



**Diseño de un sistema de posicionamiento subacuático basado en señales acústicas y
machine learning**

Soria Arízaga, Christian Javier

Departamento de Eléctrica, Electrónica y Telecomunicaciones

Carrera de Ingeniería en Electrónica y Telecomunicaciones

Trabajo de titulación, previo a la obtención del título de Ingeniero en Electrónica y
Telecomunicaciones

Ing. Carrera Erazo, Enrique Vinicio, PhD

28 de julio del 2022



Tesis_3.pdf

Scanned on: 12:7 July 26, 2022 UTC



Overall Similarity Score



Results Found



Total Words in Text

Identical Words	64
Words with Minor Changes	0
Paraphrased Words	179
Omitted Words	2126





Departamento de Eléctrica, Electrónica y Telecomunicaciones

Carrera de Ingeniería en Electrónica y Telecomunicaciones

Certificación

Certifico que el trabajo de titulación: **“Diseño de un sistema de posicionamiento subacuático basado en señales acústicas y machine learning”** fue realizado por el señor **Soria Arízaga, Christian Javier** el mismo que cumple con los requisitos legales, teóricos, científicos, técnicos y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, además fue revisado y analizado en su totalidad por la herramienta de prevención y/o verificación de similitud de contenidos; razón por la cual me permito acreditar y autorizar para que se lo sustente públicamente.

Sangolquí, 28 de julio de 2022

Firma:



Ing. Carrera Erazo, Enrique Vinicio PhD.

C. C. : 1708792104



Departamento de Eléctrica, Electrónica y Telecomunicaciones

Carrera de Ingeniería en Electrónica y Telecomunicaciones

Responsabilidad de autoría

Yo, **Soria Arízaga, Christian Javier**, con cédula/cédulas de ciudadanía n° 1725263998, declaro que el contenido, ideas y criterios del trabajo de titulación: **Diseño de un sistema de posicionamiento subacuático basado en señales acústicas y machine learning** es de mi autoría y responsabilidad, cumpliendo con los requisitos legales, teóricos, científicos, técnicos, y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Sangolquí, 28 de julio de 2022

Firma

Soria Arízaga, Christian Javier

C.C.: 1725263998



Departamento de Eléctrica, Electrónica y Telecomunicaciones

Carrera de Ingeniería en Electrónica y Telecomunicaciones

Autorización de Publicación

Yo **Soria Arízaga, Christian Javier**, con cédula de ciudadanía n° 1725263998, autorizo a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de titulación: **Diseño de un sistema de posicionamiento subacuático basado en señales acústicas y machine learning** en el Repositorio Institucional, cuyo contenido, ideas y criterios son de mi responsabilidad.

Sangolquí, 28 de julio de 2022

Firma

Soria Arízaga, Christian Javier

C.C.: 1725263998

Dedicatoria

Dedico el presente trabajo de investigación a mis padres Elena y Manuel, quienes durante toda mi vida han sido un apoyo incondicional y gracias a su amor, dedicación, integridad han sido una inspiración en mi vida y un modelo a seguir. A mis hermanas Lorena y Miriam que siempre han estado dispuestas a brindar su ayuda. A mi hermano Christopher y mis sobrinos Mateo, Alejandro, Sarahí y Panchito por los cuales me he esforzado en demostrar que pueden cumplir sus metas si se lo proponen y que el principal talento que se necesita es la perseverancia.

Este logro es para ustedes.

Christian Javier Soria Arízaga

Agradecimiento

Agradezco a mis padres Elena y Manuel quienes con mucho esfuerzo, sacrificio y amor han dedicado su vida a sus hijos, han permitido que pudiera cumplir uno de mis sueños y con su ejemplo me han enseñado la importancia de los valores, del trabajo y a no rendirme. Admiro su fortaleza y dedicación.

A mis hermanas Lorena y Miriam que siempre han estado a mi lado y me han enseñado que lo más importante y valioso que tenemos es la familia.

A todos mis amigos que en algún momento me apoyaron y que gracias a su ayuda desinteresada han formado parte de mi vida compartiendo momentos buenos y malos.

Al compañero Diego Guevara por la ayuda en la grabación de los audios utilizados en el presente trabajo.

Finalmente quiero agradecer a mi tutor el Ing. Vinicio Carrera por su apoyo, consejos, experiencia y atención permanente en el desarrollo de este trabajo.

