el sistema con ponderación tardía presentan valores similares en las métricas de evaluación de la reconstrucción de la señal es evidente que la representación de las señales originales y reconstruidas en ambos sistemas van a presentar características similares. Podemos notar como en las regiones de mayor amplitud la señal reconstruida tiene una forma de onda que coincide de mejor manera con la forma de onda de la señal original que las regiones que presentan menor energía. Esto hace que la distorsión, medida como la diferencia entre la señal original y reconstruida, en las zonas de mayor energía sea ligeramente menos perceptible que en las zonas de menor energía. Como consecuencia de esto, desde el punto de vista perceptual, en estas zonas de menor energía, que coinciden con los silencios de la frase pronunciada por los hablantes, la distorsión es casi imperceptible, teniendo que prestar especial atención para captarla.

5.5.3. Fase de Evaluación del Sistema Propio con Ponderación Temprana

Por último, y de igual forma que en los dos apartados anteriores, en esta sección se realizará un análisis cuantitativo de los diferentes escenarios propuestos con distintos niveles de degradación para el sistema con la pérdida PESQ con ponderación temprana. De igual forma que para los dos apartados anteriores, el principal objetivo de esta parte es obtener una evaluación de la robustez de todos los modelos frente a los ataques, así como la imperceptibilidad de la marca de agua sobre la señal reconstruida.

En este caso, para cada modelo se van a realizar las diferentes pruebas cruzadas con los distintos niveles de degradación posibles. Con este procedimiento se consigue determinar cuál de todos los sistemas es el que mejores resultados proporciona.

Resultados de test para la SNR

Cuadro 5.11: SNR (dB) para diferentes niveles de degradación en modelos entrenados

Modelos entrenados	**Probabilidad de Ataque**			
	0 %	**10 %**	**20 %**	**30 %**
0 % Ataques	10.82	10.82	10.82	10.82
10 % Ataques	10.79	10.79	10.79	10.79
20 % Ataques	10.35	10.35	10.35	10.35
30 % Ataques	13.21	13.21	13.21	13.21

Resultados de test para la SDR

Cuadro 5.12: SNR (dB) para diferentes niveles de degradación en modelos entrenados

Modelos entrenados	**Probabilidad de Ataque**			
	0 %	**10 %**	**20 %**	**30 %**
0 % Ataques	10.7	10.7	10.7	10.7
10 % Ataques	10.63	10.63	10.63	10.63
20 % Ataques	10.14	10.14	10.14	10.14
30 % Ataques	13.42	13.42	13.42	13.42

Resultados de test para el PESQ

Cuadro 5.13: PESQ para diferentes niveles de degradación en modelos entrenados

Modelos entrenados	**Probabilidad de Ataque**			
	0 %	**10 %**	**20 %**	**30 %**
0 % Ataques	3.435	3.435	3.435	3.435
10 % Ataques	3.4	3.4	3.4	3.4
20 % Ataques	3.27	3.27	3.27	3.27
30 % Ataques	3.7	3.7	3.7	3.7

Resultados de test para el STOI

Cuadro 5.14: STOI para diferentes niveles de degradación en modelos entrenados

Modelos entrenados	**Probabilidad de Ataque**			
	0 %	**10 %**	**20 %**	**30 %**
0 % Ataques	0.974	0.974	0.974	0.974
10 % Ataques	0.973	0.973	0.973	0.973
20 % Ataques	0.97	0.97	0.97	0.97
30 % Ataques	0.985	0.985	0.985	0.985

Resultados de test para el BER

Cuadro 5.15: BER (%) para diferentes niveles de degradación en modelos entrenados

Modelos entrenados	**Probabilidad de Ataque**			
	0 %	**10 %**	**20 %**	**30 %**
0 % Ataques	0	7.8	16.29	21.9
10 % Ataques	0	3	6.29	8.3
20 % Ataques	0	1.32	3	3.31
30 % Ataques	0	0.3	0.46	0.81

Podemos apreciar que, de nuevo, el modelo que mejores resultados presenta para todos los escenarios planteados es el modelo entrenado con un 30 % de ataques. Este presenta un BER de 0.39 % en promedio, una SNR de 13.21 dB, una SDR de 13.42 dB, un PESQ de 3.7 y un STOI de 0.985. Los resultados de la reconstrucción de la señal por parte del *Embedder* en la figura 5.15. En este caso, dado que se trata del sistema que mejor reconstruye la señal, podemos comprobar viendo la figura 5.15b como la forma de onda de la señal reconstruida coincide fielmente con la de la señal original, con algunas diferencias muy leves. Esto hace que, desde el punto de vista perceptual, la distorsión sea inapreciable.

5.5.4. Análisis General de los Resultados obtenidos en la Fase de Evaluación

En este apartado se procede al análisis comparativo de todos los resultados obtenidos por parte de cada uno de los modelos mostrados. Con el fin de analizar profundamente los resultados reflejados en las tablas anteriores realizaremos un resumen de las tendencias observadas y proporcionaremos un estudio comparativo entre los modelos de cada uno de los escenarios propuestos (sistema original, sistema con ponderación tardía y sistema con ponderación temprana) con los respectivos porcentajes de ataques con los que han sido entrenados (0 %, 10 %, 20 % y 30 %).

Comparativa de los modelos con 0 % de Ataques

En los modelos entrenados con un 0 % de degradación podemos observar que tanto el sistema original como el sistema con ponderación tardía presentan unos valores de SNR, SDR, PESQ y STOI bastante similares entre sí. Esta mejora es mucho más notoria para el sistema con ponderación temprana, alcanzando unos valores de SNR, SDR, PESQ y STOI significativamente mejores que los obtenidos mediante los sistemas anteriormente mencionados.

