UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería Informática

Implementación de algoritmos de detección de anomalías sonoras y su aplicación en el contexto de la TIE

Trabajo Fin de Grado

Grado en Ciencia de Datos

AUTOR/A: Soler Sanz, Guillem

Tutor/a: Llobet Azpitarte, Rafael

Cotutor/a externo: NARANJO ALCAZAR, JAVIER

CURSO ACADÉMICO: 2023/2024

Resumen

La Técnica del Insecto Estéril (TIE) es una técnica de control biológico de plagas que se basa en la liberación al medio ambiente de ejemplares estériles de la especie cuya población se desea controlar. Dentro de la Comunitat Valenciana la Conselleria de Agricultura, Ganadería y Pesca financia un programa de la TIE para el control de la población de la especie de mosquito Aedes albopictus, comunmente conocida como mosquito tigre. De esta manera, la TIE se basa en la cría masiva dentro de una biofábrica de la especie del insecto en cuestión. Una de las etapas más importantes dentro del proceso de cría dentro de la biofábrica es la separación por sexo, o sexado, de los ejemplares del insecto. Esto es así porque las hembras de la especie son las que pican a los seres humanos, con el consiguiente riesgo de transmisión de enfermedades como el Dengue o el Zika.

Dentro de este contexto, ITI lleva adelante el proyecto MoTIA3 financiado por IVACE con fondos FEDER. Este proyecto tiene como objetivo principal la incorporación de técnicas avanzadas de industria 4.0 basadas en Inteligencia Artificial (IA) para automatizar distintos puntos del proceso de cría masiva dentro de la biofábrica. En especial, la mayor parte del esfuerzo se ha centrado en la realización de tareas de I+D para la mejora de la etapa de sexado. Dentro de las mejoras del proceso de sexado se ha trabajado sobre una etapa de control de calidad basada en la detección de mosquitas hembra dentro de los recipientes en los que se encuentran los insectos previo a su liberación al medioambiente. Esta detección se basa en un sistema de audición por computador que permite detectar el sonido producido por el vuelo de las hembras dentro de dichos recipientes.

El presente trabajo centra su desarrollo en la implementación de modelos de IA para la detección de anomalías sonoras de forma tal que puedan ser aplicados a la detección de hembras de mosquito tigre mediante el sonido producido por el aleteo de su vuelo dentro de recipientes de reducido tamaño, siendo que el requisito es que dichos recipientes contengan solo ejemplares macho de la especie. La situación habitual es que hay entre 5 y 10 hembras por cada 200-250 ejemplares macho.

Palabras clave: Inteligencia artificial, audición por computador, Técnica del Insecto Estéril (TIE) ,aprendizaje semi-supervisado , detección de anomalías sonoras

---

Abstract

The Sterile Insect Technique (SIT) is a biological pest control method based on the release of sterile specimens of the species whose population is to be controlled into the environment. In the Valencian Community, the *Conselleria* of Agriculture, Livestock and Fishery, and Fisheries funds a SIT program to control the population of the mosquito species Aedes albopictus, commonly known as the tiger mosquito. This technique relies on the mass rearing of the insect species in a biofactory. One of the most critical stages in the rearing process within the biofactory is the sex separation, or sexing, of the insect specimens. This is crucial because the females of the species are the ones that bite humans, posing the risk of transmitting diseases such as Dengue or Zika.

In this context, ITI is carrying out the MoTIA3 project, funded by IVACE with FEDER funds. The main objective of this project is to incorporate advanced Industry 4.0 techniques based on AI to automate various points of the mass rearing process within the biofactory. Most of the effort has been focused on R&D tasks to improve the sexing stage. Among the improvements in the sexing process, work has been done on a quality control stage based on detecting female mosquitoes in the containers where the insects are kept before being released into the environment. This detection is based on a computer audition system that detects the sound produced by the flight of females within these containers.

This thesis focuses on implementing AI models for detecting sound anomalies to detect female tiger mosquitoes by the sound produced by their wingbeats within small containers. The requirement is that these containers should contain only male specimens of the species. The usual situation is that there are between 5 and 10 females per 200-250 male specimens.

Key words: Artificial intelligence, computer audition, Sterile Insect Technique (SIT), semi-supervised learning, sound anomaly detection

Índice general

Índice general v
Índice de figuras vii
Índice de tablas viii

1 Introducción 1
1.1 Introducción . . . 1
1.2 Motivación . . . 3
1.3 Objetivos del Trabajo . . . 3
1.4 Asignaturas relacionadas . . . 4
1.5 Estructura de la memoria . . . 4

2 Estado del arte 7
2.1 Estado del arte biológico . . . 7
2.1.1 La Técnica del Insecto Estéril (TIE) . . . 7
2.1.2 Dispositivo Fay-Morlan . . . 8
2.2 Estado del arte técnico . . . 9
2.2.1 Marco de la Inteligencia Artificial . . . 10
2.2.2 Aprendizaje Automático y Aprendizaje Profundo . . . 10
2.2.3 Fundamentos del Aprendizaje Automático . . . 11
2.2.4 Fundamentos del Aprendizaje Profundo . . . 13
2.2.5 Detección de anomalías . . . 17
2.2.6 Flujo de un sistema de Audición por Computador . . . 18

3 Problemática: Descripción y metodologia 23
3.1 Detalles Experimentales . . . 23
3.1.1 Configuración de Grabación . . . 23
3.1.2 Procedimiento de Grabación . . . 24
3.1.3 Conjunto de Datos . . . 25
3.1.4 Preprocesado de los datos . . . 27
3.1.5 Entorno de Trabajo y Herramientas Utilizadas . . . 28
3.2 Implementación y desarrollo del modelo . . . 28
3.2.1 Puntos clave SPADE . . . 29
3.2.2 Desarrollo del modelo SPADE . . . 29
3.2.3 Implementación del modelo . . . 33
3.2.4 Especificaciones técnicas . . . 35
3.2.5 Iteración del Algoritmo SPADE . . . 36
3.3 Entrenamiento y evaluación de los modelos . . . 37
3.3.1 Métricas de Evaluación . . . 37

4 Resultados 39
4.1 Resultados . . . 39
4.1.1 Análisis de los resultados . . . 39
4.1.2 Data Toy Car . . . 44
4.1.3 Análisis de los resultados . . . 44

5 Conclusiones y trabajo futuro 49

5.1 Conclusiones 49
5.2 Trabajo futuro 50

Apéndices

A Resultados **55**
A.1 Análisis de resultados 55
A.1.1 75-25 55
A.1.2 90-10 56
A.1.3 50-50 58

B Objetivos de Desarrollo Sostenible (ODS) **61**
B.1 Grado de relación del trabajo con los Objetivos de Desarrollo Sostenible (ODS) 61
B.1.1 Salud y Bienestar 62
B.1.2 Industria, Innovación e Infraestructuras 62
B.1.3 Vida de Ecosistemas Terrestres 62

Índice de figuras

1.1 Infografia del proyecto MoTIA. 2

2.1 Etapas biológicas del mosquito Aedes. [11] 8
2.2 Dispositivo Fay-Morlan. [15] 9
2.3 Tipos de aprendizaje automático. a) Aprendizaje semi-supervisado; b) Aprendizaje no supervisado; c) Aprendizaje supervisado [41]. 11
2.4 Representación de un SVM. [25]. 12
2.5 Arquitectura de una CNN.[37] 15
2.6 Diagrama de bloques del SRR compuesto por el *representation learner*, el refinamiento de datos y los bloques OCC finales.[46] 19
2.7 Arquitectura del modelo SPADE. [47] 20

3.1 (a) contenedor de liberación (b) ejemplares de mosquito adulto dentro del contenedor de liberación (c) configuración de grabación con la caja de aislamiento abierta (d) configuración final de grabación 24
3.2 Procedimiento de registro llevado a cabo 24
3.3 Base de Datos de los audios de los mósquitos. 25
3.4 Visualización de un audio en la aplicación *Audacity*. 26
3.5 Espectograma de bote con hembras y machos. Marcado con un rectángulo se puede apreciar frecuencialmente el aleteo de al menos una hembra. . . 26
3.6 Visualización de un *Mel-spectogram*. 27
3.7 *Autoencoder architecture* 30
3.8 Proceso de pseudo-labeling y determinación de umbrales utilizando la distancia de Wasserstein. 31
3.9 *Convolutional Neural Network (CNN) Encoder Architecture* 33
3.10 *Convolutional Neural Network (CNN) Decoder Architecture* 33
3.11 *Anomaly Predictor Architecture* 34
3.12 Diagrama del procedimiento iterativo del algoritmo SPADE. 36

4.1 Visualización UMAP de datos normales y anómalos etiquetados. 43
4.2 Visualización UMAP con datos etiquetados y las predicciones del modelo . 43
4.3 Evolución de las funciones de pérdida por cada epoch. 45
4.4 Visualización UMAP de distintas iteraciones del mismo modelo. 46
4.5 Evolución de las funciones de pérdida por cada epoch. 47
4.6 Visualización UMAP de distintas iteraciones del mismo modelo. 48

Índice de tablas

3.1 Resumen de Audios de Mosquitos 25

4.1 Distribución detallada de los distintos conjuntos de datos utilizados en los análisis. 40
4.2 Resultados variando el número de características. 40
4.3 Resultados con diferentes valores de *Alpha*(α) y *Beta*(β) 41
4.4 Resultados con diferentes números de características 41
4.5 Distribución detallada de archivos de audio en el conjunto de datos de ToyCar en el primer experimento. 45
4.6 Distribución detallada de archivos de audio en el conjunto de datos de ToyCar en el segundo experimento. 47

A.1 Resultados con diferentes números de características 55
A.2 Resultados con diferentes valores de *Alpha*(α) y *Beta*(β) 56
A.3 Resultados con diferentes números de OCCs utilizando 512 características 56
A.4 Resultados con diferentes números de características (90-10) 57
A.5 Resultados con diferentes valores de *Alpha*(α) y *Beta*(β) (90-10) 57
A.6 Resultados con diferentes números de OCCs utilizando 64 características (90-10) 58
A.7 Resultados con diferentes números de características (50-50) 58
A.8 Resultados con diferentes valores de *Alpha*(α) y *Beta*(β) (50-50) 59
A.9 Resultados con diferentes números de OCCs utilizando 64 características (50-50) 59

CAPÍTULO 1
Introducción

1.1 Introducción

La Técnica del Insecto Estéril (TIE) es una técnica de control biológico de plagas basada en la liberación masiva, mediante botes de liberación, al ambiente de insectos estériles cuya población se desea controlar. En este trabajo en concreto, se trata del mosquito *Aedes albopictus* (mosquito tigre). El proceso completo de la TIE (para el mosquito tigre) implica la cría masiva dentro de una biofábrica, la clasificación de los especímenes por sexo, la esterilización y la subsiguiente liberación de los machos estériles en el ambiente. En el caso del *Aedes albopictus* se evita la liberación de especímenes femeninos porque, a diferencia de los machos, las hembras pican, con el riesgo asociado de transmisión de enfermedades. En el caso de las biofábricas de mosquitos *Aedes* para la TIE, el punto crucial de todo el proceso es la separación por sexo. Este proceso se realiza actualmente mediante una combinación de dispositivos mecánicos (se asume que las hembras son más grandes que los machos) y sistemas de visión manuales gestionados por parte de técnicos de laboratorio. A día de hoy, hay una posibilidad elevada en términos biológicos de falsos negativos (hembras clasificadas como machos), por lo que una última etapa de verificación antes de su liberación en el ambiente sería una gran mejora del proceso. La verificación se pretende realizar en fase adulta y explotando la característica que la frecuencia en el aleteo de los mosquitos machos adultos es diferente al producido por las hembras (ver Figura 3.5). Así pues, en este TFG se pretende crear un modelo de audición por computador, que se encargue de aprovechar esta característica para detectar la presencia de hembras en contenedores antes de la liberación ambiental. Este documento presenta un estudio para la detección de hembras(con un enfoque de detección de anomalías) en recipientes de liberación de mosquitos *Aedes* para programas de la TIE utilizando técnicas de audición por computador.

Este TFG se concentra en la detección de hembras en contenedores designados para la liberación de machos estériles, una tarea crucial dentro del ámbito más amplio de control de plagas. Sin embargo, esta línea de investigación forma parte de un proyecto de I+D más amplio denominado MoTIA (*Aplicación de técnicas de industria 4.0 a la producción de mosquitos tigre para la Técnica del Insecto Estéril*), el cual se propone establecer un sistema integral para la gestión y control efectivo de procesos de la TIE empleando técnicas de Inteligencia Artificial (IA). Como se ilustra en la Figura 1.1, el proyecto abarca diversas fases, cada una dirigida a un aspecto particular de la técnica de la TIE, la fase que se va a abordar en este proyecto es la que aparece señalada en rojo: *Sistema de control de calidad mediante audición por computador basada en IA.*

Actualmente para el sexado, se utiliza un dispositivo de Fay-Morlan para la clasificación de los mosquitos según su tamaño asumiendo la hipótesis de que las hembras en esta-

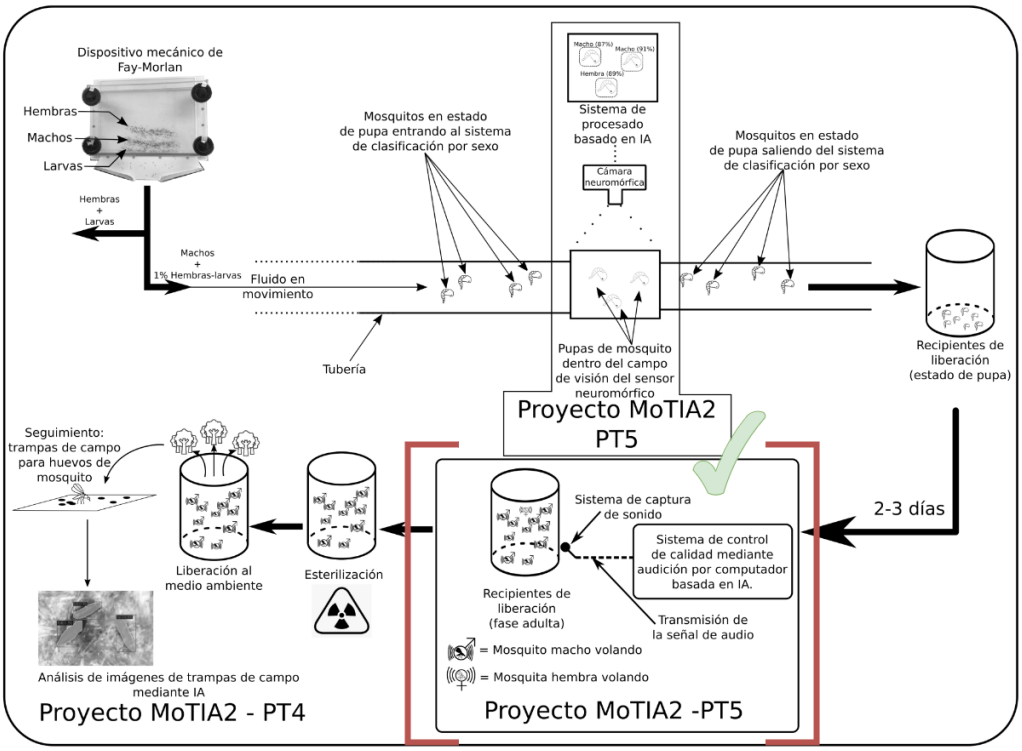


Figura 1.1: Infografia del proyecto MoTIA.

do de pupa son más grandes que los machos. Si bien existe una contaminación cruzada (hembras sexadas como mahcos), esta separación es esencial para el proceso subsiguiente, en el cual los insectos son dirigidos hacia un sistema para su tratamiento (esterilización) y liberación (en botes de 500 individuos). Cabe destacar, que a fecha de la presentación de este trabajo, el proceso de sexado se realiza únicamente con este dispositivo mecánico. El resto de procesos que se explican a continuación se encuentran en fase de investigación. El objetivo sería reducir el margen de error introducido por el dispositivo Fay-Morlan con distintas verificaciones en distintos puntos de la madurez de los insectos (pupa o fase adulta).

La primera fase en investigación introduce una innovación significativa mediante el uso de tecnología de visión por computador para la observación detallada de los mosquitos en fase pupa una vez han sido separados por el dispositivo Fay-Morlan. Para ello, se emplea una cámara neuromórfica para la toma de datos y un algortimo de visión por computador que se encarga de detectar y clasificar las pupas de mosquitos según su sexo. Este avance se concibe como una mejora al proceso de sexado del dispositivo Fay-Morlan.

Después de este análisis visual, los mosquitos son conducidos hacia contenedores específicos (en este punto, los contenedores sólo deberían estar compuestos por mosquitos machos), preparándolos para su liberación controlada en el entorno natural. Es en este momento donde se desea añadir un control de calidad basado en un sistema de audición por computador que sea capaz de detectar si los contenedores están únicamente compuestos por mosquitos machos. Para ello, se aprovecha el fenómeno de que el aleteo entre machos y hembras es distinto. Desde un punto de vista técnico, el reto se enfoca como una problema de detección de anomalías donde el aleteo de machos es el comportamiento normal.

Finalmente, tras la liberación de los machos esterilizados, el proyecto incorpora un sistema de análisis continuo de trampas de huevos de mosquitos y su análisis mediante

tecnología de visión por computador. El sistema analiza si los huevos presentes en la trampa han eclosionado o no. La no eclosión indica que la población de mosquitos se estaría reduciendo ya que los machos estériles estarían copulando con las hembras silvetres. Este último paso de control cierra el ciclo, permitiendo evaluar y ajustar las estrategias de control de poblaciones de mosquitos de una forma más eficiente que la realizada a día de hoy incorporando tecnologías punteras del estado del arte.

En conclusión, el presente trabajo se enfoca en el desarrollo de un modelo de audición por computador con técnicas de IA, diseñado específicamente para identificar la presencia de mosquitos hembras en botes de liberación y, de esta manera, prevenir su liberación accidental. Sin embargo, es importante destacar que este esfuerzo se enmarca dentro del proyecto MoTIA, una iniciativa que fusiona tecnología con estrategias biológicas para lograr un manejo sostenible y ético de las poblaciones de mosquitos. Más información sobre la TIE se presenta en la Sección 2.1.1

1.2 Motivación

El control de plagas de insectos es un desafío global que afecta la salud pública, la seguridad alimentaria y el medio ambiente. Las enfermedades transmitidas por mosquitos, como el dengue, la malaria y el virus del Zika, representan una amenaza significativa para la salud de la población mundial, especialmente en regiones tropicales y subtropicales. Además, las plagas agrícolas pueden causar pérdidas económicas significativas al destruir cultivos y reducir la producción de alimentos. En este contexto, la Técnica del Insecto Estéril (TIE) es considerada como una estrategia prometedora para el control de plagas, ofreciendo una alternativa sostenible y respetuosa con el medio ambiente en contrapartida con otros metodos de control de plagas como los pesticidas químicos.

La técnica de la TIE se basa en la liberación de insectos estériles (en nustro contexto sólo machos) de la especie objetivo en el medio ambiente, con el objetivo de competir con los machos fértiles y reducir la capacidad reproductiva de la población. Sin embargo, uno de los mayores desafíos en la implementación de la TIE sobre el mosquito tigre es garantizar que solo se liberen machos estériles en el medio ambiente, ya que la liberación de hembras podría comprometer la efectividad del programa y aumentar el riesgo de transmisión de enfermedades. Por lo tanto, es crucial desarrollar métodos precisos y eficientes para detectar y separar hembras en los procesos de producción y liberación de insectos estériles.

Este trabajo está motivado por la necesidad de abordar este desafío y contribuir al avance del conocimiento científico en el campo del control de plagas de insectos. Al mejorar la precisión y eficiencia de la detección y separación de hembras en los programas de la TIE, en el contexto del mosquito tigre, se pueden mejorar la efectividad y la sostenibilidad de los programas de control de plagas, lo que tiene el potencial de generar impactos significativos en la salud pública, la seguridad alimentaria y el medio ambiente.

1.3 Objetivos del Trabajo

El objetivo principal de este trabajo se centra en mejorar el proceso de la TIE añadiendo un nuevo punto de análisis mediante señales acústicas previo a la liberación de los insectos al medio ambiente. Este análisis acústico se puede entender como un control de calidad ya que se valida si el sexado previo, proceso que queda fuera del marco de este trabajo, se ha realizado correctamente. Los objetivos específicos incluyen:

1. Creación de una base de datos de audios de sonidos de aleteo de mosquitos *Aedes* que permita caracterizar el vuelo. Para ello, se realizaron grabaciones con botes de liberación con un ratio de 100 % machos (comportamiento normal) y un ratio del 25 % hembras y 75 % machos (comportamiento anómalo).

2. Implementación de un sistema de audición por computador para detección de anomalías sonoras. Teniendo como caso de uso la detección de la presencia de hembras en botes de liberación en el marco de la TIE aplicada al mosquito tigre. El análisis se puede definir como un control de calidad, ya que se va a realizar en un punto donde no debería haber hembras. Esto significaría un último paso de verificación, que pretende refinar el sexado de la máquina Fay-Morlan o el sistema de visión.

1.4 Asignaturas relacionadas

Los conocimientos adquiridos durante el grado de Ciencia de Datos en la Universitat Politècnica de València (UPV) jugaron un papel importante en la realización de este proyecto. Las habilidades desarrolladas en asignaturas como *Análisis exploratorio de datos* y *Visualización* contribuyeron significativamente al proceso de análisis y procesamiento de los datos. Este trabajo también se beneficia enormemente de la formación en programación adquirida en cursos como *Fundamentos de programación*, *Programación* y *Estructuras de datos*, entre otros, que proporcionan el conocimiento necesario para el desarrollo de proyectos como este.

La construcción de modelos de aprendizaje automático es otro aspecto importante del proyecto, respaldado por temas como *Modelos predictivos y descriptivos* y *Evaluación, despliegue y monitorización de modelos*. Además, el conocimiento adquirido en *Proyectos* y *Gestión de proyectos* es muy útil para la gestión de proyectos y los métodos de desarrollo de proyectos. Estos temas brindaron una base sólida para organizar y desarrollar proyectos de manera efectiva, demostrando la amplitud e importancia de la formación obtenida en el grado de Ciencia de Datos para realizar el trabajo.

1.5 Estructura de la memoria

La memoria de este trabajo está organizada de manera lógica y secuencial para guiar al lector a través de las diferentes etapas del estudio y desarrollo realizados. A continuación, se describe cómo está estructurado el documento y el contenido de cada sección principal.