Christian Javier Soria Arízaga

Índice de contenido

Dedicatoria.....	6
Agradecimiento.....	7
Resumen	15
Abstract.....	16
Introducción	17
Antecedentes	17
Justificación.....	18
Alcance del proyecto	18
Objetivos	19
General.....	19
Específicos	19
Marco teórico	20
Posicionamiento acuático.....	20
Navegación estimada e inercial	20
Navegación geofísica.....	21
Navegación acústica.....	21
Long base line	22
Short base line	23
Ultra short baseline	23
Preprocesamiento de señales	24

Procesamiento en frames	24
Transformada rápida de Fourier.....	25
Muestreo.....	25
Análisis de componentes principales PCA	25
Machine learning	26
Aprendizaje no supervisado	26
Agrupación jerárquica.....	27
Agrupación particional	27
Aprendizaje supervisado.....	27
Técnicas de Aprendizaje Supervisado.....	28
Regresión lineal	28
Arboles de decisión.....	28
Máquina de vectores de soporte	28
Procesos gaussianos.....	29
Bagging y Boosting	29
Redes neuronales.....	30
Evaluación de modelos	30
Error en regresión lineal.....	30
Error cuadrático medio MSE (Mean Square Error)	31
R cuadrado.....	31
Error absoluto medio MAE (Mean Absolute Error)	32

	10
Métodos de validación	32
Validación Cruzada K-fold	32
Holdout.....	33
Metodología	34
Creación de base de datos.....	34
Escenario.....	34
Vehículo acuático.....	34
Hidrófonos	35
Grabación de audio.....	37
Preprocesamiento	37
Recorte	37
Filtrado.....	40
Análisis de la señal	43
Etiquetado	44
Entrenamiento.....	46
Resultados.....	47
Análisis de resultados.....	47
Entrenamiento en el dominio del tiempo	47
Entrenamiento inicial (dominio del tiempo)	47
Entrenamientos posteriores (dominio del tiempo).....	52
Entrenamiento en el dominio de la frecuencia.....	53

Entrenamiento inicial (dominio de la frecuencia).....	54
Entrenamientos posteriores (dominio de la frecuencia)	59
Comparación	63
Conclusiones	66
Trabajos futuros	67
Referencias.....	68

Índice de Tablas

Tabla 1 Posición de los hidrófonos (formación triangular)	35
Tabla 2 Posición de los hidrófonos (formación cuadrada)	36
Tabla 3 Configuración final para cada arreglo	62
Tabla 4 Error cuadrático medio bidimensional	65

Índice de Figuras

Figura 1 Piscina empleada como escenario de pruebas	34
Figura 2 Vehículo acuático PowerVision PowerRay	35
Figura 3 Arreglo de hidrófonos triangular	36
Figura 4 Arreglo de hidrófonos cuadrado	36
Figura 5 Señal original obtenida de un hidrófono	38
Figura 6 Potencia promedio (dB) de la señal original	38
Figura 7 Potencia promedio (señal recortada)	39
Figura 8 Señal original recortada	40
Figura 9 Espectrograma de señales grabadas	40
Figura 10 Ruido presente en las grabaciones	41
Figura 11 Espectrograma de señal filtrada y remuestreada	42
Figura 12 Señal original preprocesada (recorte, filtrado, remuestreo)	42
Figura 13 Diagrama del proceso de entrenamiento	43
Figura 14 Representación del escenario	44
Figura 15 Trayectorias trazadas en AutoCAD	44
Figura 16 Trayectorias en Matlab	45
Figura 17 RMSE de técnicas empleadas (Cuadrado eje X) a 50 ms	48
Figura 18 Modelo Cuadrado (eje X) entrenado con Bagged trees a 50 ms	49
Figura 19 RMSE de técnicas empleadas (Cuadrado eje Y) a 50 ms	49
Figura 20 Modelo Cuadrado (eje Y) entrenado con Bagged trees a 50 ms	50
Figura 21 RMSE de técnicas empleadas (Triangular eje X) a 50 ms	50
Figura 22 Modelo Triangular (eje X) entrenado con Bagged trees a 50 ms	51
Figura 23 RMSE de técnicas empleadas (Triangular eje Y) a 50 ms	51
Figura 24 Modelo Triangular (eje Y) entrenado con Bagged tres a 50 ms	52