Sin embargo, los modelos entrenados con un 0 % de ataques son los que peores resultados de adaptación presentan cuando realizamos las pruebas cruzadas con los diferentes niveles de degradación posible. En el sistema original el BER pasa de valer 0 % en el caso de una degradación del 0 % a un 4.2 %. Lo mismo para el modelo con ponderación tardía que pasa de 0 % a 21.4 % y para el modelo con ponderación temprana pasando de un sólido 0 % a un nefasto 21.9 %. Esto sugiere que estos modelos no son capaces de manejar ataques, algo normal ya que su entrenamiento fue realizado sin ellos y que, además, son sumamente sensibles a los incrementos en el porcentaje

de ataques.

Comparativa de los modelos con 10 % de Ataques

Si analizamos las métricas de evaluación de la señal de voz podemos decir que todas se mantienen en el mismo rango de valores para cada modelo. Sin embargo, de la misma forma que para los modelos anteriores, el sistema con ponderación temprana presenta una mejora notoria en métricas como la SNR, SDR, PESQ y STOI. Estas métricas tienen unos valores de 7.14 dB, 6.54 dB, 2.22 y 0.93 y 7.84 dB, 7.35 dB, 2.46 y 0.95 (SNR, SDR, PESQ y STOI respectivamente) para los sistema original y con ponderación tardía respectivamente. Como vemos, el sistema con ponderación tardía mejora la reconstrucción de la señal en cierta medida pero no tiene punto de comparación con la reconstrucción proporcionada por el sistema con ponderación temprana, arrojando unos valores de 10.79 dB para la SNR, 10.63 dB para la SDR, 3.4 para el PESQ y 0.973 para el STOI.

Sin embargo, los sistemas con ponderación tardía y temprana no terminan de adaptarse del todo al aumento de la degradación ya que, cuando realizamos pruebas cruzadas, el BER pasa de 0 % a 10.35 % en el sistema con ponderación tardía y de 0 % a 8.3 %. Este BER no es el adecuado para un sistema de marcado de agua, que debería de mantenerse siempre por debajo del 1 %. También le ocurre lo mismo al sistema original, que pasa de un BER de 0.7 % a uno de 7 %. Estos resultados sugieren que los sistemas entrenados con un 10 % de ataques tampoco se adaptan adecuadamente a un aumento en el nivel de degradación. Aunque los resultados son mejores que para los modelos entrenados con un 0 % de ataques, estos no son del todo favorables.

Comparativa de los modelos con 20 % de Ataques

Las métricas de evaluación de la señal de voz mantienen valores constantes en todos los modelos. Ocurre lo mismo que con los dos anteriores. Los sistemas original y con ponderación tardía presentan resultados notoriamente peores en la reconstrucción de la señal que el modelo con ponderación temprana. Estos pasan de tener un valor de 7.14 dB de SNR, 6.52 dB de SDR, 2.27 de PESQ y 0.935 de STOI en el sistema original y 7.89 dB de SNR, 7.32 dB de SDR, 2.36 de PESQ y 0.945 de STOI en el sistema con ponderación tardía a tener una SNR de 10.35, una SDR de 10.14 dB, un PESQ de 3.27 y un STOI de 0.97. Comparando resultados se deja en claro que el modelo con ponderación temprana reconstruye la señal con mejor calidad.

A pesar de la mejor reconstrucción de la señal, los sistemas con ponderación tardía y temprana sufren un claro *trade-off* entre calidad de reconstrucción de la señal de voz y calidad de reconstrucción de la marca de agua. Ambos modelos presentan un BER demasiado alto para un sistema de marcado de agua mientras que el sistema original se mantiene por debajo del 1 % en todos los casos. Se puede notar una clara dicotomía entre lo buena que es la reconstrucción de la señal y lo buena que es la reconstrucción de la marca de agua. Esta dualidad hace que tengamos que elegir entre imperceptibilidad y robustez ya que en este caso no pueden satisfacerse los dos requisitos.

Comparativa de los modelos con 30 % de Ataques

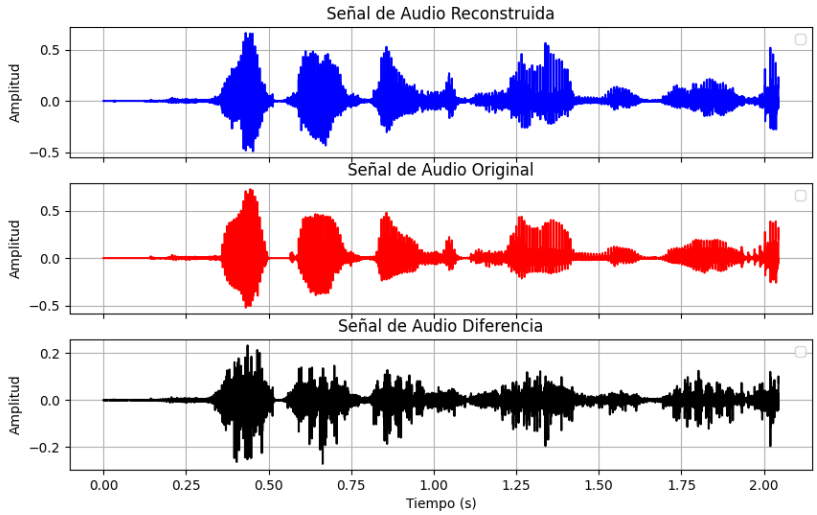
En relación con las métricas de SNR y SDR, se observa que los valores permanecen esencialmente estables en todos los modelos y a través de todos los niveles de degradación. Aunque estos valores son constantes dentro de cada modelo específico, hay una diferencia significativa entre los modelos originales mencionados en [1] y aquellos evaluados bajo la metodología de ponderación tardía en comparación con los modelos sometidos a ponderación temprana. Esta variabilidad entre los enfoques de modelado subraya la influencia de las técnicas de entrenamiento y ponderación en la consistencia de las métricas de calidad de la señal. Observando los resultados podemos dejar en claro que hay una mejora notable en estas métricas de evaluación, aumentando de 7.24 dB y 6.63 dB en el sistema original y de 7.96 dB y 7.40 dB para el sistema con ponderación tardía a 13.21 dB y 13.42 dB en el sistema con ponderación temprana para la SNR y la SDR respectivamente.