La memoria comienza con una Introducción, donde se presenta el contexto general del trabajo. Esta sección incluye una introducción al tema, la motivación detrás del proyecto, los objetivos que se pretenden alcanzar y una breve descripción de las asignaturas académicas relacionadas con el contenido del trabajo. En este apartado también se explica la estructura de la memoria para proporcionar una visión general de cómo está organizado el documento.

El segundo capítulo está dedicado al Estado del Arte. Esta sección se divide en dos partes principales: el estado del arte biológico y el estado del arte técnico. En la primera parte, se discuten técnicas biológicas relevantes, como la Técnica del Insecto Estéril (TIE) y dispositivos específicos como el Dispositivo Fay-Morlan. En la segunda parte, se exploran aspectos técnicos, incluyendo el marco de la inteligencia artificial, los fundamentos del aprendizaje automático y profundo, así como técnicas de detección de anomalías y el flujo de un sistema de audición por computador. Esta revisión exhaustiva del conocimiento actual proporciona una base sólida sobre la cual se construye el trabajo.

El tercer capítulo se centra en la Problemática: Descripción y Metodología. Aquí se detallan los problemas específicos que se abordan en el trabajo y se describe la metodología utilizada. Se incluyen detalles experimentales como la configuración de grabación, el procedimiento de grabación, el conjunto de datos utilizado y el preprocesado de estos datos. Además, se describe el entorno de trabajo y las herramientas utilizadas. Esta sección también abarca la implementación y desarrollo del modelo SPADE, destacando puntos clave en su desarrollo, especificaciones técnicas y el proceso iterativo seguido en el desarrollo del algoritmo. Finalmente, se describe cómo se entrenan y evalúan los modelos, incluyendo las métricas de evaluación utilizadas.

En el cuarto capítulo se presentan los Resultados obtenidos. Esta sección incluye un análisis detallado de los resultados obtenidos de los experimentos y evaluaciones realizadas. Se analizan los datos obtenidos y se discuten en detalle los resultados relacionados con el conjunto de datos específico utilizado, como el 'Data Toy Car'.

Finalmente, la memoria concluye con un capítulo de Conclusiones y Trabajo Futuro. En esta sección se resumen los hallazgos principales y se presentan las conclusiones derivadas del trabajo realizado. Además, se proponen posibles direcciones para futuras investigaciones y mejoras que podrían realizarse en base a los resultados obtenidos.

Esta estructura está diseñada para proporcionar una narrativa clara desde el contexto y fundamentos teóricos hasta la implementación práctica y los resultados obtenidos.

CAPÍTULO 2
Estado del arte

2.1 Estado del arte biológico

En esta sección se proporciona un resumen del estado del arte actual de la Técnica del Insecto Estéril (TIE) para el control de plagas de insectos. Se presentarán los principios fundamentales de la TIE, así como las metodologías y dispositivos utilizados para la separación y liberación de insectos estériles, con un enfoque particular en la especie de mosquito *Aedes albopictus*.

2.1.1. La Técnica del Insecto Estéril (TIE)

La Técnica del Insecto Estéril (TIE) es una estrategia de control de plagas de insectos que se basa en la liberación de insectos estériles de la especie objetivo en el medio ambiente. El objetivo principal de la TIE es reducir la capacidad reproductiva de la población de insectos mediante la competencia entre los insectos estériles y fértiles. En este TFG se realiza el estudio sobre datos de la TIE aplicados al mosquito tigre (*Aedes albopictus*). En la TIE (en el contexto del mosquito tigre), se liberan al medio ambiente machos ya que las hembras son las que pican, y por tanto, son las que pueden transmitir enfermedades tan peligrosas como el Zika o el Dengue [33]. Al saturar el área con machos estériles, se reduce la probabilidad de que las hembras se apareen con machos fértiles y se produzcan descendientes, lo que eventualmente lleva a una disminución de la población de insectos objetivo [12]. La esterilización se realiza mediante la radiación de ejemplares criados en biofábricas [4].

La razón por la cual han surgido técnicas como la TIE, que buscan controlar la población de insectos, reside en la gestión de poblaciones de insectos que eventualmente pueden ser vectores de transmisión de enfermedades. La gran diferencia entre la TIE y el resto de estrategias de control de plagas es evitar el uso de insecticidas y otras sustancias tóxicas. Lo que convierte esta estrategia en una de las más respetuosas con el medio ambiente.

El sexado de los mosquitos desempeña un papel fundamental en los programas de control y erradicación de estas plagas en el contexto de la TIE aplicada al mosquito tigre. Este proceso de diferenciación sexual es crucial para asegurar que solo se liberen mosquitos machos, que no representan una amenaza directa para la salud pública. Después del sexado, los mosquitos son trasladados a contenedores de liberación. Nuestro trabajo se centra en analizar estos contenedores de liberación para verificar la ausencia de hembras; en caso de detectar alguna, el contenedor se descartará automáticamente. Este enfoque pretende garantizar que solo se liberen al exterior mosquitos machos [34].

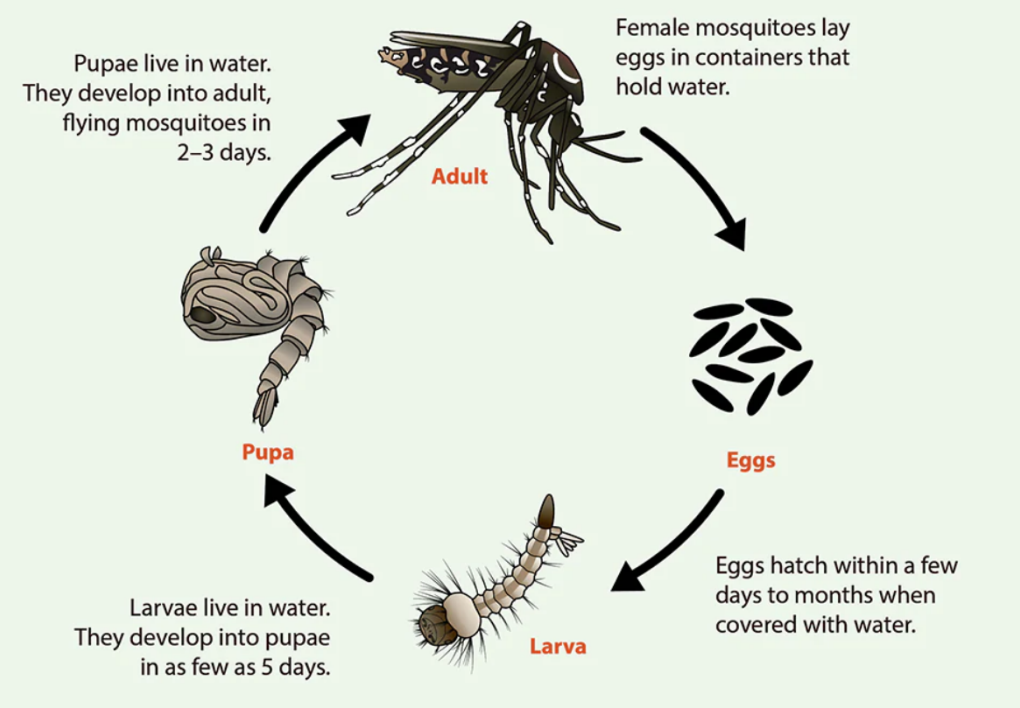


Figura 2.1: Etapas biológicas del mosquito Aedes. [11]

Este trabajo trabaja con señales de audio estando el mósquito *Aedes* en fase adulta. No obstante, en la Figura 2.1. se puede observar los diferentes estados por los que pasa el mosquito, importantes para entender las diferentes partes que forman el proyecto MoTIA[1]. Este proyecto además de incluir la verificación en fase adulta, incluye el sexado en fase de pupa y análisis de huevos en trampas de campo.

2.1.2. Dispositivo Fay-Morlan

El dispositivo Fay-Morlan es un instrumento utilizado en el proceso de producción de insectos estériles para separar machos de hembras en etapa de pupa, antes de que alcancen la etapa adulta (ver Figura 2.1). La separación de sexos en la etapa de pupa es crucial porque es en este momento cuando los insectos son estériles y aún no han alcanzado la capacidad de reproducirse [15].

La máquina de Fay-Morlan utiliza principios físicos para separar las pupas de machos y hembras en función de diferencias en el tamaño. Básicamente,tal y como se puede ver en la Figura 2.2, consiste en una placa de vidrio especial que tiene una superficie lisa y está recubierta con una capa de líquido. Las pupas se colocan en la placa de vidrio, y debido a diferencias en su tamaño y forma, las pupas de machos y hembras reaccionan de manera diferente al contacto con la superficie líquida. Esto permite que las pupas de machos y hembras se separen. Aunque por lo general las pupas hembras suelen ser más grandes, hay casos en los que no es así, por eso, a día de hoy, se cuelan hembras en la liberación, creando una contaminación cruzada. Desde un punto de vista técnico, existe la posibilidad de falsos negativos [8].

[1] `https://www.iti.es/proyectosidi/MoTIA-3/`

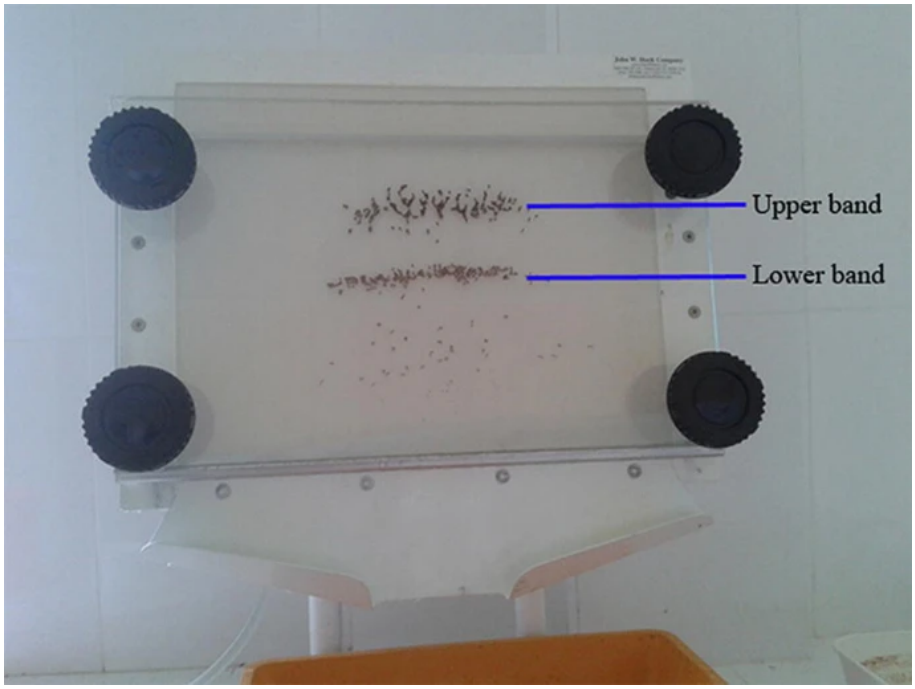


Figura 2.2: Dispositivo Fay-Morlan. [15] .

No obstante, la técnica de Fay-Morlan es una herramienta valiosa en la producción de insectos estériles para programas de control de plagas, ya que permite una separación de sexos con un grado de confianza aceptable a día de hoy. Además, al ser un método físico, no implica el uso de productos químicos o manipulaciones genéticas, lo que lo hace respetuoso con el medio ambiente y seguro para su aplicación en el control de plagas. Actualmente, el proceso de sexado de la TIE se realiza únicamente con este dispositivo. Así pues, existe la necesidad de refinar este proceso de sexado con el objetivo de automatizar y ampliar la producción de insectos estériles [15]. El proyecto MoTIA tiene como objetivo mejorar el proceso de sexado empleando técnicas de Industria 4.0 e Inteligencia Artificial. Para ello, se esta diseñando un algortimo de visión por computador para el sexado de pupas una vez han sido separadas por la máquina Fay-Morlan y un algoritmo de verificiación en fase adulta en el bote de liberación empleando señales acústicas.

2.2 Estado del arte técnico

La detección de anomalías representa un área de gran interés en el aprendizaje automático, ofreciendo aplicaciones transversales en sectores tan variados como el financiero para la detección de fraudes o el industrial para la identificación de defectos en procesos de fabricación. En el ámbito de la detección de anomalías, una característica distintiva es la presencia exclusiva de una clase representando el comportamiento normal en el conjunto de datos de entrenamiento, contrastando con la inclusión de anomalías únicamente en el conjunto de validación o test [7]. Cuando el conjunto de entrenamiento también incorpora ejemplos de anomalías, el problema se categoriza como de detección de anomalías semi-supervisada [35] (la cantidad de anomalías debe ser significativamente menor al comportamiento normal). En este escenario, el aprendizaje automático se posiciona como una herramienta robusta, destacando por su habilidad para extraer conocimientos a partir de los datos y aplicar estos aprendizajes para realizar predicciones o tomar decisiones fundamentadas en la información procesada. La etapa de validación que se implementa

en este TFG se implementa como un detector de anomalías donde el comportamiento normal es el aleteo de machos y la presencia de aleteo de alguna hembra la anomalía.

2.2.1. Marco de la Inteligencia Artificial

Actualmente, una amplia gama de soluciones tecnológicas integran métodos o técnicas de Inteligencia Artificial (IA). Aunque la aplicación es reciente, los fundamentos teóricos de la IA se establecieron ya en la década de 1940 [43]. La visión original se centraba en replicar el cerebro humano a través de la simulación del comportamiento neuronal. Sin embargo, hoy en día, la IA se ha infiltrado en prácticamente todos los sectores, transformando su aplicación y percepción completamente [36].

Dada la prevalencia de la Inteligencia Artificial en este proyecto, resulta pertinente ofrecer una descripción de la misma y explorar las diversas ramas que la componen. Esta comprensión nos permitirá apreciar mejor cómo la IA contribuye y potencia nuestras soluciones tecnológicas.

2.2.2. Aprendizaje Automático y Aprendizaje Profundo

El auge de la Inteligencia Artificial ha conducido al desarrollo de numerosos algoritmos que se pueden clasificar en diferentes categorías según su diseño. Por un lado, los modelos de Aprendizaje Automático, o *Machine Learning* (ML) en inglés, son algoritmos matemáticos que *aprenden* a través de los datos que se les proporcionan en fase de entrenamiento. Estos modelos están definidos por parámetros ajustables cuyos valores se modifican y calculan basándose en los datos de entrenamiento disponibles. Los algoritmos de ML no necesariamente se basan en redes neuronales; las Máquinas de Soporte Vectorial [5] y los Árboles de Decisión [16] son claros ejemplos de ello. Por otro lado, los modelos de Aprendizaje Profundo *(Deep Learning, DL)*, ampliamente difundidos, se fundamentan en redes neuronales [38] que cuentan con al menos una capa intermedia. Las soluciones de DL se basan en diversos tipos de redes neuronales, ya sean Redes Neuronales Profundas *(Deep Neural Networks* DNNs) [22], Redes Neuronales Convolucionales *(Convolutional Neural Networks* CNNs) [45] o Redes Neuronales Recurrentes (*Recurrent Neural Networks* RNNs). El diseño de las redes: número de capas intermedias, número de neuronas, activaciones, etc. viene determinado en parte por el conjunto de datos de entrenamiento.

Desde la perspectiva de su implementación, los algoritmos de Aprendizaje Automático (ML) mayoritariamente necesitan inicialmente un preprocesado y una limpieza de datos, conocido como ingeniería de características o *feature engineering* en inglés. En contraste, en el contexto de Aprendizaje Profundo (DL), este paso preliminar de ingeniería de características podría ser innecesario para muchas aplicaciones. En el marco de este TFG, ha sido necesario el uso de la ingeniería de características debido a que el audio ha sido procesado y transformado a una señal bidimensional mediante la extracción del espectrograma. Si bien se puede utilizar el propio audio para entrenar redes 1D [30], las señales 2D muestran mejores resultados. [9] Las CNNs son un tipo de red neuronal profunda capaces de extraer directamente las características (patrones) de la señal bidimensional (mayoritariamente imágenes) para posteriormente realizar la clasificación o detección. Por lo tanto, la implementación de algoritmos de DL puede ser considerablemente más rápida en dominios como la visión por computador. Sin embargo, en algunos campos específicos como el audio o la escucha automática, sigue siendo necesario un proceso de ingeniería de características (conversión de la señal de audio a una señal bidimensional).

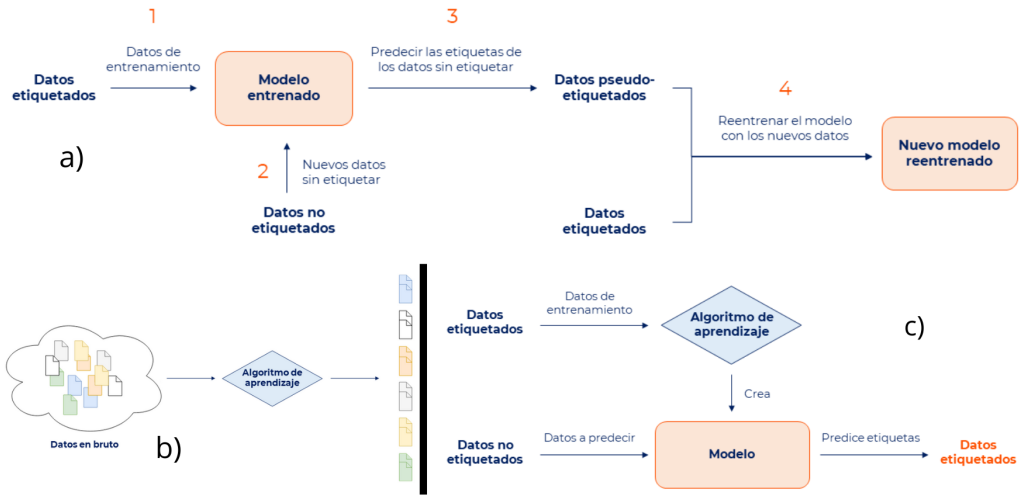


Figura 2.3: Tipos de aprendizaje automático. a) Aprendizaje semi-supervisado; b) Aprendizaje no supervisado; c) Aprendizaje supervisado [41].

Ambos algoritmos, ML y DL, pueden clasificarse de la siguiente manera según la información disponible en el conjunto de datos (ver Figura 2.3):

- Supervisado: los modelos son entrenados utilizando un conjunto de datos etiquetado, es decir, cada entrada de datos viene acompañada de la salida deseada (clase o etiqueta). El objetivo principal es aprender una función que, dada una entrada nueva, prediga la salida correspondiente. Este enfoque es común en tareas como la clasificación [16] y la regresión [27], donde el modelo utiliza los ejemplos de entrenamiento para generalizar a situaciones no vistas anteriormente.

- No supervisado: se utiliza cuando los datos de entrenamiento no están etiquetados. La meta es descubrir patrones subyacentes o estructuras en los datos. Las técnicas comunes incluyen el agrupamiento y la reducción de dimensionalidad. Este tipo de aprendizaje es útil para explorar la estructura de los datos y para encontrar agrupaciones naturales o características nuevas que describan los datos de manera más efectiva [41].

- Aprendizaje semi-supervisado: combina datos etiquetados y no etiquetados, aprovechando la información disponible en ambos para mejorar el rendimiento del modelo y reducir la necesidad de datos etiquetados, que suelen ser costosos de obtener [41] .

En el marco de este TFG, se ha trabajado con datos semi-supervisados. En la Sección 3.2.2 se presenta la implementación del algoritmo SPADE (*Semi-supervised Anomaly Detection under Distribution Mismatch*) cuyo objetivo consiste en la detección de anomalías en un conjunto de datos semi-supervisado.

2.2.3. Fundamentos del Aprendizaje Automático

El Aprendizaje Automático constituye una faceta esencial de la investigación contemporánea en inteligencia artificial, proporcionando las bases metodológicas y algorítmicas para muchos sistemas avanzados de análisis de datos. En el contexto de este proyecto,

se han empleado varias técnicas de aprendizaje automático, siendo las Máquinas de Vectores de Soporte (SVM) [44] una de las principales herramientas utilizadas debido a su eficacia en la clasificación y regresión en escenarios de alta dimensión.

SVM (Máquinas de Vectores de Soporte)

En el ámbito del aprendizaje supervisado, las Máquinas de Vectores de Soporte (SVM) constituyen una herramienta avanzada ampliamente utilizada para abordar problemas tanto de clasificación como de regresión. Los SVM operan bajo el principio de identificar un hiperplano que separa las clases de datos de manera óptima, donde la optimización se focaliza en maximizar el margen entre las clases más cercanas a este hiperplano. Este margen es crítico pues define la robustez del modelo frente a nuevas instancias de datos.

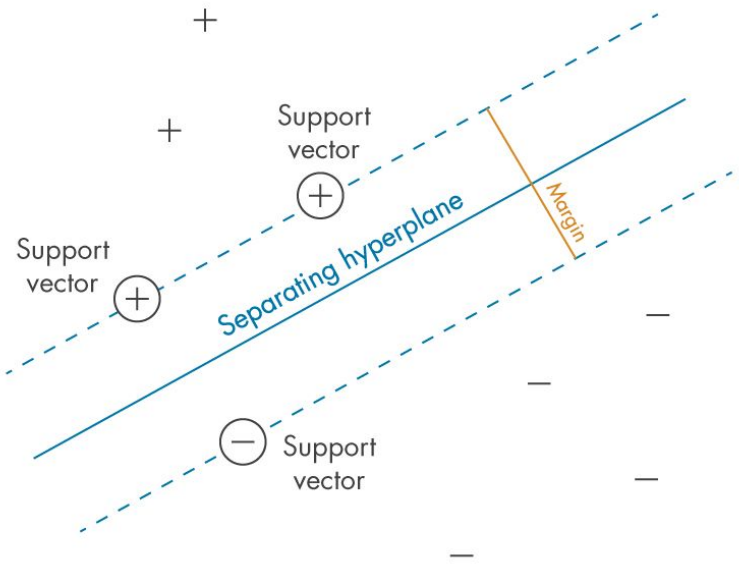


Figura 2.4: Representación de un SVM. [25].

El hiperplano de decisión en SVM se construye con la ayuda de las instancias denominadas vectores de soporte (ver Figura 2.4), que son aquellos puntos de datos que están más próximos al hiperplano y que efectivamente determinan su orientación y posición en el espacio de características. La selección de estos vectores es el resultado del proceso de entrenamiento, el cual involucra resolver un problema de optimización cuadrática.

