UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería Informática

Implementación de un algoritmo de aprendizaje activo sobre datos ambientales del puerto de Valencia

Trabajo Fin de Grado

Grado en Ciencia de Datos

AUTOR/A: Ribes Serrano, Ruben

Tutor/a: Morillas Gómez, Samuel

Cotutor/a externo: Naranjo Alcázar, Javier

CURSO ACADÉMICO: 2023/2024

Resum

El rendiment de solucions basades en tècniques d'Intel.ligència Artificial (IA) depèn, en gran mesura, del conjunt de dades utilitzat per al seu entrenament. No solament la quantitat de dades, sinó la qualitat d'aquestes. L'objectiu principal d'aquest projecte és la implementació d'un algorisme d'Aprenentatge Actiu (AL per les seues sigles en anglés) en un context on la quantitat de dades disponibles no etiquetades és molt major que el pressupost d'etiquetatge (en aquest projecte es treballarà amb senyals d'àudio ambientals). Per tant, la duració total del conjunt de dades és molt major al temps disposat per etiquetar-lo. L'aplicació d'aquestes tècniques garanteix que les dades a etiquetar siguen interessants en termes de diversitat, per tal d'optimitzar un entrenament d'un algorisme d'IA amb la quantitat de dades etiquetades. Es pot entendre l'AL com un mètode per a l'optimització dels recursos d'etiquetatge. Cal destacar que per a l'algoritme d'AL s'han utilitzat tècniques de *Deep* i *Machine Learning*. Com a cas pràctic d'ús s'han utilitzat gravacions realitzades en el port de València obtingudes durant l'execució del projecte Soroll-IA2 financiat per el Instituto Valenciano de Competitividad Empresarial (IVACE) i el Fondo Europeo de Desarrollo Regional (FEDER). El projecte s'ha realitzat durant les pràctiques de Grau a l'Institut Tecnològic d'Informàtica (ITI).

Paraules clau: Audició per Computador, Xarxes Neuronals Profundes, Aprenentatge Actiu, Sons Ambientals

Resumen

El rendimiento de las soluciones basadas en técnicas de Inteligencia Artificial (IA) dependen, en gran medida, del conjunto de datos utilizado para su entrenamiento. No sólo la cantidad de los datos, si no la calidad de estos. El objetivo principal de este proyecto es la implementación de un algoritmo de Aprendizaje Activo (*AL* por sus siglas en inglés) en un contexto donde la cantidad de datos disponibles no etiquetados es mucho mayor que el presupuesto de etiquetado (en este proyecto se trabajarán con señales de audio ambientales). Por tanto, la duración total del conjunto de datos es mucho mayor al tiempo que se puede emplear para etiquetarlo. La puesta en marcha de estas técnicas garantiza que los datos que se vayan a etiquetar sean interesantes y ricos en términos de diversidad para así poder optimizar un entrenamiento de un algoritmo de IA con la cantidad de datos etiquetados. Se puede entender el *AL* como un método para la optimización de los recursos de etiquetado. Cabe destacar que para el algoritmo de *AL* se han utilizado técnicas de *Deep* y *Machine Learning*. Como caso práctico de uso se han utilizado grabaciones realizadas en el puerto de Valencia obtenidas durante la ejecución del proyecto Soroll-IA2 financiado por el Instituto Valenciano de Competitividad Empresarial (IVACE) y el Fondo Europeo de Desarrollo Regional (FEDER). El proyecto se ha realizado durante las prácticas de Grado en el Instituto Tecnológico de Informática (ITI).

Palabras clave: Audición por Computador, Redes Neuronales Profundas, Aprendizaje Activo, Sonidos Ambientales

Abstract

The performance of Artificial Intelligence (AI)-based solutions largely depends on the dataset used for their training. Not only the amount of data, but also the quality of the data. The main objective of this project is the implementation of an Active Learning (AL) algorithm in a context where the amount of available unlabeled data exceeds the labeling budget (in this project, will be worked on enviromental audio signals). Therefore, the

total duration of the dataset is much bigger than the available time for labeling it. The deployment of these techniques ensures that the data that will be labeled are interesting in terms of diversity, thus optimizing an AI algorithm with the labeled data as train data. AL can be understood as a method for optimizing labeling resources. It is worth noting that Deep and Machine Learning techniques have been used for the AL algorithm. As a practical use case, recordings made in the port of Valencia have been employed obtained during the implementation of the Soroll-IA2 project funded by the Instituto Valenciano de Competitividad Empresarial (IVACE) and the European Regional Development Fund (ERDF). The project has been carried out during the Bachelor's degree internships at the *Instituto Tecnológico de Informática* (ITI).

Key words: Machine Listening, Deep Neural Networks, Active Learning, Ambient Sounds

Índice general

Índice de figuras

Índice de formulas

Índice de tablas

CAPÍTULO 1
Introducción

La Inteligencia Artificial (IA) se ha convertido en una herramienta indispensable en múltiples áreas, abarcando desde la optimización de procesos industriales hasta el análisis de grandes volúmenes de datos para extraer patrones y realizar predicciones en la actualidad. Las predicciones por parte de un modelo de IA permiten la realización de tareas de forma automática sin la intervención de un ser humano.

Es imprescindible destacar que la IA promete revolucionar la manera en que entendemos y interactuamos con nuestro entorno, tanto de forma acústica como visual. Si nos centramos en la parte acústica, nos encontramos con el campo de la IA llamado Audición por Computador (*Machine Listening* en inglés). Este campo trata de implementar soluciones capaces de obtener información relevante a partir de señales de audio. En este estudio se trabajará con datos de audio genéricos, es decir, aquellos que no contienen componentes musicales ni de habla humana.

Algunas de las aplicaciones que se pueden realizar con este tipo de audios son: clasificación de escenas sonoras [6], detección de eventos sonoros [7], definición del audio mediante una oración [8] o la detección de anomalías sonoras en máquinas [9], entre otras.

Este trabajo ha sido realizado en el marco del proyecto de investigación Soroll-IA2[1] (IMDEEA/2023/91), realizado durante mis prácticas en el centro tecnológico privado Instituto Tencológico de Informática (ITI). El objetivo del proyecto es crear una base de datos de audios genéricos en un entorno portuario. Para ello, se han desplegado varios nodos acústicos en distintas ubicaciones del puerto de Valencia. El proyecto ha sido financiado por el Instituto Valenciano de Competitividad Empresarial (IVACE) y el Fondo Europeo de Desarrollo Regional (FEDER). Dada la gran cantidad de datos que se esperan recopilar, se considera imposible etiquetarlos todos manualmente. Este Trabajo Final de Grado tiene como objetivo la investigación e implementación de técnicas que optimicen el proceso de etiquetado. Es decir, técnicas que de forma inteligente seleccionen qué muestras son las más relevantes para ser etiquetadas por un humano y que optimicen el rendimiento de un modelo de IA futuro. Por lo tanto, se investigarán técnicas de Aprendizaje Activo (*AL*, por sus siglas en inglés). Se conoce como Aprendizaje Activo al conjunto de técnicas destinadas a maximizar el rendimiento de un algoritmo de IA para un problema en específico, con la limitación de los recursos disponibles para el etiquetado (horas totales que se pueden etiquetar).

[1] https://www.iti.es/proyectosidi/soroll-ia2/

1.1 Motivación

Actualmente, el desarrollo comercial de modelos de *Machine Listening* presenta un gran problema: **la carencia de grandes volúmenes de datos de audio etiquetados públicos que puedan ser utilizados con fines comerciales**.

La grabación de estos datos no implica una gran complicación técnica, debido a la disposición de nuevas tecnologías de grabación y almacenamiento. No obstante, existe un problema técnico que se encuentra durante el proceso de etiquetado de estos audios. El etiquetado manual es una tarea tediosa y costosa, tanto en términos de tiempo como de coste y en contextos con grandes volúmenes de datos es imposible etiquetar todo el conjunto.

Ante este escenario, se pretende investigar e implementar una solución basada en Aprendizaje Activo *(Active Learning* en inglés) para optimizar el proceso de etiquetado de datos. El Aprendizaje Activo es una subrama del aprendizaje automático (*Machine Learning* en inglés), que a su vez, se trata de una rama de la IA basada en la creación de algoritmos que permiten a los sistemas informáticos aprender a partir de los datos. El *Active Learning* permite seleccionar de forma autónoma e inteligente los datos que, una vez etiquetados manualmente por una persona, deberían ser más beneficiosos para el aprendizaje de una solución basada en IA. Esto significa que, en lugar de etiquetar un gran conjunto de datos de manera aleatoria, el algoritmo identifica aquellos audios que realmente considera que contribuyen a mejorar el rendimiento y la precisión de un modelo futuro.

1.2 Objetivos

El propósito de este trabajo es abordar un desafío significativo del etiquetado de ficheros de audio en el contexto en el que las horas de grabación son mucho mayores que las horas disponibles para el etiquetado de los mismos. El conjunto de datos se trata de grabaciones diarias obtenidas del puerto de Valencia. Los audios fueron recogidos con el despliegue de nodos de grabación en 4 localizaciones distintas. La grabación era contínua almacenando ficheros de audio de 10 segundos.

Este trabajo se centra en el desarrollo e implementación de un modelo de Aprendizaje Activo que permita optimizar el proceso de etiquetado. Dado el volumen de los datos recogido, que se tratan de grabaciones contínuas, es decir, 24 horas de audio diariamente, la tarea manual de etiquetado total del conjunto de datos es descartada. Por tanto, existe la necesidad de una solución que optimice el etiquetado de tantos datos. El principal objetivo es desarrollar y aplicar un algoritmo de *Active Learning* presente en el estado del arte de esta área de conocimiento. A continuación, se detallan los objetivos específicos que se abarcaran este proyecto.

- **Implementación de un algoritmo del estado del arte**: el primer objetivo consiste en la implementación del algoritmo de *Active Learning* presentado en la siguiente colección de artículos científicos: "*Active learning for sound event classification by clustering unlabeled data*" [1], "*An active learning method using clustering and committee-based sample selection for sound event classification*" [2] y "*Active learning for sound event detection*" [4].
- **Creación de flujo de trabajo que sea capaz de lidiar con la cantidad de datos generada en nuestro contexto**: el segundo objetivo consiste en el desarollo de un flujo de trabajo que incorpore el algoritmo de *Active Learning*. Los algortimos de *Active*

Learning presentes en la literatura han sido testeados con bases de datos pequeñas y públicas. Como consecuencia, estos algoritmos no son capaces de lidiar con la cantidad de datos presente en este proyecto. Por tanto, se ha tenido que implementar un proceso que incorpora técnicas para filtrar y optimizar las muestras que van a ser procesadas por el propio algoritmo de *Active Learning*.

1.3 Estructura de la memoria

La memoria se estructura mediante 5 capítulos:

- **Capítulo 1, Introducción:** en este capítulo se introduce el trabajo realizado, empezando con una explicación de la motivación detrás del proyecto, los objetivos específicos que se pretenden alcanzar y una descripción de la estructura del documento, detallando de forma breve el contenido de cada uno.
- **Capítulo 2, Estado del arte:** durante este capítulo se realiza una revisión de la literatura existente y se presentan los conceptos y tecnologías que han resultado de gran utilidad para el trabajo. Esto incluye un análisis de los algoritmos de Aprendizaje Activo existentes y que se han considerado relevantes para el desarrollo de este proyecto.
- **Capítulo 3, Metodología:** en esta parte de la memoria, se describe la metodología empleada en el desarrollo del proyecto. Durante el capítulo, se detallan los procesos de recolección de datos y del sistema de etiquetado, incluyendo las bases de datos utilizadas y la presentación del proceso de etiquetado. También se explica el enfoque de Aprendizaje Activo adoptado y proporcionando detalles del experimento.
- **Capítulo 4, Resultados:** se mostrarán y explicarán los resultados obtenidos del proceso de etiquetado además de realizar un análisis visual del comportamiento del algoritmo de *Active Learning*.
- **Capítulo 5, Conclusiones:** en este último capítulo, se resumen las principales conclusiones del trabajo, destacando los resultados más importantes y el cumplimiento de los objetivos planteados. También se discuten posibles mejoras para trabajos futuros y la relación del trabajo con los estudios cursados.

CAPÍTULO 2
Estado del arte

Este capítulo va a constar de dos grandes bloques: trabajos relacionados con el Aprendizaje Activo (ver sección 2.1) y un bloque relacionado con la redes neuronales convolucionales para la clasificación de audio. Siendo la red conocida como *PANNs* [10] el centro de la explicación (ver sección 2.2).

En el primer bloque se presentan y explican las distintas implementaciones de *Active Learning* existentes como *MAL* (ver subsección 2.1.1), *MAL-MF* (ver subsección 2.1.2) y *Active Learning for Sound Event Detection* (ver subsección 2.1.3). A continuación, en el segundo bloque se introducirá un modelo de redes neuronales convolucionales llamado *PANNs* el cual va a ser necesario para nuestro proceso de Aprendizaje Activo.

Por último, se explicará más en detalle la idea del algoritmo a conseguir, usando la información de los estudios anteriores, además de las diferencias que propone frente a los algoritmos existentes.

2.1 Algoritmos de Aprendizaje Activo existentes

Para el desarrollo de modelos de clasificación y detección de sonido, es imprescindible contar con datos de audio etiquetados para su entrenamiento. Aunque el proceso de grabación es sencillo, el hecho es que, el etiquetado de los mismos es muy costoso en términos de tiempo ya que exige un tiempo o superior a su duración para poder ser etiquetados. Durante este punto hablaremos del presupuesto de etiquetado (*labeling budget* en inglés, *LB* por sus siglas) que se refiere al número de horas o muestras que se pueden etiquetar dentro del conjunto de datos total. En la Fórmula 2.1 se define el término *LB*.

$$LB = HD_{\text{etiquetador}} \times N = \frac{HD_{\text{etiquetador}}}{DE_{\text{audio}}} \times N = NA_{\text{etiquetador}} \times N \tag{2.1}$$

Donde $HD_{\text{etiquetador}}$ hace referencia al número de horas disponible de etiquetado de un etiqutador, N al número de etiqutadores disponibles, DE_{audio} el tiempo que cuesta etiquetadar un audio y $NA_{\text{etiquetador}}$ el número de audio que puede etiquetar un etiqueador. Como se puede apreciar, el *LB* se puede definir en función del tiempo o en función al número de audios (diviendo el tiempo total por una estimación de cuanto tiempo se tarda en etiquetador un audio).

Dado el presupuesto limitado para el etiquetado y la imposibilidad de etiquetar todas las horas disponibles, es imprescindible optimizar la selección de muestras para el etiquetado manual. Con este fin, se ha implementado una estrategia de Aprendizaje Activo diseñada para seleccionar de manera óptima las muestras que deben ser etiquetadas. Es-

ta metodología mejora significativamente la selección aleatoria, asegurando un uso más eficiente de los recursos disponibles y maximizando el beneficio del proceso de etiquetado en términos de calidad y efectividad. Se puede definir al Aprendizaje Activo como el campo de la IA que tiene como objetivo optimizar el presupuesto de etiquetado obteniendo el mejor rendimiento posible de un algoritmo de IA para un problema concreto con el menor número de muestras etiquetadas. Las distintas técnicas de elección de las muestras se explicarán en los siguientes apartados de esta memoria. En este contexto, el proceso de *Active Learning* se realizará de forma cíclica. En cada ciclo/itearción el algoritmo de *Active Learning* propondrá un conjunto de audios a etiquetar. Una vez etiquetados, estos audios ya revisados por un ser humano se utilizarán en el mismo algoritmo para afinar la selección de forma cíclica. El uso de iteraciones viene motivado por la idea de que se dispone de un tiempo limitado de etiquetado por semana/día y no de una disponiblidad total para el etiquetado. Dado un número de horas totales para etiquetar, éstas se dividen de forma semanal y cada semana, el algoritmo de *Active Learning* se lanza, propone la muestras a etiqutear de acuerdo al *LB* de esa semana y se etiquetan.

2.1.1. *Medoid-Based Active Learning* (MAL)

El primer trabajo de *Active Learning* estudiado propone el algoritmo llamado aprendizaje activo basado en medioides (*MAL*, por sus siglas en inglés) [1]. De acuerdo con los autores, este algortimo es útil en el caso de que el *labeling budget* sea bajo. La idea principal de esta solución se basa en llevar a cabo un aprendizaje no supervisado llamado *K*-medioides en cada iteración.

Un método de aprendizaje no supervisado es una técnica de *Machine Learning* en la que el algoritmo debe encontrar patrones existentes en un conjunto de datos no etiquetado. Por otro lado, el aprendizaje supervisado se puede definir como el conjunto de soluciones que tienen como objetivo encontrar relaciones entre los datos de entrada y de salida (conocidas como etiquetas o clases). Una de las técnicas por las cuales el aprendizaje no supervisado consigue su objetivo de encontrar patrones y estructuras en los datos es mediante la agrupación. Esta técnica es conocida como algoritmos de agrupamiento (*clustering* en inglés) y consiste en agrupar los datos en grupos (*clusters* en inglés) de datos similares en función de sus características. El algoritmo de agrupamiento *K*-mediodes consiste en dividir los datos en un número de grupos preestablecido, *K*, en los que cada grupo viene definido por un medioide. El medioide se trata del dato dentro de un grupo cuya suma de diferencias con el resto de datos de la agrupación, es mínima. Se puede definir el medioide como el dato más representativo de un *cluster*.

Una de las limitaciones del algoritmo reside en el establecimiento del número de grupos por parte del usuario (*K*). Un número bajo puede conllevar a una agrupación errónea donde distintos grupos pueden ser agrupados al mismo. Por contrapartida, un valor de *K* elevado puede conducir a subdiviones del mismo grupo. Tal y como se presenta en el trabajo [1], un valor óptimo puede ser $K = n/4$ donde n es el número total de muestras presentes en el conjunto de datos.

La primera iteración del algoritmo *MAL* consiste en la aplicación del algoritmo *K*-mediodes para poder obtener *K clusters* sobre el conjunto de datos marcados como no escuchados (*unlistened* en inglés). Cabe destacar que en esta iteración, todos los audios están marcados como no escuchados. Una vez se obtienen los grupos, el medioide de cada grupo será el seleccionado para etiquetar manualmente.