Ocurre lo mismo con las métricas PESQ y STOI. A pesar de que en los sistemas original y con ponderación tardía estas métricas presentan unos valores aceptables, tanto el PESQ como el STOI, pero las obtenidas con el sistema con ponderación temprana son significativamente mejores, sobre todo el PESQ, que pasa de valer 2.23 en el sistema original y 2.44 en el sistema con ponderación tardía, a aumentar a 3.7 en el sistema con ponderación temprana.

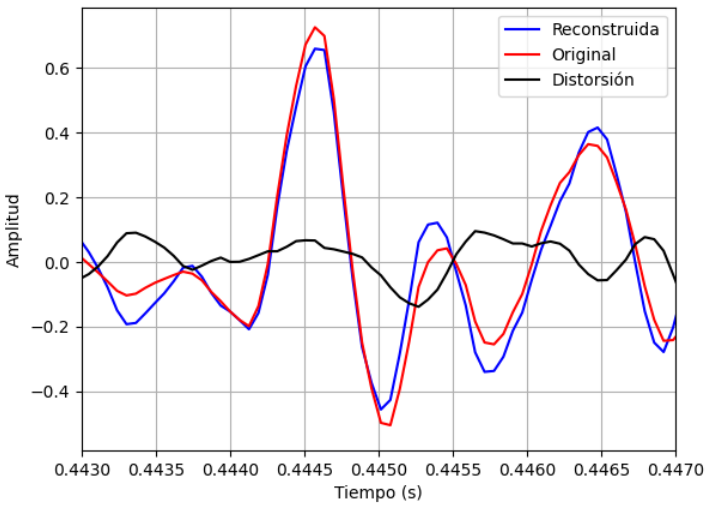
En cuanto a la Tasa de Error de Bit (BER), el sistema original destaca por ofrecer los resultados más favorables, con un promedio de apenas 0.2 %. Por otro lado, el sistema que utiliza ponderación tardía muestra los peores resultados en esta métrica. Esto indica que la estrategia de ponderación tardía no es la más adecuada, a pesar de que pueda dar una ligera mejora en la reconstrucción de la señal. Existe un compromiso claro, un *trade-off*, entre la calidad de reconstrucción de la señal y la eficiencia en términos de BER

cuando se aplica esta metodología. Este peor desempeño podría deberse a que el modelo está insuficientemente adaptado a la introducción tardía de pérdidas basadas en el PESQ, lo que en términos generales convierte a este sistema en el menos efectivo. Sin embargo, este fenómeno no ocurre en el sistema con ponderación temprana ya que este presenta un BER de 0.39 % en promedio. Aunque el sistema original presenta un BER menor, el sistema con ponderación temprana presenta un BER menor al 1 %, sugiriendo buenos resultados en términos de robustez.

Por tanto, dado que el sistema con ponderación temprana es el que mejor reconstruye la señal de voz con mucha diferencia y proporciona una robustez sobresaliente en la detección de la marca de agua, este será sistema el que mejor se adapta a un escenario real y el que mejor cumple las premisas de imperceptibilidad y robustez esenciales en los sistemas de marcado de agua.

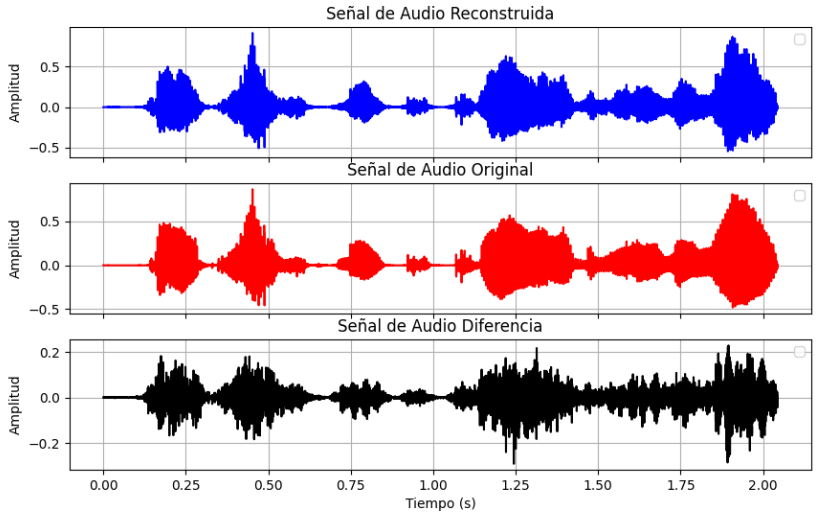


(a) Señales Reconstruida (azul), Original (roja) y Distorsión (negra).