Una de las fortalezas de las SVM es su versatilidad en el manejo de problemas no lineales a través del uso de funciones *kernel*, que proyectan los datos en espacios de mayor dimensionalidad donde la separación lineal es posible. Entre las funciones *kernel* más comunes se encuentran la lineal, la polinómica y la de base radial (RBF). La elección de la función *kernel* adecuada es decisiva y debe ser ajustada según la naturaleza del problema específico.

El rendimiento y la eficiencia computacional de los SVM los hacen adecuados para una variedad de aplicaciones prácticas, que van desde el reconocimiento de patrones visuales y auditivos hasta el procesamiento del lenguaje natural[2].

[2] `https://es.mathworks.com/discovery/support-vector-machine.html`

OCSVM (Máquinas de Vectores de Soporte de Una Clase)

En estrecha relación con las Máquinas de Vectores de Soporte (SVM) descritas anteriormente, las Máquinas de Vectores de Soporte de Una Clase (OCSVM, por sus siglas en inglés) se especializan en una variante del aprendizaje no-supervisado orientada hacia la detección de anomalías. A diferencia de las SVM tradicionales, que se enfocan en maximizar el margen entre dos o más clases de datos, las OCSVM están diseñadas para identificar y describir una sola clase de comportamiento normal [39].

El objetivo principal de las OCSVM es establecer un límite de decisión que separe los datos de la clase normal de todas las posibles anomalías, que no están presentes durante el entrenamiento. Esto se logra mediante la construcción de un hiperplano en un espacio de características transformado, de manera similar a las SVM, pero con el enfoque único de encapsular la mayoría de los datos de entrenamiento de la clase normal dentro de este límite, mientras se minimiza el riesgo de incluir puntos anómalos.

En nuestro contexto, las OCSVM son idoneas ya que como se ha comentado anteriormente se busca un modelo que nos permita detectar el aleteo de una hembra entre muchos machos. Por lo tanto, utilizaremos las OCSVM como parte fundamental de nuestro algoritmo SPADE (*Semi-supervised Anomaly Detection under Distribution Mismatch*), el cual se expondrá en la Sección 3.2.2 [42].

2.2.4. Fundamentos del Aprendizaje Profundo

Las arquitecturas de Aprendizaje Profundo utilizadas en este trabajo incluyen las Redes Neuronales Profundas (DNNs) y las Redes Neuronales Convolucionales (CNNs). Una DNN se define como una red neuronal que incluye múltiples capas intermedias ocultas. Por otro lado, una CNN es una arquitectura especializada para procesar datos con una estructura de rejilla, como imágenes, utilizando capas que emplean convoluciones para capturar características espaciales [14].

Ecuación de los nodos y funciones de activación

Un nodo es un punto de cálculo donde las entradas son ponderadas por un vector de pesos, que se ajustan durante la etapa de entrenamiento. Para lograr un comportamiento no lineal en la red, comúnmente se aplica una función de activación no lineal. Así, la operación matemática en un solo nodo se describe con la ecuación [14]:

$$z = f(a) = f(w \cdot x + b) = f\left(\sum_{i=1}^{n} w_i x_i + b\right)$$

donde x representa la entrada, w los pesos ajustables, b un sesgo ajustable conocido como *bias* y f denota una función de activación. Las funciones de activación utilizadas en este trabajo son las Unidades Lineales Rectificadas (ReLU) [28]. La activación ReLU evita el problema de desvanecimiento del gradiente, un fenómeno conocido que ocurre cuando el gradiente disminuye drásticamente conforme se propaga hacia atrás a través de la red. Además, son computacionalmente más eficientes, permitiendo que la red aprenda los parámetros mucho más rápido que otras activaciones. La ecuación de la función de activación ReLU es:

$$\text{ReLU}(a) = \text{máx}(0, a)$$

Otras funciones de activación muy importantes y que se han utilizado durante el proyecto son las llamadas sigmoidal y softmax [6]. La función softmax toma un vector de entrada con K posiciones y lo normaliza en una distribución que consiste en K probabilidades proporcionales a los exponentes de los números de entrada:

$$\text{softmax}(a)_i = \frac{e^{a_i}}{\sum_{k=1}^{K} e^{a_k}}$$

donde la entrada es el vector K-dimensional $[a_1 \dots a_K]$ y $\text{softmax}(a)_i$ es el valor de salida para el coeficiente vectorial i-ésimo. Softmax se utiliza en problemas de clasificación multiclase [14].

Por otro lado, la función sigmoidal mapea el rango $(-\infty, +\infty)$ al intervalo $[0, 1]$, transformando cualquier valor real en otro valor real que puede interpretarse como una probabilidad:

$$\text{sigmoid}(a) = \frac{1}{1 + e^{-a}}$$

donde $\text{sigmoid}(a)$ es el valor de salida correspondiente a cualquier valor real a en el rango $(-\infty, +\infty)$ [14].

Funciones de coste

Los valores de los parámetros de la red, es decir, los valores de los coeficientes lineales que componen los nodos de la red, se calculan mediante procedimientos de optimización iterativos que toman en cuenta el conjunto de muestras de entrenamiento y la salida de red esperada para esas muestras de entrada. La llamada función de coste se evalúa en cada iteración del procedimiento. Esta función es una representación matemática compleja de la diferencia entre la salida esperada de la red y la salida observada en cada iteración del procedimiento. La optimización estándar de coeficientes se basa en un descenso de gradiente iterativo destinado a encontrar un mínimo global de la función de coste.

Las funciones de coste utilizadas en este TFG constan de la función de error cuadrático medio (*mean squared error* MSE) [14] y la entropía cruzada binaria o *binary cross-entropy* en inglés (BCE) [27].

MSE es la elección más adecuada cuando se calcula el error en la predicción de valores reales. Su ecuación es:

$$L_{\text{mse}} = \frac{1}{N} \sum_{i=1}^{N} (X_i - \hat{X}_i)^2$$

donde N es el número de muestras de entrenamiento, y X_i y $\hat{X}_i$ representan los valores reales objetivos y predichos para la i-ésima muestra de entrenamiento, respectivamente.

En nuestro caso, un detector de anomalías, la entropía cruzada binaria (BCE) es una opción idónea para modelar este tipo de problemas. La capa de salida de nuestra red está compuesta por K neuronas independientes, cada una con una activación sigmoidal. Esto permite que cada neurona determine la probabilidad de que una instancia específica sea anómala. La función de coste BCE, L_{bce}, se define matemáticamente de la siguiente manera:

$$L_{\text{bce}} = -\frac{1}{NK} \sum_{i=1}^{N} \sum_{k=1}^{K} [y_{i,k} \log \hat{y}_{i,k} + (1 - y_{i,k}) \log(1 - \hat{y}_{i,k})]$$

donde N es el número de muestras, $y_{i,k}$ es una variable binaria que indica si la muestra i debe estar etiquetada con la etiqueta k y $\hat{y}_{i,k}$ es la probabilidad asignada por la red a la muestra i de estar etiquetada con la etiqueta k.

En la Sección 3.2.4 se detallan cómo estas funciones matemáticas son adaptadas para nuestro problema en concreto.

Introducción a las Redes Neuronales Convolucionales (CNN)

Las Redes Neuronales Convolucionales (CNN) representan una evolución significativa dentro del ámbito de las redes neuronales profundas, diferenciándose principalmente por su estructura única que utiliza *kernels* o filtros en lugar de neuronas tradicionales [23] (ver Figura 2.5). Estas redes están diseñadas para procesar datos de manera que emulen la percepción visual humana, lo que las hace particularmente efectivas para tareas que involucran imágenes y señales bidimensionales.

La capacidad de las CNN para procesar y analizar imágenes se ha extendido con éxito al campo del procesamiento de señales de audio mediante el uso de espectrogramas [21]. Estos son representaciones visuales de las frecuencias presentes en una señal a lo largo del tiempo, proporcionando una estructura bidimensional que es similar a la de las imágenes. Este paralelismo permite que las CNN extraigan características relevantes de los espectrogramas de manera eficiente.

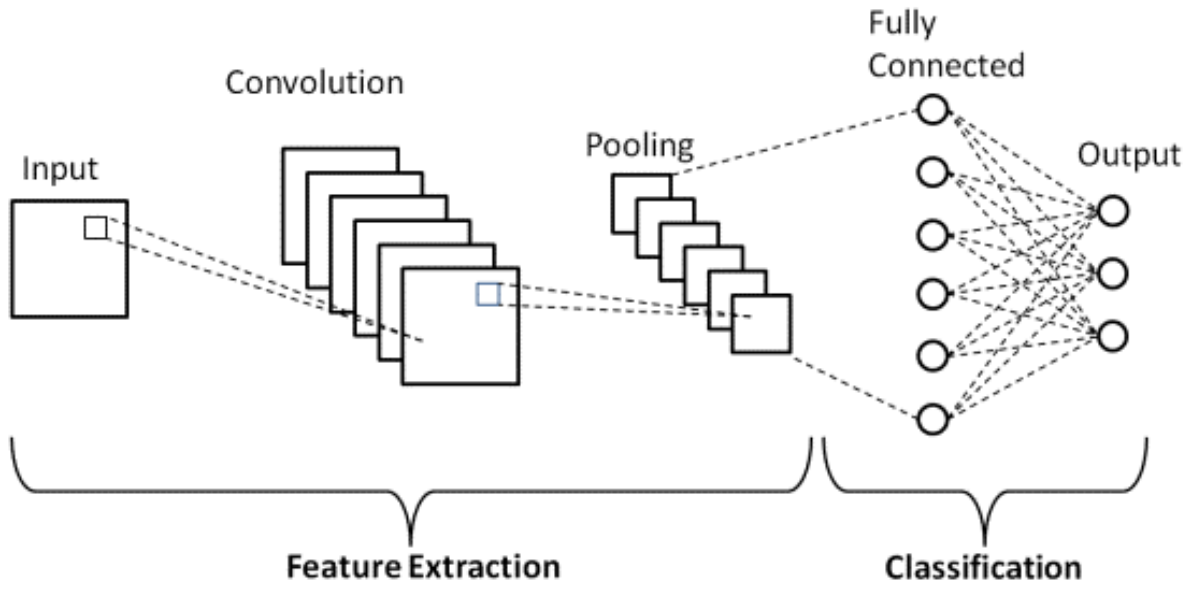


Figura 2.5: Arquitectura de una CNN.[37]

Al aplicar CNN a espectrogramas, se puede explotar su habilidad para identificar patrones locales y jerárquicos en los datos. En un espectrograma, las relaciones entre frecuencias adyacentes y cómo cambian a lo largo del tiempo son fundamentales para comprender fenómenos como el habla o la música [2]. Tal y como se puede ver en la Figura 2.5, la CNN mediante sus capas convolucionales y de agrupamiento (*pooling*), aprende filtros que responden específicamente a ciertas frecuencias y sus variaciones, lo que es ideal para tareas como el reconocimiento de voz [14], la detección de eventos sonoros [21] o incluso diagnósticos médicos basados en señales acústicas [14].

El entrenamiento de CNN para espectrogramas se lleva a cabo de manera similar al entrenamiento para imágenes, donde los *kernels* convolucionales se ajustan para detectar

características significativas en el dominio de la frecuencia. Las capas subsecuentes construyen una representación cada vez más abstracta de los datos originales, permitiendo a la red diferenciar entre diversas clases de sonidos o predecir parámetros acústicos con alta precisión.

En resumen, las CNN transforman la manera en que se interpreta y se procesa grandes conjuntos de datos visuales y auditivos, facilitando avances significativos en múltiples campos de la tecnología y la investigación.

Componentes de una CNN

En las capas convolucionales que forman las CNN, los coeficientes lineales de la ecuación son reemplazados por un *kernel* convolucional diseñado específicamente para identificar patrones particulares en la señal de entrada. Aunque se pueden derivar ecuaciones generales para las capas convolucionales, este trabajo se centra en datos de entrada 2D. Para un caso particular de entradas 2D y un único *kernel*, K, la ecuación sería:

$$z(x,y) = f\left(\sum_{i=1}^{kH}\sum j = 1^{kW} K_{i,j} a_{x+i-1,y+j-1} + b_{x,y}\right)$$

donde $z(x,y)$ es la salida del nodo, K es un *kernel* 2D de tamaño $kH \times kW$, a es la entrada 2D, b es un sesgo fijo y $f(.)$ es una función de activación.

Una capa convolucional se define por los siguientes parámetros:

- Tamaño del *kernel*: los tamaños más comunes para las CNN con entradas 2D son (3×3) o (5×5), aunque en algunas CNN se usan tamaños de (7×7) o (11×11).
- Salto o *Stride*: representa el desplazamiento o salto del filtro a medida que se mueve a través de la imagen de entrada. El stride más común es $(1,1)$.
- Número de canales (C): número de filtros que forman una capa convolucional.
- Relleno o *Padding*: determina cómo se procesan los bordes de la imagen. Si no se añade *padding*, el tamaño de la salida se reduce en comparación con la entrada. Para mantener el mismo tamaño entre entrada y salida, los bordes de la entrada deben ser rellenados, ya sea con ceros o replicando los valores del borde de la entrada, tantas veces como sea necesario según el tamaño del *kernel*.

Dado una entrada de tamaño $nH \times nW$ y un *kernel*, K, de tamaño $kH \times kW$, el tamaño del mapa de características, también conocido como mapa de salida, sería:

$$\text{tamaño del mapa de características} = \left(\left\lfloor \frac{nH + 2p - kH}{s} \right\rfloor + 1\right) \times \left(\left\lfloor \frac{nW + 2p - kW}{s} \right\rfloor + 1\right)$$

donde s denota el stride y p el padding, ambos supuestos iguales en ambas direcciones.

Además de las capas convolucionales, las CNN están compuestas por otros tipos de capas:

- Capas de normalización por lotes o *BatchNormalization* [19]: Esta capa permite estandarizar las características de entrada para tener una media cero ($\mu = 0$) y una varianza unitaria ($\sigma^2 = 1$) a través de múltiples observaciones. Consiste en añadir un paso de normalización dentro de una red neuronal profunda durante la alimentación de un lote (*batch*), dada por la siguiente fórmula:

$$BN(x) = \gamma \times \frac{x - \mu_B}{\sigma_B} + \beta \tag{2.1}$$

donde μ_B es la media y σ_B la desviación estándar para la distribución del *batch*. El resultado es una nueva distribución con media cero y desviación estándar igual a uno. Se recupera este grado de libertad incluyendo un parámetro de escala elemental γ y un parámetro de desplazamiento β que tienen la misma distribución que x. Ambos son parámetros que necesitan ser aprendidos como parte del entrenamiento.

En redes neuronales convolucionales, la normalización de lotes se aplica en una base por canal, lo que significa que la media y la desviación utilizadas para normalizar se obtienen de la distribución propia del canal y no de la distribución entre todos los canales juntos.

En general, la normalización de lotes ayuda al entrenamiento a converger más rápidamente y con una curva más suave.

- Capas de agrupamiento (*Pooling*): el número de filtros de las capas convolucionales suele aumentar a medida que se profundiza en la CNN. En el caso de que esta capa no reduzca la dimensionalidad (nH o nW), puede ser necesario que las últimas capas procesen matrices 3D del mismo ancho y alto que la entrada a la primera capa y con un aumento sustancial en la tercera dimensión. Dado que este procesamiento de datos no es factible, se requiere una capa de submuestreo, conocidas como capas de agrupamiento. Al igual que las capas convolucionales, tienen un campo receptivo de cierto tamaño conocido como tamaño del *pool* [21]. La reducción de la dimensionalidad se realiza tomando un único valor del receptivo (*MaxPooling*). Las opciones más comunes son tomar el valor más alto o el promedio. Además, existen agrupamientos globales donde el tamaño del pool es el ancho y la altura de la matriz tridimensional, logrando así un único valor por filtro (*GlobalPoling*).

- Capas de abandono (*Dropout*): esta idea implica introducir ruido mientras se calcula cada capa interna durante la propagación hacia adelante. Esto literalmente elimina algunas neuronas durante el entrenamiento, de ahí el nombre de *dropout*. A lo largo del entrenamiento, el *dropout* estándar consiste en dejar de usar algunos nodos antes de calcular la siguiente capa en la capa.

- Aplanamiento (Flatten): en muchas ocasiones, las CNN deben llevar a cabo tareas de clasificación. Esta tarea se lleva a cabo por capas completamente conectadas. Así, para lograr una representación unidimensional de una señal tridimensional ($H \times W \times C$), se realiza un remuestreo conocido como aplanamiento, donde todos los valores se colocan de manera unidimensional.

- Capas completamente conectadas: las primeras CNN implementaban capas completamente conectadas ya que se encargaban de establecer las relaciones entre los mapas de características generados por las capas convolucionales anteriores y la salida final de la clasificación. Esta tendencia sigue en uso [14]. Sin embargo, existe una tendencia paralela que implementa redes completamente convolucionales para la clasificación [21].

2.2.5. Detección de anomalías

Tal y como se ha expuesto con anterioridad, en este trabajo se utilizará la IA para diseñar un detector de anomalías de audio por computador, que a su vez servirá como un de

control de calidad. Esto es debido a que será el último paso antes de la liberación de los mosquitos *Aedes*, y servirá como última instancia para detectar una posible hembra antes de liberar el conjunto de machos al medio ambiente. La liberación se realiza en botes de 500 unidades. Así, muestras de audio que provengan de botes donde hayan 500 machos serán consideradas comportamiento normal. Por contrapartida, muestras de botes con algún porcentaje de hembras se consideraran anomalías [18].

A continuación, se van a exponer los distintas modelos que durante el proceso de trabajo se han considerado y/o se han llegado a implementar/utilizar.

2.2.6. Flujo de un sistema de Audición por Computador

La audición por computador o *Machine Listening* en inglés, se refiere al estudio y desarrollo de algoritmos capaces de extraer información relevante de datos de audio. Las aplicaciones de la audición por computación abarcan desde la clasificación de eventos acústicos [13] hasta la detección de anomalías [3], adaptándose a diferentes tipos de señales de audio como música [3] o sonidos ambientales [13].

Como ya se ha comentado con anterioridad, las Redes Neuronales Convolucionales (CNN) han demostrado ser particularmente eficaces para trabajar con datos estructurados espacialmente como imágenes y, por extensión, espectrogramas de audio. Estos últimos representan una señal de audio en dos dimensiones, donde una dimensión corresponde a la frecuencia y la otra al tiempo. La capacidad de las CNN para detectar patrones locales y jerárquicos en los espectrogramas las hace ideales para tareas como el reconocimiento de voz y la detección de eventos sonoros, aprovechando su habilidad para aprender filtros que responden a características específicas de frecuencia y sus variaciones a lo largo del tiempo.

Al aplicar las CNN a espectrogramas, se utiliza su capacidad de identificar patrones y características clave dentro de estructuras complejas. En el dominio del audio, esto incluye la habilidad para discernir entre diferentes tipos de sonidos, reconocer patrones de habla o diagnosticar condiciones a partir de señales acústicas en aplicaciones médicas. Las CNN procesan estos espectrogramas mediante una serie de capas convolucionales y de agrupamiento, construyendo una representación cada vez más abstracta de la señal original y permitiendo clasificaciones precisas o predicciones acústicas detalladas.

La estructura de nuestra parte del proyecto englobada dentro del marco de MoTIA se puede describir en varias etapas clave, desde la captura del audio hasta la notificación al usuario:

1. **Captura de Audio:** La primera etapa implica la captura de audio a través de micrófonos, donde el rango dinámico y la tasa de muestreo determinan la calidad y resolución del audio capturado.

2. **Segmentación:** Seguidamente, el audio es segmentado para su procesamiento, utilizando técnicas que pueden incluir la selección previa del audio antes de su segmentación.

3. **Extracción de Características:** Tradicionalmente, se transforma la señal de audio en una representación espectro-temporal antes de aplicar técnicas de clasificación. Sin embargo, enfoques más recientes alimentan señales de audio directamente a las CNN [17], que actúan tanto como extractores de características como clasificadores.

4. **Sistema de IA:** Las redes CNN procesan la entrada de audio transformada para extraer información relevante, empleando su capacidad para identificar patrones complejos dentro de los datos.

5. **Notificación al Usuario:** Finalmente, los resultados son comunicados al usuario a través de notificaciones visuales o alertas sonoras, completando el ciclo de interacción de la aplicación de escucha de máquinas.

Este enfoque integrado no solo mejora la capacidad de procesamiento y análisis de audio sino que también abre nuevas posibilidades en la interacción inteligente entre humanos y máquinas en entornos variados. Más adelante en la Sección 3.1 se expondrá de manera detallada cómo ha sido llevada a cabo cada una de las distintas partes que se mencionan [29].

Self-Supervise, Refine, Repeat (SRR)

El algoritmo SRR hace referencia a un paper [46] en el que se expone una metodología iterativa para la detección de anomalías en la que los modelos se auto-supervisan, se refinan a través de la retroalimentación de sus predicciones y se repiten estos pasos para mejorar continuamente su rendimiento. Esta técnica fue utilizada en el anterior proyecto de MoTIA. Este modelo fue desarrollado antes del inicio del TFG. Sin embargo, es usado como punto de partida para el modelo final.

El método SRR propuesto para la detección de anomalías sin supervisión aborda la problemática inherente a la detección de anomalías en conjuntos de datos no etiquetados. En el núcleo de la propuesta SRR reside la necesidad de mejorar la robustez de los clasificadores de una sola clase (*One-Class Classifier*, OCCs), los cuales son ampliamente utilizados en este dominio. A diferencia de los enfoques tradicionales que asumen conjuntos de datos completamente negativos (es decir, sin anomalías), SRR aborda de manera directa la problemática de conjuntos de datos que pueden contener tanto muestras normales como anómalas. Esto se logra mediante un proceso de refinamiento de datos iterativo basado en un conjunto de OCCs.