El método de inicialización del algoritmo *K*-medioides es el conocido como "más lejano primero"(*farthest-first* en inglés) [11, 12]. Este método tiene como objetivo optimizar la selección de medioides iniciales. Para ello, se parte de un dato aleatorio y se genera un

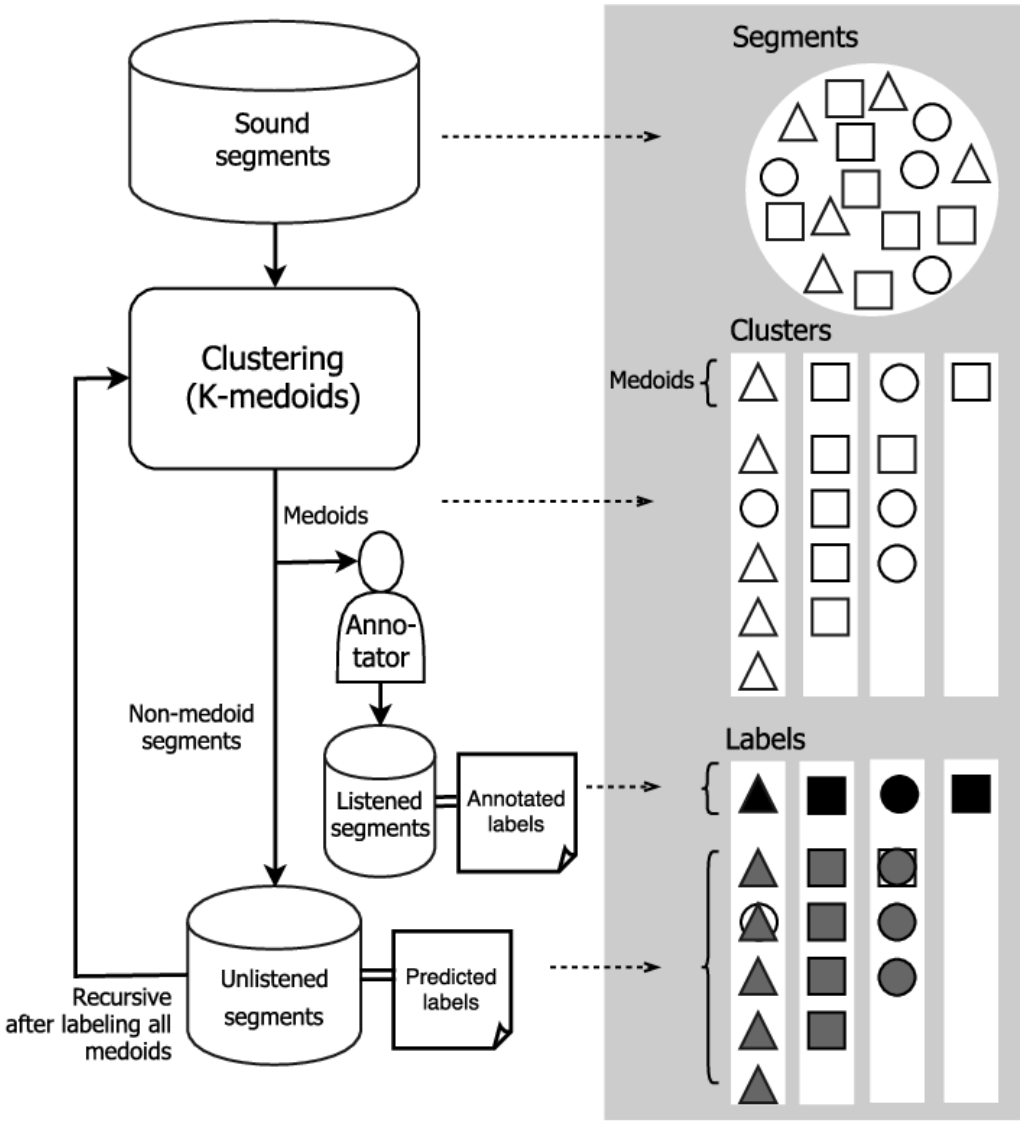

Figura 2.1: Vista del algoritmo *MAL* [1].

grupo de medioides. Desde ese punto inicial, se van añadiendo al grupo los datos más alejados. El proceso de adición finaliza cuando se obtienen K muestras en el grupo. Un mayor detalle sobre este proceso se puede encontrar en los trabajos [11, 12]. Una vez obtenido este conjunto inicial, todas los datos disponibles se asignan a su medioide más cercano. Realizado este paso, cada grupo que se ha creado actualiza su medioide por el dato dentro del grupo el cual la suma de distancias a los demás datos sea mínima. Estos dos pasos anteriores se realizan de forma iterativa hasta que la actualización de los medioides ya no mejore la suma de diferencias con los demás miembros de la agrupación.

Ya con el conjunto de medioides final, cada medioide será etiquetado de forma manual asignando una etiqueta de dentro de un conjunto de etiquetas predefinido. Los medioides se presentan al etiquetador en orden descendente según el tamaño del grupo, es decir de medioide que forma parte de un grupo grande a medioide que forma parte de grupos más pequeños. La etiqueta asignada al medioide será propagada a los demás elementos del mismo grupo.

Tras el etiquetado de los medioides, estos se marcan como "escuchado" (*listened* en inglés). Este proceso de *Active Learning* se repite para el conjunto de datos que estan marcados como *unlistened* hasta la finalización del *LB*. De esta forma, en las siguientes

iteraciones, se propondrán nuevos medioides que no han sido etiquetados manualmente con anterioridad.

De acuerdo con el artículo [1], todos los datos están marcados de forma inicial como *unlistened* y "no etiquetados"(*unlabeled* en inglés). Cada vez que se etiqueta manualmente un medioide, este pasará a ser *listened* y "etiquetado"(*labeled* en inglés), mientras que la etiqueta de este medioide que es propagada a los demás miembros del grupo hará que estos datos esten marcados como *labeled* y *unlistened*. Todas las ejecuciónes del *Active Learning* se haran sobre los datos marcados como *unlistened*.

La gran ventaja de este algoritmo reside en que desde la primera iteración, se tiene un conjunto de datos etiquetado, ya sean estas etiquetas obtenidas de forma manual por un etiquetador o por la propagación de éstas primeras. Una vez finalizado el presupuesto de etiquetado, cada muestra tiene asiganada una etiqueta, ya haya sido porque ha sido considerada medioide o por el simple hecho de pertener a un clúster. La principal limitación de este algoritmo reside en el valor fijo y predefinido del valor de K que no aporta versatilidad a la creación de nuevas etiquetas o a un funcionamiento erróneo en caso de que la elección del valor de K no haya sido correcta desde un inicio.

En la Figura 2.1 se puede observar de forma visual la metodología seguida por el algoritmo *MAL* a partir del conjunto de datos de audio inicial.

2.1.2. *Medoid-Based Active Learning with Mismatch-First* MAL-MF

El siguiente algoritmo que se va a presentar se trata de una evolución del anterior algoritmo llamado *MAL* propuesta por los mismos autores [2] definida como *MAL-MF*. La principal mejora de esta propuesta reside en la optimización respecto al uso de medioides etiquetados para iteraciones futuras. En *MAL*, una vez los medioides han sido etiquetados manualmente, el algoritmo repite de nuevo todo el proceso sobre los datos no etiquetados pero con la particularidad de que estos datos etiquetados, no se reutilizan para poder mejorar la selección de datos a etiquetar.

La metodología de *MAL-MF* se divide en dos fases que se van a explicar a continuación. Se puede ver este procedimiento ilustrado en la Figura 2.2.

1. La primera fase es, de forma muy similar, el método *MAL* con la particularidad de que el valor K se define de una manera distinta. En esta nueva actualización el tamaño medio de clústers, KI, varia según el conjunto de datos presentado. Cabe recordar que en la Sección 2.1.1 el valor de K se establecía a $n/4$.

    - De forma simplificada, esta primera fase que se puede ver en la Figura 2.2 en verde, se realiza un K-medioides donde se obtienen un conjunto de muestras (medioides) con la mínima distancia entre los datos de su propio grupo. Estas muestras seleccionadas, son las primeras que pasan por el proceso de etiquetado. l igual que MAL, las etiquetas son propagadas a los miembros que componen el clúser.
    - Para obtener el nuevo valor de K en establecido para *MAL-MF* se propone usar el método llamado "prueba de vecinidad cercana" (*median neighborhood test method* en inglés). Este método, como se explica de forma más detallada en [2], se usa con la principal idea de intentar estimar el tamaño de la mayor agrupación de forma fiable. Se utiliza el etiquetado manual de un pequeño número de datos, empezando por seleccionar un dato pivote p que tenga una distancia media a su vecino más cercano. Luego, se inicializa un contador i y se incrementa cada vez que un vecino i de p tenga la misma etiqueta. Cuando esto no ocurre, se establece el valor de KI como i.

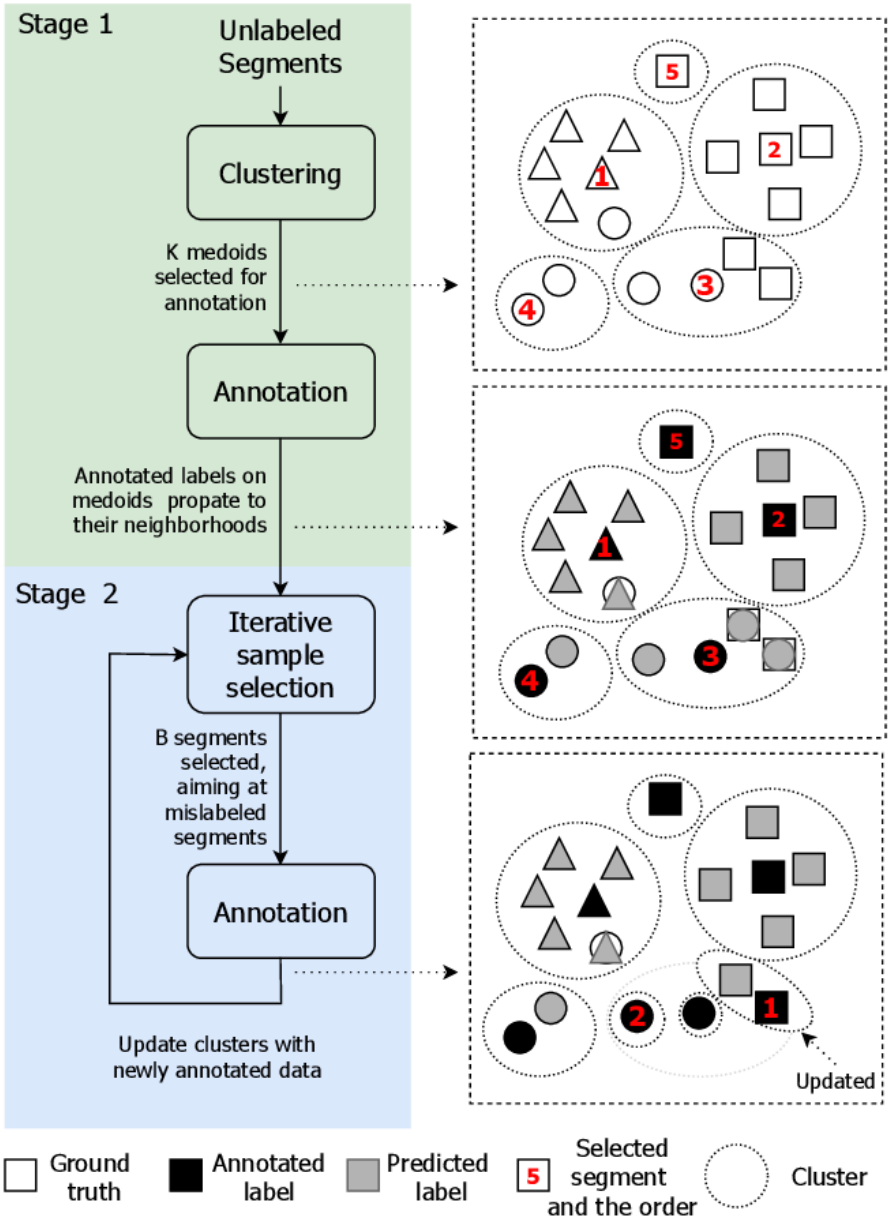

Figura 2.2: Funcionamiento del método *MAL-MF* [2].

2. La segunda fase consiste en un proceso iterativo de recomendación de audios a etiquetar que son seleccionados basándose en la técnica *Mismatch-First Farthest-Search*.

   - Se puede apreciar en la Figura 2.2 que en el inicio de la segunda fase del algoritmo parte de un conjunto de datos (resultado de la primera fase). Este lote de muestras han sido etiquetadas por los etiquetadores, además de contener también, muestras con etiquetas "predichas" fruto de la propagación de las etiquetas de los etiquetadores.
   - El proceso iterativo consiste en, que a partir de un conjunto de datos con etiquetas, se utiliza un criterio basado en la discrepancia entre clasificadores para la elección de nuevos lotes de datos para etiquetar (*Mismatch-First*). El método propuesto usa dos clasificadores: un clasificador que predice etiquetas según su vecino más cercano (que se tratan de las etiquetas propagadas anteriores) y un modelo de clasificación entrenado con los datos etiquetados disponibles hasta el momento.

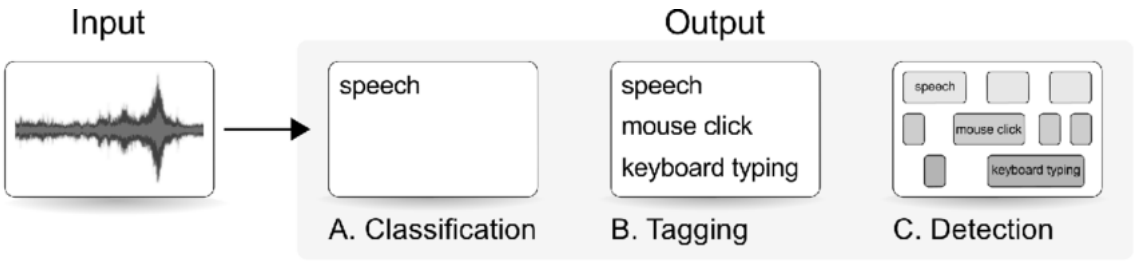

Figura 2.3: Ejemplo de clasificación, etiquetado y detección de eventos sonoros [3].

- Como se ha comentado en el párrafo anterior, el criterio principal para la selección de nuevos datos para etiquetar se basará en la discrepancia entre los dos clasificadores presentados conocido como *Mismatch-First* en inglés. Sin embargo, si el presupuesto de etiquetado es menor al número de datos seleccionados para etiquetar, se recurrirá a otro criterio para la selección, el llamado *Farthest-Traversal*. Este segundo criterio ordenará los datos seleccionados por la discrepancia de acuerdo a la distancia de la muestra a su dato etiquetado más cercano. El orden se hace de forma descendente, es decir, las muestras discrepantes con datos etiquetados más lejanos son las primeras en el ránking. Con esta aproximación, se asume que éste será más probable de que esté mal etiquetado. En el caso de que el presupuesto de etiquetado sea mayor a la cantidad de predicciones con discrepancia, se obtendrán los siguientes datos según el criterio de *Farthest-Traversal* pero sobre el conjunto de datos que no han tenido discrepancia en su predicción.
- Este proceso se repite de forma iterativa hasta finalizar el *LB*.

2.1.3. Aprendizaje activo para detección de eventos sonoros

Este último algoritmo presentado en [4] se entiende como la última versión realizada por los autores de los algoritmos de *Active Learning* explicados anteriormente. El artículo propone un algoritmo de Aprendizaje Activo para detección de eventos sonoros (*SED*, según sus siglas en inglés)

Hasta ahora, un algoritmo de Aprendizaje Activo no se había aplicado en modelos *SED* ya que la mayoría de estudios previos se habían hecho sobre clasificación de sonidos. La diferencia entre un problema de clasificación (o etiquetado) y uno de detección reside en que en el primero la salida del modelo sólo hace referencia a las clases presentes en el clip de audio mientras que en el último, el modelo debe predecir el tiempo de inicio y fin de cada una de las clases presentes. En la Figura 2.3 se muestran las salidas de modelos de audición por computador dependiendo el problema que se pretenda solucionar. El desarrollo de un modelo *SED* se hace basado en aprendizaje supervisado, exigiendo una gran cantidad de datos etiquetados para su entrenamiento, lo que resulta siendo un proceso costoso tanto en tiempo como en recursos.

El nuevo sistema propuesto incluye unas novedades considerables a comparación con los algoritmos mostrados en los puntos anteriores. Primeramente, se introduce un nuevo concepto basado en detección de puntos de cambio (*change point detection*) que consiste en evitar la generación de segmentos de audios los cuales tengan eventos sonoros fragmentados. Se usa este método para poder hacer una partición en segmentos de tamaño variable de los audios no etiquetados del conjunto de datos. Seguidamente, la selección de segmentos de audio para su etiquetado, se hace según el principio llamado *Mismatch-First Farthest-Traversal* (*MFFT*, según sus siglas en inglés). Principio que ha resultado efectivo en anteriores algoritmos de Aprendizaje Activo como en *MAL-MF*.

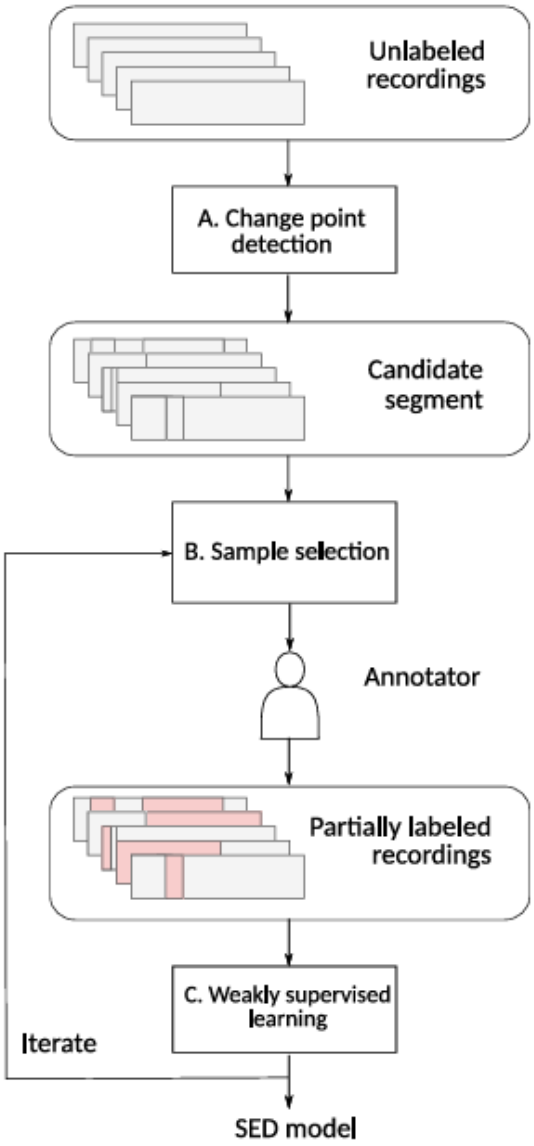

Figura 2.4: Vista previa del sistema de Aprendizaje Activo propuesto [4].

Según esta metodologia se elimina el *clustering* propuesto originalmente en *MAL*. Como resultado, en este proceso no se necesita el valor del número de clusters K como hiperparámetro.

El proceso de *Active Learning* usando el principio de *Mismatch-First Farthest-Traversal* se puede apreciar de una forma visual en la Figura 2.5. Inicialmente, se realiza la detección de puntos de cambios sobre el conjunto de audios no etiquetados, separando cada audio en segmentos candidatos a poder ser seleccionados para su posterior etiquetado. Al tratarse de un proceso iterativo, en cada iteración se conseguirá un lote de segmentos a etiquetar. Para cada iteración el lote propuesto será etiquetado por los etiquetadores y el modelo *SED* será entrenado con estos mismos datos etiquetados. La selección de las muestras se basa en el principio MFFT siendo *Mismatch-First* como el criterio principal priorizando la discrepancia entre predicciones y *Farthest-Traversal* como criterio secundario maximizando la diversidad entre las muestras.