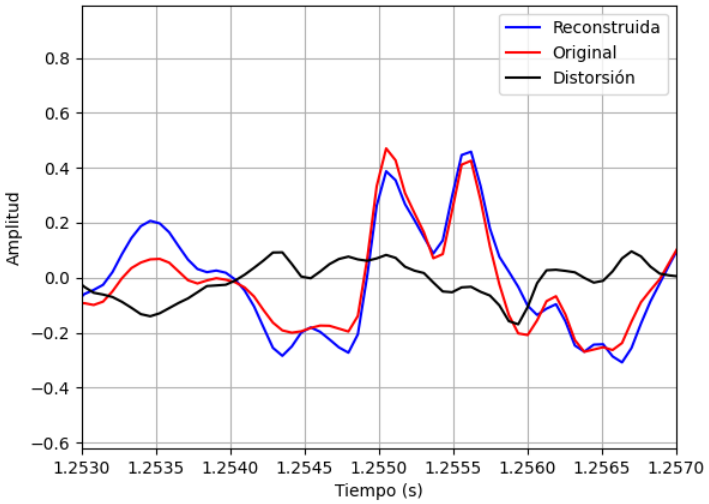


(b) Zoom de las señales.

Figura 5.13: Resultados del *Embedder*.

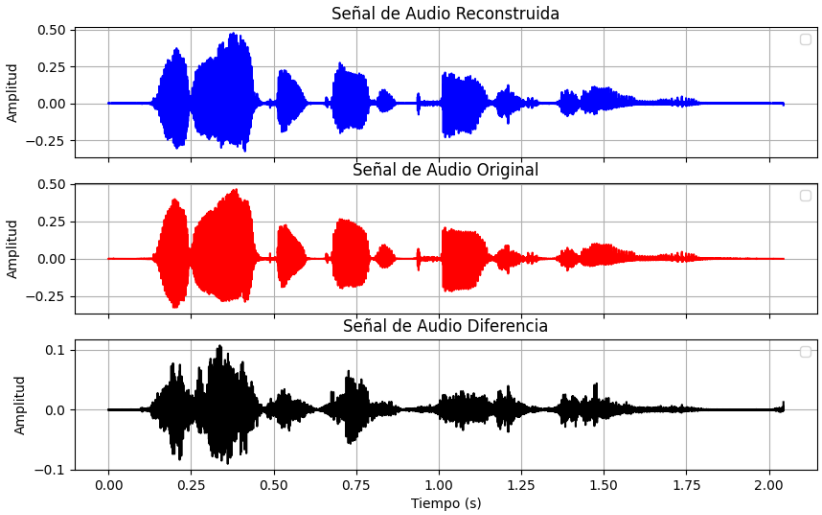


(a) Señales Reconstruida (azul), Original (roja) y Distorsión (negra).

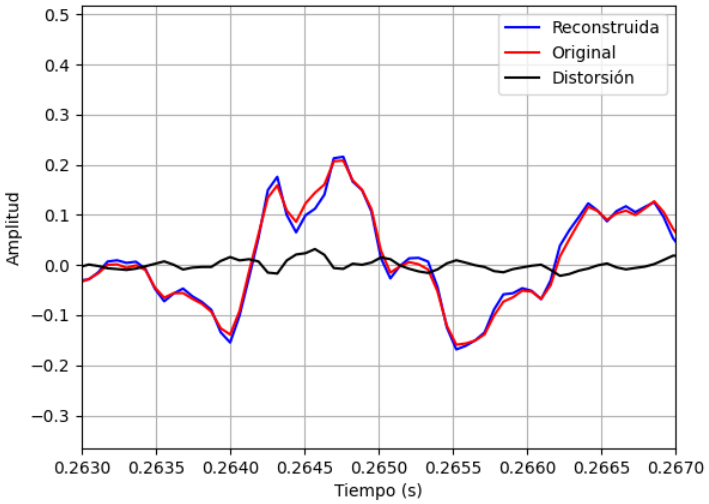


(b) Zoom de las señales.

Figura 5.14: Resultados del *Embedder*. Ponderación Tardía.



(a) Señales Reconstruida (azul), Original (roja) y Distorsión (negra).



(b) Zoom de las señales.

Figura 5.15: Resultados del *Embedder*. Ponderación Temprana.

Capítulo 6

Conclusiones y Trabajo Futuro

En este capítulo final, se procederá a realizar una evaluación exhaustiva del grado de cumplimiento de los objetivos inicialmente establecidos para este proyecto. Adicionalmente, se discutirán las conclusiones derivadas tanto de estos objetivos como de los resultados obtenidos por el sistema, proporcionando una visión integral y detallada del desempeño y las implicaciones del proyecto en su conjunto.

6.1. Conclusión

La infracción de la normativa vigente sobre derechos de autor, propiedad intelectual y *Copyright*, junto con la suplantación de identidad y la propagación de información falsa, están experimentando una creciente tendencia al alza en la actualidad [1]. La reciente aparición de las tecnologías de inteligencia artificial ha generado un nicho de actos impúdicos, como los *deepfakes*, que provocan malestar y vulnerabilidad entre los usuarios. Es de una gran importancia que los usuarios estén plenamente protegidos en estos aspectos fundamentales y el *watermarking* surge como una técnica que nos garantiza en gran medida este objetivo.

- En este trabajo se ha presentado un sistema de marcado de agua basado en aprendizaje profundo y en redes neuronales. El sistema propuesto consiste en dos redes neuronales cada una de ellas con una tarea conceptualmente opuesta a la otra. La primera de ellas es a la que hemos denominado *Embedder* durante toda la memoria y cuya tarea principal es la de incrustar la marca de agua en la señal de voz. La segunda es

el *Detector* y su tarea es la de detectar la marca de agua oculta por el *Embedder*. Ambas redes neuronales fueron entrenadas conjuntamente pues el rendimiento de una depende de la otra y viceversa. El proceso conjunto de entrenamiento ha sido diseñado cuidadosamente parar garantizar la convergencia de ambas redes neuronales. En la primera fase del entrenamiento se le dió más peso al *Detector* para que sea capaz de detectar la marca de agua incrustada en el *Embedder* ya que este, en el proceso de reconstruir la señal portadora, tiende a minimizar las distorsiones introducidas sobre la señal pudiendo borrar la marca de agua en este proceso. En la segunda fase, cuando ya se garantizó la convergencia del *Detector* y la detección de la marca de agua por parte del mismo, comenzó a tomar el protagonismo el *Embedder* para garantizar una reconstrucción de la señal aceptable.