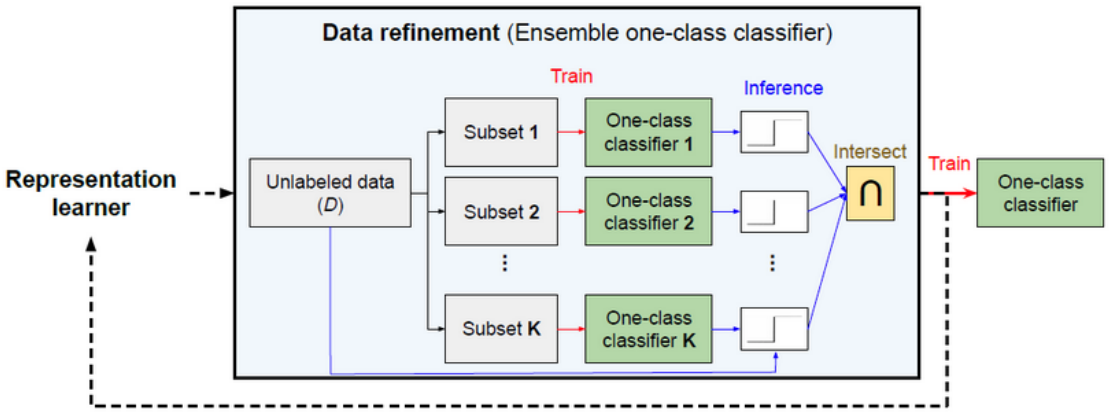


Figura 2.6: Diagrama de bloques del SRR compuesto por el *representation learner*, el refinamiento de datos y los bloques OCC finales.[46]

El proceso de refinamiento de datos de SRR se fundamenta en la generación de *pseudo-labels* para el conjunto de datos de entrenamiento no etiquetado. En lugar de utilizar un único clasificador para este fin, SRR emplea un conjunto de clasificadores OCCs, cada uno entrenado en subconjuntos disjuntos de datos de entrenamiento. La combinación de las predicciones (sobre todo el conjunto) de estos clasificadores permite la generación de *pseudo-labels*más robustas y precisas. Es decir, si bien cada OCC se entrena sobre un conjunto de los datos de entrenamiento, los *pseudo-labels* se obtienen de la concordancia de

todos o varios OCCs sobre todo el conjunto de entrenamiento. Esa es la idea fundamental tras el SRR. Se considerará una anomalía si todos los OCCs (entrenados con subgrupos distintos) consideran que es una anomalía. Una representación del algoritmo se puede visualizar en la Figura 2.6.

Además, SRR integra el aprendizaje de representaciones autodirigido en el proceso de refinamiento de datos, pero con una adaptación clave para su aplicación en la detección de anomalías en audio. En nuestra implementación, se emplea el modelo *Bootstrap Your Own Latent for Audio* (BYOL-A) [32]. Este modelo se considera un extractor de características diseñado específicamente para audio, basado en el aprendizaje auto-supervisado o *self-supervised* en inglés.

Una característica distintiva de SRR es su capacidad para ajustarse a diferentes proporciones de anomalías en los conjuntos de datos de entrenamiento. Esto se logra mediante la modulación de parámetros clave, como el número de clasificadores de una clase (OCC) en el conjunto y el umbral de decisión utilizado para la generación de etiquetas pseudo-anómalas.

En resumen, al incorporar BYOL-A para la extracción de características de audio, el método SRR ofrece una solución innovadora para la detección de anomalías sin supervisión en el ámbito del audio. Este enfoque combina el refinamiento iterativo de datos con el aprendizaje de representaciones autodirigido y la utilización de un conjunto de clasificadores OCC, destacándose por su robustez y capacidad para adaptarse a una variedad de condiciones y escenarios de aplicación en el mundo real.

Semi-supervised Anomaly Detection under Distribution Mismatch (SPADE)

Para la realización de este trabajo se decidió explotar la posibilidad de disponer de botes de liberación con un porcentaje de hembras. Es decir, disponer de una cantidad (aunque reducida) de comportamientos anómalos etiquetados. Esto es posible gracias a la empresa colaboradora TRAGSA del proyecto MoTIA. Esta empresa se encarga del proceso biológico de la TIE y de la crianza de mosquitos. Así pues, se decidió implementar una mejora de SRR semi-supervisada. El algoritmo es conocido como SPADE. Este se presenta como un modelo robusto al cambio de dominio (la distribución de los datos cambia durante el tiempo) y este es un fenómeno presente en el proyecto MoTIA ya que el aleteo de los mosquitos varía por múltiples factores como la época del año, estado de madurez, alimentación, etc. A través de estem modelo, se aborda este desafío crítico pero poco explorado del desajuste de distribución.

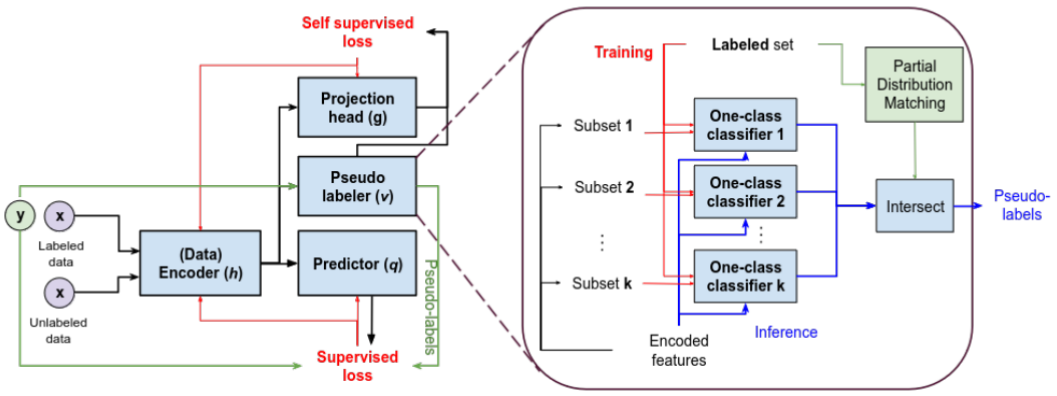


Figura 2.7: Arquitectura del modelo SPADE. [47]

Otro factor clave en la elección de SPADE fue la similitud entre el funcionamiento de este y el funcionamiento de SRR, dado que este se había implementado con anterioridad. Este funcionamiento nos permite abordar de una manera eficaz la escasez de datos etiquetados mediante una metodología que ajusta dinámicamente los hiperparámetros críticos sin depender de conjuntos de validación externos. Esta capacidad para autoajustarse y su desempeño superior en la identificación de anomalías en datos complejos ilustran por qué SPADE ha sido preferido sobre otras metodologías. La Figura 2.7 muestra el diagrama de bloques del modelo. La implementación del modelo, y cada uno de los bloques que lo componen, se explica en la Sección 3.1.

Con el desarrollo de SPADE se espera mejorar la precisión y eficacia de la detección de anomalías sino también en la adaptabilidad y escalabilidad del sistema frente a variados desafíos de datos.

CAPÍTULO 3
Problemática: Descripción y metodologia

En esta sección se expondrán los distintos pasos realizados para desarrollar el proyecto. Se empezará exponiendo los detalles experimentales, que incluyen el proceso de obtención de los datos de audios: su método de grabación, el etiquetado manual subsiguiente y la creación de la base de datos con la información relevante sobre los mismos.

A continuación, se profundizará en el modelo SPADE y su implementación.Se anaizarán los puntos clave del modelo, así como su desarrollo y las diversas arquitecturas que lo componen.

Por último, se presentará una ejecución del algoritmo con el fin de exponer claramente su funcionamiento.

3.1 Detalles Experimentales

3.1.1. Configuración de Grabación

El equipo de grabación consta de 3 componentes principales:

- Contenedor de liberación: consiste en un diseño tubular de dos piezas donde la pieza externa es un tubo de conexión de PVC de 8.8 cm de diámetro y 12.5 cm de altura, mientras que la pieza interna consta de ocho tiras radiales de metacrilato interconectadas por dos anillos que encajan perfectamente en la pieza externa. Este contenedor se puede visualizar en la Figura 3.1-a). Cada recipiente se llena con 250 mosquitos para la experimentación en este TFG. La Figura 3.1-b) muestra varios mosquitos dentro de uno de los contenedores de liberación utilizados en el experimento.

- Dispositivo de grabación: el micrófono utilizado es el ReSpeaker 4-Mic Array[1] que se coloca debajo del contenedor. El propósito de utilizar una matriz de micrófonos es obtener una mejor información espacial, ya que un mosquito puede estar volando desde un lado o esquina particular. Con esta configuración, se obtiene un mayor contexto espacial del aleteo de los mosquitos. La frecuencia de muestreo durante el proceso de grabación es de 44.1 kHz. Como se explica en la Sección 3.1.4, la señal se convierte a mono promediando los 4 canales.

[1] https://wiki.seeedstudio.com/ReSpeaker_4_Mic_Array_for_Raspberry_Pi/

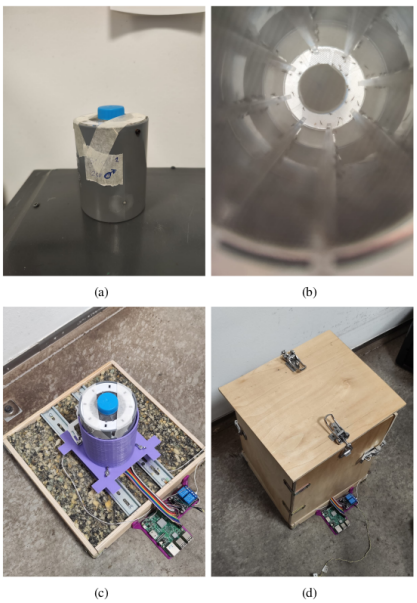


Figura 3.1: (a) contenedor de liberación (b) ejemplares de mosquito adulto dentro del contenedor de liberación (c) configuración de grabación con la caja de aislamiento abierta (d) configuración final de grabación

- Caja de aislamiento: se ha construido una caja de madera para aislar el sonido del aleteo de los mosquitos del resto de fuentes de audio presentes en la sala de grabación. La imagen 3.1-c) muestra el interior de la caja. El contenedor de liberación se coloca en el centro de la caja de madera y se sujeta en un soporte impreso en 3D. Debajo del contenedor, se coloca la matriz de micrófonos. La imagen 3.1-d) muestra la configuración final en el momento de la grabación. La Raspberry Pi permanece fuera de la caja para evitar cualquier ruido eléctrico producido por ella.

3.1.2. Procedimiento de Grabación

El proceso de grabación se llevó a cabo en las instalaciones dedicadas al proceso de la TIE de la Comunitat Valenciana. Los mosquitos fueron criados y separados por sexo en etapa de pupa por un grupo de expertos siguiendo los protocolos actuales.

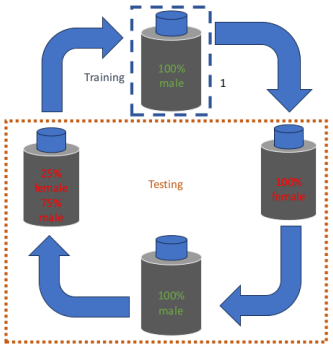


Figura 3.2: Procedimiento de registro llevado a cabo

El orden de los contenedores durante la grabación fue el siguiente: contenedor de machos (conjunto de entrenamiento), contenedor de hembras, contenedor de machos (para prueba) y contenedor mixto (25 % de hembras y 75 % de machos).

El ciclo de grabación se realizó dos veces, produciendo un total de 16 grabaciones de 30 segundos por recipiente por día. Se graban dos lotes de 8 clips en lugar de 16 de una vez para evitar que los mosquitos se acostumbren al equipo. Se comprobó empíricamente que después de 4 minutos dentro del equipo, los mosquitos se habituaban y volaban mucho menos.Para una mejor comprensión, la Figura 3.2 muestra un ejemplo del proceso de grabación diario.

3.1.3. Conjunto de Datos

El conjunto de datos utilizado para este estudio consta de 2 horas y 8 minutos de grabación. La grabación de cada contenedor corresponde a un total de 32 minutos. Por lo tanto, hay 8 minutos de grabación por contenedor por día. Cabe destacar que las sesiones de entrenamiento son independientes entre días, ya que los mosquitos pueden variar en su comportamiento, debido a que como se observa en la Figura 2.1 la vida de los mósquitos es muy corta y de un día para otro puede haber diferencias de vitalidad/energía en su aleteo según en que fase de su corta vida se encuentren.

Categoría	Número de Audios	Porcentaje
Normal	1922	73.8 %
Anomalía	682	26.2 %
Total	**2604**	**100 %**

Tabla 3.1: Resumen de Audios de Mosquitos

Cada sistema ha sido entrenado en consecuencia utilizando parte de los minutos disponibles en el contenedor de entrenamiento . Cada clip tiene una duración de 30 segundos, de estos clips se ha segmentado manualmente 6 segmentos de 5 segundos sin solapamiento. Por lo tanto, el conjunto de entrenamiento consiste en 2604 clips de duración de 5 segundos (ver Tabla 3.1).

Con el próposito de cumplir el objetivo específico de crear una base de datos (ver Figura 3.3) con la información pertinente de cada uno de los audios y su correspondiente etiqueta (anómalo o normal), se ha llevado a cabo un meticuloso proceso de análisis de datos acústicos de mosquitos *Aedes*. Este trabajo se ha enfocado mayormente en distinguir entre los patrones de vuelo de machos y hembras.

	id_container [PK] character varying	nmale character varying	nlady character varying	observations character varying
1	0101	250	0	todo machos
2	0201	250	0	"todo machos"
3	0301	250	0	todo machos y se percibio una hembra que se comprobo que habia
4	0401	250	0	todo machos
5	0402	0	250	todo hembras
6	0403	125	125	mix 50%
7	0404	60	190	mix 75% hembras
8	0405	190	60	mix 25% hembras

Figura 3.3: Base de Datos de los audios de los mósquitos.

Cabe destacar que la preparación del conjunto de datos ha sido fundamental para el desarrollo de un modelo predictivo. Cada audio fue grabado en condiciones controladas para garantizar la presencia de solo mosquitos machos o de ambos, según el caso. No

obstante, este trabajo podía tener errores ya que realizar una separación perfecta entre machos y hembras, en estos momentos, es un proceso laborioso e incierto. Es por eso que una vez grabados los audios había que hacer un posterior análisis de los espectogramas para cercionarse de que en los audios no había errores. Para garantizar este análisis, se realizó todo el proceso de segmentación de los audios de manera manual, los audios de 30 segundos fueron divididos de manera manual en intervalos de cinco segundos. Este procedimiento detallado fue crucial para asegurar la precisión en la clasificación y etiquetado de cada muestra.

El análisis individualizado de los segmentos permitió identificar con precisión la presencia de errores en los datos, que incluían la presencia de hembras en botes donde a priori solo había machos y ruidos de fondo indeseados como conversaciones y vibraciones mecánicas. Estos factores podrían haber confundido al modelo, por lo que su identificación y exclusión fueron esenciales. El etiquetado de los datos se realizó con extremo cuidado, asignando la etiqueta de *normal* a los audios que solamente evidenciaban la actividad de los machos y *anomalía* cuando se detectaba la presencia de hembras. Para el correcto etiquetado de los audios se ha utilizado la herramienta *Audacity* [2]. Esta herramienta nos permite visualizar de manera clara los cuatros canales que componen cada señal de audio(ver Figura 3.4), de esta manera se pudo realizar una separación fiable y precisa de los distintos segmentos del audio.

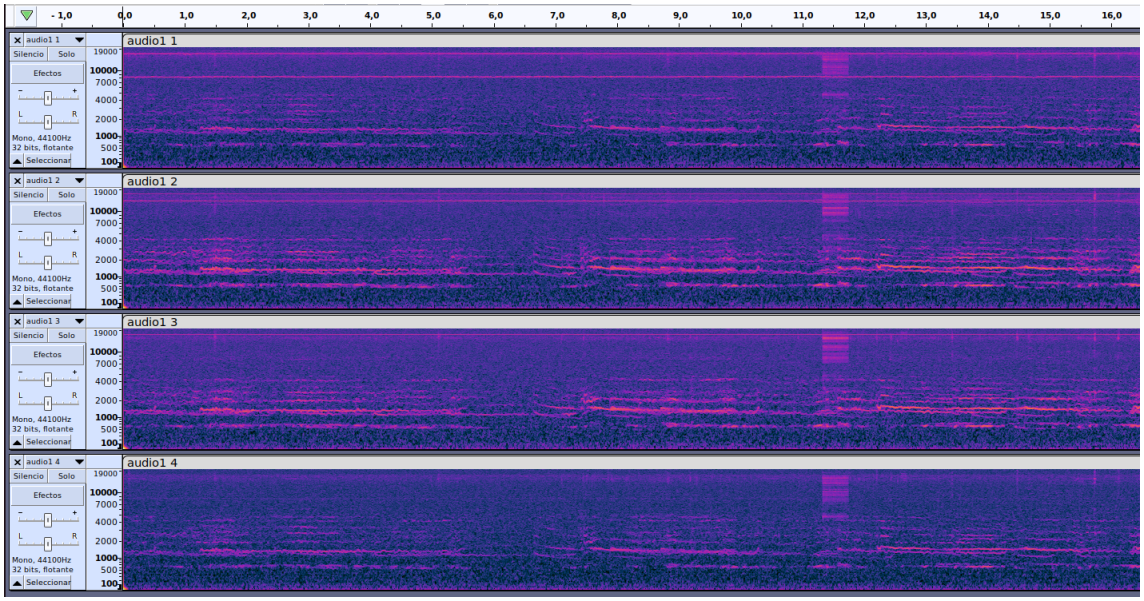


Figura 3.4: Visualización de un audio en la aplicación *Audacity*.

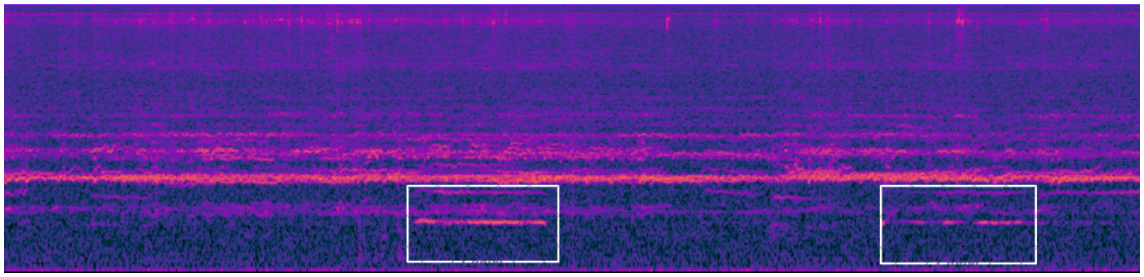


Figura 3.5: Espectograma de bote con hembras y machos. Marcado con un rectángulo se puede apreciar frecuencialmente el aleteo de al menos una hembra.

[2]https://www.audacityteam.org/

Es importante destacar que en los botes donde coexistían machos y hembras, había que asegurarse mediante la observación del espectograma que la hembra volaba. Ya que, en el caso de que la hembra se encuentre en el bote pero su aleteo no se vea reflejado en el espectrograma, se añadirían falsas anomalías al modelo. En la Figura 3.5 se puede observar como la línea discontinua que aparece en la parte de abajo del spectograma (resaltada con un círculo) corresponde a la frecuencia de aleteo de al menos una hembra. En este caso se observa que el aleteo no es constante, esto es otro factor que conlleva que los audios necesiten de una revisión exhaustiva.

Como resultado de este riguroso proceso, tal y como se puede ver en la Tabla 3.1, se compiló un conjunto de datos con 1922 audios de vuelos de machos y 682 audios donde se observaban hembras. Este conjunto nos servirá como base sólida para entrenar y validar el modelo predictivo.

3.1.4. Preprocesado de los datos

En el campo del procesamiento de señales de audio, la generación de espectrogramas basada en el banco de filtro *Mel* es una técnica esencial para la transformación de datos para su análisis mediante aprendizaje automático [24, 10, 40]. Tal y como se puede observar en la Figura 3.6, el espectograma es una representación bidimensional que descompone una señal de tiempo continuo en sus componentes frecuenciales para distintas ventanas temporales de análisis, distribuyendo el tiempo en el eje horizontal y las frecuencias, ajustadas a la escala de *Mel*, en el eje vertical. Si bien hay casos donde los ejes se invierten, la implementación más común es la explicada con anterioridad.

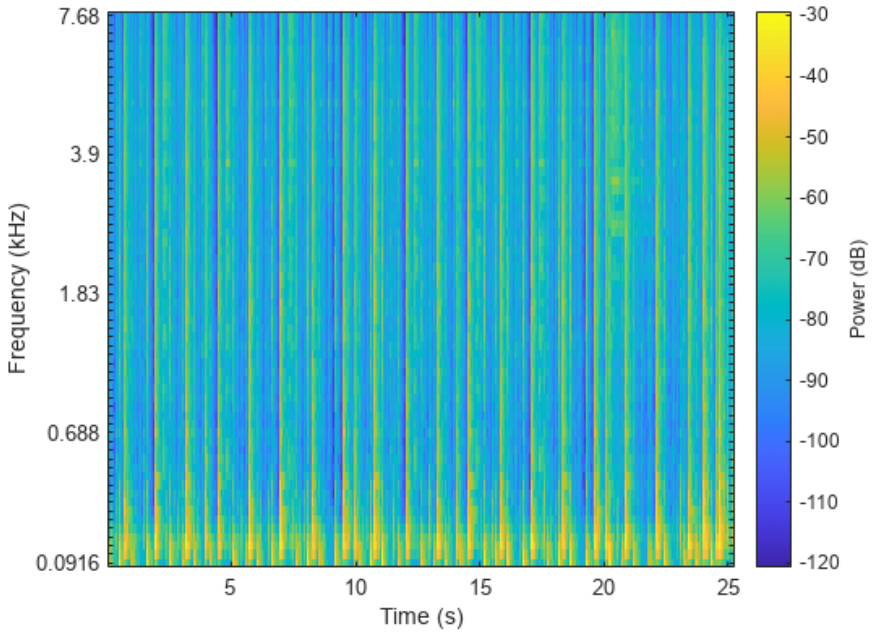


Figura 3.6: Visualización de un *Mel-spectogram*.

Como se explica en la Sección 3.1.1, la señal original contiene cuatro canales de audio que se convierten a una señal mono mediante el promedio de estos canales. Esta conversión asegura que toda la información relevante de los diferentes canales se integre en una única señal de audio, simplificando así el análisis posterior.

La estandarización de espectrogramas de Mel se efectúa mediante una normalización de magnitudes espectrales, asegurando la comparabilidad entre diversas señales. Este ajuste minimiza las variaciones no esenciales y resalta las diferencias discriminatorias entre categorías acústicas.

El aprovechamiento de espectrogramas de Mel normalizados por modelos de aprendizaje automático incrementa la precisión en la identificación de patrones sonoros. En la práctica, esto facilita la extracción de características distintivas en aplicaciones tales como reconocimiento vocal, clasificación sonora y detección de anomalías auditivas.

3.1.5. Entorno de Trabajo y Herramientas Utilizadas

Durante el desarrollo de este proyecto, se utilizó un entorno de trabajo especializado para manejar las demandas computacionales específicas, especialmente relacionadas con el procesamiento de datos y el aprendizaje automático.