En la primera iteración, no existen segmentos etiquetados, por esta razón, se realiza directamente el segundo criterio (*Farthest-Traversal*) ya mencionado para seleccionar un

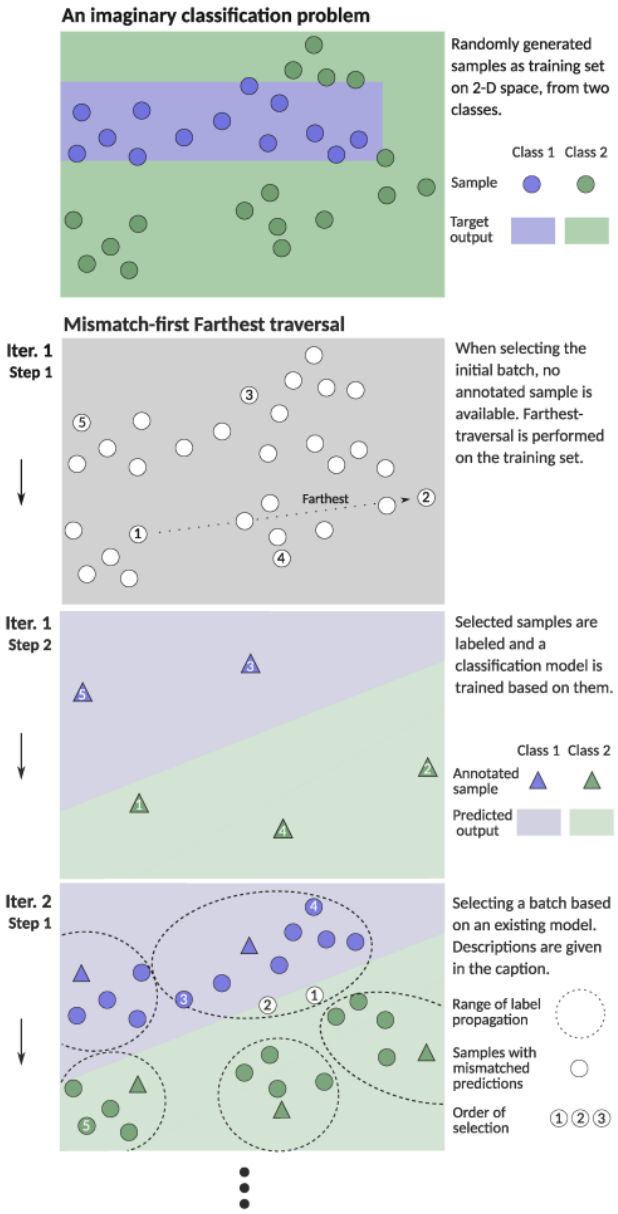

Figura 2.5: Proceso del *Mismatch-First Farthest-Traversal* para un problema de clasificación binario [4].

conjunto de muestras que se encuentren más alejadas entre ellas. Una vez esta primera selección de muestras es etiquetada, en las siguientes iteraciones dos tipos de predicciones se realizan para cada muestra no etiquetada. Por un lado, una predicción es realizada por un modelo *SED* entrenado con los datos etiquetados hasta el momento, y por otro lado, otra predicción es realizada basándose en el método del vecino más cercano, asignando a los segmentos no etiquetados la etiqueta de la muestra etiquetada más cercana. El orden con el que se muestran las muestras propuestas depende de varios factores. Primero, basándose en el primer criterio (*Mismatch-First*), las muestras se mostraran según las similitudes entre predicciones, siendo las muestras con menos similitud las primeras. Si a las predicciones realizadas por el modelo *SED* las indicamos como A_x y a las predicciones basadas en el vecino más cercano como B_x, en un problema de multi-etiqueta (el mismo audio puede contener varias etiquetas a la vez), la similitud entre estas dos predicciones será medida por el índice de Jaccard mostrado en la ecuación 2.2

$$J(x) = \begin{cases} \frac{|A_x \cap B_x|}{|A_x \cup B_x|}, & \text{si } A_x \cup B_x \neq \emptyset \\ 1, & \text{si } A_x \cup B_x = \emptyset \end{cases} \tag{2.2}$$

En el caso de que existiese múltiples muestras se tengan un valor de similitud en su predicción igual, se pasará al segundo criterio (*Farthest-First*), que seleccionará, dentro de estas muestras de igual similitud, la más alejada.

2.1.4. Otros proyectos relacionados con el *Active Learning*

Además de los algoritmos de Aprendizaje Activo presentados anteriormente, que serán los cuales van a tomarse como referencia para el desarrollo de este TFG, existen otros enfoques de *Active Learning* relevantes en el dominio de la audición por computador.El trabajo presentado en [13] propone un algortimo de *Active Learning* basado en la incertidumbre de un clasificador. La idea principal de estos métodos basados en la incertidumbre es que las muestras no etiquetadas con la menor probabilidad de clasificación a una clase serían principalmente candidatas a ser propuestas para su etiquetado. A medida de que nuevas muestras son etiquetadas, el clasificador se entrena con estos nuevos datos. Distintos trabajos que incluyen este método pueden verse en [14, 15, 16]. Sin embargo, aunque los enfoques basados en la incertidumbre proporcionen muestras a etiquetar muy significativas, pueden llevar a sesgos debido a errores de etiquetado. Para evitarlo, en [17] se propone una solución que consiste en etiquetar muestras con un nivel de incertidumbre bajo cada número específico de iteraciones (muestras que el clasificador predice con una alta probabilidad).

2.2 PANNs (Pretrained Audio Neural Network)

La extracción de información relevante de señales de audio es una tarea fundamental con multitud de aplicaciones.Algunos ejemplos pueden ser asistetes domésticos[18], monitorización de tráfico [19, 20] o sistemas de detección de fallos [21], entre otras. El campo de estudio que tiene este objetivo es conocido como Audición por Computador o *Machine Listening* en inglés. Algunas soluciones de *Machine Listening* pueden ir desde el reconocimiento del habla [22] hasta la clasificación de eventos en sonidos ambientales [23]. En el contexto de este TFG, se trabajará con audios ambientales (sin componentes de habla) obtenidos en el puerto de Valencia. Al igual que en otros campos de conocimiento, como por ejemplo, la visión por computador, las soluciones basadas en redes neuronales convolucionales (*CNN* por sus siglas en inglés) se encuentran en el estado del arte [24, 25, 21]

Una red neuronal convolucional es un tipo de red neuronal que esta especialmente diseñada para procesar con datos bidimensionales. Estas redes están basadas en filtros (bidimensionales) que extraen patrones [5]. Estas redes fueron diseñadas especialmente para trabajar sobre imágenes. Debido al buen rendimiento de estas redes sobre imágenes, las soluciones de *Machine Listening* han emulado su flujo de trabajo. Para ello, el primer paso de muchas soluciones de *Machine Listening* consiste en transformar la señal de audio unidimensional (1D) a una señal bidimensional (2D) [24]. Normalmente, la representación bidimensional elegida es el espectrograma sobre el banco de filtro Mel [26]. El espectograma sobre un banco de filtros Mel es una representación que mapea la variación de la frecuencia de un sonido durante el tiempo. Para ello, se extraen las componentes frecuenciales sobre una pequeña ventana temporal que se va moviendo por todo el audio. En la Figura 2.6 se visualiza un espectrograma de un audio del puerto de Valencia.

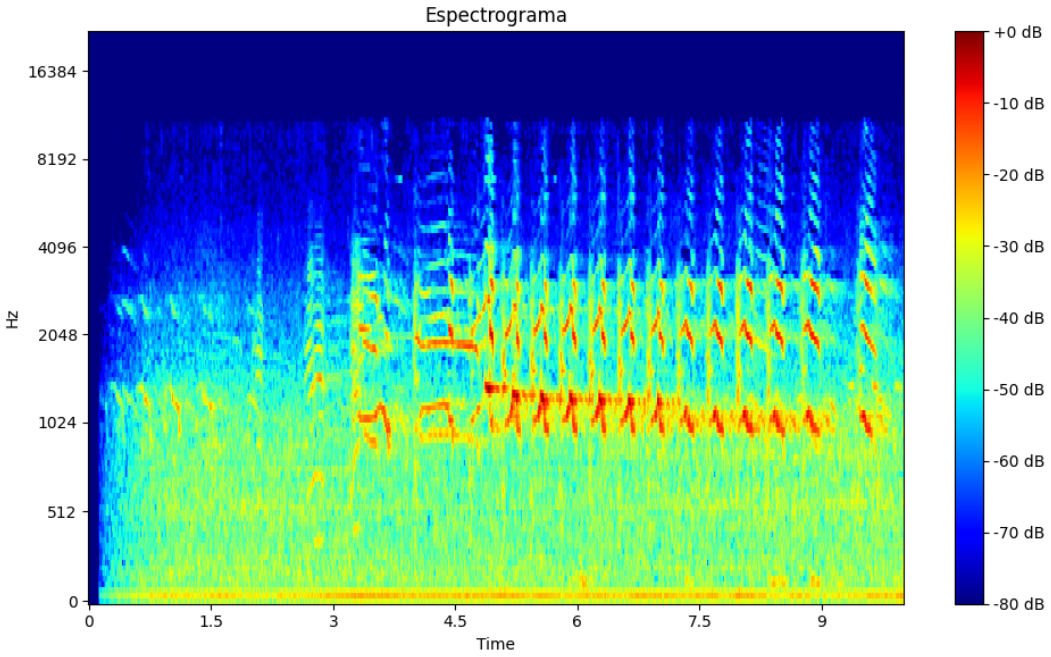

Figura 2.6: Espectograma de un audio grabado en el puerto de Valencia n el que se puede apreciar el sonido emitido por una gaviota

Esta representación (el espectrograma basado en el banco de filtro Mel) es la entrada de la *CNN*. Esta señal bidimensional es procesada por distintas capas convoluciones compuestas por filtros que extraen patrones locales. Esta información obtenida se va combinando de forma progresiva en las siguientes capas, permitiendo así, que la red aprenda patrones cada vez más complejos.

Las *CNNs* también incluyen capas de submuestreo o *pooling* en inglés cuya función es reducir la dimensionalidad de los mapas de características obtenidos después de las capas convolucionales. Además, comúnmente, las *CNNs* acaban con capas completamente conectadas que tienen como objetivo mapear las características extraídas por las capas convolucionales en una salida deseada, como sería el caso de una etiqueta de clasificación. En la Figura 2.7 se puede ver la arquitectura de la red *CNN* conocida como *LeNet* [5] que se muestra como ejemplo de una arquitectura CNN estándar.

Entre las arquitecturas de *CNN* para audición por computador más destacadas en el estado del arte se encuentran *SincNet* [27], *VGGish* [28] o *PANNs* (*Pretrained Audio Neural Networks*) [10]. Debido a los resultados [29, 30, 31, 32] y a la versatilidad a la hora de implementarla por la disposición de un repositorio público[1], se ha decidido utilizar la red conocida como *PANNs* en diferentes puntos de la solución de *Active Learning* (su uso se explica más en detalle en la Sección 3.3 del Capítulo 3). *PANNs* consta de un conjunto de CNNs con distintas arquitecturas, estando algunas de ellas ya entrenadas sobre la base de datos pública Audioset [33].

AudioSet se trata de un conjunto de datos de audio masivo con millones de clips de audio de 10 segundos de duración clasificados en 527 clases. Este conjunto de datos fue creado por *Google* extrayendo la señal de audio de videos de *YouTube*. Las 527 etiquetas abarcan una amplia gama de eventos sonoros incluyendo sonidos de animales, sonidos de naturaleza, sonidos de objetos cotidianos, instrumentos musicales, entre otros. La disponibilidad pública y la diversidad de etiquetas que presenta *AudioSet* convierten el conjunto de datos en un activo valioso para la investigación en el campo del procesamiento

[1] https://github.com/qiuqiangkong/audioset_tagging_cnn

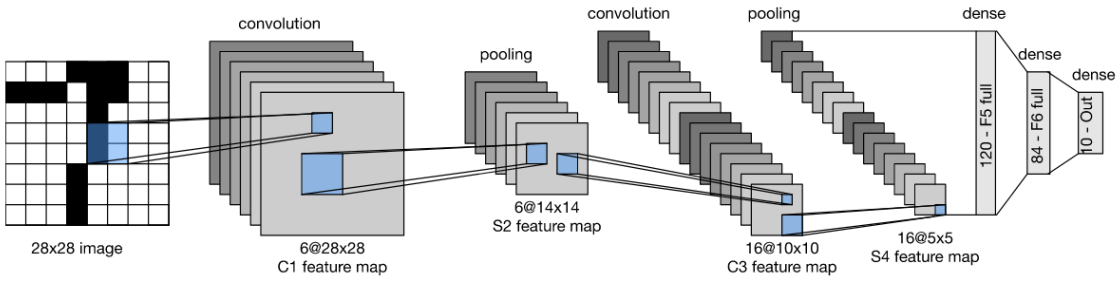

Figura 2.7: Arquitectura de LeNet [5] donde se pueden ver las distintas capas convolucionales.

de audio. Cabe destacar que Audioset sólo puede ser utilizado para fines académicos y no se puede comercializar ninguna solución que haya sido entrenada con este conjunto de datos. Este fenómeno hace que sea de vital importancia la disponiblidad de una base de datos propia (por parte de cualquier empresa/compañía) para poder comercializar soluciones de *Machine Listening*. La base de datos más amplia presente en el estado del arte no puede ser utilizada para tales fines.

La métrica usada para evaluar los distintos modelos presentes en *PANNs* se trata del promedio medio de precisión, más comúnmente conocida por sus siglas en inglés *mAP*. El uso de esta métrica es habitual para evaluar el rendimiento de modelos de clasificación. Se calcula como el promedio del valor de precisión media (*Average Precision, AP*) para cada clase presente en el conjunto de datos. La fórmula para el cálculo de la *mAP* se puede ver en la ecuación 2.3 donde n es el número total de clases presentes en el conjunto de datos y AP_i es el *Average Precision* para la clase i.

$$mAP = \frac{1}{n}\sum_{i=1}^{n} AP_i \tag{2.3}$$

Las tres redes principales que componen *PANNs* son las nombradas como *Cnn14*, *Cnn10* y *Cnn6* las cuales se diferencian por el número de capas convolucionales que las componen. La red neuronal convolucional que mejor resultado ha obtenido ha sido la *Cnn14* entrenada con un *mAP* (*Mean average precision*) de 0.439 sobre el conjunto de datos de *Audioset*. Esta métrica refleja la eficacia de *Cnn14* en la tarea de reconocimiento de audio, consolidándola como la opción más efectiva dentro del marco de *PANNs* para la clasificación de sonidos en entornos complejos y diversos.

La red *Cnn14* entrenada con el conjunto de datos *Audioset* ha sido utilizada en este TFG para:

- La extracción del vector de características (obtención de la representación interna de la red antes de la capa de clasificación) de los datos de audio. Este vector se utiliza para entrenar unos de los clasificadores del algoritmo de *Active Learning*.

- Obtención de la predicción del audio, es decir, la etiqueta que define al audio.

El motivo de estos cálculos se explicaran más en detalle en el Capítulo 3 en la Sección 3.3.

2.3 Innovaciones y contribuciones del proyecto

La principal contribución de este TFG reside en implementar y adaptar un algoritmo de *Active Learning* del estado del arte [4]. Debido a la gran cantidad de datos disponibles provenientes de las grabaciones en el entorno portuario, se ha tenido que implementar un pre-procesado (diseñado en el marco de este TFG) previo al algoritmo de *Active Learning*. Para este pre-procesado, se ha utilizado una red pre-entrenada de *PANNs*, en concreto, la red *Cnn14*.

A modo de resumen, el objetivo de este TFG es conseguir un proceso de Aprendizaje Activo que se use regularmente y que sea capaz de trabajar con grandes volúmenes de datos. Este planteamiento responde a la necesidad de ampliar las limitaciones de los actuales algoritmos, que se han implementado para problemas con datos no mayores a 25 horas de audio. Al contrario, nuestro proyecto va a trabajar sobre datos obtenidos de grabaciones continuas en el puerto de Valencia. Por lo tanto, será necesario hacer una adaptación para poder gestionar esta magnitud de información, la cual se realizará, mediante el uso de *PANNs* .

En la próxima sección del proyecto, *Metodología*, se detallará la implementación realizada para alcanzar los objetivos propuestos del proyecto.

CAPÍTULO 3
Metodología

En este capítulo se van a presentar los pasos que se han llevado a cabo para implementar un sistema de Aprendizaje Activo en el caso de uso práctico de datos provinientes del entorno portuario de Valencia. Los pasos seguidos buscan la optimización del entrenamiento de un futuro modelo de audición por computador proponiendo el etiquetado de las muestras que sean más informativas para un modelo de IA y evitando así redundancias o posibles sobreajustes, es decir, un proceso de selección inteligente al contrario de lo que podría ser una selección aleatoria.

Cabe destacar que el trabajo que se ha desarrollado en este proyecto ha sido la realización de un proceso de *Active Learning* para la obtención de datos etiquetados con la suficiente calidad como para poder entrenar un clasificador propio que se adapte al contexto de los datos recogidos, el de una zona portuaria.

El marco de este TFG engloba el uso y tratamiento de los datos provenientes del puerto de Valencia. Cabe destacar que la fase de recolección de los datos se realizó antes del inicio de este TFG. Sin embargo, se presenta de una breve explicación del proceso de recolección de audio para que la explicación del proceso de *Active Learning* sea entendible y así tener, además, una visión global del proyecto Soroll-IA2 (presentado en el Capítulo 1).

3.1 Recolección de datos

El primer paso para poder realizar cualquier proyecto de IA es la recolección de los datos. En algunas aplicaciones o contexto, el uso de datos públicos es suficiente. No obstante, este no es el escenario en este TFG ya que no existen bases de datos públicas con audios portuarios. Además, este proceso se hace de forma propia e independiente ya que así se pueden desplegar soluciones con total libertad puesto que los datos no son propiedad de un externo. La idea es obtener datos que tengan un contexto en específico, el contexto sería un entorno portuario, en este caso, el puerto de Valencia.

Para poder obtener los datos, se procedió a hacer un despliegue estratégico de micrófonos, también nombrados como nodos, en puntos específicos del puerto de Valencia que han sido cuidadosamente seleccionados. Se decidió posicionar 4 nodos tal y como se pueden apreciar en la Figura 3.1. Algunos de los eventos sonoros interesantes que se pretenden captar con estos nodos son: trenes de mercancías, maquinária portuaria, construcción, buques de carga, entre otros.

Los nodos desplegados en el puerto se tratan de dispositivos IoT (Internet de las Cosas). Un dispositivo IoT consta de componentes electrónicos que le permiten conectarse a Internet y recopilar, enviar y recibir datos, además de poder ser monitorizados de forma remota. Para este estudio, se usará como dispositivo de IoT una Raspberry Pi que estará

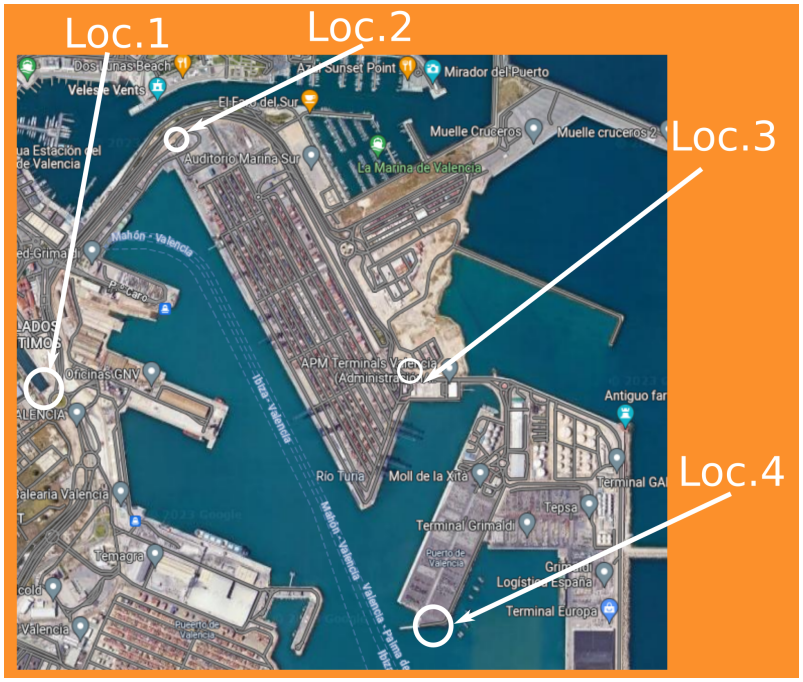

Figura 3.1: Localización de los nodos desplegados en el puerto de Valencia.