- Al sistema inicialmente descrito en el párrafo anterior, se le incorporó una función de pérdida adicional, denominada Pérdida PESQ (Perceptual Evaluation of Speech Quality Loss). Esta función se sumó a las pérdidas generadas por el *Embedder*, con el objetivo de optimizar la calidad perceptual de las señales de voz procesadas. Se han explorado dos escenarios distintos respecto a la implementación de la pérdida PESQ en el proceso de entrenamiento. En el primero, denominado "ponderación temprana", la pérdida PESQ se integra desde las etapas iniciales del entrenamiento. En el segundo escenario, llamado "ponderación tardía", la pérdida PESQ se incorpora únicamente en las fases finales del entrenamiento. Esta diferenciación permitió evaluar el impacto temporal de la pérdida PESQ en la optimización del modelo. Los resultados que se obtuvieron mediante cada uno de los enfoques mencionados (sistema original, sistema con ponderación tardía y sistema con ponderación temprana) arrojan las siguientes conclusiones particulares:

  - La novedad de esta filosofía implementada a partir de la pérdida basada en PESQ nos permite afirmar que esta función de pérdida da buenos resultados de cara a reconstruir la señal, especialmente en el sistema con ponderación temprana. Sin embargo, una mejor reconstrucción de la señal de voz conlleva un claro *trade-off* que afecta negativamente al BER cometido en la reconstrucción de la marca de agua. Este efecto es especialmente notable en el sistema con ponderación tardía, ya que mejora la reconstrucción de la señal en comparación con el sistema original, pero presenta un BER considerablemente superior.
  - El sistema original nos ofreció buenos resultados, aunque con un gran margen de mejora, cumpliendo con los dos requisitos principales: imperceptibilidad y robustez. Aunque el sistema con pon-

deración tardía mejora ligeramente la reconstrucción de la señal, cumpliendo con la imperceptibilidad, este esquema pone en serio compromiso al BER, no satisfaciendo la robustez necesaria. El sistema con ponderación temprana es el que destacó por encima del resto. La reconstrucción de la señal de voz es mejor que en el sistema original y en el que tiene ponderación tardía, superando a ambos en cuanto a la imperceptibilidad se refiere.

  - Perceptualmente, cuando escuchamos las señales de voz reconstruidas por el *Embedder*, a duras penas logra apreciarse la distorsión provocada por la inserción de la marca de agua en los sistemas original y con ponderación tardía, salvo en las zonas silenciosas de la señal de voz. Esta distorsión es completamente imperceptible en el sistema con ponderación temprana, siendo realmente complicado captar las diferencias entre la señal original y la reconstruida.

- En relación con los ataques diseñados para degradar la señal y complicar la detección de marcas de agua, se seleccionaron una gama que es particularmente representativa de los desafíos actuales en los sistemas de comunicaciones. El espectro de posibles ataques es considerablemente amplio, y sería factible dedicar un estudio independiente exclusivamente a evaluar los efectos de los distintos tipos de ataques en estos sistemas.

- Podemos afirmar que el sistema cumple satisfactoriamente con los requisitos de imperceptibilidad y robustez en mayor o menor medida. Los resultados obtenidos en el capítulo anterior muestran la efectividad del sistema en preservar la calidad de la marca de agua incrustada en el *Embedder* a pesar de las degradaciones introducidas antes de llegar al *Detector* y, a su vez, garantiza que la distorsión provocada por la marca de agua sobre la señal reconstruida sea prácticamente imperceptible para el usuario final.

- Por último, cabe mencionar que las modificaciones implementadas para integrar la métrica de pérdida basada en el PESQ al modelo original han tenido como objetivo principal mejorar la reconstrucción de la señal sin afectar significativamente al BER. Dicha integración ha resultado efectiva en el sistema de ponderación temprana, mientras que en el sistema de ponderación tardía los resultados no han sido tan favorables. Los resultados observados en el capítulo anterior han proporcionado las siguientes conclusiones particulares:

  - El sistema original nos ofrecía un BER bastante sólido como pudo verse en el capítulo anterior. El modelo óptimo en términos globales fue el entrenado con un 30 % de ataques, en el cual el BER

aumenta desde un 0 % en un escenario con un 0 % de ataques hasta un 0.2 % en un escenario con un 30 % de ataques.

- El sistema con ponderación tardía que mostró los mejores resultados en términos del BER también fue el entrenado con un 30 % de probabilidad de ataque, aunque no alcanza los mismos niveles de desempeño que el sistema original en términos de robustez. Este modelo pasa de un 0 % de BER cuando nos encontramos en un escenario con un 0 % de ataques y aumenta hasta un 5.6 % cuando tenemos un 30 % de probabilidad de ataques. El *trade-off* que existe en este esquema entre la mejora en la reconstrucción de la señal y la degradación del BER hace que este planteamiento resulte inviable de cara a una aplicación real.
- El sistema con ponderación temprana que presentó los mejores resultados fue el modelo entrenado con un 30 % de ataques. Este sistema, como se mencionó anteriormente, superó significativamente a los dos anteriores en la reconstrucción de la señal. El BER varía de un 0 % con un 0 % de ataques a un 0.81 % con un 30 % de ataques. Aunque el BER es ligeramente superior al del sistema original, este se mantiene por debajo del 1 %, lo que lo convierte en la mejor aproximación en términos globales.