El proyecto se desarrolló utilizando *Visual Studio Code*, un editor de código que soporta una variedad de lenguajes de programación y herramientas. Para garantizar la consistencia del entorno de desarrollo a través de diferentes plataformas y facilitar la configuración, se empleó *Docker*.

Dado que el proyecto requería altas capacidades de cómputo, especialmente para el entrenamiento de modelos de aprendizaje profundo, se optó por llevar a cabo el desarrollo en un servidor equipado con una GPU. Esta configuración permitió aprovechar la aceleración por hardware para mejorar significativamente los tiempos de entrenamiento y ejecución de los modelos. Se utilizaron herramientas de conexión y manejo remoto de servidores.

El proyecto hizo uso intensivo de varias librerías especializadas en procesamiento de datos y aprendizaje automático, entre las que destacan:

- **PyTorch**[3]: esta librería de aprendizaje automático fue utilizada para crear todos los módulos necesarios del modelo SPADE (*decoder*, *Encoder*, *Predictor*), facilitando la implementación de técnicas avanzadas de aprendizaje profundo.
- **Librosa**[4]: librería empleada para el procesamiento de audio. Librosa fue utilizada para transformar señales de audio en espectrogramas log-Mel, un formato adecuado para el análisis y procesamiento de señales de audio en contextos de aprendizaje automático. [26]
- **Pyod**[5]: biblioteca especializada en la detección de anomalías. Pyod fue fundamental para implementar y evaluar los OCCs en el módulo definido como *Pseudo-Labeler*, se utilizaron los OCSVM, explicados anteriormente en la Sección 2.2.3. [48]

3.2 Implementación y desarrollo del modelo

En esta sección, se presenta una descripción detallada y precisa de los componentes del modelo SPADE. Como se ilustró en la Figura 2.7, el modelo SPADE está estructurado entorno a cuatro módulos principales: *Encoder*, *Decoder*, *Predictor* y *Pseudo-Labeler*.

[3]https://pytorch.org/
[4]https://librosa.org/
[5]https://pyod.readthedocs.io/en/latest/

Además de estos cuatro módulos, se detallarán las transformaciones aplicadas a los datos, el funcionamiento del algoritmo a lo largo de su ciclo de ejecución y los puntos claves para la elección de SPADE.

3.2.1. Puntos clave SPADE

Al enfrentar el reto de la detección de anomalías en el contexto de la TIE, se realizó una meticulosa evaluación de las herramientas disponibles, culminando en la selección de SPADE por razones como las que se detallan a continuación:

- **Gestión de Desajuste de Distribución:**

  La gestión del desajuste de distribución implementada por SPADE surge para abordar un problema específico que surge al trabajar con datos que pueden varias por distintas razones. Como bien se ha dicho, en este proyecto, se va a trabajar con datos tanto etiquetados como no etiquetados.

  Uno de los problemas más comunes en este tipo de configuración es el "desajuste de distribución". Esto se refiere a situaciones donde la forma en que se distribuyen o varían los datos etiquetados es diferente de la distribución de los datos no etiquetados. Además, al trabajar con datos biológicos (audio de aleteo de mosquitos) se ha podido comprobar que la distribución de los datos puede variar debido a la alimentación, el clima, u otros factores. Este fenómeno fue estudiado con anterioridad en proyectos previos donde se analizaron estos comportamientos. SPADE debería ser capaz de detectar y ajustarse a estas diferencias, lo que lo hace muy valioso para asegurarse de que las predicciones o detecciones de anomalías sean precisas y reflejen correctamente las variaciones naturales en los datos.

- **Enfoque Semi-Supervisado:**

  Gracias al trabajo conjunto con expertos en la TIE, es posible trabajar en un marco de aprendizaje semi-supervisado (disponiendo de una cantidad de datos etiquetados). Este fenómeno será explotado por el mecanismo de *pseudo-labeling*, que se explicará más adelante, basado en un ensamble de clasificadores de una clase (OCCs).

- **Mejora sobre SRR:**

  El algoritmo SPADE fue presentado como una mejora de SRR (algoritmo desarrollado en el proyecto previamante). Así pues, es natural también implementar esta solución en este contexto.

- **Estudio/Selección de hiperparámetros:**

  Se propone un método novedoso de emparejamiento parcial para la selección de hiperparámetros sin necesidad de un conjunto de validación. Esto es vital ya que la selección convencional de hiperparámetros depende de conjuntos de validación, a menudo no disponibles en situaciones reales con datos etiquetados limitados.

3.2.2. Desarrollo del modelo SPADE

El algoritmo SPADE tiene como objetivo el refinamiento de un *Predictor* que sea capaz de detectar anomalías de forma semi-supervisada. Para ello, emplea un módulo definido como *Encoder* que debe extraer patrones significativos de la entrada. La salida del *Encoder* es, a su vez, la entrada del *Predictor*.

El entrenamiento del *Encoder* se realiza mediante una cabeza de proyección, que en nuestro caso corresponde a un *Decoder* con la misma arquitectura que el *Encoder*. Este proceso

permite al *Encoder* aprender representaciones robustas de los datos, facilitando la identificación de patrones anómalos.

Para el refinamiento de las muestras no etiquetadas, se implementa un *Pseudo-Labeler* que asigna etiquetas provisionales a los datos sin etiquetar. Este *Pseudo-Labeler* utiliza la información aprendida por el *Encoder* para generar etiquetas que guíen el entrenamiento del *Predictor*. Al iterar este proceso, el algoritmo SPADE mejora progresivamente su capacidad para identificar anomalías, aprovechando tanto los datos etiquetados como los no etiquetados de manera eficaz.

En resumen, SPADE integra cuatro modulos diferentes que le permiten lograr un sistema eficiente para la detección semi-supervisada de anomalías. A continuación se expondrán los distintos modulos de manera detallada.

Autoencoder

El *Autoencoder* es utilizado para la creación de representaciones abstractas del problema a resolver. La arquitectura del *Autoencoder* es utilizada como *Encoder* (h) y *Projection Head (Decoder)* (g) en esta implementación. El proceso del *Autoencoder* en SPADE implica la transformación de datos de entrada en un espacio de representación más abstracto y luego la reconstrucción de estos datos a partir de esa representación. Este proceso se realiza en dos etapas: el *Encoder* y el *Decoder*.

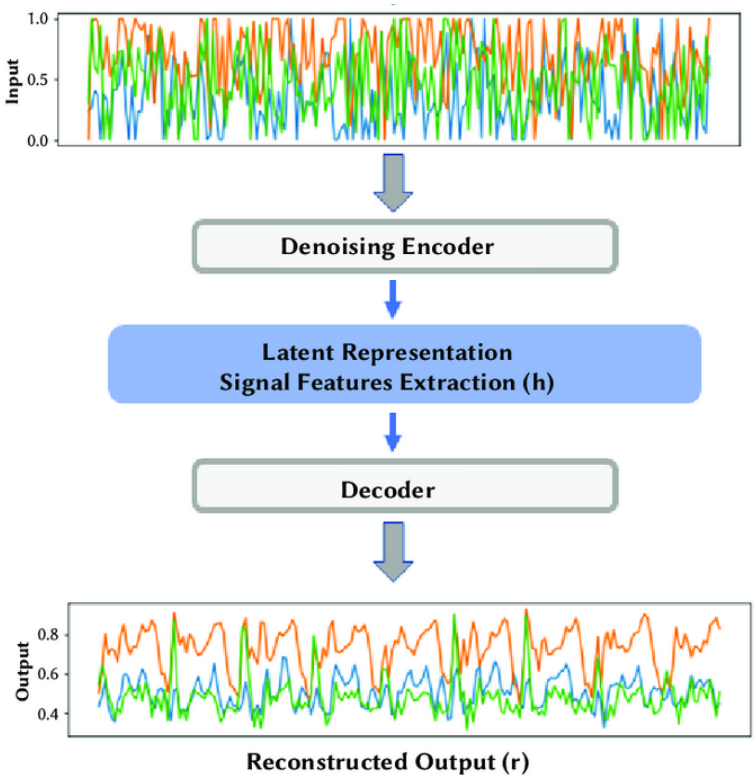


Figura 3.7: *Autoencoder architecture*

-Encoder:

En esta etapa, tal y como se observa con claridad en la Figura 3.7, los datos de entrada, los espectrogramas (x), se transforman en un vector de características que encapsulan la información en una dimensionalidad menor. El *Encoder*, compuesto por capas convolucionales, toma los espectrogramas como entrada y los mapea a un espacio de caracterís-

ticas. Estas características representan una versión comprimida y abstracta de los datos originales, lo que permite una representación más eficiente y útil para el procesamiento posterior.

Este conjunto de características ($h(x)$) que se obtiene del *Encoder* es clave, las características son utilizadas en distintos modulos de SPADE; el *Predictor* y el *Pseudo-Labeler*. El *Predictor* utiliza los vectores de características para detectar las anomalías.

***-Decoder*:**

Utiliza los datos que el *Encoder* ha transformado en el espacio de características ($h(x)$) para generar una reconstrucción de los datos originales (x'). Idealmente, esta reconstrucción debería ser una versión fiel a los datos de entrada, de manera que minimice la pérdida de información durante el proceso de codificación y decodificación.

Con el objetivo de lograr una reconstrucción fiel de los datos se utilizara un criterio de pérdida, que se explicará en más detalle en la Sección 3.2.4, como el error cuadrático medio (MSE, por sus siglas en inglés), entre la entrada original (x) y la salida reconstruida (x'). Cabe destacar que este criterio de pérdida se utiliza para ajustar los parámetros del modelo durante el entrenamiento, de modo que la discrepancia entre la entrada y la salida se minimice gradualmente. En resumen, el proceso de optimización del modelo busca encontrar los parámetros del *Encoder* y *Decoder* que minimizan la diferencia entre la entrada y la salida reconstruida, lo que garantiza una reconstrucción precisa de los datos originales.

Pseudo-Labeler

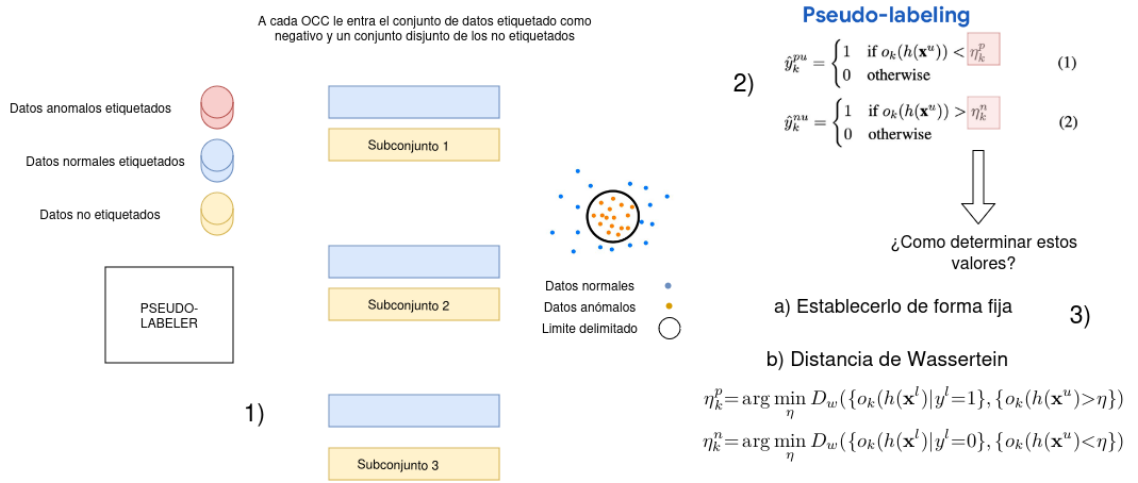


Figura 3.8: Proceso de pseudo-labeling y determinación de umbrales utilizando la distancia de Wasserstein.

Este módulo tiene como objetivo el aprovechamiento de las muestras no etiquetadas en fase de entrenamiento. Para ello, este módulo predice las muestras anómalas y normales sobre el conjunto no etiquetado. Para ello, se implementan una conjunto de clasificadores de una clase que son alimentados con los datos normales etiquetados y un grupo disjunto de las muestras no etiquetadas a cada clasificador. Después, una vez entrenados, se predice una muestra no etiquetada sobre todos los clasficadores (sólo 1 de ellos ha visto esa muestra en fase de entrenamiento) y se determina si es anómala o normal según un criterio de concordancia (ver Figura 3.8).

La determinación de los umbrales (para considerar comportamientos normales o anómalos) es crítica para el *Pseudo-Labeler* en SPADE y se realiza utilizando la distancia de Wasserstein (ver Figura 3.8). Este proceso permite una selección adaptativa de umbrales sin necesidad de un conjunto de validación dedicado.

Los umbrales η_p para etiquetas pseudo-positivas (anomalías) y η_n para etiquetas pseudo-negativas (normales) se determinan minimizando la distancia de Wasserstein D_W entre las distribuciones de puntuaciones de datos etiquetados y los no etiquetados, suponiendo que las distribuciones son similares. Es decir, las anomalías (o comportamientos normales) etiquetadas, deben tener una distribución similar a las anomalías no etiquetadas.[47] A continuación, se muestra el proceso de umbrales para las anomalías (siendo igual para los datos normales):

1. Se calculan las puntuaciones de anomalías para los datos etiquetados (x_l) y no etiquetados (x_u).
2. Como se asume que la distribución de los datos etiquetados positivos será similar a la de los datos positivos no etiquetados. Se hace un barrido de umbrales (η) según la ecuación en la Figura 3.8 para minimizar la distancia de Wasserstein D_W entre las puntuaciones de anomalías de los datos etiquetados positivos ($y^l = 1$) y los posibles datos positivos no etiquetados. En este caso, se considerarán anomalías si las muestras tienen una probabilidad de anomalía superior a η. Así, obtenemos el valor de η_p.

Para las etiquetas pseudo-negativas se asume el mismo criterio sobre las distribuciones y se utiliza (η_n) para minimizar la distancia de Wasserstein D_W entre las puntuaciones de anomalías de los datos etiquetados negativos ($y^l = 0$) y los posibles datos negativos no etiquetados. En este caso, las probabilidades menores a η son consideradas muestras normales.

Este método asegura que SPADE adapte eficientemente los umbrales de decisión para el pseudo-labeling. Como se mencionó en la Sección 3.2.2, esta funcionalidad es clave para evitar el problema del desajuste de distribución.

Una vez que los datos no etiquetados han sido etiquetados por el *Pseudo-Labeler*, se calcula una función de pérdida entre estas etiquetas y las etiquetas del *Predictor* (para discernir si las muestras no etiquetadas han sido predecidas de igual forma por ambos módulos), proporcionando una mayor cantidad de datos a partir de los cuales el modelo puede aprender. Este ciclo iterativo de entrenamiento y pseudo-labeling permite que SPADE se adapte y ajuste continuamente a nuevas informaciones y patrones. Esto será explicado con más detalle en la Sección 3.2.5 .

Todo esto convierte al *Pseudo-Labeler* en una parte fundamental del modelo. Su papel no solo consiste en complementar los datos etiquetados con nuevos etiquetados, sino que también mejora continuamente la capacidad del *Predictor* para discernir entre lo normal y lo anómalo.

Predictor

Este módulo es el que finalmente predice si los datos son normales o anómalos. Este componente es clave porque se encarga de interpretar tanto los datos originalmente etiquetados como aquellos etiquetados por el *Pseudo-Labeler*, y hace predicciones basadas en el conjunto enriquecido de información disponible.

El *Predictor* se ajusta con los datos etiquetados manualmente y los datos no supervisados pero etiquetados por el *psuedo-labeler* en cada iteración de entrenamiento.

A medida que el proceso de *pseudo-labeling* proporciona nuevos datos dentro del ciclo de entrenamiento, el *Predictor* se enfrenta al desafío de integrar esta nueva información en su marco de aprendizaje. En cada iteración los nuevos datos etiquetados permiten al *Predictor* ajustar y afinar sus parámetros.

SPADE emplea una función de pérdida (más adelante explicada en Sección 3.2.4) compuesta por dos términos para el *Predictor*. Uno de ellos corresponde a la pérdida de las predicciones del *Predictor* respecto a las etiquetas reales. La otra a las *pseudo-labels*, proporcionando un feedback constante que guía la optimización de sus parámetros.

3.2.3. Implementación del modelo

En la arquitectura de SPADE, distintos módulos trabajan de manera conjunta. Cada módulo ha sido cuidadosamente desarrollado e implementado para cumplir con funciones específicas, contribuyendo así al rendimiento del modelo. Esta sección desglosa las arquitecturas de los distintos componentes de SPADE.

Arquitectura del *Autoencoder*

En la arquitectura *Autoencoder*, se ha decidido utilizar una arquitectura que no fuese muy compleja ni muy profunda. Esto fundamentalmente viene motivado por el tamaño del conjunto de datos utilizado para entrenamiento.

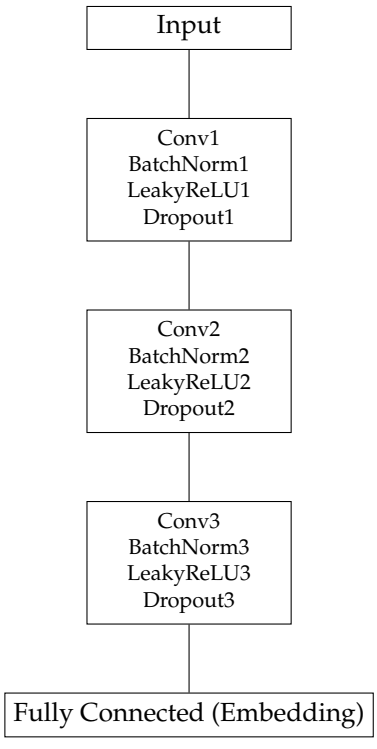


Figura 3.9: *Convolutional Neural Network (CNN) Encoder Architecture*

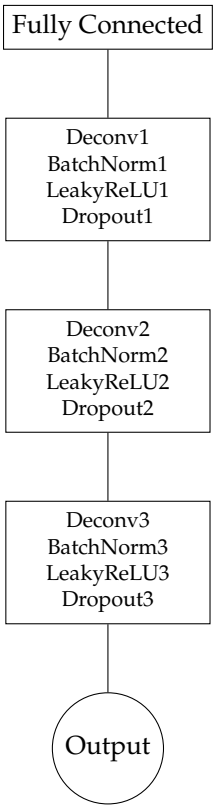


Figura 3.10: *Convolutional Neural Network (CNN) Decoder Architecture*

Tal y como se observa en las figuras anteriores, se utilizan capas Convolucionales de 2 dimensiones, ya que tal y como se ha explicado en la Sección 3.1.4, los spectogramas son representaciones bidimensionales de los audios. También se utilizan capas de normali-

zación *(BatchNorm2d)* y la función de activación, previamente explicada en la Sección 2.2.4, *LeakyReLU*. Al final de la arquitectura *Encoder* se observa una capa linear, esta capa se utiliza para transformar la salida en un vector de las características deseadas. Se ha experimentado con capas de distinto tamaño: 64, 128 o 256.

Las arquitecturas en las Figuras 3.10 y 3.9 tienen la misma estructura pero de manera inversa. Esto se debe a que, tal y como se ha visto en la Figura 3.7, el *Decoder* se encarga de realizar el proceso inverso, recibe el vector de características y tiene que intentar reconstruir la entrada del *Decoder*, es decir, los spectogramas.

Arquitectura del *Predictor*

La arquitectura del *Predictor* está especialmente diseñada para manejar las carecteristicas del audio extraídas por el *Decoder*. Más adelante, en la Sección 4 se observa como se ha probado el modelo con distintos tamaños de vectores de características. Como consecuencia, se ha adaptado el *Predictor* para que en función del tamaño del vector de características de entrada, las capas lineares tengan capacidad para adaptar sus dimensiones.

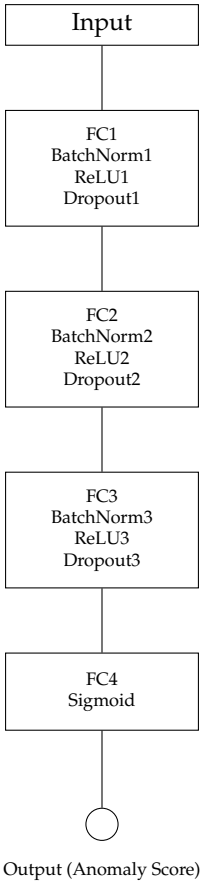


Figura 3.11: *Anomaly Predictor Architecture*

La red que se observa en la Figura 3.11 incluye normalización por lotes (*BatchNorm1d*) después de cada capa lineal, lo cual estabiliza el aprendizaje y acelera la convergencia. También utiliza la función de activación ReLU, previamente explicada en la Sección 2.2.4, después de cada normalización.

La última capa de salida utiliza una función sigmoidal para clasificar entradas como normales o anómalas, ideal para la clasificación binaria requerida en detección de anomalías de audio.

Arquitectura del *Pseudo-labeler*

En el *Pseudo-Labeler* empleamos una arquitectura basada en máquinas de vectores de soporte de una clase (OCSVM). Las OCSVM, tal y como ya se ha comentado en la Sección 2.2.3, se utilizan para entrenar clasificadores especializados en detectar anomalías en los datos. En este contexto, cada OCSVM se entrena con un conjunto disjunto de muestras etiquetadas y no etiquetadas, aprendiendo a diferenciar comportamientos normales de anómalos. Esta arquitectura es esencial para el funcionamiento del modelo SPADE .

3.2.4. Especificaciones técnicas

Funciones de pérdida

El modelo de detección de anomalías $q(h(\cdot))$ (salida del *Predictor*) se entrena utilizando tres funciones de pérdida anteriormente comentadas en la Sección 2.2.4:

- Entropía cruzada binaria (BCE) sobre datos etiquetados.
- BCE sobre *pseudo-labels*.
- Pérdida sobre los datos reconstruidos(MSE).

A continuación, se describen las formulaciones de pérdida. La pérdida BCE sobre los datos etiquetados se propone como:

$$L_Y^l = \mathbb{E}\left[L_{BCE}(q(h(x_l)), y_l)\right],$$

y la pérdida BCE sobre los *pseudo-labels* como:

$$L_Y^u = \mathbb{E}\left[L_{BCE}(q(h(x_u)), \hat{y}_u)\right],$$

donde h hace referencia al *Encoder*, q al *Predictor*, y_l a los etiquetas supervisadas y $\hat{y}_u$ a los *pseudo-labels*. Por tanto, $q(h(x_l))$ es la salida de la sigmoide del *Predictor* sobre el conjunto de datos etiquteados y $q(h(x_u))$ sobre el conjunto de datos no etiquetado. Con estas dos funciones, se calcula el error de los dos conjuntos de datos: el etiquetado y el no etiquetado. Para el no etiquetado, se comparan las etiquetas generadas por el *Pseudo-Labeler*.