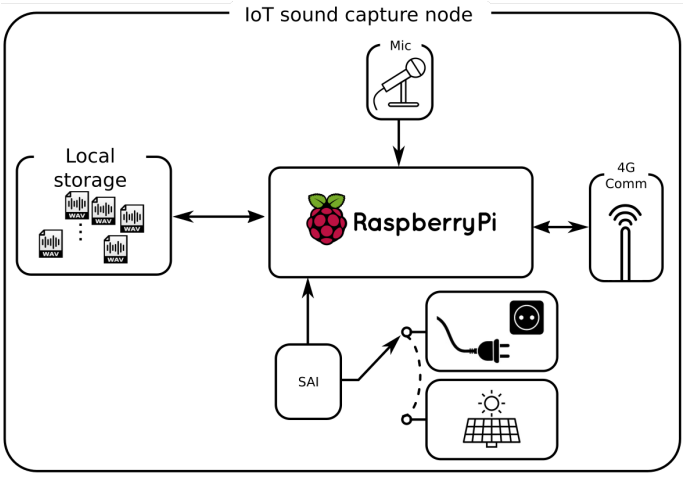

Figura 3.2: Diseño y programación del montaje de un nodo.

programada para poder grabar y almacenar clips de audio de 10 segundos de forma continua sin solape. Estos clips se almacenarán una tarjeta de memoria en el formato *Wav*. El nodo incluye Sistema de Alimentación Ininterrumpida (SAI) para el apagado seguro del dispositivo en caso de corte abrupto de la corriente eléctrica. Además, en caso de que no se disponga de toma de corriente, el nodo se alimenta con paneles solares y el SAI es el encargado del encendido y el apago del nodo según el nivel de batería. En la Figura 3.2 se muestra un diagrama de bloques de los componentes del nodo. Los nodos presentes en la *Loc.1* y *Loc.2* son nodos con toma de luz y los otros dos con paneles solares (ver Figura 3.1)

La conectividad a Internet de este dispositivo permite la monitorización del estado de la batería, el nivel de almacenamiento y la temperatura. Cuando el almacenamiento esta casi lleno, se realiza una visita al puerto para reemplazar las tarjetas de memoria.

En la Figura 3.3 se observa la instalación de un nodo con paneles solares (*Loc.4*). Como se puede apreciar, los componentes se encuentra dentro de una caja industrial que los

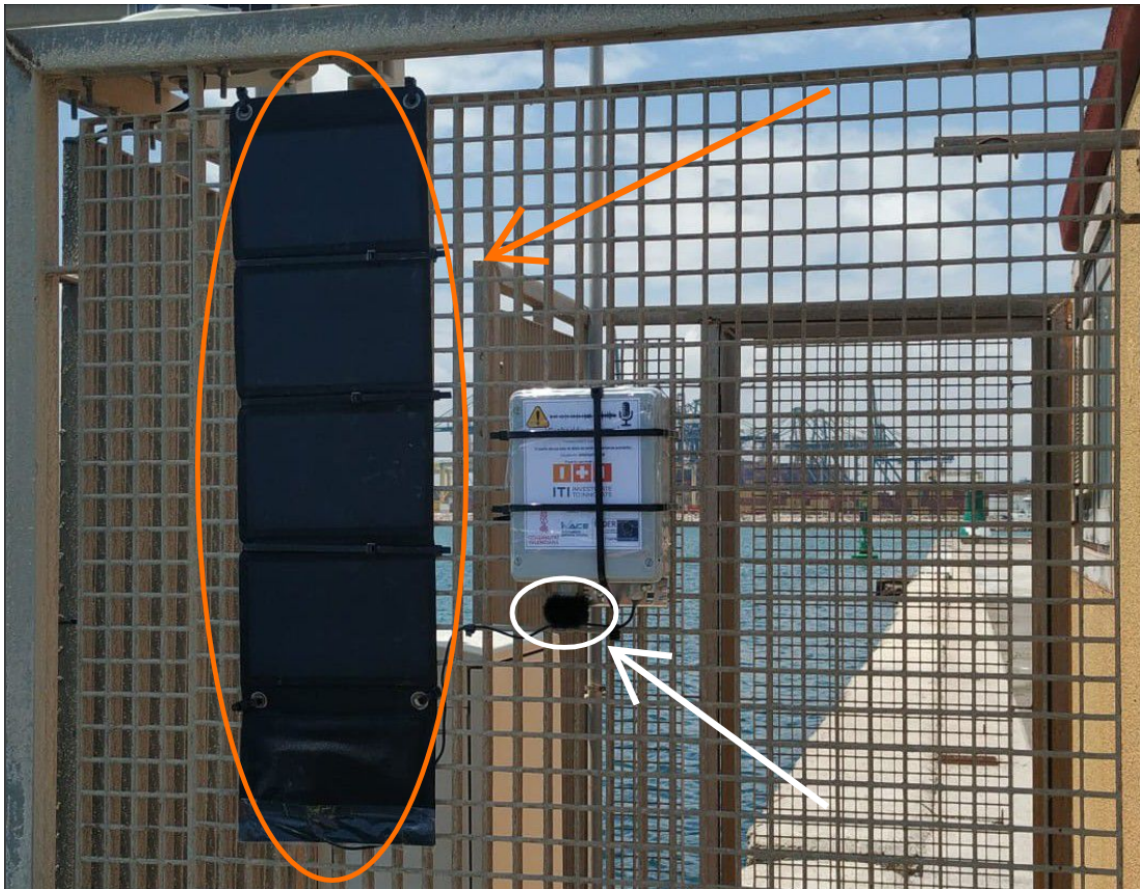

Figura 3.3: Foto de un nodo desplegado en el puerto de Valencia en el muelle de la Xità, donde la flecha naranja apunta a los paneles solares y la blanca al micrófono.

aísla. El micro sale de la caja en la parte inferior de la misma y los paneles solares se encuentran a la izquierda de la caja.

3.2 Implementación del flujo de etiquetado

Para el desarrollo de nuestro proyecto, se ha decidido implementar dos bases de datos. Una de ellas para el registro de las iteraciones del proceso de *Active Learning* y otra donde se almacena la información del etiquetado (ver Subsección 3.2.1). Además, se ha elaborado un proceso de etiquetado por grupos para tener una validación cruzada de cada etiquetador, intentando minimizar sesgos o etiquetas mal asignadas de forma puntual por alguno de ellos (ver Subsección 3.2.2). Este enfoque ha permitido la creación de un flujo de trabajo continuo con el propósito de etiquetar de forma eficiente los datos propuestos por el algoritmo de Aprendizaje Activo (ver Sección 3.3).

3.2.1. Bases de datos

La información relevante sobre el proyecto se almacena en dos bases de datos llamadas ***SOROLL-IA*** y ***AL***.

Por un lado, la primera base de datos (*SOROLL-IA*) contiene toda la información necesaria para el futuro desarrollo de soluciones de IA de forma personalizada (el diseño e implementación de esta base de datos está fuera del marco de este TFG). A modo resumen, a continuación se presenta en detalle cada una de las tablas que la componen:

- *Nodes*: Información asociada a cada dispositivo de grabación IoT desplegado.
- *Paths*: Rutas hacia las carpetas donde los audios se encuentran almacenados.
- *Ontology*: Las etiquetas que se encuentran en la base de datos.
- *Labelers*: Información (Nombre e Id) sobre los etiquetadores existentes en el proyecto.
- *Audios*: Información sobre los audios recogidos por los nodos como el nombre del archivo, la frecuencia de muestreo, la duración del audio...
- *Chunks*: Información sobre el proceso de etiquetado es almacenada en esta tabla. Cada fila de la tabla hace referencia a una etiqueta propuesta por un etiquetador en un audio. Cada etiquetador puede proponer una etiquetas distintas para un mismo audio, indicando el inicio de esta etiqueta y el final.

Por otro lado la segunda base de datos (*AL*) se enfoca en información relevante del algoritmo de *Active Learning*, registrando las ejecuciones realizadas, los datos usados en cada ejecución y los audios propuestos a etiquetar. La base de datos contiene las dos tablas presentadas en el diagrama de la Figura 3.4. Se procede a realizar una descripción breve de la información contenida en cada tabla.

- *ALPreprocessing*: Información relevante sobre las ejecuciones del algoritmo de *Active Learning* como la ventana temporal del análisis, el número de audios propuestos, el número de particiones (dato que se explicará más en detalle en la Sección 3.3) y el procentage de datos etiquetados para el análisis.
- *WavsProposed*: Información sobre los audios propuestos por el algoritmo de Aprendizaje Activo. Esta tabla contiene campos como la etiqueta del audio una vez ha sido etiquetado (*mediod*), cuántos etiquetadores han etiquetado este audio (*max_labelers*) además del porcentage de concordancia para la etiqueta (*agreement*, el nombre del audio (*wav_name*) y el identificador del nodo *id_node*.

Todos estos campos se explicarán y detallarán en la Sección 3.3.

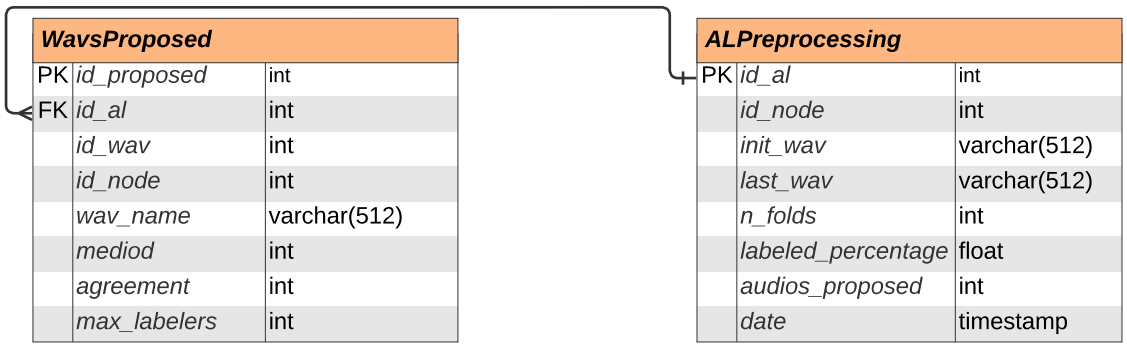

Figura 3.4: Diagrama de la base de datos de *AL*

3.2.2. Proceso de etiquetado

El proceso del *Active Learning* (ver Sección 3.3) se va a llevar a cabo cada semana, es decir, cada semana varios etiquetadores realizarán el proceso de etiquetado manual de

los nuevos datos propuestos por el algoritmo. Para su ejecución, se define una ventana temporal (entre que días y meses se pretende analizar) y un nodo especifico a estudiar para que el algoritmo recomiende qué audios son los más interesantes para el etiquetado dentro de los requisitos previamente mencionados. Los audios propuestos por el algoritmo de *Active Learning* son divididos en dos grupos disjuntos. Puesto que los etiquetadores disponbiles son 5, éstos son divididos en dos grupos de 3 y 2 individuos. El *labeling budget* de cada grupo se establece a 200 audios semanales por grupo, es decir, 40 audios al día durante 5 días (el algoritmo de *Active Learning* debe proponer 400 audios semanales a etiquetar). Cada uno de estos audios es etiquetado por los miembros del grupo mediante una herramienta web externa que facilita el etiquetado de cada audio. Cada etiquetador tiene la posibilidad de seleccionar distintas ventanas temporales del audio para asignar las etiquetas que considere necesarias, pudiendo sobreponer unas etiquetas sobre otras. Las etiquetas disponibles, inicialmente, serán las presentes en el conjunto de datos de *Audioset*. Sin embargo, a medida de que se van etiquetando audios semana tras semana, se va considerando la posibilidad de ampliar el conjunto de etiquetas para incluir etiquetas nuevas que sean relevantes. De esta forma, se garantiza que el sistema de etiquetado se va adaptando a las necesidades del proyecto, como es en nuestro caso, un entorno portuario.

3.3 Aprendizaje Activo

El mayor desafío que presenta este trabajo es procesar un conjunto de datos de grandes dimensiones recogidos del puerto de Valencia. Los algoritmos de *Active Learning* presentados en la Sección 2.1 han sido testeados con conjuntos de datos mucho más pequeños que con los que este trabajo presenta. Por esta razón, es necesario incorporar el algoritmo de Aprendizaje Activo [4] a un flujo más complejo para lidiar con este entorno.

La implementación de este algoritmo de *Active Learning* parte del código proporcionado en el artículo[4] que tiene implementado el algoritmo *MAL-MF* aunque con ligeras modificaciones con respecto a la teoría del algoritmo presentada en el artículo. La modificación más destacable se trata que, en la primera fase del *MAL-MF*, aún sin tener ningún dato con etiqueta, se realiza un *MAL* el cual tenia la novedad de que el valor de K del K-medioides que se realizaba, variaba con el tamaño del conjunto de datos. Sin embargo, el código no implementa esa mejora y se establece K a 4, como en la primera implementación de *MAL*.

Esta implementación del algoritmo, aplica técnicas de propagación tanto por el uso de K-medioides como por la técnica de vecinos más cercanos. Estas técnicas necesitan de operaciones que requieren una capacidad de cómputo a la altura del tamaño de los datos procesados, como es para el caso del cálculo de una matriz de distancias. La matriz de distancias calculada se usa para poder identificar que muestras tienen más semejanzas entre sí a partir del vector de características proporcionado por cada muestra. El tamaño de la matriz escala con el volumen del conjunto de datos, incrementando a su vez, la demanda de recursos computacionales.

Al trabajar con un conjunto de datos con un tamaño extenso, se requiere de una gran cantidad de recursos computacionales para hacer una matriz de distancias de todo el conjunto de datos, además de que, de forma periódica, este conjunto de datos irá aumentando su tamaño. Para manejar con este problema, se realiza una preselección de los datos que se vayan a pasar al algoritmo de *Active Learning* para su procesado. Esta selección se va a hacer manteniendo la idea principal del algoritmo de Aprendizaje Activo, el cuál nos indica que aquellos datos que nos interesa estudiar y etiquetar son aquellos que presenten más variabilidad dentro del conjunto de datos presentado.

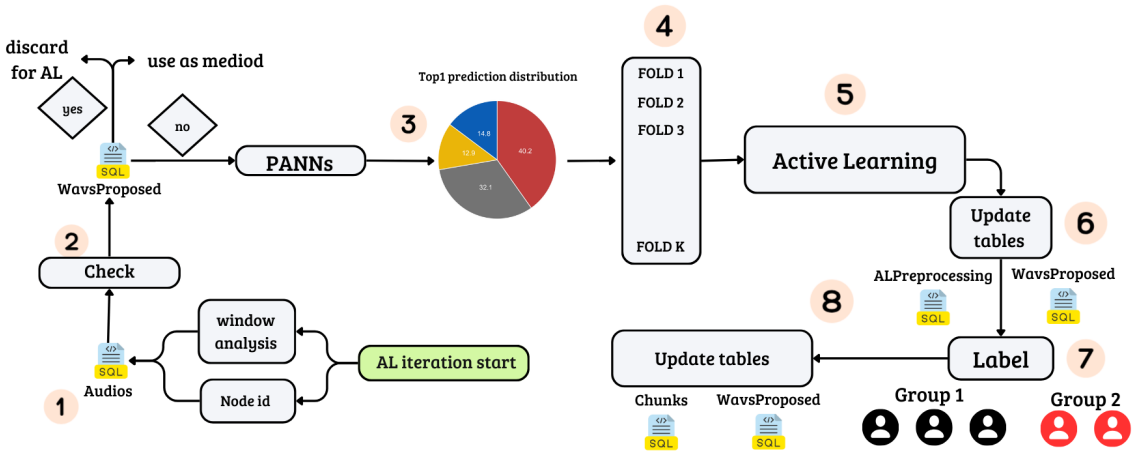

Figura 3.5: Vista del flujo de trabajo del *Active Learning*

Para explicar de manera adecuada y clara el procedimiento seguido para poder conseguir un algoritmo de *Active Learning* para procesar un gran volúmen de datos se va a seguir los pasos mostrados en la Figura 3.5. Los pasos enumerados a continuación coinciden con los pasos en la imagen.

1. En primer lugar, debido a la gran cantidad de datos disponibles, el marco de trabajo del *Active Learning* debe realizarse sobre una ventana temporal y espacial, escogiendo, de esta manera, un rango de tiempo y un micrófono (nodo) específico. Esta selección de los datos se hará haciendo una consulta a la tabla *Wavs* de la base de datos de *SOROLL-IA*.

2. A continuación, se realiza una comprobación de los datos seleccionados con la tabla *WavsProposed* de la base de datos de *AL* con el fin de descartar las muestras que, en anteriores ejecuciones del algoritmo hayan sido propuestas. Sin embargo, esta comprobación no solo se hace para descartar la repetición de muestras, sino también se hace para obtener las muestras que tienen una clase establecida (medioide), ya que se usarán como datos de entrenamiento.

3. Se usa la red conocida como *PANNs* pre-entrenada con la base de datos *Audioset* para obtener la clase de cada muestra que no hayan sido propuesta en iteraciones anteriores. El uso de *PANNs* se centra concretamente en la red neuronal convolucional *Cnn14* explicada en la Sección 2.2. La clase asignada será la clase que la *Cnn14* considere con una mayor probabilidad. Además, aprovechando la predicción de las muestras, el vector de características (mapa de caracteríticas interno de la penúltima capa) de cada muestra será almacenado provisionalmente para su uso futuro.

4. Considerando que el número total de muestras sigue siendo lo suficientemente elevado como para se den complicaciones a nivel computacional para el procesamiento de estas, se decide realizar una división del conjunto de datos en k particiones (*folds*, en inglés). El valor de k dependerá del número de muestras totales y del número de muestras que se considere adecuado para cada partición. Estas particiones se realizaran enfocándose en la variabilidad/incertidumbre ordenado en orden descendente entre *folds*. La organización de las particiones se hace de tal manera que en el primer *fold* se encuentran los datos que contengan una mayor variabilidad, mientras tanto, en el último *fold* se encontraran los datos que presentan una menor variabilidad. La mayor variabilidad e incertidumbre se realiza siguiendo mediante una serie de reglas. Si una clase es asignada a pocas muestras y la cantidad de asignaciones en menor al número de carpetas: todas las muestras serán asignadas a la

primera carpeta. En caso de que una clase tenga un número de asignaciones mayor al número de carpetas, éstas muestras se ordenan en orden ascendente entre carpetas. Así, asignando las muestras con una menor probabilidad de clasificación a las primeras carpetas. Siendo la primera carpeta aquella con más incertidumbre (con muestras asignadas a clases comunes pero con baja probabilidad) y mayor variabilidad (muestras extrañas se asignan a esta carpeta). Así, se consiguen subconjuntos disjuntos ricos para el *Active Learning* en orden descendente, siendo las primeras carpetas las más interesantes dentro de la ventana de análisis.

5. Se lanza el algoritmo de *Active Learning* sobre la primera carpeta. En caso de disponer de un *LB* elevado, se podría lanzar el *Active Learning* sobre tantas carpetas como se desease. En nuestro caso de uso, sólo se lanzaba el modelo de *Active Learning* sobre la primera carpeta. Luego de la ejecución del algoritmo, se habrá propuesto un nuevo conjunto de datos para etiquetar. La forma con la cual estos datos son propuestos, es mediante el método introducido en la Sección 2.1.2 del Capítulo 2 [2]. Los dos clasificadores usados para realizar esta técnica son, por una parte, la propagación de etiquetas según sus vecinos más cercanos y, por otra parte, una Regresión Logística.

6. Las tablas *WavsProposed* y *ALPreprocessing* se actualizan con la información resultante de esta ejecución del *Active Learning*. La primera tabla añadirá los nuevos audios propuestos, mientras que la segunda insertará un nuevo registro con la información relevante sobre la ejecución. La tabla *WavsProposed* insertará nuevos audios, con los campos *mediod*,*agreement* y *max_labelers* vacíos.

7. Con los nuevos audios propuestos, se procede al proceso de etiquetado, por parte de los dos grupos de etiquetadores formados por 3 miembros el Grupo 1 y 2 miembros el Grupo 2.