- Aunque los rendimientos obtenidos en los tres sistemas evaluados son técnicamente aceptables, existe un considerable margen de mejora. Este potencial está limitado por el tamaño de nuestra base de datos, que contiene únicamente tres horas de audio, en comparación con la extensa base de datos de más de 800 horas utilizada en la referencia principal [1]. Además, las restricciones computacionales impuestas por la plataforma Google Colab han condicionado las fases de entrenamiento, validación y evaluación, limitando la duración y profundidad de los entrenamientos para adaptarse a las políticas de uso de la plataforma.

6.2. Trabajo Futuro

El trabajo presentado en esta memoria tiene un potencial de alcance significativo en el campo del procesamiento de señales y autenticación del contenido digital mediante técnicas de aprendizaje profundo y redes neuronales. La técnica propuesta de marcado de agua sobre señales de voz con redes neuronales aparenta tener la capacidad de ofrecer un balance optimizado entre impercetibilidad, robustez y capacidad en comparación con los métodos existentes [1].

De cara al trabajo futuro, hay varias direcciones que se podrían explorar y a las que se podría dedicar otro trabajo completamente diferente a este:

- **Implementación de nuevos ataques y evaluación de la robustez del sistema ante estos**: Aunque el sistema que se propone en este trabajo ya presenta una robustez más que aceptable para su correcto funcionamiento frente a los ataques más frecuentes en los sistemas de comunicación, el continuo desarrollo del presente proyecto para mejorar la resistencia frente a ataques más sofisticados, como los ataques de desincronización, sería crucial y algo bastante novedoso en este campo. El hecho de hacer frente a los ataques de desincronización, como en [25], y otros ataques emergentes nos obligaría a mejorar la capacidad de nuestro sistema.

- **Optimización y eficiencia:** Reducir en parte la complejidad computacional del sistema y mejorar la eficiencia del entrenamiento y de la operabilidad del sistema nos permitiría incluir su implementación en sistemas y plataformas de recursos limitados, como dispositivos móviles o sistemas embebidos.

- **Aplicaciones en tiempo real:** Adaptar el sistema implementado para abordar aplicaciones en tiempo real como transmisiones en vivo y comunicaciones seguras o firmas de documentos en tiempo real, como en [26].

- **Generalización a otros tipos de medio:** Aunque el enfoque actual está plenamente centrado en las señales de voz, el espectro de aplicación podría ampliarse a diferentes tipos de contenidos multimedia como música o vídeo, como el sistema presentado en [27]. Este enfoque podría expandir significativamente la utilidad de este proyecto

  .

Con este trabajo presentamos una base rígida para las posibles investigaciones futuras y nos permite establecer un marco referencial para explorar y expandir los hallazgos en el campo del *watermarking* basado en redes neuronales.

.

Capítulo 7

Presupuesto y Cronograma de Trabajo

En este capítulo abordaremos dos aspectos fundamentales para la planificación y desarrollo exitoso del trabajo expuesto en la presente memoria: el presupuesto y el cronograma de trabajo. La elaboración de cualquier proyecto requiere de un análisis presupuestario que dé una aproximación del valor económico del mismo. En la descripción del presupuesto especificaremos los recursos económicos necesarios, así como su justificación y distribución. El cronograma de trabajo también es una parte esencial ya que nos permite desglosar las etapas del proyecto, los plazos que se han de cumplir y los hitos alcanzados a lo largo del desarrollo del proyecto.

7.1. Presupuesto del Trabajo

En este apartado se exponen tanto los costes que se corresponden con los recursos humanos así como los costes de los materiales empleados para la confección de este trabajo.

7.1.1. Costes de Recursos Humanos

Para el cálculo del coste de los recursos humanos empleados para el desarrollo de este trabajo tenemos que contemplar, por una parte, el coste que supone un ingeniero senior experimentado, cuyas tareas principales van desde la revisión del trabajo hasta la dirección del mismo y, por la otra parte, el coste de un ingeniero junior, cuya tarea principal es desarrollar el proyecto según las indicaciones de los ingenieros senior experimentados.

El coste por hora de un ingeniero senior es de 65€, mientras que el coste por hora de un ingeniero junior es de 50€.

El cálculo del coste total se ha calculado haciendo un recuento de todas las horas empleadas por el alumno para desarrollar este trabajo al completo junto con las reuniones llevadas a cabo con los tutores, en las que se daban indicaciones de cómo debía de desarrollarse el proyecto. Aproximadamente todas las reuniones tenían una duración de una hora.

Concepto	**Ingeniero**	**Tiempo invertido (h)**	**Precio unitario (€ / h)**	**Total sin IVA (€)**
Formación y documentación	Junior	100	50	5000
Diseño e implementación de los sistemas de watermarking	Junior	230	50	11.500
Análisis de los resultados	Junior	25	50	1250
Redacción de la memoria	Junior	50	50	2500
Revisión de la memoria	Junior	20	50	1000
Revisión y dirección del trabajo (tutor 1)	Senior	68	65	4420
Revisión y dirección del trabajo (tutor 2)	Senior	68	65	4420
TOTAL		**561**		**27.590**

Cuadro 7.1: Desglose de los costes de recursos humanos.

En el cuadro 7.1 podemos ver un desglose del coste que ha supuesto la confección de este trabajo. Como podemos ver se han empleado un total de 561 horas para desarrollarlo al completo y ha supuesto un coste monetario de 27.590,0 euros.

7.1.2. Coste del Material

El presente proyecto ha sido desarrollado en la plataforma de programación colaborativa *Google Colab*. Esta plataforma presta a los usuarios una GPU en la nube de uso temporal que permite realizar entrenamiento de modelos durante un periodo de tiempo. Dado que este método era bastante limitado, el alumno optó por pagar la suscripción mensual de la plataforma, que daba libre acceso a las GPUs sin límite de uso. La suscripción tuvo un coste de 11.65 €/mes y se mantuvo durante 6 meses, por tanto, tuvo un coste total de 69,9 €. La base de datos utilizada para el entrenamiento de los modelos fue proporcionada por los tutores por lo que se asume un coste nulo por parte de la misma. La memoria fue escrita en la plataforma Overleaf, también gratuita y sin límite de uso.