Para mejorar la calidad del *Encoder* h, se usa una pérdida de reconstrucción:

$$L_R = \mathbb{E}\left[L_{MSE}(x, g(h(x)))\right],$$

donde g hace referencia el *Decoder*.

En general, el *Encoder* h, el *Predictor* q y el módulo de reconstrucción g son entrenados resolviendo el siguiente problema de optimización:

$$\min_{h,g,q} \left\{ L_Y^l + \alpha L_Y^u + \beta L_R \right\},$$

donde α, β son hiperparámetros ajustables ($\alpha = \beta = 1.0$ para la mayoría de los experimentos).

3.2.5. Iteración del Algoritmo SPADE

El algoritmo SPADE, diseñado para la detección semi-supervisada de anomalías, opera a través de un procedimiento iterativo donde cada componente tiene un rol específico en la mejora progresiva de la detección. A continuación, se describen los pasos clave del algoritmo, tal como se ilustran en la Figura 3.12.

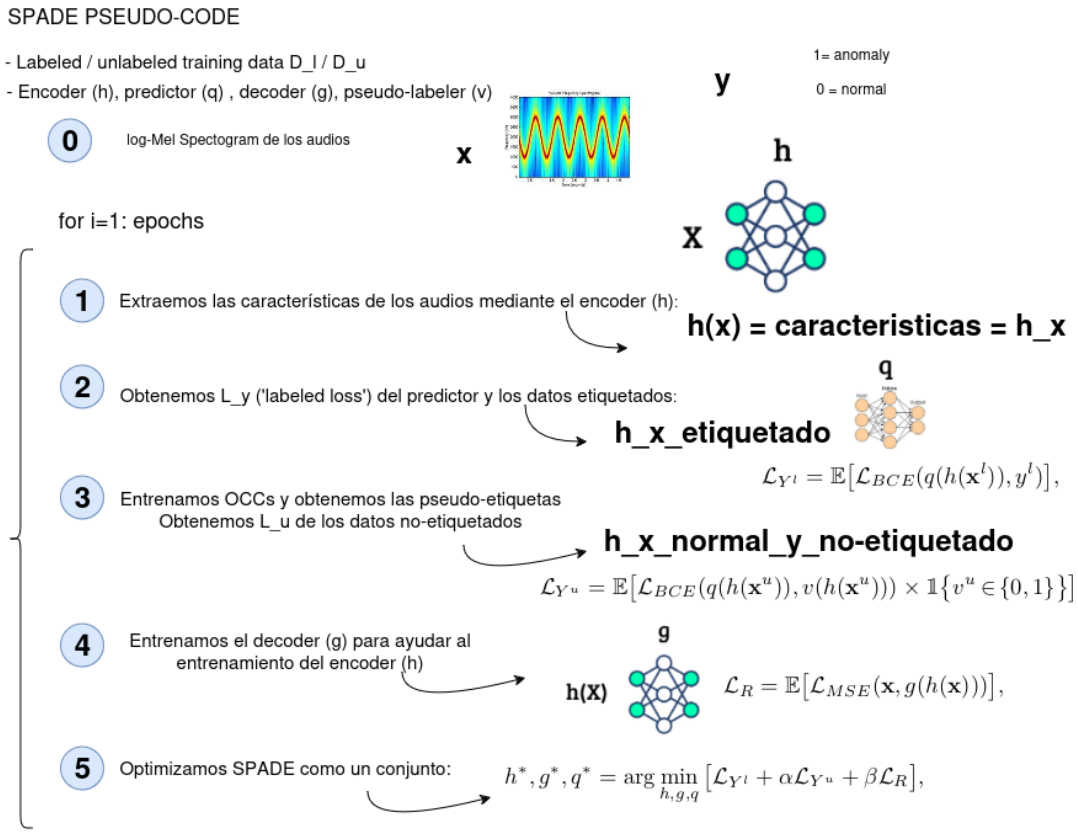


Figura 3.12: Diagrama del procedimiento iterativo del algoritmo SPADE.

Cada uno de los siguientes pasos se puede observar en la Figura 3.12, de esta manera se espera que la comprensión de los distintos pasos realizados a lo largo del algoritmo sea clara:

0. **Transformación de Audios**: Como se puede ver, el primer paso es convertir los audios iniciales en espectogramas log-Mel. Este proceso facilita la extracción de características relevantes para la detección de anomalías.

1. **Extracción de Características**: Una vez dentro del bucle que realiza el método, se observa como el *Encoder* (h), procesa los datos de entrada transformándolos en vectores de características codificadas, esenciales para la clasificación posterior.

2. **Se obtiene (L_y)**: Tal y como se ha expuesto anteriormente enla Sección 3.2.4 las diferentes funciones de perdida realizan un papel clave en SPADE, en este punto del algoritmo se obtendra la pérdida (L_y) , donde se pone a prueba la clasificación del *Predictor* por primera vez.

3. **Entrenamiento de Clasificadores**: En este punto, los vectores de características alimentan un conjunto de clasificadores de una clase (OCCs), que aprenden a diferenciar entre comportamientos normales y anómalos. El *Pseudo-Labeler* en este punto

utiliza el método de *Partial Matching*(detallado en la Sección 3.2.2) para establecer umbrales (η_p y η_n) utilizados para asignar *pseudo-labels* a los datos no etiquetados. Este proceso se guía por la función de pérdida no etiquetada (L_u), que ayuda a optimizar mediante la comparación con las etiquetas establecidas por el *Predictor*.

4. **Función de pérdida (L_r)**: En último lugar, calcula la función de perdida (L_r) entre el *Encoder* (h) y el *Decoder* (g), de esta manera se mejora el entrenamiento del *Encoder*, parte clave del algoritmo.

5. **Entrenamiento Iterativo**: Para finalizar, se entrena iterativamente a todos los componentes del modelo (g, h, y q), utilizando tanto datos etiquetados como *pseudo-labels* . La Figura subraya la importancia de la función de pérdida combinada (L), que incluye términos de pérdida etiquetada, no etiquetada y de reconstrucción (L_r) para mejorar la precisión del modelo. Este ciclo de evaluación, ajuste y reentrenamiento se repite hasta alcanzar la convergencia, mejorando así la detección de anomalías en cada iteración, tal y como se indica en el diagrama.

Esta metodología integrada de aprendizaje supervisado y autoaprendizaje debería permitir a SPADE adaptarse continuamente y mejorar su capacidad para detectar anomalías, aprovechando de manera eficiente tanto los datos etiquetados como los patrones aprendidos de los datos no etiquetados, tal y como se detalla en la Figura 3.12.

3.3 Entrenamiento y evaluación de los modelos

3.3.1. Métricas de Evaluación

Las métricas de evaluación juegan un papel crucial en la valoración de la eficacia de los modelos de clasificación, proporcionando una visión cuantitativa del rendimiento y ayudando a determinar si se ajustan adecuadamente a la problemática dada. En este proyecto se han implementado diversas métricas para medir la calidad del modelo. Antes de exponer las métricas se van a definir los siguientes conceptos básicos para un mejor entendimiento de las métricas utilizadas:

- **Verdaderos Positivos (TP)**: Casos en los que el modelo predice correctamente la clase positiva.
- **Falsos Positivos (FP)**: Casos en los que el modelo predice incorrectamente como clase positiva.
- **Verdaderos Negativos (TN)**: Casos en los que el modelo predice correctamente la clase negativa.
- **Falsos Negativos (FN)**: Casos en los que el modelo predice incorrectamente como clase negativa.

A partir de estos valores fundamentales se definen las siguientes métricas:

- **Sensibilidad (Recall)**: Esta métrica evalúa la capacidad del modelo para identificar correctamente todos los casos positivos. Se calcula mediante la fórmula:

$$\text{Sensibilidad} = \frac{TP}{TP + FN}$$

- **Precisión**: La precisión mide la exactitud de las predicciones positivas del modelo. Se calcula como:

$$\text{Precisión} = \frac{TP}{TP + FP}$$

- **Puntuación F1 (F1-score)**: Esta métrica combina la precisión y la sensibilidad en una medida armonizada, calculada por:

$$\text{F1-score} = 2 \times \frac{\text{Precisión} \times \text{Sensibilidad}}{\text{Precisión} + \text{Sensibilidad}}$$

- **Exactitud (Accuracy)**: La exactitud mide la proporción de predicciones correctas tanto positivas como negativas del modelo. Su fórmula es:

$$\text{Exactitud} = \frac{TP + TN}{TP + FP + FN + TN}$$

- **Área Bajo la Curva ROC (AUC)**: Esta métrica proporciona una visión global de la capacidad del modelo para discriminar entre las clases a lo largo de todos los umbrales posibles de clasificación. El AUC es especialmente útil para comparar diferentes modelos, independientemente de los puntos de corte. Un AUC de 1 representa un modelo perfecto, mientras que un valor de 0.5 sugiere un rendimiento no mejor que el azar. Se calcula utilizando la curva ROC, que grafica la tasa de verdaderos positivos (sensibilidad) contra la tasa de falsos positivos (1 - especificidad) a varios umbrales de decisión:

$$\text{AUC} = \int_0^1 \text{TPR}(t)\, dt$$

donde TPR(t) representa la tasa de verdaderos positivos para un umbral específico t.

CAPÍTULO 4
Resultados

4.1 Resultados

La evaluación del rendimiento del modelo presentado en este trabajo refleja su complejidad y capacidad de adaptación frente a diversas situaciones. La configuración flexible del modelo nos ha permitido explorar un amplio abanico de opciones y configuraciones en el apartado de resultados. Esta versatilidad, sin embargo, implica también una demanda significativa en términos de tiempo computacional, ya que cada ejecución del modelo requiere un período considerable de tiempo para su completa evaluación.

Para garantizar una validación exhaustiva de la capacidad del modelo para adaptarse y generalizar su aprendizaje, se ha procedido a testearlo utilizando dos conjuntos de datos distintos. Esta estrategia no solo ha permitido ajustar y afinar los parámetros del modelo para cada *dataset* específico, sino que también ha facilitado la realización de una comparativa detallada que subraya las fortalezas y limitaciones del modelo en escenarios variados.

La inclusión de múltiples conjuntos de datos en nuestra evaluación refuerza la relevancia de los resultados obtenidos, proporcionando una visión más amplia del modelo en condiciones divergentes. En las siguientes subsecciones, se detallarán los resultados obtenidos, discutiendo las métricas de desempeño alcanzadas, las particularidades observadas en cada conjunto de datos, y las implicaciones prácticas de estos hallazgos para futuras aplicaciones y desarrollos del modelo.

4.1.1. Análisis de los resultados

Tal y como ya se ha expuesto el entorno *SPADE* nos permite variar una gran cantidad de parámetros, estos nos dejan observar como se comporta y como cada uno de ellos afecta a la obtención de resultados.

Debido a la limitación de tiempo se ha decidido probar cuatro enfoques distintos:

1. Como ya se ha expuesto dentro del modelo existen cuatro conjuntos de datos. Por una parte, está el conjunto de datos de entrenamiento y el conjunto de datos de test. Dentro del conjunto de entrenamiento se tiene, conjunto de datos etiquetados y no-etiquetados. Uno de los diferentes parametros que se ha ido probando ha sido el porcentaje de datos etiquetados y no-etiquetados. Para cada uno de estos distintos porcentajes se han realizado los distintos análisis:

    a) Variando el número de características que nos proporcionaba el *Encoder*

    b) Variando el número de OCC del *Pseudo-Labeler*

c) Variando los parámetros *Alpha*(α) y *Beta*(β) de la función de optimización final, vista en la Sección 3.12

2. Comparativa entre dos conjuntos de datos ; *ToyCar dataset* y los audios utilizados en MoTIA.

Los porcentajes de muestras etiquetadas han sido:

Porcentaje de datos	Tipo de dato	Train		Test
		Etiquetado	No-Etiquetado	
90 % Etiquetado - 10 % No-Etiquetado	Normales	1383	154	385
	Anómalos	490	55	137
75 % Etiquetado - 25 % No-Etiquetado	Normales	1152	385	385
	Anómalos	171	408	137
50 % Etiquetado - 50 % No-Etiquetado	Normales	768	769	385
	Anómalos	171	272	273

Tabla 4.1: Distribución detallada de los distintos conjuntos de datos utilizados en los análisis.

Con el objetivo de no excederse en el número de tablas utilizadas se ha preferido solamente exponer los mejores modelos en cada uno de los distintos escenarios.

Si se desea observar más resultados, en el Apéndice A aparecen todos los resultados de todos los modelos probados.

Número de Características del *Embedding*

Parámetros					Métricas		
Etiquetado / No-Etiquetado	*Alpha*(α)	*Beta*(β)	No. OCCs	No. Features	AUC	F1 Score	Accuracy
0.75-0.25	1	1	30	64	0.824	0.623	0.750
0.75-0.25	1	1	30	256	0.816	0.600	0.660
0.9-0.1	1	1	30	256	0.823	0.579	0.558
0.9-0.1	1	1	30	512	0.806	0.549	0.560

Tabla 4.2: Resultados variando el número de características.

Al analizar los resultados, se observa que los modelos con vectores de características de tamaño moderado, específicamente aquellos con 64 características, tienden a ofrecer los mejores rendimientos en términos de AUC, F1 y Accuracy. Es importante destacar que se realizaron pruebas con vectores de características más pequeños, pero el rendimiento del modelo disminuyó significativamente indicando que los vectores de características más pequeños que 64 no son capaces de capturar la información de los audios. Por otro lado, valores muy grandes como 1024 y 2048, introducen muchos valores cercanos a 0 en el vector que no proporcionan información relevante al modelo. Se puede apreciar que a medida que aumenta el número de muestras anómalas, el vector debe ser más compacto.

Diferentes *Alpha*(α)-*Beta*(β)

Se experimentaron cuatro valores diferentes para los parámetros *Alpha*(α) y *Beta*(β): [1, 1.3, 1.6, 1.9]. En total, se obtuvieron 16 combinaciones de resultados, pero aquí se presentan solo los 5 mejores para una mayor claridad y concisión. La siguiente tabla muestra estos resultados, manteniendo constante el número de características en 64, dado que en la sección anterior se determinó que este valor ofrecía el mejor rendimiento.

Parámetros					Métricas		
Etiquetado / No-Etiquetado	***Alpha*(α)**	***Beta*(β)**	**Nº OCCs**	**Nº características**	**AUC**	**F1**	**Accuracy**
0.75-0.25	1.9	1.6	30	64	0.861	0.655	0.81
0.9-0.1	1	1.3	30	64	0.866	0.641	0.665
0.5-0.5	1.9	1.6	30	64	0.830	0.640	0.891
0.5-0.5	1.3	1.9	30	64	0.830	0.654	0.730
0.75-0.25	1	1.6	30	64	0.838	0.623	0.76

Tabla 4.3: Resultados con diferentes valores de *Alpha*(α) y *Beta*(β)

La variación de los valores de *Alpha*(α) y *Beta*(β) muestra una ligera tendencia a mejorar el rendimiento del modelo en términos de AUC, F1 y Accuracy. Hasta ahora, solo se han explorado valores relativamente pequeños para estos parámetros. Un futuro experimento interesante sería probar valores más grandes para *Alpha*(α) y *Beta*(β), con el fin de observar cómo afectan la función de pérdida y el comportamiento del modelo (o reducir con valores inferiores a 1). La exploración de una gama más amplia de valores podría ofrecer información valiosa sobre la relación entre estos parámetros y la capacidad del modelo para generalizar en diferentes conjuntos de datos.

Diferente número de OCCs

Otro aspecto crucial del algoritmo es cómo se comporta el modelo al variar la cantidad de OCCs (*One-Class Classifiers*). En teoría, un mayor número de OCCs debería proporcionar una mayor seguridad al modelo para generar *pseudo-labels*. A continuación se presentan los resultados obtenidos al variar el número de OCCs:

Parámetros					Métricas		
Etiquetado / No-Etiquetado	***Alpha*(α)**	***Beta*(β)**	**Nº OCCs**	**Nº características**	**AUC**	**F1**	**Accuracy**
0.75-0.25	1	1	30	512	0.89	0.67	0.83
0.5-0.5	1	1	25	64	0.791	0.604	0.757
0.9-0.1	1	1	25	64	0.888	0.655	0.753
0.5-0.5	1	1	40	64	0.818	0.601	0.723

Tabla 4.4: Resultados con diferentes números de características

En este caso, al aumentar el número de OCCs, se realizaron pruebas con vectores de diferentes valores de carecterísticas, la tendencia general, tal y como se observa en la tabla, seguía evidenciando que los vectores de 64 carecterísticas eran más apropiados. No obstante, con 512 y el conjunto de datos 0.75/O.25 se obtuvo un muy buen resultado. Así pues, se pudo encontrar un punto de fricción entre el número de OCCs que debe ser investigado como trabajo futuro entre la dimensión de la salida del *Encoder* y el número de OCCs.

Análisis global de los resultados

Los resultados indican que el modelo parece funcionar mejor con vectores de pocas características, valores de *Alpha*(α) y *Beta*(β) mayores a 1, y un valor de OCCs cercano a 30.

La implementación y evaluación de estos experimentos ha requerido un esfuerzo computacional considerable. Aunque las tablas solo aparezcan algunos de los resultados, más abajo en el Apéndice A se pueden observar todos los experimentos que se han llevado a cabo.

En resumen, estos experimentos proporcionan una visión valiosa sobre cómo diferentes parámetros afectan el rendimiento del modelo. A medida que se sigue explorando y ajustando estos parámetros, se espera mejorar la capacidad del modelo para generalizar y predecir con precisión. La próxima fase de este trabajo podría incluir la evaluación de combinaciones más complejas de parámetros y la incorporación de técnicas avanzadas para la optimización del modelo.

Además, cabe destacar que el conjunto de datos (por motivos de recolección de datos ajenos a este TFG) parece pequeño para un algoritmo de semejante complejidad. Se requerirán muchos más datos para poder obtener una solución generalizable de SPADE en el marco de la TIE.

Análisis visual empleando UMAP

En esta sección vamos a tratar de exponer de una manera más visual los resultados obtenidos en el modelo. Vamos a presentar algunos gráficos obtenidos a partir de las diferentes pruebas realizadas donde se observa como se distribuyen los datos anómalos y normales en un plano bidimensional[1]. La distribución viene dada por las características extraídas del *Encoder*.

Primero se va a exponer un gráfico donde solo se observa como los datos etiquetados estarían distribuidos en un espacio bidimensional según las características obtenidas por el *Encoder*. De esta manera se puede observar con claridad si el modelo consigue separar de manera adecuada los datos. (ver Figura 4.1)

[1] `https://umap-learn.readthedocs.io/en/latest/`

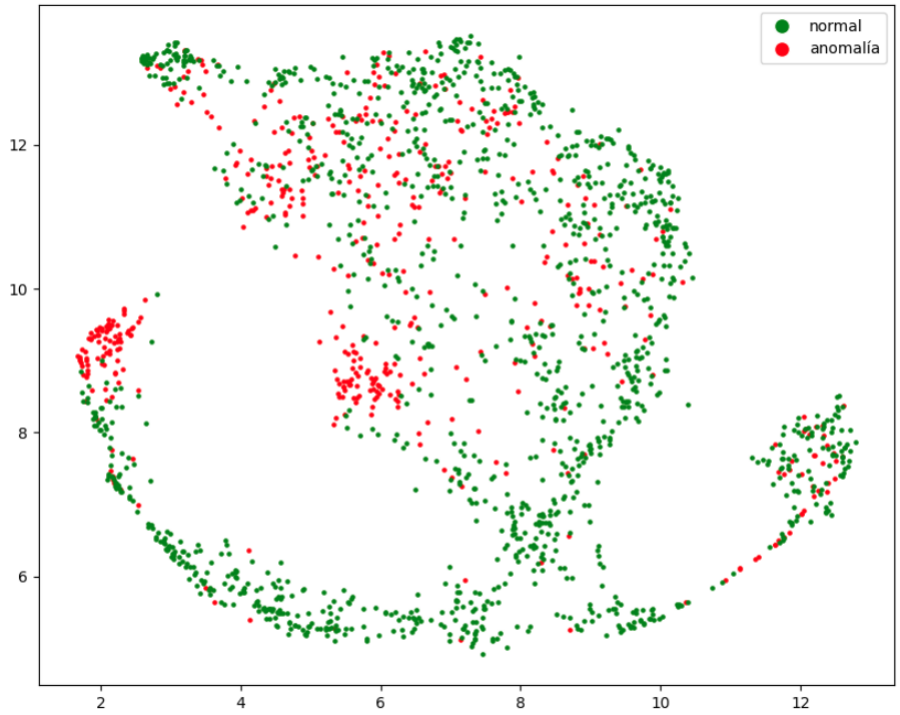


Figura 4.1: Visualización UMAP de datos normales y anómalos etiquetados.

En este primer gráfico donde solo observamos los datos anómalos y normales etiquetados, vemos como se observa una tendencia a separar los datos a derecha y a izquierda de la imagen. Pese a que se divisa una separación se puede observar como todavía existen zonas donde ambos tipos de audios se encuentran mezclados entre sí.

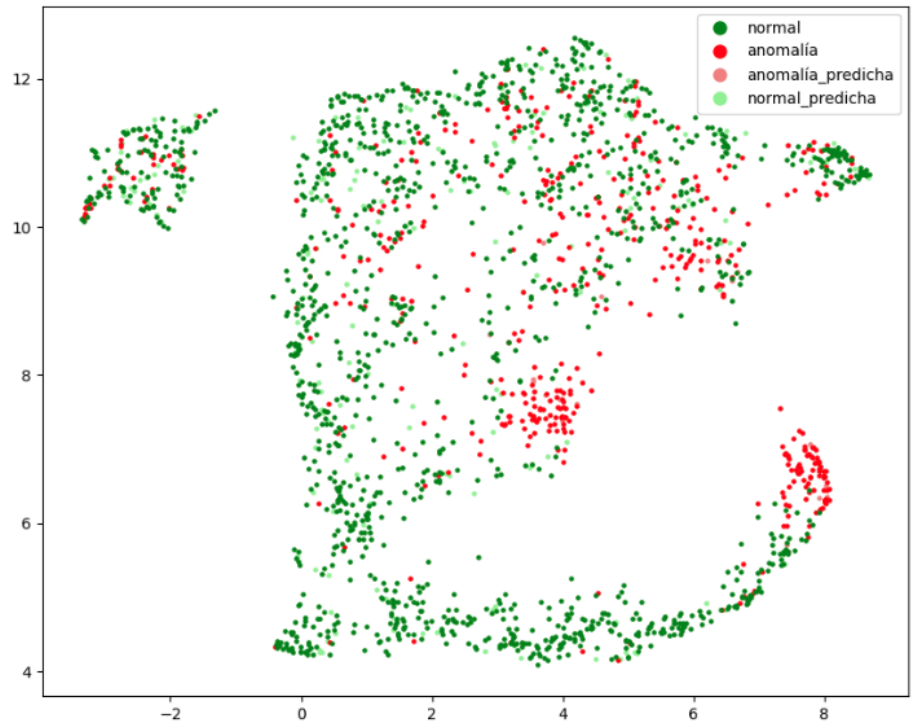


Figura 4.2: Visualización UMAP con datos etiquetados y las predicciones del modelo .