8. Una vez el proceso de etiquetado se concluye, la información de resultante del proceso de etiquetado se añade a la tabla *Chunks* de la base de datos de *SOROLL-IA*. A continuación, con esta información, los campos *mediod*, *agreement* y *max_labelers* de la tabla *WavsProposed* que se encontraban vacíos, pueden ser actualizados. Al poder realizar un etiquetado multi-etiqueta (el mismo audio puede contener varias clases), se ha diseñado un criterio para definir la clase medioide de la muestra. Si este criterio de concordancia se cumple, se escogería una etiqueta representativa para el campo *mediod*. La necesidad de escoger tan solo una etiqueta que represente el audio viene dada por el uso de medioides en el *Active Learning*. Como se ha explicado en el Capítulo 2, el uso de datos etiquetados de forma manual servirá para que el algoritmo pueda proponer, en las siguientes iteraciones, datos a etiquetar más precisos según los datos ya etiquetados. El algoritmo del estado del arte utilizado implementa una propagación de etiquetas que exige un único valor de etiqueta/-clase por muestra. El criterio de concordancia establecido es de 2/3, es decir, los etiquetadores de un grupo deben de coincidir, en 2/3, con la etiqueta propuesta. Dos tercios de los etiquetadores del grupo deben coincidir con la existencia de esa clase específica en el audio. En el caso de que existan múltiples etiquetas de concordancia, la seleccionada será aquella con una duración mayor, considerando que es la etiqueta más presente en el audio. A modo de ilustración, se puede ver la Figura 3.6. Cuando una etiqueta cumple con el criterio de concordancia establecido, se actualiza las columnas *mediod*, *agreement* y *max_labelers* de la tabla *WavsProposed* para garantizar su uso en futuras ejecuciones del algoritmo de *Active Learning*

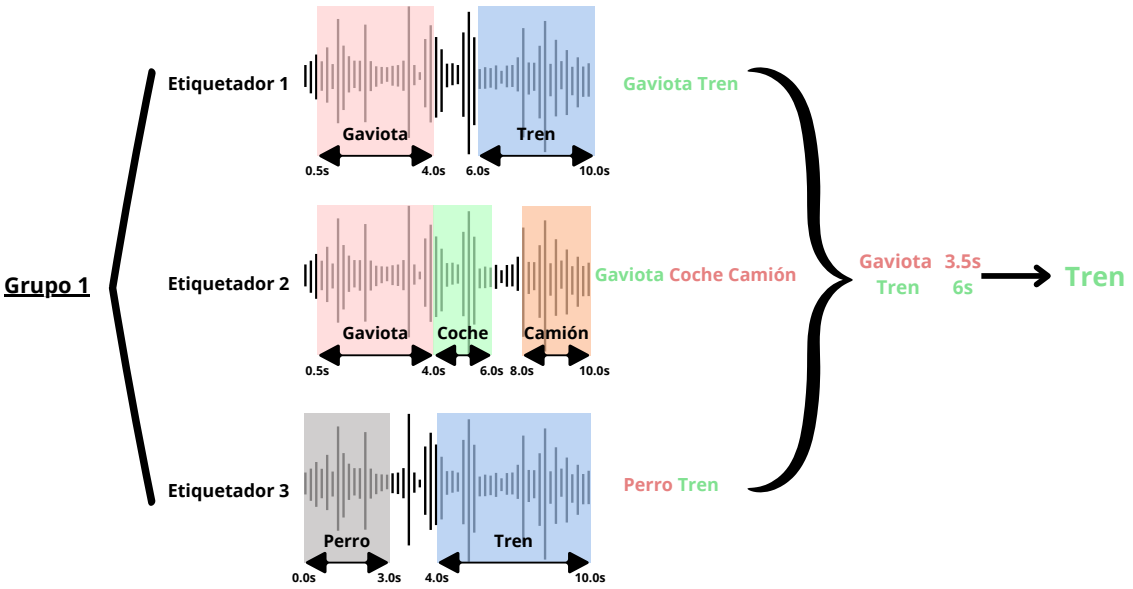

Figura 3.6: Ejemplo de selección de un medioide para un mismo audio dentro del grupo 1.

3.4 Detalles del experimento

En esta sección se va a detallar el marco de trabajo que se ha seguido para llevar al cabo el proyecto: la configuración y herramientas utilizadas. Este análisis permitirá comprender mejor la forma en la cual se ha trabajado durante el proyecto y el porqué del uso de las herramientas presentes.

3.4.1. Entorno de trabajo

Se ha optado por trabajar con el lenguaje de progamación *Python* a través del editor de código fuente *Visual Studio Code* (*VSCode*) en una máquina local para la implementación del trabajo. Para el proceso de entrenamiento del algoritmo de *Active Learning*, se ha utilizado un ordenador con una *GPU*,(Unidad de Procesamiento Gráfico) por sus siglas en inglés, mediante una conexión a una máquina remota (esta máquina ha sido proporcionada como infraestructura de la empresa mediante el servicio de *DATAHUB*[1]. Se puede apreciar información más técnica sobre la *GPU* usada en la Figura 3.7. El uso principal de este procesador viene dado por su gran capacidad para realizar cálculos paralelos de manera eficiente, lo que acelera significativamente el procesamiento de modelos de IA y grandes volúmenes de datos. Al tener que enfrentarse a grandes volúmenes de datos además del uso de *PANNS* para el preprocesado de estos, se consideró oportuno utilizar este tipo de procesador para agilizar el trabajo.

Para la gestión de dependencias y contenerización, se ha optado por el uso de la herramienta *Docker*. El despliegue de contenedores de *Docker* permite la creación de entornos de trabajo aislados y reproducibles para problemas específicos. La configuración de contenedores que corresponde a este proyecto viene dada por 3 contenedores:

- Servicio de PGAdmin: Proporciona una interfaz de usuario capaz de administrar y visualizar la base de datos usada en el proyecto, facilitando de esta forma la gestión de los datos. El acceso a esta interfaz se hace mediante un puerto de la máquina remota.

[1] https://datahub.iti.upv.es/

```
+-----------------------------------------------------------------------------------------+
| NVIDIA-SMI 550.67                 Driver Version: 550.67         CUDA Version: 12.4     |
|-----------------------------------------+------------------------+----------------------+
| GPU  Name                 Persistence-M | Bus-Id          Disp.A | Volatile Uncorr. ECC |
| Fan  Temp   Perf          Pwr:Usage/Cap |           Memory-Usage | GPU-Util  Compute M. |
|                                         |                        |               MIG M. |
|=========================================+========================+======================|
|   0  Tesla V100S-PCIE-32GB          Off |   00000000:00:10.0 Off |                    0 |
| N/A   37C    P0             38W /  250W |    1630MiB /  32768MiB |      0%      Default |
|                                         |                        |                  N/A |
+-----------------------------------------+------------------------+----------------------+

+-----------------------------------------------------------------------------------------+
| Processes:                                                                              |
|  GPU   GI   CI        PID   Type   Process name                              GPU Memory |
|        ID   ID                                                               Usage      |
|=========================================================================================|
|    0   N/A  N/A   2651124      C   python3                                     1626MiB |
+-----------------------------------------------------------------------------------------+
```

Figura 3.7: Especificaciones técnicas de la máquina remota *GPU* proporcionada por *DATAHUB*

- Servicio de base de datos (Postgres): Aloja la base de datos del proyecto. Permitiendo un acceso eficiente y seguro a los datos necesarios para el correcto funcionamiento
- Aplicación (código de Python): contiene la aplicación principal de este trabajo, es decir, el desarrollo del algoritmo de *Active Learning*. El acceso a este contenedor permite tener un entorno de ejecución controlado para el desarrollo y despliegue del algoritmo deseado

La organización modular que nos proporciona el uso de estos contenedores nos brinda flexibilidad y escalabilidad en el desarrollo y despliegue de ejecuciones de algoritmos del mismo estilo. La forma con la cual se ha podido desplegar estos contenedores, ha sido mediante la herramienta *Docker Compose*. Esta herramienta permite definir y gestionar aplicaciones multi-contenedor. Permite crear y conectar varios contenedores independientes, asegurando una comunicación eficiente entre ellos, funcionalidad que ha resultado de utilidad para la creación y el uso de los contenedores anteriormente mencionados.

Una representación visual de la arquitectura del proyecto se observa en la Figura 3.8. Desde la máquina local de trabajo, se establece una conexión a una máquina remota que cuenta con una GPU (sólo se trabaja con terminal en la máquina remota), la cual aloja tres contenedores Docker desplegados. Cada uno de estos contenedores desempeña una función específica, siendo destacable el rol del contenedor App (Aplicación), donde reside el algoritmo de *Active Learning*.

3.4.2. Librerías usadas

En el contexto del proyecto, si nos centramos en el desarrollo e implementación de un algoritmo de Aprendizaje Activo, es imprescindible destacar el papel fundamental del uso del lenguaje de programación Python[2]. La elección de Python para este proyecto no solo se debe a su popularidad y versatilidad, sino también a su gran utilidad en problemas de *Machine Learning*. En los últimos años se ha convertido en uno de los lenguajes de programación más usados en este ámbito, esto se debe a que Python ofrece una amplia variedad de librerías y herramientas diseñadas específicamente para el desarrollo y la

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

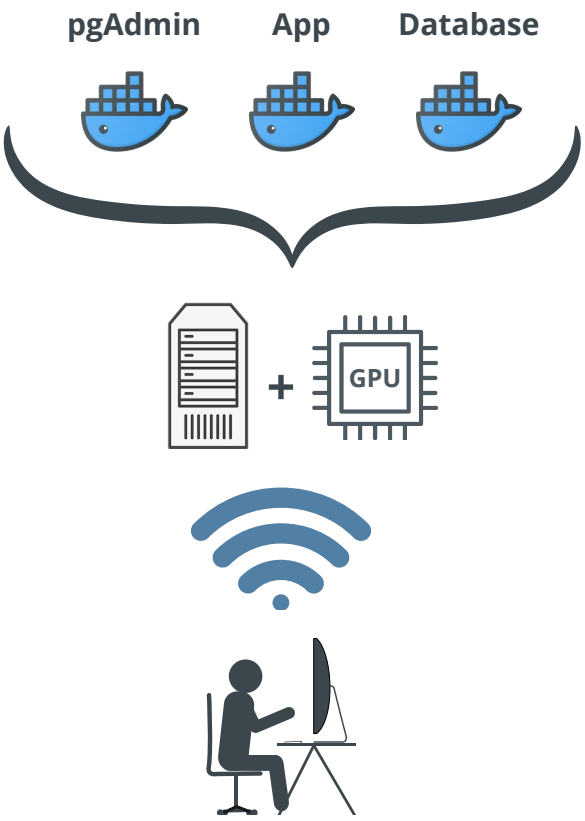

Figura 3.8: Vista de las conexiones realizadas durante el proceso de desarrollo y despliegue del algoritmo de Aprendizaje Activo

implementación de algoritmos de *Machine Learning*, lo que lo convierte en la opción ideal para este tipo de proyectos.

A continuación, se analizarán algunas de las librerías utilizadas durante el desarrollo de este TFG:

- **NumPy**[3] : Manejo de matrices y realización de operaciones matemáticas de manera eficiente.
- **PyTorch**[4] : Implementación de modelos de Deep Learning en una GPU.
- **Librosa**[5] : Manipulación de ficheros de audio.
- **H5py**[6] : Almacenamiento y gestión de datos en archivos .hdf5, eficiente para grandes conjuntos de datos como matrices o vectores.
- **SQLAlchemy**[7] : Creación, acceso y gestión de bases de datos PostgreSQL.

[3] https://numpy.org/
[4] https://pytorch.org/
[5] https://librosa.org/
[6] https://www.h5py.org/
[7] https://www.sqlalchemy.org/

- **Pandas**[8] : Conversión de consultas SQLAlchemy a DataFrames para facilitar el análisis y manipulación de datos.
- **Matplotlib**[9] : Creación de gráficos para visualizar los resultados del estudio.
- **Seaborn**[10] : Creación de gráficos detallados junto con Matplotlib, mejorando la comprensión de los resultados obtenidos.
- **UMAP**[11]: Reducción de dimensiones para visualización y análisis de datos no lineales.

[8] https://pandas.pydata.org/
[9] https://matplotlib.org/
[10] https://seaborn.pydata.org/
[11] https://umap-learn.readthedocs.io/en/latest/

CAPÍTULO 4
Resultados

En este capítulo, se presentarán los resultados obtenidos a lo largo de cinco meses de etiquetado en los que se ha empleado el algortimo de *Active Learning* para la selección de muestras. Cabe destacar que el objetivo de este capítulo no es demostrar el funcionamiento del algoritmo de Aprendizaje Activo, sino mostrar los resultados del flujo de trabajo (número de audios etiquetados, gráficas de los audios seleccionados, etc.). Además, al ser un caso práctico real de creación de una base de datos de audio desde cero, no se dispone de un *ground-truth* para poder obtener métricas del algoritmo de Aprendizaje Activo. Es decir, no se dispone de un conjunto de validación o test.

El proceso de *Active Learning* se ha realizado de forma semanal, es decir, cada semana se lanzaba una iteración de *Active Learning* sobre una ventana de análisis y se sugerian una cantidad específica de datos a etiquetar. El resultado de este proceso es un conjunto determinado de audios que serán etiquetados por los etiquetadores durante la semana. Debido a la duración del proceso, el algoritmo de *Active Learning* se lanzaba cada viernes para que el lunes de la semana siguiente estuvisen propuestos los datos a etiquetar durante la semana. Los audios recomendados por el algoritmo pasaran por un proceso de etiquetado (proceso introducido en la Sección 3.2 del Capítulo 3) formado por dos grupos de etiquetadores. Un grupo formado por tres etiquetadores y otro por dos.

En la Tabla 4.1 se puede observar la información correspondiente a cada iteración del *Active Learning* (se entiende como iteración a cada vez que el algoritmo ha sido ejecutado). En esta tabla, se muestra que cada iteración está asociada a una ventana temporal y un nodo determinado de datos a analizar, además de mostrar los audios propuestos para etiquetar en la semana siguiente. Como se puede apreciar, el tamaño de las ventanas temporales procesadas varía, desde ventanas de 15 días, como en la iteración 4, hasta ventanas de 2 días, como en la iteración 6. Este cambio en el tamaño de las ventanas se debe a la falta de audios en distintos meses, especialmente en los meses de verano en los que aparecieron ciertos impedimentos para poder grabar el mes completo.

Para el proceso de etiquetado, se consideró conveniente crear la clase *Doubt* (Duda). Los etiquetadores podían usarla y definir eventos que no consideraran estar seguros de etiquetar con alguna clase disponible (el conjunto de etiquetas iniciales fue el presente en *Audioset*). Teóricamente, cada 10 iteraciones del algoritmo, se realiza una semana de resolución de dudas. Esta semana, consistía en que cada etiquetador volvía a escuchar y etiquetar aquellos audios que anteriormente habían marcado como Duda para, ver si, con una mayor experiencia en el proceso de etiquetado, eran capaces de resolverlos. En la Tabla 4.1 se puede comprobar que, entre las iteraciones 10 y 11, se realizó una iteración de Dudas para intentar resolver los audios conflictivos de las primeras 10 iteraciones.

Adicionalmente, de forma semanal, cada viernes se concretaba una reunión entre los etiquetadores (ambos grupos) con el fin de resolver dudas presentadas durante el proceso

de etiquetado de los anteriores días y sugerir nuevas etiquetas. El proceso de etiquetado durante estos meses ha resultado en 7340 datos etiquetados que corresponden a 20 horas de audios etiquetados en total. El esfuerzo de etiquetado ha supuesto unas 91 horas de etiquetado por etiquetador (suponiendo que cada audio cuesta de etiquetar 45 segundos).

Los resultados que se expondrán en las siguientes secciones abarcan varios aspectos presentados durante el proyecto. En primer lugar, se proporcionará un análisis detallado de la información resultante del proceso de etiquetado de las etiquetas propuestas por el algoritmo del *Active Learning* (ver sección 4.1). Además, se presentará un análisis visualmente informativo del rendimiento del algoritmo a través de las múltiples iteraciones realizadas (ver sección 4.2). Este análisis visual será fundamental para comprender cómo el algoritmo ha evolucionado y cómo se ha adaptado a medida que se ha ejecutado repetidamente, proporcionando una perspectiva completa sobre su comportamiento y efectividad a lo largo de estos meses de etiquetado.

Iteración	Nodo	Año	Mes	Dia Inicial	Dia Final	Audios propuestos
1	1	2023	Julio	21	31	60
2	1	2023	Agosto	01	07	320
3	1	2023	Septiembre	01	15	320
4	1	2023	Octubre	15	30	400
5	1	2023	Noviembre	01	07	400
6	2	2023	Junio	02	04	400
7	2	2023	Julio	21	31	400
8	2	2023	Agosto	01	15	400
9	2	2023	Septiembre	15	30	400
10	2	2023	Octubre	01	15	400
Dudas	1,2	-	-	-	-	-
11	2	2023	Noviembre	15	30	400
12	2	2023	Diciembre	01	15	320
13	2	2024	Enero	01	15	320
14	2	2023	Agosto	15	30	400
15	2	2023	Septiembre	01	15	400
16	2	2023	Octubre	15	24	400
17	2	2023	Octubre	25	31	400
18	2	2023	Noviembre	01	15	400
19	2	2023	Diciembre	15	31	400
20	2	2023	Enero	15	31	400
21	2	2023	Septiembre	15	30	400

Tabla 4.1: Tabla informativa sobre las iteraciones del *Active Learning*. Indicando en la columna Iteración, con un valor identificativo de cada vez que el algoritmo se ha ejecutado, junto con información sobre el Nodo, Año, Mes, Dia Inicial y Dia Final de la ventana de datos seleccionada para su análisis. En la columna Audios Propuestos se encuentran el número de audios que el algoritmo ha propuesto para esa ejecución.

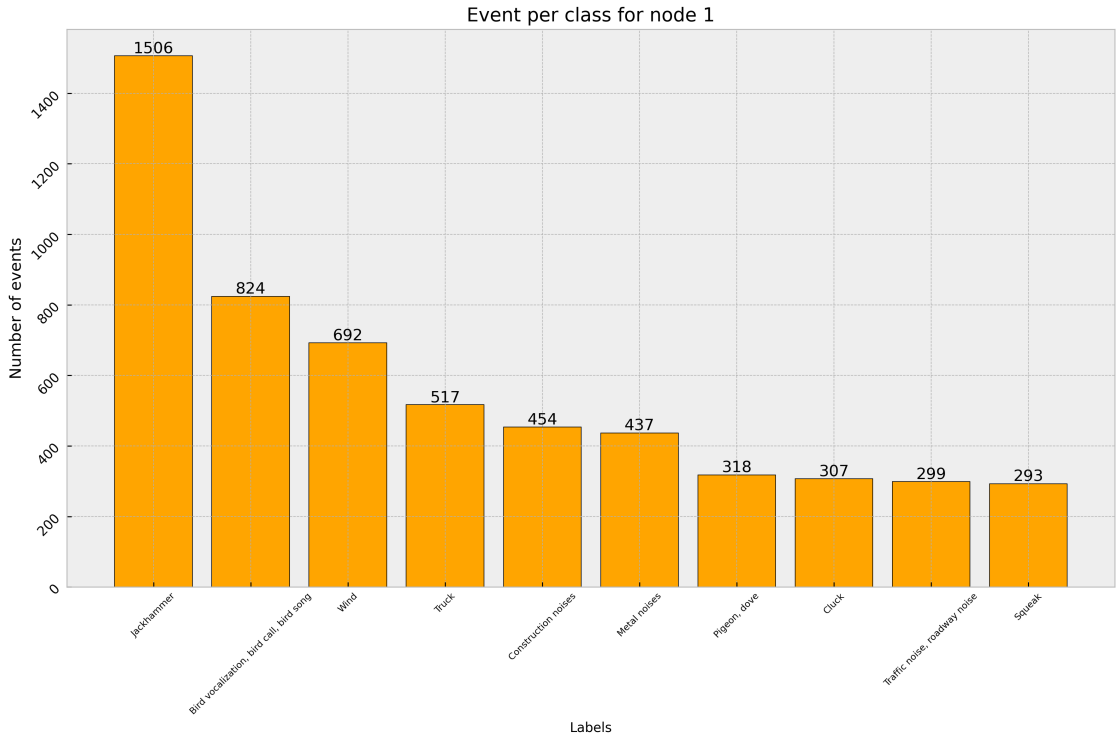

Figura 4.1: Diagrama de barras del número de etiquetas clasificadas por los etiquetadores para el nodo 1.