Respecto al hardware utilizado para el desarrollo del trabajo, se hizo uso de un portátil **HP 15-fc0091ns**, con un procesador AMD Ryzen 5, 1 TB de almacenamiento y 16 GB de memoria RAM, obtenido por el alumno y con un coste de amortización de 181,74 euros.

Añadiendo todos estos costes al coste total humanitario concluimos que el coste total del proyecto es de **27.841,64 euros**.

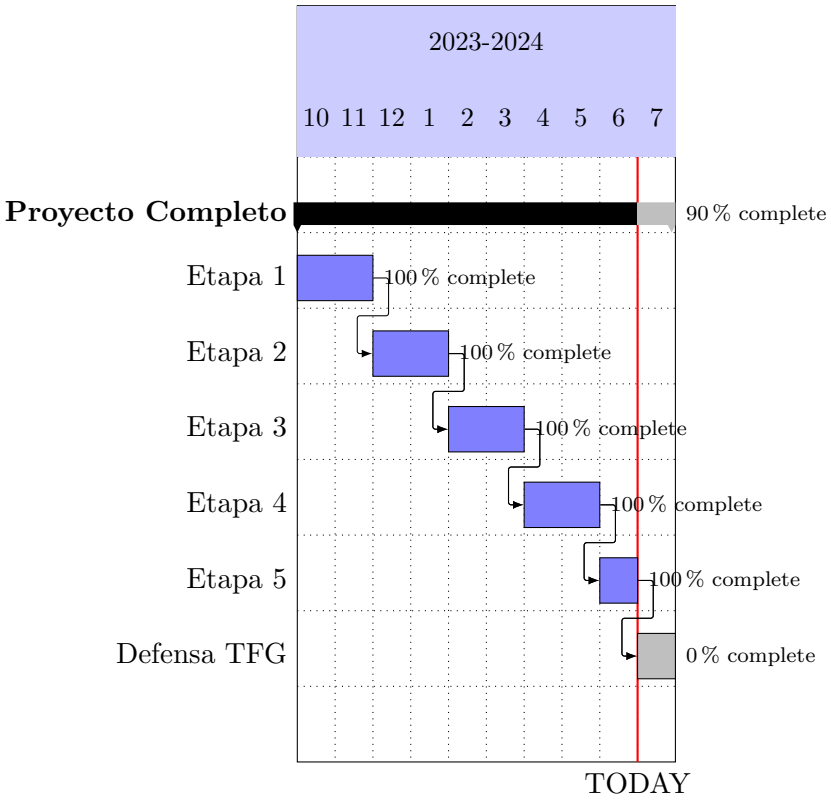
7.2. Cronograma de Trabajo

Al comienzo del desarrollo de este trabajo, hacia octubre de 2023, se estructuró y planificó el proyecto y las tareas que era necesario realizar para la confección y la consecución de los objetivos marcados por el trabajo. Este trabajo fue dividido esencialmente en 5 etapas diferenciadas.

- **Primera Etapa:** Esta primera etapa se desarrolló entre los meses de octubre y diciembre de 2023. En ella se realizó un estudio exhaustivo sobre el *watermarking*, con el fin de obtener una base de información sólida para desarrollar el trabajo sin problemas de concepto o teóricos.
- **Segunda Etapa:** Posterior a la recopilación de información, entre diciembre de 2023 y enero de 2024, se realizaron unos cursos introductorios al lenguaje de programación *Pytorch* sobre el que se desarrolla el código del trabajo realizado. El alumno se familiarizó con el entorno de trabajo Google Colab y con el lenguaje de programación utilizado para implementar redes neuronales
- **Tercera Etapa:** En esta etapa, entre febrero y mayo de 2024, se realizaron los diseños de las redes *Embedder* y *Detector* y se entrenaron

para los diferentes escenarios propuestos y se obtuvieron los resultados que se presentaron en capítulos anteriores.

- **Cuarta Etapa:** Entre los meses de mayo y junio de 2024 se redactó la memoria del trabajo y fue presentada a los tutores del mismo.
- **Quinta Etapa:** En las últimas semanas de junio, antes de la entrega, se hizo un análisis de la memoria redactada, corrigiendo aspectos de la misma e incluyendo partes nuevas que no se encontraban anteriormente.



Bibliografía

[1] Kosta Pavlović, Slavko Kovačević, Igor Djurović, and Adam Wojciechowski. Robust speech watermarking by a jointly trained embedder and detector using a dnn. *Digital Signal Processing*, 122:103381, 2022.

[2] Michael Arnold, Martin Schmucker, and Stephen D Wolthusen. *Techniques and applications of digital watermarking and content protection*. Artech House, 2002.

[3] Lukas Tegendal. Watermarking in audio using deep learning, 2019.

[4] Guang Hua, Jiwu Huang, Yun Q Shi, Jonathan Goh, and Vrizlynn LL Thing. Twenty years of digital audio watermarking—a comprehensive review. *Signal processing*, 128:222–242, 2016.

[5] William M. Fisher Jonathan G. Fiscus David S. Pallett Nancy L. Dahlgren Victor Zue John S. Garofolo, Lori F. Lamel. Timit acoustic-phonetic continuous speech corpus. *LDC93S1*, 1993.

[6] Qiu Yang, Yana Zhang, Cheng Yang, and Wei Li. Information entropy used in digital watermarking. In *2012 Symposium on Photonics and Optoelectronics*, pages 1–4. IEEE, 2012.