Por otro lado, en este segundo *umap* (ver Figura 4.2) además de tener información sobre los datos etiquetados también tendrá información de las etiquetas predichas por el *Predictor*. Es decir, por una parte aparecerán los datos etiquetados con sus respectivas etiquetas y por otro lado los datos no-etiquetados con la etiqueta que el *Predictor* ha determinado.

Tal y como se observa existe una tendencia de las etiquetas anómalas sobre la parte izquierda del modelo. No obstante, se sigue sin observar una distinción clara y notoria entre ambos conjuntos, habiendo zonas donde los audios anómalos y normales se encuentran superpuestos. Las etiquetas predichas por el *Predictor* se encuentran cerca de las datos etiquetados, no obstante al haber zonas donde los puntos anómalos y normales se encuentran tan juntos, el modelo podría estar equivocandose. Así, se puede discernir que los datos normales y anóamlos son similares desde un punto de vista del modelo y abre la puerta a mejorar o pensar algún preprocesado de los datos distinto.

4.1.2. Data Toy Car

Además del *dataset* de la TIE, también se ha decidido validar el algoritmo SPADE con el conjunto de datos *Dataset Toy Car* [20]. Este *dataset* incluye grabaciones de sonidos normales y anómalos de un coche de juguete (definido con el término *Machine* en el propio *dataset*).

Las grabaciones provienen de 4 jueguetes distintos (*Machine01-Machine04*) del mismo modelo. Cada audio tiene una duración de 10 segundos. Primero, se probaron las 4 máquinas por separado (creando un detector de anomalías por unidad), pero dado que es un *dataset* complejo y reducido (con las máquinas por separado solamente se tenía un *dataset* de 700 audios por máquina) los resultados no fueron los deseados. Así pues, se decidió crear dos *datasets* agrupando los jueguetes de dos en dos. En el primero de ellos se agruparon los audios de los juguetes identificados como uno y dos, y en el segundo *dataset* los jueguetes identificados como tres y cuatro. De esta forma se consigue un conjunto de datos más grande con el que poder entrenar y validar los datos.

Con esta peculiaridad, en este caso había 2 tipos de audios normales y anómalos dentro del *dataset* (provenientes de distintos jueguetes). Esto implicaba que el modelo tenía que ser capaz de identificar patrones anómalos independientemente del juguete.

4.1.3. Análisis de los resultados

Machine01 - Machine02

Primero analizaremos los resultados obtenidos utilizando solamente los audios de los dos primeros tipos .

La separación de datos realizada en este conjunto ha sido un partición *train/test*(80 %/20 %). La partición de *train* a su vez es dividida con una proporción No-Etiquetado/Etiquetado de 75 %/25 % (ver Tabla 4.5)

Tipo de máquina	Tipo de dato	Train		Test
		Etiquetado	No-Etiquetado	
Máquina 1	Normales	599	200	200
	Anormales	167	55	55
Máquina 2	Normales	598	199	199
	Anormales	154	51	51

Tabla 4.5: Distribución detallada de archivos de audio en el conjunto de datos de ToyCar en el primer experimento.

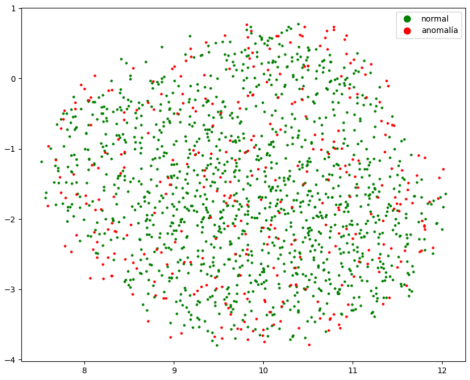
Con este *dataset* se realizaron diversos experimentos, al igual que se ha expuesto en la Sección 4.1.1, cambiando los valores de distintos parámetros como son el número de características, el número de clasificadores y los valores de *Alpha*(α) y *Beta*(β) . No obstante, en este caso la obtención de resultados no ha tenido una gran variabilidad, la incapacidad del modelo por discernir entre anomalias y normales, en este *dataset* ha sido notoria.

Figura 4.3: Evolución de las funciones de pérdida por cada epoch.

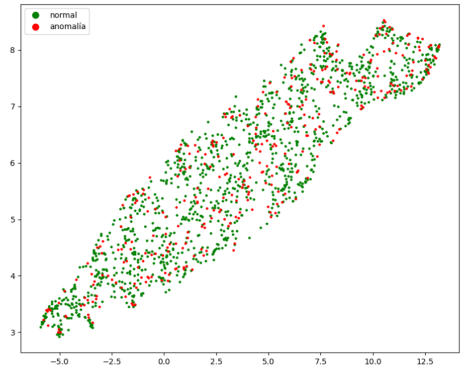
El hecho de que las funciones de pérdida ((L_y),(L_u) y (L_r) muestren una disminución (ver Figura 4.3) indica que el modelo está siendo entrenado correctamente desde un punto de vista técnico. Tal y como se puede observar, existe una disminución considerable de los valores de coste (L_u) y (L_r), valores que se obtienen a partir de las etiquetas del *Predictor*,los datos etiquetados y los *pseudo-labels* obtenidos mediante el *Pseudo-Labeler*. Por otra parte, el valor más elevado es el de la función de coste obtenido del *Autoencoder*, ((L_y). Esto podría deberse a una situación de *overfitting*, donde tras cada epoch las características extraídas de los datos son menos generales, lo que resulta en un aumento de la pérdida.

Para entrar más en detalle en este análisis se han creado gráficos UMAPs. Estos gráficos nos permiten observar la capacidad del modelo de separar los audios anómalos de los normales, las siguientes imágenes han sido extraídas en la iteración 0, la iteración 10 y la iteración 25 del modelo. Se expone de esta forma para que se pueda observar como el modelo va ajustando sus parámetros iteración tras iteración con el objetivo de visualizar/divisar diferencias entre los dos tipos de audios.

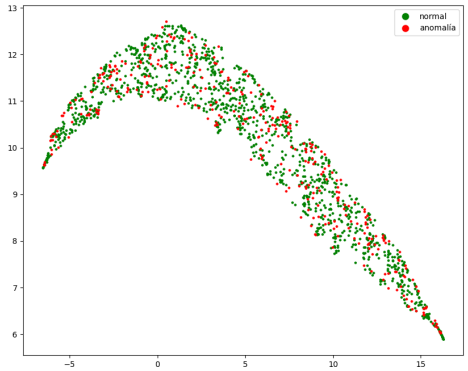
Tal y como se aprecia en las tres imágenes 4.4a,4.4b y 4.4c, el modelo es incapaz de separar de forma correcta los audios de la clase normal y la clase anómala, se observa como en todas las zonas hay un solapamiento de las dos clases, el umap nos permite visualizar que los comportamientos normales y anómalos son similares.



(a) UMAP 0



(b) UMAP 10



(c) UMAP 25

Figura 4.4: Visualización UMAP de distintas iteraciones del mismo modelo.

Machine03 - Machine04

EL último análisis realizado se ha centrado en los dos tipos de audios todavia no vistos, los audios de los tipos tres y cuatro.

La separación de datos realizada en este test ha sido muy parecida a la ya expuesta en la sección anterior. Ver Tabla 4.6 para la partición exacta.

Tipo de máquina	Tipo de dato	Train		Test
		Etiquetado	No-Etiquetado	
Máquina 3	Normales	642	145	213
	Anómalos	172	47	46
Máquina 4	Normales	634	172	194
	Anómalos	171	41	53

Tabla 4.6: Distribución detallada de archivos de audio en el conjunto de datos de ToyCar en el segundo experimento.

La distribución de los datos es muy similar a la vista anteriormente en la Tabla 4.5 y los resultados obtenidos también. Para poder hacer una comparación entre ambos resultados vamos a mostrar la gráfica vista anteriormente sobre el nuevo conjunto de datos. Tal y como se puedo observar en la Figura 4.5 el comportamiento de las distintas funciones de coste es muy similar, observando una disminución considerable de los valores de pérdida (L_u) y (L_r), y un aumento del valor (L_y) . En ambos casos observamos el mismo comportamiento, en futuros estudios este podría ser un punto por el que empezar la mejora del rendimiento.

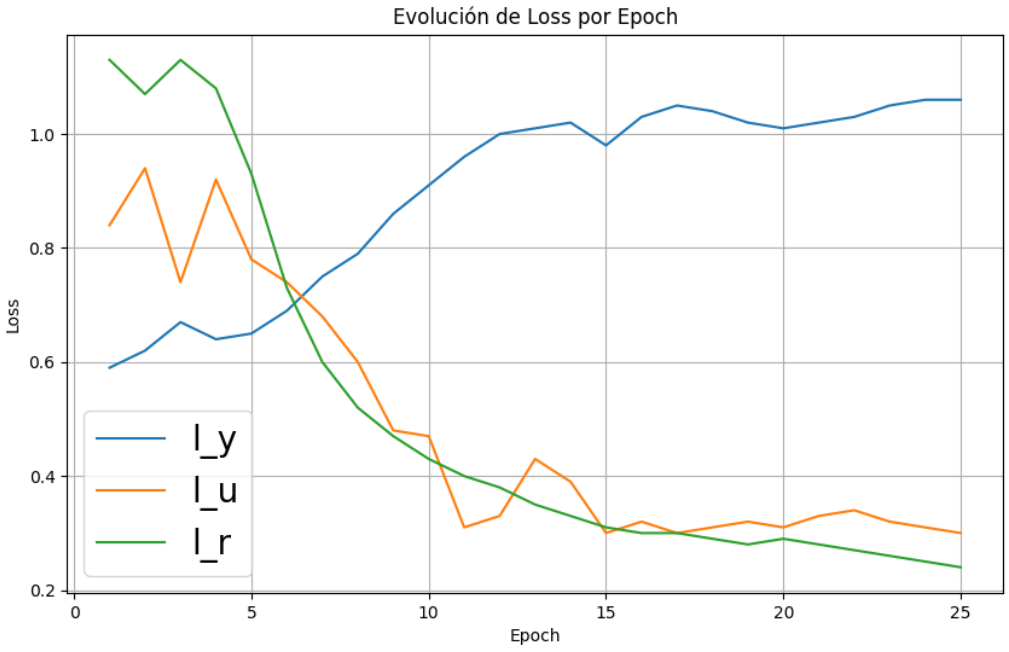
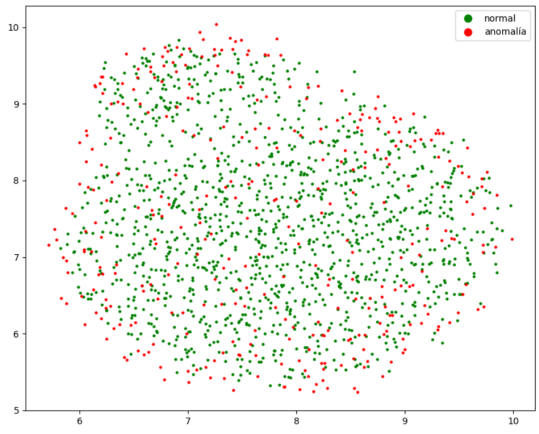
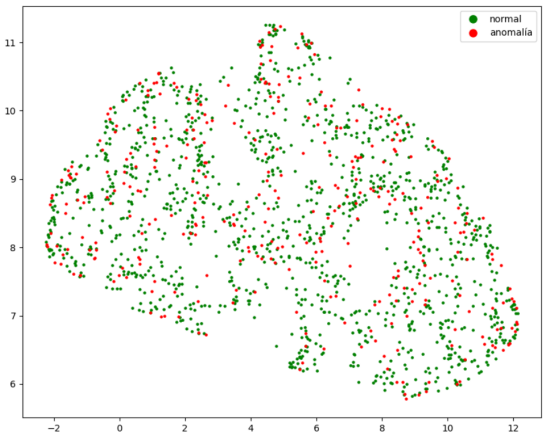


Figura 4.5: Evolución de las funciones de pérdida por cada epoch.

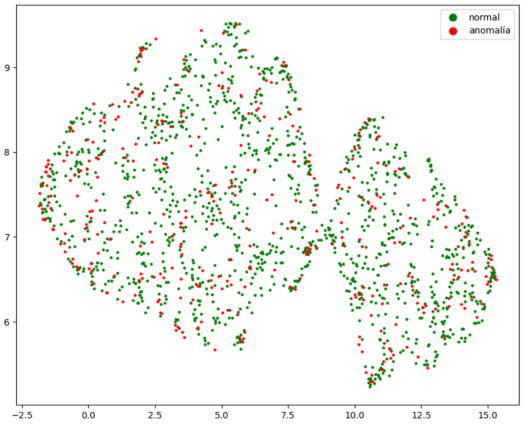
Por último, vamos a observar los UMAPs obtenidos en este análisis, vemos como en los dos primeros gráficos 4.6a y 4.6b se observa un comportamiento muy similar al observado en el análisis anterior. No obstante, en las iteraciones finales (ver Gráfica 4.6c) se empieza a divisar una división entre dos conjuntos, pese a que esto podría ser esperanzador, se sigue viendo como los datos anómalos y normales siguen estando mezclados. Estos dos grupos que podrían estar formandose podrían ser las distintas máquinas que hay en el *dataset* utilizado para este análisis.



(a) UMAP 0



(b) UMAP 10



(c) UMAP 25

Figura 4.6: Visualización UMAP de distintas iteraciones del mismo modelo.

Cabe destacar que la duración de los segmentos de audio es de 10 segundos. La anomalía podría no estar ocurriendo en todo el segmentado de audio. Es decir, supongamos que la anomalía tiene una duración de 1 segundo sobre el segmento de 10. En ese caso, el 90 % del audio, el comportamiento es normal. Con esto, se quiere recalcar que SPADE (un algoritmo pensado para imagen) puede no estar mostrando resultados satisfactorios sobre espectrograma porque no interpreta la anomalía como algo transitorio y cabría la posibilidad de cambiar su implementación para realizar un análisis temporal más exhaustivo, como por ejemplo, en ventanas o espectrogramas de 1 segundo en lugar de 5 o 10.

Tal y como se ha observado que el modelo no es capaz de separar adecuadamente el comportamiento normal de la anomalía. Esto puede deberse a que las señales son muy parecidas entre sí. Además, debido al desbalanceo del conjunto de datos (hay muchas más señales normales que anómalas), el modelo parece clasificar casi todo como normal, sin caracterizar bien la anomalía. Este análisis se utilizó para obtener *insight* y analizar SPADE para futuras modificaciones.

CAPÍTULO 5
Conclusiones y trabajo futuro

5.1 Conclusiones

Este proyecto surge de la necesidad de optimizar la Técnica de Insectos Estériles (TIE) empleando tecnologías de Inteligencia Artificial.

Como resultado de este TFG, se ha implementado un algoritmo del estado del arte para la detección de anomalías conocido como SPADE. Este algoritmo novedoso puede ser desplegado en multitud de dominios, siendo el caso de uso de este TFG la detección de hembras en botes de liberación de mosquitos tigres, dónde sólo deberían haber ejemplares machos. Como fuente de información se utiliza el sonido emitido por el aleteo de los ejemplares, ya que es distinto. El vuelo de las hembras, se considera, por tanto, el comportamiento anómalo. Además, se ha diseñado una base de datos SQL etiquetando y segmentando de forma manual clips de audio de aleteo de mosquitos.

Respecto al modelo, se ha realizado un *Grid-Search* sobre 3 elementos clave del modelo SPADE: el tamaño del *Embedding*, los valores para ponderar la función de coste y el número de clasificadores *one-class* que conforman el *pseudo-labeler* (ver Sección 4.1.1 para un análisis más detallado de los resultados). Los resultados más satisfactorios para cada experimento varían entre el un 62 % y un 67 % de F1-Score. Siendo la mejor configuración de todas un modelo SPADE con un *Embedding* compuesto por 512 neuronas, 30 clasificadores *one-class* y una ponderación de 1 sobre todas las pérdidas que componen el modelo. Con los resultados obtenidos, se puede apreciar que el número de OCCs es un valor crítico y su modificación es la que muestra mejores resultados. El tamaño del *Embedding* parece muy ligado al número de clasificadores, teniendo que ser más grande a medida que aumentan el número de clasificadores, habiendo overfitting en un punto concreto (ver Tabla A.3). El estudio de valores de ponderación sobre la función de coste no es significativo ya que los resultados son menos prometedores.

Con este análisis sobre los resultados, se podría concluir que en este caso de uso la cantidad de audios disponibles parece demasiado baja si se pretende obtener una solución comercial generalizable. Cabe destacar que los audios se toman bajo unos protocolos de sanidad estrictos. Además, el periodo de toma de datos es óptimo entre diciembre y enero (por motivos biológicos de plagas), reduciendo la ventana de toma de datos cada año. Así pues, se cree necesario aumentar el conjunto de datos para poder analizar de una forma más rigurosa el algoritmo SPADE. Con el conjunto de datos disponible a fecha de entrega de este TFG, el mejor resultado obtenido es de un F1 del 67 %.

Un factor crítico que comprometió los resultados fue la combinación de una cantidad limitada de datos y una alta variabilidad en estos. Factores como la etapa de desarrollo

de los mosquitos o la cantidad e intensidad del aleteo introdujeron una capa adicional de complejidad. El modelo enfrentó dificultades para identificar patrones consistentes.

A pesar de estos obstáculos, es importante destacar que el proyecto propone una solución innovadora durante el proceso de la TIE. Este trabajo aporta conocimientos (*know-how*) y establece una base sólida para futuras investigaciones y desarrollos metodológicos y tecnológicos.

Como resumen, en este TFG, se ha implementado y desarollado de una forma parametrizable el algoritmo del estado del arte SPADE. Este algoritmo se ha testeado en dos casos concretos. Sin embargo, puede ser utilizado (si fuese necesario) en otros contextos y escenarios donde sí se disponga de una base de datos considerable. Para su desarrollo, se han aplicado técnicas de contenerización para que su despligue futuro sea lo más eficiente y rápido posible. La implementación de SPADE supone un gran *know-how* en el equipo de trabajo con el que he realizado este TFG.

5.2 Trabajo futuro

Aunque los resultados no han sido tan extremadamente satisfactorios (F1 de 67 % en el mejor de los casos), la implementación del modelo SPADE representa un logro significativo. Esta innovadora arquitectura se compone de diversas unidades independientes, lo que permite la flexibilidad para cambiar componentes específicos. Por ejemplo, si se considera que el *Predictor* no es suficientemente capaz de separar las clases, se podría sustituir por otro. De igual manera, si las características extraídas por el *Encoder* no cumplen con las expectativas, se podrían utilizar otros extractores de características, como BYOL-A [32], opciones exploradas a lo largo del desarrollo del trabajo.

Además, las posibles mejoras también se relacionan con la cantidad y calidad de los audios grabados. La grabación de un mayor número de audios para entrenar los algoritmos de aprendizaje profundo probablemente resultaría en mejoras significativas en los resultados.

Una buena forma de abordar este problema en el futuro sería mediante un solución centrada en el trabajo de los datos (*Data-Centric*) [1] [31]. En lugar de enfocarse exclusivamente en mejorar y experimentar con los algoritmos de aprendizaje automático, se pondría énfasis en la calidad, diversidad y cantidad de los datos utilizados para entrenar el modelo. Esto podría incluir la recolección de más datos representativos, la mejora de la anotación de datos y la aplicación de técnicas de aumentado de datos para enriquecer el conjunto de entrenamiento. Un enfoque centrado en los datos podría ayudar a abordar problemas de sobreajuste, mejorar la generalización del modelo y, en última instancia, conducir a un mejor desempeño en escenarios del mundo real.

Bibliografía

[1] Lora Aroyo et al. «Data-centric AI: A Conceptual Framework and Survey». En: *Journal of Data and Information Quality* 14.1 (2022), págs. 1-28.

[2] Dzmitry Bahdanau, Kyunghyun Cho y Yoshua Bengio. «Neural machine translation by jointly learning to align and translate». En: *arXiv preprint arXiv:1409.0473* (2014). URL: https://arxiv.org/abs/1409.0473.

[3] Daniele Barchiesi et al. «Acoustic scene classification: Classifying environments from the sounds they produce». En: *IEEE Signal Processing Magazine* (2015). DOI: 10.1109/MSP.2015.2398952. URL: https://ieeexplore.ieee.org/document/7078982.

[4] Romeo Bellini et al. «Pilot Field Trials with Aedes albopictus Irradiated Sterile Males in Italian Urban Areas». En: *Journal of Medical Entomology* 50.2 (2013), págs. 317-325. DOI: 10.1603/ME12048.

[5] Bernhard E Boser, Isabelle M Guyon y Vladimir N Vapnik. «A training algorithm for optimal margin classifiers». En: *Proceedings of the fifth annual workshop on Computational learning theory* (1992), págs. 144-152. DOI: 10.1145/130385.130401. URL: https://dl.acm.org/doi/10.1145/130385.130401.

[6] John S Bridle. «Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition». En: *Neurocomputing* (1990). DOI: 10.1007/978-94-009-0643-3_34. URL: https://link.springer.com/chapter/10.1007/978-3-642-76153-9_28.

[7] Varun Chandola, Arindam Banerjee y Vipin Kumar. «Anomaly detection: A survey». En: *ACM computing surveys (CSUR)* 41.3 (2009). DOI: 10.1145/1541880.1541882. URL: https://dl.acm.org/doi/10.1145/1541880.1541882.