4.1 Resultados del proceso de etiquetado

En esta sección se van a presentar los resultados obtenidos durante el proceso de etiquetado, abarcando el período desde el 8 de enero de 2024 hasta el 31 de mayo de 2024. Cabe destacar que, aunque el proceso de etiquetado continúa en progreso, se ha decidido establecer estas fechas para poder ofrecer un análisis concreto de los resultados hasta el momento para este TFG.

Los resultados presentados se van a centrar en las etiquetas propuestas por los etiquetadores durante el proceso de etiquetado del período de tiempo establecido. El equipo de etiquetado estuvo compuesto por cinco personas, divididas en dos grupos: uno de tres miembros y otro de dos.

A lo largo del período mencionado, los etiquetadores propusieron un total de 180 clases distintas para clasificar los audios propuestos por el algoritmo de Aprendizaje Activo. Este amplio conjunto de etiquetas refleja la diversidad y complejidad de los datos acústicos analizados.

En la Figura 4.1, se presentan las clases más comunes identificadas en los audios por los etiquetadores durante el análisis del nodo 1 (*Loc.1* en la Figura 3.1). Estos resultados abarcan un período de 5 semanas de etiquetado, con un total de 1500 audios presentados al proceso de etiquetado. La figura muestra la frecuencia con la que cada clase ha estado presente al menos una vez por algún etiquetador en las muestras analizadas.

Se puede apreciar la frecuente presencia de clases como el martillo neumático (*Jackhammer*), camión (*Truck*), sonidos de construcción (*Construction Noises*) y sonidos metálicos (*Metal Noises*). Estas clases están asociadas a los ruidos típicos de una obra. Esta alta presencia de sonidos relacionados con la construcción se debe a las obras que se han estado realizando cerca del nodo durante el tiempo de recolección de los datos a estudiar.

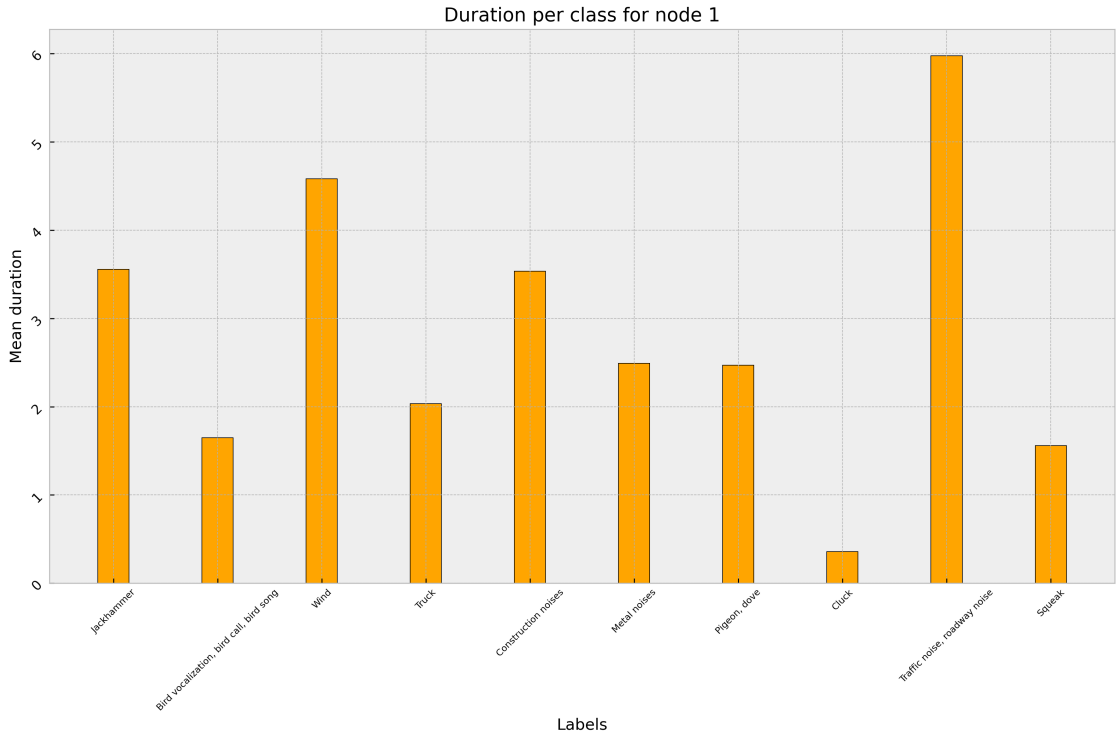

Figura 4.2: Diagrama de barras de la duración media de las etiquetas más frecuentes para el nodo 1.

Además, la proximidad del nodo a una rotonda (véase la Figura 3.1 de la Sección 3.1 para recordar la ubicación de los nodos) explica la abundancia de etiquetas como *Truck* y *Traffic noise, roadway noise* (sonido de tráfico).

Las etiquetas *Cluck* y *Squeak* se han asociado a sonidos cortos de *clicks* genéricos y chirridos, producidos por vehículos, puertas, golpes, entre otros. Por último, etiquetas como *Bird vocalization, bird call, bird song* (Canto de los pájaros), *Wind* (Viento) y *Pigeon, dove* (Paloma) están influenciadas por la naturaleza de la ubicación del nodo, su cercanía con el río y la brisa del mar, lo que justifica la presencia de estos sonidos naturales en las grabaciones.

Estos resultados ofrecen una visión detallada de los tipos de sonidos predominantes en el nodo 1, sin tener en consideración si estas etiquetas han sido validadas por el criterio de concordancia presentado en la Sección 3.3 del Capítulo 3.

Para complementar la información presentada en la Figura 4.1, la Figura 4.2 muestra la duración media de las clases más comunes. Se observa que las etiquetas relacionadas con los sonidos de obra, en general, tienen una presencia significativa a lo largo de los audios. Sin embargo, las etiquetas correspondientes al sonido del viento y el ruido del tráfico destacan por tener las duraciones más largas en las grabaciones. Esto indica que, aunque los sonidos de construcción son frecuentes, los ruidos ambientales como el viento y el tráfico son los que predominan en términos de duración en los audios analizados.

Para el nodo 2, se realizó un proceso de etiquetado durante 15 semanas de etiquetado, resultando en un total de 5840 audios etiquetados. En la Figura 4.3, se observa que, al tratarse de una nueva ubicación, las etiquetas presentes son diferentes en comparación con las mostradas en la Figura 4.1. No obstante, algunas etiquetas comunes en el nodo 1 también se encuentran en el nodo 2.

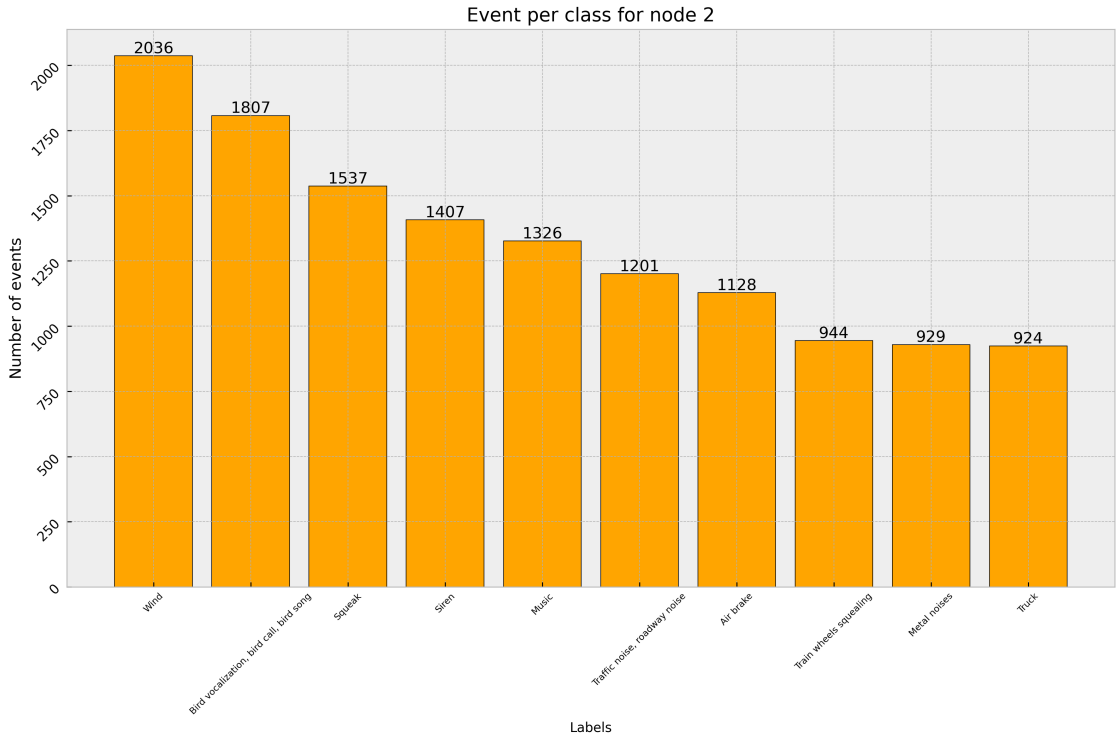

Figura 4.3: Diagrama de barras del número de etiquetas clasificadas por los etiquetadores para el nodo 2.

Las etiquetas relacionadas con el tráfico (*Traffic noise, roadway noise* y *Truck*) son bastante comunes debido a la proximidad del nodo a una carretera. La etiqueta de sonidos metálicos (*Metal noises*) está asociada a la etiqueta de camión (*Truck*), ya que cerca del nodo hay un badén que produce un sonido metálico por sus vagones cuando los camiones pasan. Por esta razón, ambas etiquetas tienen una presencia similar.

Además, la presencia de autobuses y camiones explica la frecuencia de la etiqueta *Air brake*, que se refiere al freno de aire utilizado por estos vehículos. Dado que las vías del tren están cerca, la etiqueta *Train wheels squealing*, que describe el chirrido de las ruedas del tren, es bastante común debido al paso frecuente de trenes por esa zona. La etiqueta *Squeak* también está asociada a los chirridos del tren y sus vagones.

La etiqueta *Siren* se refiere a las sirenas que suenan durante el funcionamiento de grúas y maquinaria en el puerto. Además, se incluye una etiqueta que podría no considerarse parte de un entorno portuario: *Music* (Música). La presencia de esta etiqueta se debe a que frente a la ubicación del nodo hay varios locales de ocio (bares y discotecas) que ponen música a distintas horas del día.

Concluyendo el análisis de la figura, las etiquetas más abundantes son las relacionadas con el viento y el sonido de los pájaros. Al igual que en el primer nodo, se considera que estos sonidos son normales debido a la naturaleza de la ubicación del nodo, cerca del mar.

Para complementar la información presentada en la Figura 4.3, se encuentra la Figura 4.4 que muestra las duraciones medias de las clases indicadas en la primera figura. En esta gráfica, se observa una distinción más marcada en la duración de las clases *Wind*, *Siren*, *Music*, y *Traffic noise, roadway noise*, en comparación con las demás clases. Estas cuatro clases tienen una presencia media en los audios de aproximadamente 5 segundos, lo que equivale a la mitad de la duración total de los audios.

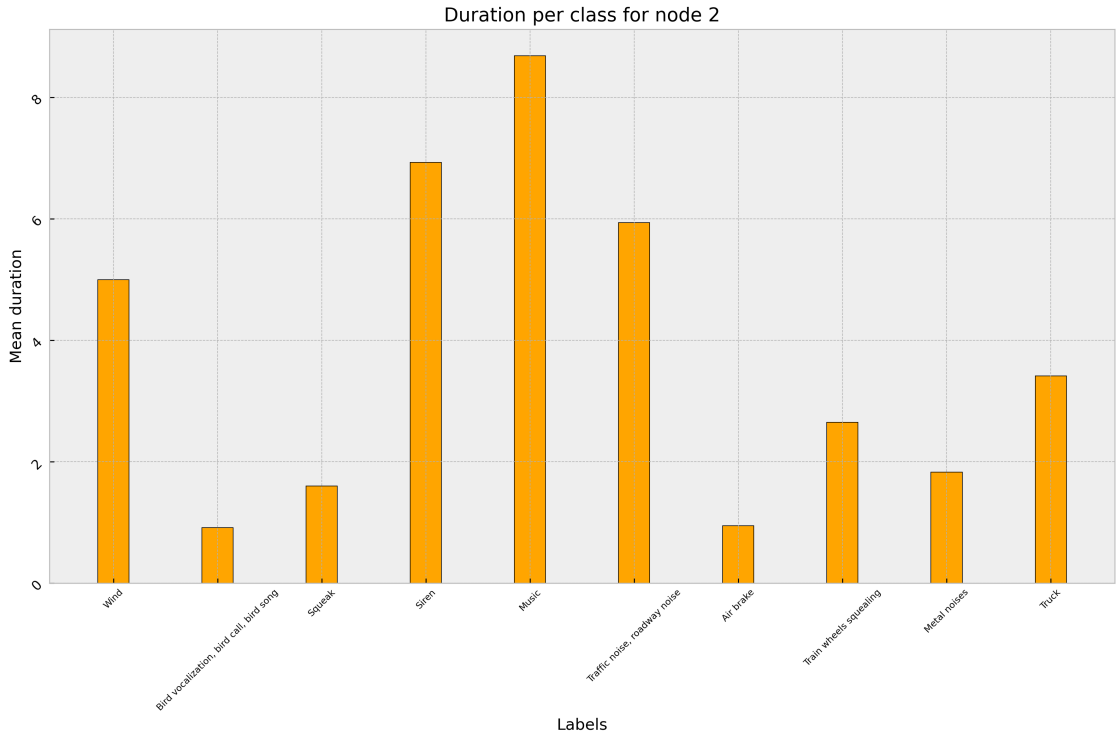

Figura 4.4: Diagrama de barras de la duración media de las etiquetas más frecuentes para el nodo 2.

Por otro lado, las demás clases presentan duraciones medias inferiores a los 5 segundos. Entre las clases con menor duración se encuentran los sonidos de pájaros (*Bird vocalization, bird call, bird song*), el freno de aire de los vehículos (*Air brake*) y los chirridos (*Squeak*). Estas observaciones sugieren que, aunque las clases de sonido de viento, sirena, música y tráfico son predominantes en términos de duración, otras clases de sonidos, aunque bastante presentes, tienden a ser más breves.

Para concluir esta sección, se presentarán figuras que muestran información sobre las etiquetas de los audios que, según el criterio de concordancia establecido en la Sección 3.3 del Capítulo 3, han sido las más comunes a lo largo de todo el proceso de etiquetado. Además, estas etiquetas (las disponibles en el momento) son las que se usaban como datos de entrenamiento para la iteración del algoritmo de *Active Learning* semanal.

En primer lugar, para el nodo 1, se registraron un total de 1173 audios que se consideran que tienen, al menos, una etiqueta válida según el criterio de concordancia establecido.

Como se puede ver en la Figura 4.5 y tal como se explicó anteriormente respecto al nodo 1 en la Figura 4.1, la presencia de obras cerca del micrófono implicaba la abundancia de etiquetas sobre sonidos genéricos de construcción. En este caso, se observan sonidos como martillos mecánicos (*Jackhammer*), pitidos de marcha atrás de maquinaria de obra (*Reversing beeps*), frenos de aire (*Air brake*), chirridos (*Squeak*) y sonidos de construcción (*Construction noises*). También se encuentran presentes sonidos relacionados con el tráfico, destacando el sonido de una motocicleta (*Motorcycle*) como el de mayor concordancia. Los sonidos de la naturaleza también aparecen, como el viento, el canto de los pájaros y, a diferencia de las figuras anteriores, el sonido del agua (*Water*) debido a la proximidad del río al nodo.

En general, no se observan diferencias significativas de concordancia entre los grupos, a pesar de que el Grupo 1 está constituido por 3 personas y el Grupo 2 por solo 2. Sin

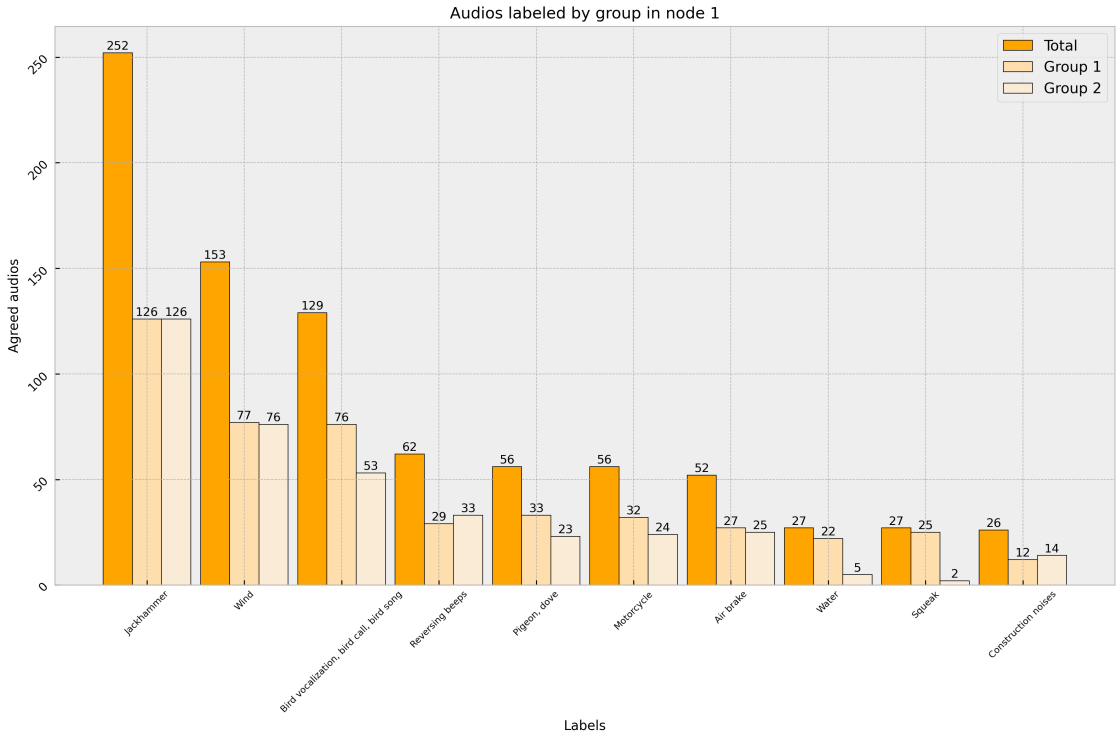

Figura 4.5: Diagrama de barras de las etiquetas válidas según el criterio de concordancia, dividido por grupos para el nodo 1.

embargo, en las clases *Water* y *Squeak* se puede notar una diferencia considerable entre grupos. En el Grupo 1, para que un audio se considere válido, al menos dos de los tres integrantes deben estar de acuerdo con la etiqueta propuesta. En cambio, en el Grupo 2, todos los integrantes deben coincidir para que una etiqueta sea considerada válida. Por esta razón, se considera que los integrantes del Grupo 1, deberían de ser los que más facilidades tienen para aportar etiquetas válidas al proyecto.

Por otro lado, para el nodo 2, se han registrado un total de 3066 audios que se consideran que tienen, al menos, una etiqueta válida según el criterio de concordancia establecido.

Las clases presentes en la Figura 4.6 son prácticamente las mismas que las mencionadas en la Figura 4.3 del mismo nodo. La única diferencia significativa es la incorporación de la clase *Railroad car, train wagon*, que se refiere a los vagones de tren. La presencia de esta clase en la Figura 4.6 y no en figuras anteriores podría deberse a la intención de ambos grupos de etiquetadores de centrarse en etiquetar con mayor precisión y cuidado todas aquellas etiquetas relacionadas con el tren, entre otras. Esta tendencia sugiere que habrá más concordancia entre los etiquetadores en este tipo de etiquetas.

A grandes rasgos, se observa que en este nodo el Grupo 1 tiende a aportar más etiquetas válidas que el Grupo 2. Esto podría atribuirse al mismo motivo mencionado en los resultados del nodo 1 en la Figura 4.5 .

Estos resultados resaltan la importancia de la concordancia entre los etiquetadores y cómo la estructura del grupo puede influir en los resultados del etiquetado.