[7] Mahbuba Begum and Mohammad Shorif Uddin. Digital image watermarking techniques: a review. *Information*, 11(2):110, 2020.

[8] I.J. Cox, J. Kilian, F.T. Leighton, and T. Shamoon. Secure spread spectrum watermarking for multimedia. *IEEE Transactions on Image Processing*, 6(12):1673–1687, 1997.

[9] Ron G Van Schyndel, Andrew Z Tirkel, and Charles F Osborne. A digital watermark. In *Proceedings of 1st international conference on image processing*, volume 2, pages 86–90. IEEE, 1994.

[10] Walter Bender, Daniel Gruhl, Norishige Morimoto, and Anthony Lu. Techniques for data hiding. *IBM systems journal*, 35(3.4):313–336, 1996.

[11] Brian Chen and Gregory W Wornell. Quantization index modulation: A class of provably good methods for digital watermarking and information embedding. *IEEE Transactions on Information theory*, 47(4):1423–1443, 2001.

[12] Mauro Barni, Fernando Pérez-González, and Benedetta Tondi. Dnn watermarking: Four challenges and a funeral. In *Proceedings of the 2021 ACM Workshop on Information Hiding and Multimedia Security*, IHMMSec '21, page 189–196, New York, NY, USA, 2021. Association for Computing Machinery.

[13] Miguel Zamorano Castaño. Análisis de señales mediante stft y wavelet: aplicación a defectología en rodamientos. Master's thesis, 2010.

[14] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015.

[15] Xiao-Xia Yin, Le Sun, Yuhan Fu, Ruiliang Lu, Yanchun Zhang, et al. U-net-based medical image segmentation. *Journal of healthcare engineering*, 2022, 2022.

[16] Oleg Evsutin, Anna Melman, and Roman Meshcheryakov. Digital steganography and watermarking for digital images: A review of current research directions. *IEEE Access*, 8:166589–166611, 2020.

[17] Zihan Wang, Olivia Byrnes, Hu Wang, Ruoxi Sun, Congbo Ma, Huaming Chen, Qi Wu, and Minhui Xue. Data hiding with deep learning: A survey unifying digital watermarking and steganography. *arXiv preprint arXiv:2107.09287*, 2021.

[18] Francisco José Núñez Sánchez-Agustino. Diseño de un sistema de reconocimiento automático de matrículas de vehículos mediante una red neuronal convolucional. 2016.

[19] Juan Manuel Martin-Doñas, Angel Manuel Gomez, Jose A. Gonzalez, and Antonio M. Peinado. A deep learning loss function based on the perceptual evaluation of the speech quality. *IEEE Signal Processing Letters*, 25(11):1680–1684, 2018.

[20] ITU-T Recommendation. Perceptual evaluation of speech quality (pesq): An objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs. *Rec. ITU-T P. 862*, 2001.

[21] Hwai-Tsu Hu and Tung-Tsun Lee. Frame-synchronized blind speech watermarking via improved adaptive mean modulation and perceptual-based additive modulation in dwt domain. *Digital Signal Processing*, 87:75–85, 2019.

[22] Martin Steinebach, Fabien AP Petitcolas, Frederic Raynal, Jana Dittmann, Caroline Fontaine, S Seibel, Nazim Fates, and Lucilla Croce Ferri. Stirmark benchmark: audio watermarking attacks. In *Proceedings international conference on information technology: coding and computing*, pages 49–54. IEEE, 2001.

[23] Slami Saadi, Ahmed Merrad, and Ali Benziane. Novel secured scheme for blind audio/speech norm-space watermarking by arnold algorithm. *Signal Processing*, 154:74–86, 2019.

[24] Jonathan Le Roux, Scott Wisdom, Hakan Erdogan, and John R. Hershey. Sdr – half-baked or well done? In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 626–630, 2019.

[25] María Alejandra Menéndez Ortiz. *Esquema robusto ante ataques de desincronización para marcas de agua en audio.* PhD thesis, Instituto Nacional de Astrofísica, Óptica y Electrónica, 2012.

[26] Víctor Sebastián Ruiz Guerra and Sebastián Paúl Troya Lanas. Implementación del servicio web'matriculación a través de internet'con seguridad single sing on y watermarking para la carrera de ingeniería electrónica y redes de información. B.S. thesis, QUITO/EPN/2007, 2007.

[27] Xinwei Liu, Jian Liu, Yang Bai, Jindong Gu, Tao Chen, Xiaojun Jia, and Xiaochun Cao. Watermark vaccine: Adversarial attacks to prevent watermark removal. In *European Conference on Computer Vision*, pages 1–17. Springer, 2022.

[28] Patrick Bas and Teddy Furon. A new measure of watermarking security: The effective key length. *IEEE Transactions on Information Forensics and Security*, 8(8):1306–1317, 2013.

Acrónimos

DNN	*Deep Neural Network*
JND	*Just Noticeable Difference*
SS	*Spread Spectrum*
LSB	*Lateral Single Band*
QIM	*Quantified Index Modulation*
DT	*Dominio del Tiempo*
DF	*Dominio de la Frecuencia*
DWT	*Discrete Wavelet Transform*
DFT	*Discrete Fourier Transform*
FrFT	*Fractional Fourier Transform*
DCT	*Discrete Cosine Transform*
STFT	*Short Time Fourier Transform*
ISTFT	*Inverse Short Time Fourier Transform*
MAE	*Mean Absolute Error*
BCE	*Binary Cross Entropy*
SNR	*Signal to Noise Ratio*
SDR	*Signal to Distortion Ratio*
PESQ	*Perceptual Evaluation of Speech Quality*
STOI	*Short-time Objective Intelligibility*
BER	*Bit Error Rate*