[8] Nicholas J. Culbert et al. «A rapid quality control test to foster the development of genetic control in mosquitoes». En: *Scientific Reports* 8.1 (2018). DOI: 10.1038/s41598-018-34501-1. URL: https://www.nature.com/articles/s41598-019-44071-z.

[9] Wei Dai et al. *Very Deep Convolutional Neural Networks for Raw Waveforms*. 2016. arXiv: 1610.00087 [cs.SD].

[10] Steven B Davis y Paul Mermelstein. «Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences». En: *IEEE Transactions on Acoustics, Speech, and Signal Processing*. Vol. 28. 4. IEEE. 1980, págs. 357-366.

[11] Centers for Disease Control y Prevention. *Life Cycle of Aedes Mosquitoes*. Accessed: 2024-05-28. 2024. URL: https://www.cdc.gov/mosquitoes/about/life-cycle-of-aedes-mosquitoes.html.

[12] FAO/IAEA. *The Sterile Insect Technique*. https://www.iaea.org/topics/sterile-insect-technique. 2019.

[13] Dimitrios Giannoulis et al. «Detection and classification of acoustic scenes and events: An IEEE AASP challenge». En: *2013 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics* (2013). URL: https://ieeexplore.ieee.org/document/6701819.

[14] Ian Goodfellow y Aaron Bengio Yoshua y Courville. *Deep Learning*. MIT Press, 2016. URL: https://www.deeplearningbook.org/.

[15] Hasantha Nayana Gunathilaka et al. «Use of mechanical and behavioural methods to eliminate female Aedes aegypti and Aedes albopictus for sterile insect technique and incompatible insect technique applications». En: *Parasites Vectors* 12 (mar. de 2019). DOI: 10.1186/s13071-019-3398-7.

[16] Trevor Hastie, Robert Tibshirani y Jerome Friedman. *The elements of statistical learning: data mining, inference, and prediction*. 2nd. Springer Science Business Media, 2009. URL: https://link.springer.com/book/10.1007/978-0-387-84858-7.

[17] Shawn Hershey et al. «CNN architectures for large-scale audio classification». En: *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2017), págs. 131-135. DOI: 10.1109/ICASSP.2017.7952132. URL: https://ieeexplore.ieee.org/document/7952132.

[18] IBM. *Anomaly Detection*. 2022. URL: https://www.ibm.com/es-es/topics/anomaly-detection.

[19] Sergey Ioffe y Christian Szegedy. «Batch normalization: Accelerating deep network training by reducing internal covariate shift». En: *arXiv preprint arXiv:1502.03167* (2015). URL: https://arxiv.org/abs/1502.03167.

[20] Yuma Koizumi et al. *ToyADMOS: A Dataset of Miniature-Machine Operating Sounds for Anomalous Sound Detection*. 2019. arXiv: 1908.03299 [eess.AS].

[21] Alex Krizhevsky, Ilya Sutskever y Geoffrey E Hinton. «ImageNet classification with deep convolutional neural networks». En: *Advances in neural information processing systems* 25 (2012), págs. 1097-1105. URL: https://dl.acm.org/doi/10.1145/3065386.

[22] Yann LeCun y Yoshua Bengio. «Convolutional networks for images, speech, and time-series». En: *The handbook of brain theory and neural networks* 3361.10 (1995), pág. 1995. URL: https://www.researchgate.net/publication/216792820_Convolutional_Networks_for_Images_Speech_and_Time-Series.

[23] Yann LeCun et al. «Gradient-based learning applied to document recognition». En: *Proceedings of the IEEE* 86.11 (1998). DOI: 10.1109/5.726791. URL: https://ieeexplore.ieee.org/document/726791.

[24] Beth Logan. «Mel frequency cepstral coefficients for music modeling». En: *International Symposium on Music Information Retrieval (ISMIR)*. Vol. 270. 2000.

[25] MathWorks. *Support Vector Machines*. 2022. URL: https://es.mathworks.com/discovery/support-vector-machine.html.

[26] Brian McFee et al. «librosa: Audio and Music Signal Analysis in Python». En: ene. de 2015, págs. 18-24. DOI: 10.25080/Majora-7b98e3ed-003.

[27] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

[28] Vinod Nair y Geoffrey E Hinton. «Rectified Linear Units Improve Restricted Boltzmann Machines». En: *Proceedings of the 27th International Conference on Machine Learning (ICML-10)* (2010). URL: https://www.cs.toronto.edu/~fritz/absps/reluICML.pdf.

[29] Javier Naranjo Alcazar. «Classification of sound scenes and events in real-world scenarios with deep learning techniques». Directores: Pedro Diego Zuccarello (dir. tes.), Máximo Cobos Serrano (codir. tes.) Tesis doct. Universitat de València, 2021. URL: https://dialnet.unirioja.es/servlet/tesis?codigo=298321.

[30] Javier Naranjo-Alcazar et al. «A Comparative Analysis of Residual Block Alternatives for End-to-End Audio Classification». En: *IEEE Access* 8 (2020), págs. 188875-188882. DOI: 10.1109/ACCESS.2020.3031685.

[31] Andrew Ng et al. «Data-Centric AI: A New Approach to Machine Learning». En: *Communications of the ACM* 64.12 (2021), págs. 36-39.

[32] Daisuke Niizumi et al. «BYOL for Audio: Self-Supervised Learning for General-Purpose Audio Representation». En: *arXiv preprint arXiv:2103.06695* (2021). URL: https://arxiv.org/pdf/2103.06695.

[33] Pan American Health Organization. *Zika virus and complications*. https://www.who.int/news-room/fact-sheets/detail/zika-virus. 2022.

[34] Philippos A Papathanos , Hervé C Bossin, Mark Q Benedict, Flaminia Catteruccia, Colin A Malcolm, Luke Alphey, Andrea Crisanti. *Sex separation strategies: past experience and new approaches*. https://pubmed.ncbi.nlm.nih.gov/19917075/. 2009.

[35] Lukas Ruff et al. «A unifying review of deep and shallow anomaly detection». En: *Proceedings of the IEEE* 109.5 (2021). URL: https://arxiv.org/abs/2009.11732.

[36] Stuart Russell y Peter Norvig. *Artificial Intelligence: A Modern Approach*. 3rd. Pearson, 2016.

[37] C Saraswathy, K Nandhini y R Kalaivani. «Traffic sign detection and recognition based on convolutional neural network». En: *International journal of health sciences* (jun. de 2022), págs. 1224-1238. DOI: 10.53730/ijhs.v6nS8.9884.

[38] Jürgen Schmidhuber. «Deep learning in neural networks: An overview». En: *Neural Networks* 61 (2015), págs. 85-117. DOI: 10.1016/j.neunet.2014.09.003. URL: https://www.sciencedirect.com/science/article/pii/S0893608014002135.

[39] Bernhard Schölkopf et al. «Estimating the support of a high-dimensional distribution». En: *Neural computation* 13.7 (2001), págs. 1443-1471.

[40] Jonathan Shen et al. «Natural TTS synthesis by conditioning wavenet on MEL spectrogram predictions». En: *arXiv preprint arXiv:1712.05884* (2018).

[41] Decide Soluciones. *Tipos de aprendizaje automático*. 2022. URL: https://decidesoluciones.es/tipos-de-aprendizaje-algoritmos-machine-learning/.

[42] David MJ Tax y Robert PW Duin. «Support vector data description». En: *Machine learning* 54.1 (2004), págs. 45-66. DOI: 10.1023/B:MACH.0000008084.60811.49. URL: https://link.springer.com/article/10.1023/B:MACH.0000008084.60811.49.

[43] Alan M. Turing. «Computing Machinery and Intelligence». En: *Mind* 59.236 (1950), págs. 433-460. DOI: 10.1093/mind/LIX.236.433.

[44] Vladimir Vapnik y Corinna Cortes. «Support-vector networks». En: *Machine learning* (1995). DOI: 10.1007/BF00994018. URL: https://link.springer.com/article/10.1007/BF00994018.

[45] Xiaolong Wang y Abhinav Gupta. «Learning deep features for scene recognition using places database». En: *Advances in neural information processing systems* 27 (2016). URL: https://dl.acm.org/doi/10.5555/2969442.2969528.

[46] Jinsung Yoon et al. «Self-supervise, Refine, Repeat: Improving Unsupervised Anomaly Detection». En: *arXiv preprint arXiv:2106.06115* (2022). URL: https://arxiv.org/pdf/2106.06115.pdf.

[47] Jinsung Yoon et al. «SPADE: Semi-supervised Anomaly Detection under Distribution Mismatch». En: *arXiv preprint arXiv:2212.00173* (2022). URL: https://arxiv.org/pdf/2212.00173.pdf.

[48] Yue Zhao, Zain Nasrullah y Zheng Li. «PyOD: A Python Toolbox for Scalable Outlier Detection». En: ene. de 2019.

APÉNDICE A
Resultados

A.1 Análisis de resultados

A.1.1. 75-25

En esta sección se analizan los resultados obtenidos con una proporción de 75 % de datos etiquetados y 25 % de datos no etiquetados. En los apartados de abajo se puede observar como la modificación de los distintos parámetros afecta al rendimiento del modelo, este rendimiento queda reflejado en las métricas que se observan *(Auc, Accuracy Y F1-score)*.

Diferentes Número de Características Features

Los resultados muestran cómo varía el rendimiento del modelo al cambiar el número de características. Se puede observar que un aumento en el número de características no siempre mejora las métricas de rendimiento. A continuación, se presentan los resultados obtenidos:

Parámetros				**Métricas**		
***Alpha*(α)**	***Beta*(β)**	**Nº OCCs**	**Nº características**	**AUC**	**F1**	**Accuracy**
1	1	30	64	0.824	0.623	0.75
1	1	30	128	0.670	0.542	0.47
1	1	30	256	0.816	0.600	0.66
1	1	30	512	0.680	0.580	0.81
1	1	30	1024	0.770	0.590	0.75

Tabla A.1: Resultados con diferentes números de características

En esta tabla, se observa que el mejor rendimiento en términos de F1 score se obtiene con 64 características, alcanzando un valor de 0.623. Sin embargo, el aumento a 512 características muestra una mejora en la precisión (Accuracy), aunque no tanto en el F1 score.

Diferentes *Alpha*(α)-*Beta*(β)

Debido a las limitaciones de tiempo, solo se probaron 4 valores diferentes para *Alpha*(α) y *Beta*(β): [1, 1.3, 1.6, 1.9]. En total se obtuvieron 16 resultados, pero solo se reflejan los 5 mejores en la tabla. En esta sección se utilizó el valor de 64 características.

Parámetros				Métricas		
***Alpha*(α)**	***Beta*(β)**	**Nº OCCs**	**Nº características**	**AUC**	**F1**	**Accuracy**
1.9	1.6	30	64	0.861	0.655	0.81
1	1.6	30	64	0.838	0.623	0.76
1.6	1	30	64	0.816	0.605	0.61
1.9	1.9	30	64	0.800	0.600	0.75
1.6	1.3	30	64	0.760	0.587	0.66

Tabla A.2: Resultados con diferentes valores de *Alpha*(α) y *Beta*(β)

Se puede observar que los mejores resultados se obtienen con *Alpha*(α) = 1.9 y *Beta*(β) = 1.6, lo que sugiere que un mayor valor de *Alpha*(α) y *Beta*(β) puede mejorar el rendimiento del modelo.

Diferente número de OCCs

Es importante comprobar cómo se comporta el modelo al cambiar la cantidad de OCCs. En teoría, cuantos más OCCs haya, más seguro estará el modelo de generar *pseudo-labels*. Los resultados obtenidos se muestran a continuación:

Parámetros				Métricas		
***Alpha*(α)**	***Beta*(β)**	**Nº OCCs**	**Nº características**	**AUC**	**F1**	**Accuracy**
1	1	10	512	0.730	0.553	0.48
1	1	15	512	0.690	0.520	0.43
1	1	20	512	0.766	0.556	0.54
1	1	25	512	0.776	0.588	0.678
1	1	30	512	0.890	0.670	0.83
1	1	35	512	0.790	0.607	0.691
1	1	40	512	0.760	0.598	0.72

Tabla A.3: Resultados con diferentes números de OCCs utilizando 512 características

En esta tabla, se observa que el mejor rendimiento en términos de F1 score se obtiene con 30 OCCs y 512 características, alcanzando un valor de 0.670. Esto indica que un número adecuado de OCCs es crucial para el rendimiento del modelo.

A.1.2. 90-10

En esta sección se analizan los resultados obtenidos con una proporción de 90 % de datos etiquetados y 10 % de datos no etiquetados. Se observa cómo varían las métricas en función del número de características utilizadas.

Diferentes Features

Los resultados muestran cómo varía el rendimiento del modelo al cambiar el número de características para la proporción 90-10.

Parámetros				Métricas		
***Alpha*(α)**	***Beta*(β)**	**Nº OCCs**	**Nº características**	**AUC**	**F1**	**Accuracy**
1	1	30	64	0.789	0.524	0.393
1	1	30	128	0.711	0.518	0.389
1	1	30	256	0.823	0.579	0.558
1	1	30	512	0.806	0.549	0.560
1	1	30	1024	0.751	0.516	0.364

Tabla A.4: Resultados con diferentes números de características (90-10)

En esta tabla, se observa que el mejor rendimiento en términos de F1 score se obtiene con 256 características, alcanzando un valor de 0.579. Sin embargo, se nota que el aumento a 512 características no mejora significativamente las métricas de rendimiento.

Diferentes *Alpha*(α)-*Beta*(β)

Debido a las limitaciones de tiempo, solo se probaron 4 valores diferentes para *Alpha*(α) y *Beta*(β): [1, 1.3, 1.6, 1.9]. Se reflejan los 5 mejores resultados. En esta sección se utilizó el valor de 64 características.

Parámetros				Métricas		
***Alpha*(α)**	***Beta*(β)**	**Nº OCCs**	**Nº características**	**AUC**	**F1**	**Accuracy**
1.3	1.3	30	64	0.781	0.547	0.556
1.9	1.6	30	64	0.759	0.561	0.598
1.3	1	30	64	0.777	0.597	0.680
1.3	1.9	30	64	0.815	0.591	0.656
1	1.3	30	64	0.866	0.641	0.665

Tabla A.5: Resultados con diferentes valores de *Alpha*(α) y *Beta*(β) (90-10)

Se puede observar que los mejores resultados se obtienen con *Alpha*(α) = 1 y *Beta*(β) = 1.3, lo que sugiere que un valor más bajo de *Alpha*(α) y un valor intermedio de *Beta*(β) pueden mejorar el rendimiento del modelo en esta proporción.

Diferente número de OCCs

En esta sección se analizan los resultados obtenidos al cambiar la cantidad de OCCs, utilizando 64 características. Este análisis es crucial para entender cómo afecta la cantidad de OCCs en la generación de *pseudo-labels*.

Parámetros				Métricas		
***Alpha*(α)**	***Beta*(β)**	**Nº OCCs**	**Nº características**	**AUC**	**F1**	**Accuracy**
1	1	10	64	0.710	0.528	0.420
1	1	15	64	0.708	0.553	0.550
1	1	20	64	0.811	0.616	0.748
1	1	25	64	0.888	0.655	0.753
1	1	30	64	0.789	0.524	0.393
1	1	35	64	0.599	0.516	0.362
1	1	40	64	0.846	0.403	0.862

Tabla A.6: Resultados con diferentes números de OCCs utilizando 64 características (90-10)

En esta tabla, se observa que el mejor rendimiento en términos de F1 score se obtiene con 25 OCCs, alcanzando un valor de 0.655. Esto indica que hay un número óptimo de OCCs que maximiza el rendimiento del modelo.

A.1.3. 50-50

En esta sección se analizan los resultados obtenidos con una proporción de 50 % de datos etiquetados y 50 % de datos no etiquetados.

Diferentes Features

Los resultados muestran cómo varía el rendimiento del modelo al cambiar el número de características para la proporción 50-50.

Parámetros				Métricas		
***Alpha*(α)**	***Beta*(β)**	**Nº OCCs**	**Nº características**	**AUC**	**F1**	**Accuracy**
1	1	30	64	0.634	0.516	0.362
1	1	30	128	0.691	0.516	0.362
1	1	30	256	0.725	0.516	0.364
1	1	30	512	0.777	0.520	0.391
1	1	30	1024	0.742	0.516	0.364

Tabla A.7: Resultados con diferentes números de características (50-50)

En esta tabla, se observa que el rendimiento del modelo no varía significativamente con el número de características, ya que las métricas de F1 score y Accuracy se mantienen bastante constantes.

Diferentes *Alpha*(α)-*Beta*(β)

Se probaron 4 valores diferentes para *Alpha*(α) y *Beta*(β): [1, 1.3, 1.6, 1.9]. Se reflejan los 5 mejores resultados. En esta sección se utilizó el valor de 64 características.

Parámetros				Métricas		
***Alpha*(α)**	***Beta*(β)**	**Nº OCCs**	**Nº características**	**AUC**	**F1**	**Accuracy**
1.3	1	30	64	0.763	0.593	0.730
1	1.9	30	64	0.818	0.603	0.786
1	1.3	30	64	0.811	0.617	0.753
1.9	1.6	30	64	0.830	0.640	0.891
1.3	1.9	30	64	0.830	0.654	0.730

Tabla A.8: Resultados con diferentes valores de *Alpha*(α) y *Beta*(β) (50-50)

Se puede observar que los mejores resultados se obtienen con *Alpha*(α) = 1.9 y *Beta*(β) = 1.6, lo que sugiere que un mayor valor de *Alpha*(α) y *Beta*(β) puede mejorar el rendimiento del modelo en esta proporción.

Diferente número de OCCs

Es crucial analizar los resultados obtenidos al cambiar la cantidad de OCCs, utilizando 64 características. Esto ayuda a entender cómo afecta la cantidad de OCCs en la generación de *pseudo-labels*.

Parámetros				Métricas		
***Alpha*(α)**	***Beta*(β)**	**Nº OCCs**	**Nº características**	**AUC**	**F1**	**Accuracy**
1	1	10	64	0.804	0.576	0.590
1	1	15	64	0.764	0.578	0.694
1	1	20	64	0.793	0.592	0.780
1	1	25	64	0.791	0.604	0.757
1	1	30	64	0.720	0.559	0.661
1	1	35	64	0.600	0.516	0.362
1	1	40	64	0.818	0.601	0.723

Tabla A.9: Resultados con diferentes números de OCCs utilizando 64 características (50-50)

En esta tabla, se observa que el mejor rendimiento en términos de F1 score se obtiene con 20 OCCs, alcanzando un valor de 0.592. Esto indica que hay un número óptimo de OCCs que maximiza el rendimiento del modelo en esta proporción.

Este apartado simplemente se ha utilizado para exponer la cantidad de pruebas que se han realizado pero tal y como el lector ha podido observar el análisis no ha sido exhaustivo.

APÉNDICE B
Objetivos de Desarrollo Sostenible (ODS)

B.1 Grado de relación del trabajo con los Objetivos de Desarrollo Sostenible (ODS)

Objetivos de Desarrollo Sostenibles	Alto	Medio	Bajo	No Procede
ODS 1. Fin de la pobreza.				✓
ODS 2. Hambre cero.				✓
ODS 3. Salud y bienestar.	✓			
ODS 4. Educación de calidad.				✓
ODS 5. Igualdad de género.				✓
ODS 6. Agua limpia y saneamiento.				✓
ODS 7. Energía asequible y no contaminante.				✓
ODS 8. Trabajo decente y crecimiento económico.				✓
ODS 9. Industria, innovación e infraestructuras.	✓			
ODS 10. Reducción de las desigualdades.				✓
ODS 11. Ciudades y comunidades sostenibles.				✓
ODS 12. Producción y consumo responsables.				✓
ODS 13. Acción por el clima.				✓
ODS 14. Vida submarina.				✓
ODS 15. Vida de ecosistemas terrestres.	✓			
ODS 16. Paz, justicia e instituciones sólidas.				✓
ODS 17. Alianzas para lograr objetivos.				✓

Reflexión sobre la relación del TFG/TFM con los ODS y con el/los ODS más relacionados.

B.1.1. Salud y Bienestar

Tal y como expongo a lo largo de la memoria, muchos científicos argumentan que la próxima pandemia a nivel mundial podría ser transmitida por los mosquitos. En este contexto, la utilización de la Técnica del Insecto Estéril (TIE) podría ser clave para la mitigación del virus y el control de la situación. En mi Trabajo de Fin de Grado (TFG), se presenta un algoritmo de audición por computador que actúa como un control de calidad y detector de anomalías justo antes de la liberación de los machos estériles al medio ambiente. Este sistema asegura que no haya hembras en los botes de liberación, un paso crucial para garantizar el éxito de la TIE. Al eliminar la posibilidad de liberar hembras, se reduce significativamente el riesgo de transmisión de enfermedades como el dengue y el zika, contribuyendo así a la salud pública y al bienestar de las comunidades afectadas.

B.1.2. Industria, Innovación e Infraestructuras

Este trabja se ha desarrollado en un contexto empresarial, y la innovación que supondría su éxito e implementación es considerable. La solución propuesta mejora la eficiencia de un proceso que se realiza diariamente en empresas dedicadas al control de mosquitos. Implementar un sistema de audición por computador para la detección de hembras en los botes de liberación no solo incrementa la precisión del proceso de sexado, sino que también reduce el tiempo y los recursos necesarios para realizar este control de calidad. Esta innovación tecnológica contribuye al avance de las infraestructuras empresariales, haciendo el proceso más fiable y eficiente. Por tanto, considero que este Objetivo de Desarrollo Sostenible (ODS) tiene una gran relación con mi trabajo, ya que promueve el uso de nuevas tecnologías y la mejora continua de los procesos industriales.

B.1.3. Vida de Ecosistemas Terrestres

En una situación en la que los mosquitos fueran los causantes de la transmisión de enfermedades, se debería actuar con rapidez para evitar una crisis de salud pública. Sin soluciones que no dañen el medio ambiente y los ecosistemas terrestres, la alternativa sería el uso masivo de pesticidas, lo cual podría tener consecuencias devastadoras para la biodiversidad y la salud del suelo y del agua. La implementación de la TIE ofrece una solución sostenible que garantiza la protección de los ecosistemas terrestres. Al liberar solo machos estériles, se controla la población de mosquitos sin el uso de químicos dañinos. Por ello, toda mejora en el proceso de la TIE, como la que propongo en mi TFG, representa un paso significativo hacia una futura solución respetuosa con el medio ambiente. Esta técnica no solo preserva la biodiversidad, sino que también contribuye a la sostenibilidad a largo plazo de los ecosistemas terrestres, demostrando cómo la tecnología puede ser aliada de la conservación ambiental.