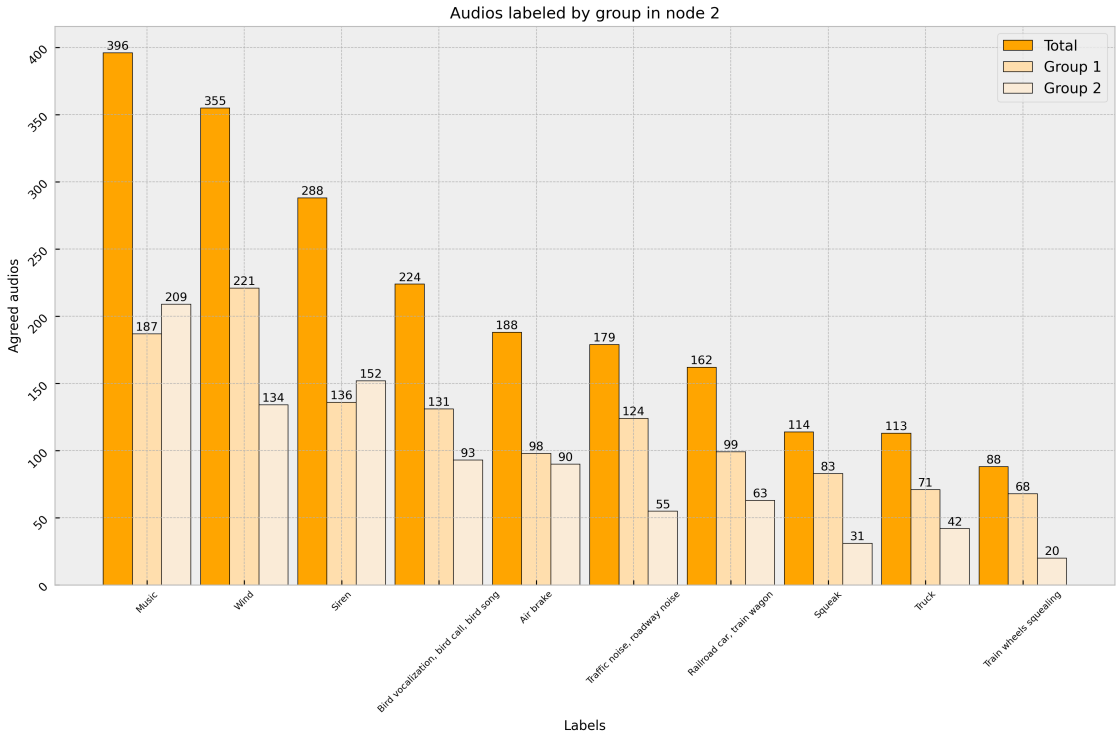

Figura 4.6: Diagrama de barras de las etiquetas válidas según el criterio de concordancia, dividido por grupos para el nodo 1.

4.2 Resultados del proceso de Aprendizaje Activo

En esta última sección, se presentan distintos gráficos que muestran los datos obtenidos en varias iteraciones del proceso de Aprendizaje Activo. Estos gráficos permiten observar los resultados obtenidos a lo largo de distintos meses de trabajo mediante el uso del algoritmo para la selección de audios a etiquetar.

Para la creación de los gráficos, se ha optado por utilizar una técnica de reducción de dimensionalidad llamada UMAP (*Uniform Manifold Approximation and Projection*) [34]. UMAP es una técnica útil para visualizar datos de altas dimensiones en espacios de menor dimensión. En este caso, se utilizará para proyectar los datos en dos dimensiones, lo que facilita la identificación de patrones y agrupamientos en el conjunto de datos presentados para cada ventana temporal.

A continuación, se mostrarán algunos gráficos generados provenientes de distintas iteraciones, explicando los patrones y agrupamientos observados, así como las implicaciones de estos resultados en el proceso de selección de datos para el etiquetado. Este análisis visual será fundamental para comprender la efectividad del Aprendizaje Activo y su impacto en la mejora del algoritmo a lo largo de este periodo de tiempo de trabajo.

Primeramente, se va a mostrar el UMAP de la primera semana de etiquetado del trabajo, es decir, la correspondiente a la iteración número 1 de la tabla 4.1, que abarca el rango de días del 21 al 31 de julio de 2023 del nodo 1. Este gráfico puede verse en la Figura 4.7

En el gráfico, se pueden observar dos clases diferenciadas: en forma de estrella y de color verde la clase *Proposed* y en forma circular y color lila la clase *Not Proposed*. Estas clases hacen referencia a los audios correspondientes de la ventana de datos establecida (presentes en la primera *fold*) y que han sido usados como entrada para el Aprendiza-

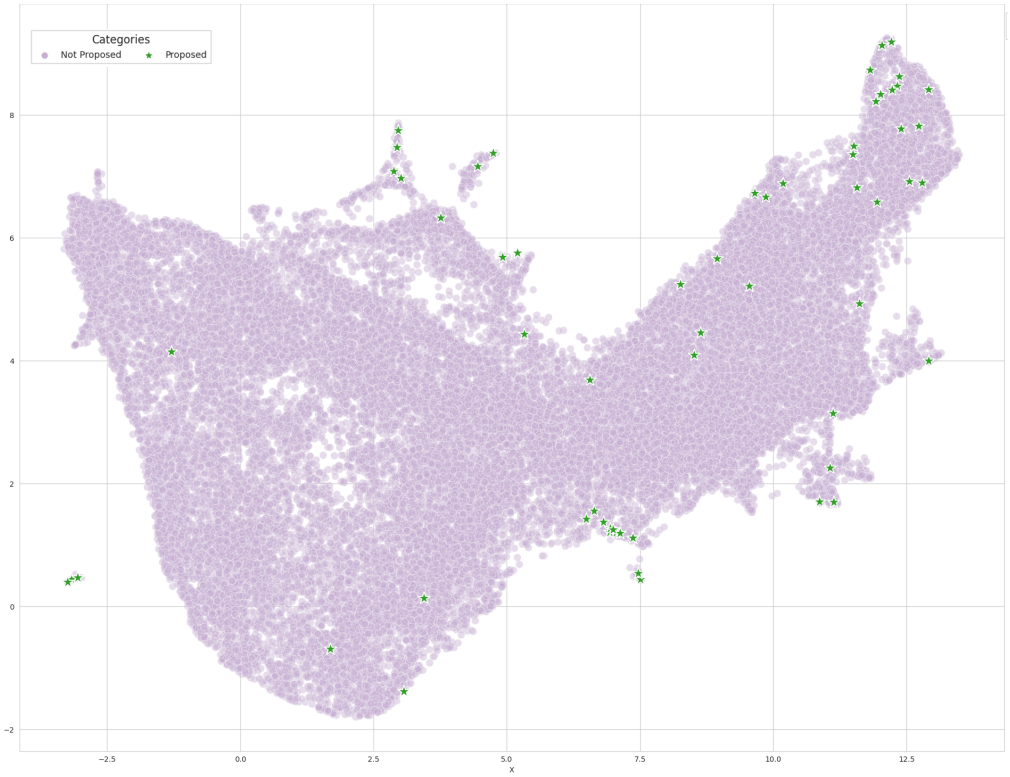

Figura 4.7: UMAP de la iteración 1

je Activo. La clase *Proposed* forma parte de los datos que el algoritmo ha considerado relevantes y ha propuesto para su etiquetado.

Como se ha explicado en capítulos anteriores, al ser esta la primera iteración, no se cuenta aún con datos etiquetados para que el Aprendizaje Activo pueda utilizar como datos de entrenamiento. Con estos datos se puede aplicar la técnica ya introducida llamada *Mismatch-First*, sin embargo, al ser la primera iteración el algoritmo ha seleccionado los datos propuestos mediante el método de *Farthest-Traversal*. Esta técnica se basa en seleccionar los puntos de datos más alejados entre sí en el espacio de características.

En esta primera visualización, el gráfico proporciona una visión clara de la ejecución del método *Farthest-Traversal* viendo como los datos seleccionados para etiquetar, se encuentran en los extremos del conjunto total de datos, es decir, alejados en el espacio de características.

Como se ha comentado en anterioridad, en la primera iteración del algoritmo aún no existen aún datos etiquetados para ayudar al Aprendizaje Activo a proponer etiquetas más relevantes. Por esta razón, el siguiente gráfico que se va a presentar, muestra información sobre la iteración 9, correspondiente a la ventana temporal del 15 de septiembre de 2023 al 30 de septiembre de 2023 del nodo 2. En esta iteración, los dos grupos de etiquetadores han etiquetado un total de 2700 audios. De estos, 2151 audios se han utilizado como medioides. La selección de estos 2151 audios se debe a la concordancia entre etiquetadores, tal como se explicó en la sección 3.3.

A continuación, en la Figura 4.8, además de las dos clases presentadas en el gráfico anterior, podemos observar una tercera clase denominada *Medoids*, representada de forma circular y de color naranja, que hace referencia a los datos etiquetados en anteriores iteraciones que son usados para entrenar el algoritmo. Se puede apreciar que la presencia de esta clase es similar a los datos propuestos en la iteración actual.

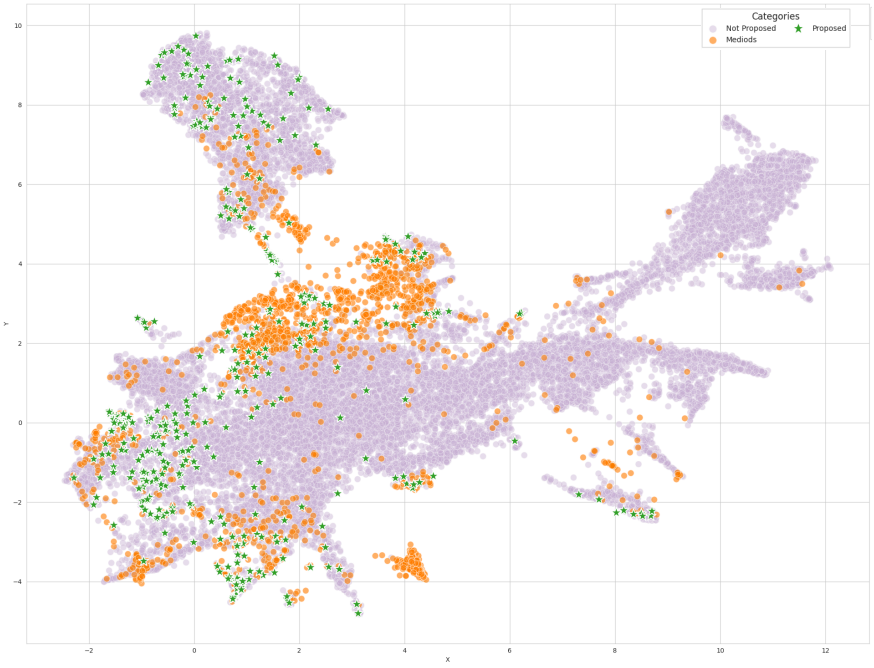

Figura 4.8: UMAP de la iteración 9

Se puede llegar a detectar zonas donde el algoritmo considere que los datos etiquetados son suficientes como para establecer una frontera clara, haciendo que ya no considere relevante proponer nuevos audios para etiquetar por la región. Un ejemplo claro de esto es la zona lateral derecha superior del gráfico. En esta área, la cantidad de datos etiquetados ha sido suficiente para que el algoritmo pueda establecer fronteras precisas entre las clases.

Sin embargo, también existen áreas en las que, a pesar de haber muchos datos ya etiquetados, el algoritmo sigue proponiendo nuevos datos para etiquetar. Un ejemplo de esto es la zona central del gráfico. Esta persistencia del algoritmo en proponer nuevos datos podría deberse a que se trata de una zona conflictiva donde conviven muchas clases distintas. La complejidad de esta región hace difícil para el algoritmo establecer una frontera clara entre las diferentes clases, por lo que requiere más datos para poder mejorar la separación y clasificación en esa área.

La identificación de estas zonas conflictivas y la necesidad de más datos en dichas áreas suma importancia al proceso iterativo en el Aprendizaje Activo. Al proporcionar continuamente nuevos datos, el algoritmo puede mejorar su capacidad para diferenciar entre clases y aumentar la precisión de sus predicciones futuras.

Para verificar las afirmaciones planteadas en el párrafo anterior, se ha decidido obtener un gráfico de clases según *PANNs*, concretamente mediante el uso del modelo *Cnn14*. El objetivo de este gráfico es encontrar una explicación a las zonas donde el algoritmo de Aprendizaje Activo propone audios para su etiquetado.

Este gráfico se muestra en la Figura 4.9, donde se puede apreciar cómo en la zona lateral derecha, mencionada anteriormente como una región donde el algoritmo había considerado,en esta iteración, más fácil la distinción entre clases, se ve esta “simple” separación a simple vista. Esta separación respalda la idea de que el algoritmo ha establecido con éxito una frontera precisa entre las clases en esa región, reduciendo así la necesidad de proponer nuevos audios para etiquetar.

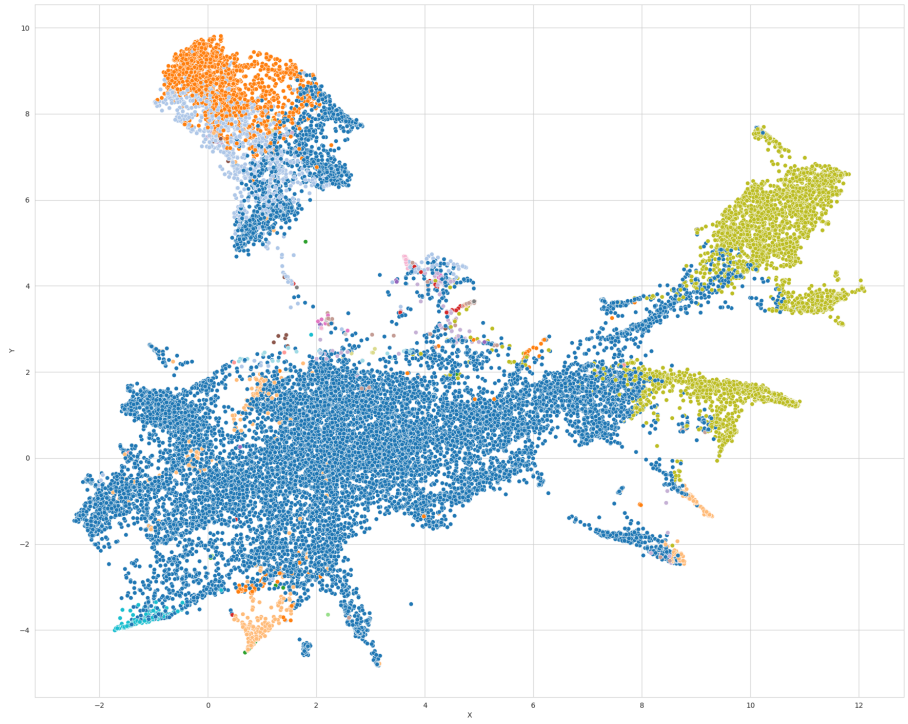

Figura 4.9: Gráfico de etiquetas según *PANNs* para el conjunto de datos de la iteración 9 donde cada dato de un color significa una clase distinta. Por ejemplo, los puntos verdes pertenecen a la clase Silencio, mientras que el azul predominante es la clase Vehículo y el naranja superior hace referencia a la clase Música.

En cambio, en la zona central, donde se consideraba que habría una frontera entre clases menos clara, se puede observar que conviven muchas clases distintas en un pequeño espacio. En consecuencia, el algoritmo sigue proponiendo datos de audio en esa área, ya que, incluso con los datos etiquetados hasta el momento, no es suficiente para distinguir bien las clases en esa zona. En la Figura 4.10, se puede apreciar mejor la multitud de clases presentes en la zona mencionada anteriormente, comparándola con los datos propuestos y los medioides del gráfico UMAP.

Por último, se mostrarán el gráfico UMAP y el de clases según *PANNs* en la Figura 4.11, correspondientes a una de las iteraciones más actualizada hasta la fecha, la número 20. Esta iteración abarca la ventana temporal del 15/01/2024 al 31/01/2024 del nodo 2.

Como se ha comentado en los gráficos anteriores, el algoritmo sigue proponiendo muestras en zonas que generan incertidumbre, es decir, en áreas donde se encuentran muchas clases distintas y es complicado establecer una división clara entre ellas.

La diferencia en esta iteración es que, al tener más datos etiquetados que se usan como datos de entrenamiento para el algoritmo, hay zonas que a simple vista se podrían considerar conflictivas, como la zona derecha redondeada de negro donde se observan muchas clases. Sin embargo, según lo que se puede ver, el algoritmo ya cuenta con suficientes muestras en esta área y, actualmente, no propone muchas muestras adicionales para etiquetar en esa región.

No obstante, hay otras zonas en las que, a pesar de tener muchos datos etiquetados de iteraciones anteriores, el algoritmo sigue insistiendo en proponer más datos para etiquetar. Un ejemplo de esto es la zona inferior redondeada de rojo, donde, aunque ya existen numerosos datos etiquetados, el algoritmo sigue identificando la necesidad de más datos para mejorar la separación y clasificación en esa área.

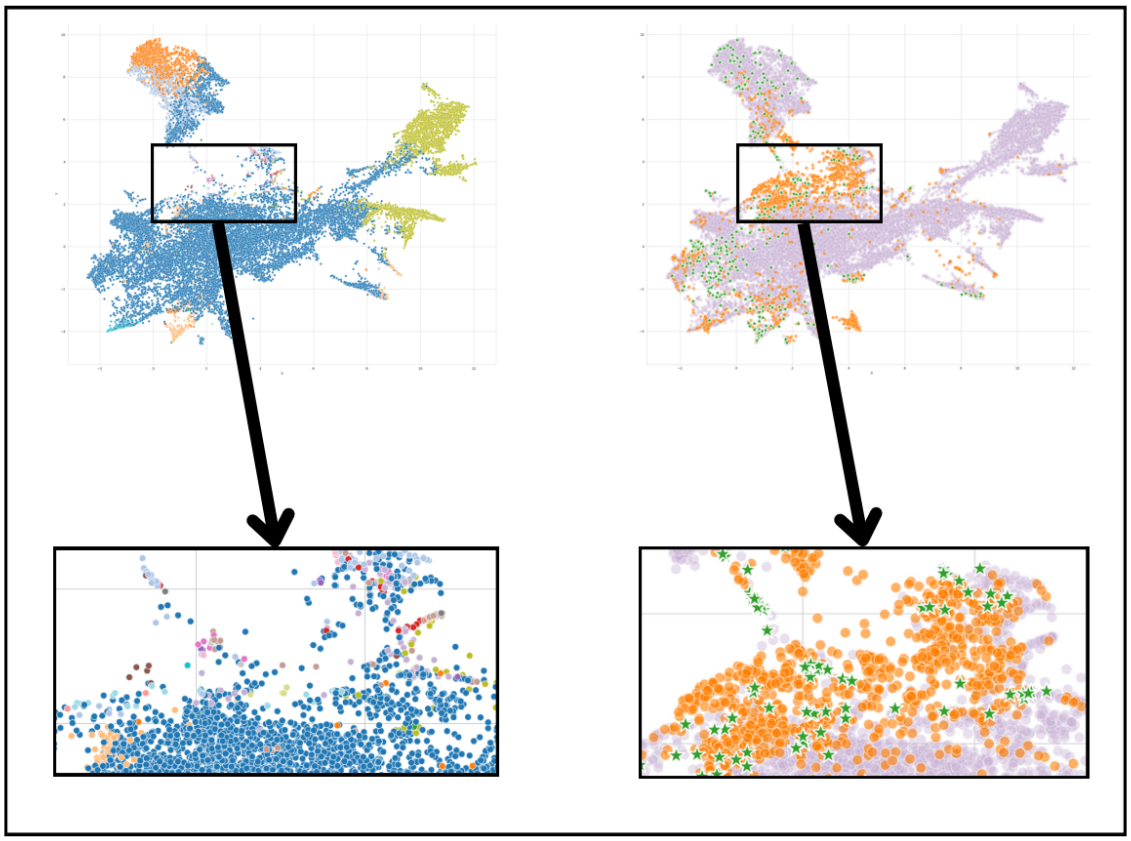

Figura 4.10: Comparación de la zona centro de los gráficos

Esta persistencia en ciertas áreas destaca la complejidad del conjunto de datos y la necesidad continua de proporcionar al algoritmo nuevas muestras etiquetadas. De esta forma, se creará una frontera cada vez más clara en según que zonas para poder dar paso al algoritmo, a proponer audios dentro de otras zonas menos conflictivas.

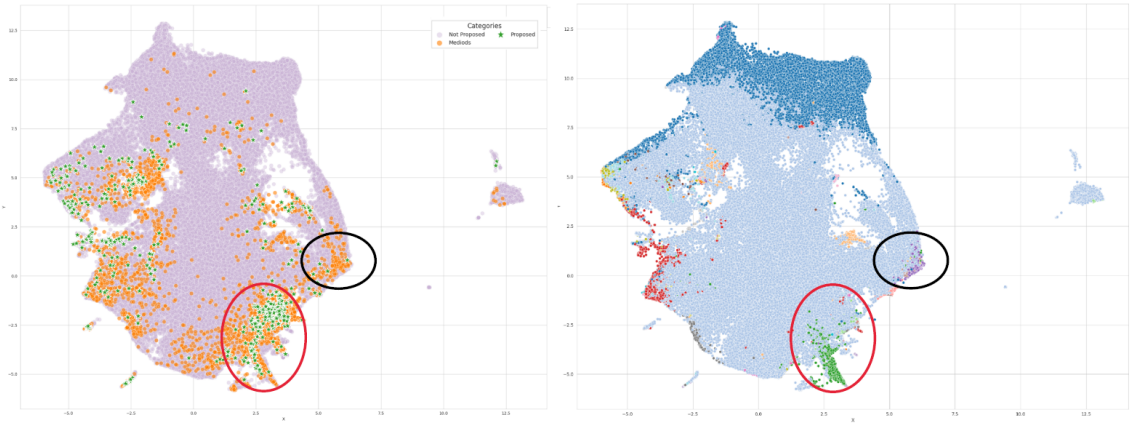

Figura 4.11: Comparación UMAP y gráfico de clases para la iteración número 20.

CAPÍTULO 5
Conclusiones

Uno de los objetivos principales de este trabajo era implementar un algoritmo de Aprendizaje Activo presentado en el estado del arte. Este objetivo se formuló con la idea de poder seleccionar muestras de forma inteligente, dentro de un conjunto de datos no etiquetado, aquellos datos con mayor relevancia. Evitando de esta forma, el proceso de etiquetar el conjunto de datos de forma aleatoria, tratando de optimizar al mázimo los recursos disponibless.

Además, se planteó como objetivo adicional adaptar este algoritmo a un flujo de trabajo real con un conjunto de datos dinámico y de grandes dimensiones, como es el caso de la construcción de una base de datos de sonidos ambientales a partir de datos de audio recolectados en un entorno portuario.

Para garantizar que el algoritmo de *Active Learning* funcione para grandes volúmenes de datos, se realizó una preselección de los datos a etiquetar dentro de una ventana temporal y espacial. Asimismo, se llevó a cabo una selección inteligente de los datos basándose en la variabilidad e incertidumbre de los mismos. De esta forma, independientemente del tamaño del conjunto de datos, el flujo de trabajo sería capaz de procesarlo y proponer un lote de muestras para etiquetar.

Los datos propuestos por el algoritmo eran etiquetados por un equipo de etiquetadores, lo que permitía al algoritmo aprender de estas nuevas muestras etiquetadas. Gracias a la implementación del algoritmo y al proceso de etiquetado de los datos propuestos, se han conseguido un total de 20 horas de datos de audio etiquetados durante un periodo de trabajo de 5 meses.

Con la información obtenida a partir de estos datos etiquetados ,se ha logrado analizar dónde se ubican dentro del espacio de características. El análisis de estos datos etiquetados dentro del conjunto total demuestra que su selección sirve para identificar y definir las fronteras entre clases, facilitando así el establecimiento de fronteras claras entre clases. El etiquetado de estos datos propuestos servirá como conjunto de entrenamiento para futuros algoritmos o modelos, demostrando que, sin necesidad de etiquetar todo el conjunto de datos en su totalidad, es posible obtener suficiente información mediante el etiquetado de pocas muestras aplicando una solución como la mostrada mediante el algoritmo de Aprendizaje Activo. Esto permite establecer diferencias significativas entre clases y optimizar el uso de recursos en proyectos de etiquetado de grandes volúmenes de datos.

5.1 Trabajo futuro

A lo largo de este proyecto, se ha mostrado un marco de trabajo para la implementación de un algoritmo de Aprendizaje Activo. Sin embargo, existen diversas ideas y direcciones futuras que podrían mejorar este flujo de trabajo, así como áreas adicionales de investigación que han surgido a raíz de esta implementación.

Actualmente, el criterio de concordancia para seleccionar los medioides se basa en la validación de etiquetas cuando más de 2/3 de los etiquetadores coinciden en que una etiqueta está presente, además de ser la etiqueta más larga durante el audio (véase la Sección 3.3 y la Figura 3.6). Se sugiere una mejora y comparación en este criterio utilizando el llamado *Intersection Over Union* (*IOU*). Este método mediría el grado de similitud entre las etiquetas propuestas por los distintos etiquetadores de un grupo. La idea es considerar cuál es la etiqueta que más se repite entre los etiquetadores, proporcionando así una métrica más precisa y robusta para la selección de medioides. Además, se compararía el uso de este método con el actual para poder ver cuál de los dos métodos proporciona al proyecto unos mejores resultados.

El flujo de trabajo continuará en el futuro, obteniendo más datos con etiquetas válidas por parte de ambos grupos. A medida que se obtenga un número significativo de datos, se podrá considerar la posibilidad de utilizarlos como conjunto de entrenamiento para un clasificador. Este conjunto de datos ampliado permitirá entrenar una red neuronal basada en, la ya conocida, *PANNs* , como podría ser una *Cnn14*, que se ha utilizado en diferentes etapas del trabajo.

El desarrollo de una red neuronal adaptada específicamente al trabajo presentado permitirá mejorar la precisión y eficiencia del algoritmo de *Active Learning*. Esta nueva red podría reemplazar la red entrenada por *Audioset* en la preselección de datos que sirven como entrada al algoritmo de *Active Learning*. Asimismo, se podría sustituir el uso de la Regresión Logística como clasificador de comparación en el uso del *Mismatch-First Farthest-Traversal*.

5.2 Relación del trabajo con los estudios cursados

En este capítulo se detallan los conocimientos y habilidades adquiridos durante el grado cursado que se han sido de utilidad a la hora de llevar a cabo este TFG. A continuación se van a destacar algunas asignaturas que se considera que han tomado un papel importante en el desarrollo de este proyecto.

- **Fundamentos de la programación, Programación y Estructuras de Datos** : Estas asignaturas proporcionan habilidades necesarias para escribir y entender código en Python. Esta base ha sido esencial para programar el algoritmo de Aprendizaje Activo y poder trabajar con grandes conjuntos de datos de una manera eficiente.

- **Infraestructura para el procesamiento de datos** : Asignatura donde se adquirieron conocimientos sobre el uso de contenedores Docker y el trabajo con GPUs. Estas herramientas han sido importantes para crear un entorno de trabajo aislado y eficiente dentro del desarrollo del proyecto.

- **Bases de datos y Gestión de datos** : Asignaturas que proporcionan un conocimiento del funcionamiento de las bases de datos y su almacenamiento ayudando a garantizar una gestión eficiente y segura de los datos.

- **Análisis matemático, Álgebra Lineal y Algorítmica** : Asignaturas que han aportado una base matemática sólida para el entendimiento de problemas de algoritmos de *Machine Learning* y *Deep Learning*
- **Modelos descriptivos y predictivos I y II y Evaluación, despliegue y monitorización de modelos** : Estas asignaturas ofrecieron los fundamentos teóricos y prácticos sobre los modelos de *Machine Learning* y *Deep Learning*.
- **Proyecto I, II y III** : Asignaturas que fomentaron habilidades como el trabajo en equipo y la gestión de proyectos, en este caso, pudiendose ver aplicado durante el desarrollo del trabajo en Instituto Tecnológico de Informática dentro de un proyecto.

En conjunto, las asignaturas cursadas han proporcionado una base sólida de conocimientos y habilidades que han sido directamente aplicables al desarrollo del TFG. Esto no solo ha permitido abordar el problema planteado de manera efectiva, sino también demostrar la relevancia de la formación académica en la resolución de problemas reales en el ámbito de la IA.

Bibliografía

[1] Z. Shuyang, T. Heittola, and T. Virtanen, "Active learning for sound event classification by clustering unlabeled data," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 751–755.

[2] ——, "An active learning method using clustering and committee-based sample selection for sound event classification," in *2018 16th International Workshop on Acoustic Signal Enhancement (IWAENC)*. IEEE, 2018, pp. 116–120.

[3] A. Mesaros, T. Heittola, and T. Virtanen, "Metrics for polyphonic sound event detection," *Applied Sciences*, vol. 6, p. 162, 05 2016.

[4] Z. Shuyang, T. Heittola, and T. Virtanen, "Active learning for sound event detection," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 2895–2905, 2020.

[5] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[6] I. McLoughlin, H. Zhang, Z. Xie, Y. Song, and W. Xiao, "Robust sound event classification using deep neural networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 3, pp. 540–552, 2015.

[7] D. De Benito-Gorrón, D. Ramos, and D. T. Toledano, "A multi-resolution crnn-based approach for semi-supervised sound event detection in dcase 2020 challenge," *IEEE Access*, vol. 9, pp. 89 029–89 042, 2021.

[8] H. Xie, O. Räsänen, K. Drossos, and T. Virtanen, "Unsupervised audio-caption aligning learns correspondences between individual sound events and textual phrases," in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 8867–8871.

[9] A. Jalali, A. Schindler, and B. Haslhofer, "Dcase challenge 2020: Unsupervised anomalous sound detection of machinery with deep autoencoders," *Detection and Classification of Acoustic Scenes and Events*, 2020.

[10] Q. Kong, Y. Cao, T. Iqbal, Y. Wang, W. Wang, and M. D. Plumbley, "Panns: Large-scale pretrained audio neural networks for audio pattern recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 2880–2894, 2020.

[11] S. Basu, A. Banerjee, and R. J. Mooney, "Active semi-supervision for pairwise constrained clustering," in *Proceedings of the 2004 SIAM international conference on data mining*. SIAM, 2004, pp. 333–344.

[12] D. S. Hochbaum and D. B. Shmoys, "A best possible heuristic for the k-center problem," *Mathematics of operations research*, vol. 10, no. 2, pp. 180–184, 1985.

[13] B. Settles, "Active learning literature survey," 2009.

[14] S. Shishkin, D. Hollosi, S. Doclo, and S. Goetze, "Active learning for sound event classification using monte-carlo dropout and pann embeddings," in *Proceedings of the 6th Workshop on Detection and Classication of Acoustic Scenes and Events (DCASE 2021)*. DCASE, 2021, pp. 150–154.

[15] M. Meire, J. Zegers, and P. Karsmakers, "Active learning in sound-based bearing fault detection," in *Proceedings of the 8th Detection and Classification of Acoustic Scenes and Events 2023 Workshop (DCASE2023)*, Tampere, Finland, September 2023, pp. 111–115.

[16] S. Shishkin, D. Hollosi, S. Goetze, and S. Doclo, "Active learning for sound event classification using bayesian neural networks with gaussian variational posterior," in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2024, pp. 896–900.

[17] Y. Wang, A. E. Mendez Mendez, M. Cartwright, and J. P. Bello, "Active learning for efficient audio annotation and classification with a large amount of unlabeled data," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 880–884.

[18] R. Jiménez-Moreno, J. O. Pinzón-Arenas, and C. G. Pachón-Suescún, "Assistant robot through deep learning," *Int. J. Electr. Comput. Eng*, vol. 10, pp. 1053–1062, 2020.

[19] S. Damiano and T. van Waterschoot, "Pyroadacoustics: a Road Acoustics Simulator Based on Variable Length Delay Lines," in *Proceedings of the 25th International Conference on Digital Audio Effects (DAFx20in22)*, Vienna, Austria, September 2022, pp. 216–223.

[20] S. Damiano, L. Bondi, S. Ghaffarzadegan, A. Guntoro, and T. van Waterschoot, "Can synthetic data boost the training of deep acoustic vehicle counting networks?" in *Proceedings of the 2024 International Conference on Acoustics, Speech and Signal Processing (ICASSP) (accepted)*, Seoul, South Korea, April 2024.

[21] K. Wilkinghoff, "Self-supervised learning for anomalous sound detection," in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2024, pp. 276–280.

[22] M. Malik, M. K. Malik, K. Mehmood, and I. Makhdoom, "Automatic speech recognition: a survey," *Multimedia Tools and Applications*, vol. 80, pp. 9411–9457, 2021.

[23] X. Dong, B. Yin, Y. Cong, Z. Du, and X. Huang, "Environment sound event classification with a two-stream convolutional neural network," *IEEE Access*, vol. 8, pp. 125 714–125 721, 2020.

[24] J. Naranjo-Alcazar, S. Perez-Castanos, P. Zuccarello, and M. Cobos, "Acoustic scene classification with squeeze-excitation residual networks," *IEEE Access*, vol. 8, pp. 112 287–112 296, 2020.

[25] L. Gao, Q. Mao, and M. Dong, "On local temporal embedding for semi-supervised sound event detection," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2024.

[26] M. Lesnichaia, V. Mikhailava, N. Bogach, Y. Lezhenin, J. Blake, and E. Pyshkin, "Classification of accented english using cnn model trained on amplitude mel-spectrograms." in *INTERSPEECH*, 2022, pp. 3669–3673.

[27] M. Ravanelli and Y. Bengio, "Speaker recognition from raw waveform with sincnet," in *2018 IEEE Spoken Language Technology Workshop (SLT)*, 2018, pp. 1021–1028.

[28] S. Hershey, S. Chaudhuri, D. P. W. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold, M. Slaney, R. J. Weiss, and K. Wilson, "Cnn architectures for large-scale audio classification," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 131–135.

[29] A. Singh, H. Liu, and M. D. Plumbley, "E-panns: Sound recognition using efficient pre-trained audio neural networks," in *INTER-NOISE and NOISE-CON Congress and Conference Proceedings*, vol. 268, no. 1. Institute of Noise Control Engineering, 2023, pp. 7220–7228.

[30] L. Xu, L. Wang, S. Bi, H. Liu, and J. Wang, "Semi-supervised sound event detection with pre-trained model," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

[31] X. Liu, X. Mei, Q. Huang, J. Sun, J. Zhao, H. Liu, M. D. Plumbley, V. Kilic, and W. Wang, "Leveraging pre-trained bert for audio captioning," in *2022 30th European Signal Processing Conference (EUSIPCO)*. IEEE, 2022, pp. 1145–1149.

[32] S. Park and G. Kim, "Pretrained network-based sound event recognition for audio surveillance applications," in *2021 International Conference on Information and Communication Technology Convergence (ICTC)*. IEEE, 2021, pp. 1306–1309.

[33] J. F. Gemmeke, D. P. W. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio set: An ontology and human-labeled dataset for audio events," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 776–780.

[34] L. McInnes, J. Healy, and J. Melville, "Umap: Uniform manifold approximation and projection for dimension reduction," *arXiv preprint arXiv:1802.03426*, 2018.

APÉNDICE A

Relación del proyecto con los Objetivos de Desarrollo Sostenible

El 25 de septiembre de 2015, los líderes mundiales adoptaron un conjunto de objetivos globales para erradicar la pobreza, proteger el planeta y asegurar la prosperidad para todos como parte de una nueva agenda de desarrollo sostenible. Cada objetivo tiene metas específicas que deben alcanzarse en los próximos 15 años.

En la siguiente tabla se muestra el nivel de impacto que tiene el trabajo realizado con los Objetivos de Desarrollo Sostenible (ODS).

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

Tabla A.1: Relación del trabajo con los Objetivos de Desarrollo Sostenible (ODS)

El proyecto presentado se centra en la implementación de un algoritmo de Aprendizaje Activo para mejorar la eficiencia en la etiquetación de datos ambientales, se considera relevante destacar su relación con varios Objetivos de Desarrollo Sostenible (ODS) establecidos por las Naciones Unidas. A continuación, se describe cuales han sido los ODS que se consideran que tienen relación con el TFG:

- **ODS 8 - Trabajo decente y crecimiento económico** : Este Objetivo de Desarrollo Sostenible se centra en promover el crecimiento económico sostenido, inclusivo y

sostenible, el empleo pleno y productivo, y el trabajo decente para todos. El desarrollo de este proyecto esta relacionado con el objetivo debido a que se ha llevado a cabo con colaboración con una empresa, el Instituto Tecnológico de Informática, formando parte de un proyecto más amplio. La mejora en la eficiencia de la recolección y análisis de datos se puede considerar que incrementa la productividad, además del desarrollo de habilidades especializadas entre los empleados, promoviendo un entorno de trabajo más innovador y dinámico.

- **ODS 9 - Industria, innovación e infraestructura** : Este Objetivo de Desarrollo Sostenible se centra en la construcción de infraestructuras resilientes, la promoción de la industrialización inclusiva y sostenible, y el fomento de la innovación. El proyecto presentado implementa un algoritmo de Aprendizaje Activo para mejorar la eficiencia en la etiquetación de datos en un contexto de señales de audio ambientales. El trabajo se alinea con este objetivo al introducir técnicas de Inteligencia Artificial para optimizar los recursos de etiquetado, reduciendo costos y tiempo. Esta innovación tecnológica no solo mejora la precisión y eficiencia de los modelos de IA, sino que también contribuye al desarrollo de infraestructuras de datos más eficientes, necesarias para la toma de decisiones informadas en la industria. Además, la aplicación práctica de este algoritmo en el puerto de Valencia demuestra cómo las tecnologías innovadoras pueden ser utilizadas para gestionar y mejorar operaciones industriales, apoyando así una industrialización más sostenible y fortaleciendo la infraestructura tecnológica.

- **ODS 11 - Ciudades y comunidades sostenibles** : El siguiente ODS busca lograr que las ciudades y los asentamientos humanos sean inclusivos, seguros, resilientes y sostenibles. El proyecto presentado, que implementa un algoritmo de Aprendizaje Activo (AL) para optimizar la etiquetación de señales de audio ambientales, se relaciona estrechamente con este objetivo al facilitar la monitorización continua del entorno urbano en el puerto de Valencia. Esta tecnología avanzada permite identificar y gestionar fuentes de ruido y contaminación acústica, mejorando así la calidad de vida de los habitantes. Al optimizar la recolección y etiquetado de datos ambientales, el proyecto contribuye a una gestión más eficiente de los recursos urbanos, proporcionando a las autoridades locales información precisa para diseñar políticas efectivas de gestión ambiental. De esta manera, se promueve el desarrollo de ciudades más sostenibles y resilientes, al tiempo que se fomenta una mejor planificación urbana y la reducción de los impactos ambientales negativos

En conclusión, el proyecto presentado demuestra cómo la implementación de tecnologías avanzadas de IA y Aprendizaje Activo puede tener un impacto significativo en varios aspectos del desarrollo sostenible. Al optimizar la etiquetación de datos ambientales y proporcionar herramientas para una mejor gestión y planificación urbana, el proyecto no solo mejora la eficiencia y precisión en el manejo de datos, sino que también contribuye a la sostenibilidad industrial y la creación de ciudades más resilientes, además de haber sido un proyecto dessarrollado en colaboración de la empresa ITI. De esta manera, se evidencia una clara alineación con los ODS 8,9 y 11, dando importancia a la relevancia y el potencial de las innovaciones tecnológicas en la promoción de un desarrollo sostenible y equitativo.