Figura 25 RMSE por variación de frame para Bagged trees	52
Figura 26 Modelo Cuadrado (eje Y) entrenado con Bagged trees 20 ms	53
Figura 27 Comparativa entre tamaños de frame	54
Figura 28 RMSE de técnicas en frecuencia (Cuadrado eje X) a 50 ms	55
Figura 29 Modelo Cuadrado (eje X) entrenado en frecuencia con P.G. Exponencial a 50 ms...	55
Figura 30 RMSE de técnicas en frecuencia (Cuadrado eje Y) a 50 ms	56
Figura 31 Modelo Cuadrado (eje Y) entrenado con P.G. Racional Quadratico a 50 ms	56
Figura 32 RMSE de técnicas en frecuencia (Triangular eje X) a 50 ms.....	57
Figura 33 Modelo Triangular (eje X) entrenado con P.G. Exponencial a 50 ms.....	57
Figura 34 RMSE de técnicas en frecuencia (Triangular eje Y) a 50 ms.....	58
Figura 35 Modelo Triangular (eje Y) entrenado con P.G. Cuadrático Racional a 50 ms	59
Figura 36 RMSE variando el tamaño de frame (Arreglo cuadrado)	60
Figura 37 RMSE variando el tamaño de frame (Arreglo triangular)	60
Figura 38 RMSE con variación de PCA (arreglo cuadrado)	61
Figura 39 RMSE con variación de PCA (arreglo triangular)	62
Figura 40 Diagrama de dispersión Arreglo Cuadrado eje X (izq) y eje Y (der)	63
Figura 41 Diagrama de dispersión Arreglo Triangular eje X (izq) y eje Y (der)	63
Figura 42 Trayectorias predichas por los modelos propuestos.....	64

Resumen

La navegación subacuática ha sido un tema que ha tomado relevancia en los últimos años debido al interés en conocer mejor este entorno y sobre todo en cómo aprovecharlo. Aplicaciones como las expediciones científicas, la explotación de recursos naturales, las comunicaciones submarinas, el control y vigilancia territorial son algunos ejemplos de la importancia de contar con vehículos subacuáticos que permitan realizar estas tareas en función de su ubicación. Sin embargo, este entorno presenta varias limitaciones como la visibilidad, turbulencia y la imposibilidad de emplear ondas electromagnéticas. Por otro lado, las tecnologías basadas en inteligencia artificial (i.e., *machine/deep learning*) están teniendo un gran impacto en todas las áreas de ciencia e ingeniería, permitiendo encontrar soluciones eficientes a un sinnúmero de aplicaciones que no pueden ser resueltas de forma tradicional. De esta forma, en el presente trabajo de investigación se propone desarrollar un sistema de posicionamiento de un dron en un entorno subacuático controlado, mediante la utilización de técnicas de *machine learning* utilizando el software Matlab; para cumplir con este propósito se plantea utilizar una base de datos que contiene señales grabadas de diferentes trayectorias realizadas por el vehículo subacuático y utilizarlas para entrenar el sistema de posicionamiento, el cual se procederá a entrenar con diversas técnicas de *machine learning* para observar qué técnicas son las más indicadas para este tipo de aplicaciones, adicionalmente se plantea variar parámetros como el tamaño de *frame*, número de componentes, dominio del tiempo y frecuencia para observar cómo estas variaciones afectan al desempeño del modelo entrenado y así determinar qué técnicas y parámetros tienen un mayor potencial para aplicaciones de posicionamiento subacuático.

Palabras clave: Aprendizaje de máquina, posicionamiento subacuático, ondas acústicas, sonar pasivo, drones subacuáticos.

Abstract

Underwater navigation has been a topic that has taken relevance in recent years due to the interest in knowing better this environment and especially how to take advantage of it.

Applications such as scientific expeditions, exploitation of natural resources, underwater communications, territorial control and surveillance are some examples of the importance of having underwater vehicles to perform these tasks depending on their location. However, this environment presents several limitations such as visibility, turbulence and the impossibility of using electromagnetic waves. On the other hand, technologies based on artificial intelligence (i.e., machine/deep learning) are having a great impact in all areas of science and engineering, allowing to find efficient solutions to countless applications that cannot be solved in a traditional way.

Thus, in this research we propose to develop a positioning system for a drone in a controlled underwater environment, through the use of machine learning techniques using Matlab software; To fulfill this purpose, it is proposed to use a database containing recorded signals from different trajectories performed by the underwater vehicle and use them to train the positioning system, which will be trained with various machine learning techniques to observe which techniques are the most suitable for this type of applications, additionally it is proposed to vary parameters such as frame size, number of components, time domain and frequency to observe how these variations affect the performance of the trained model and thus determine which techniques and parameters have greater potential for underwater positioning applications.

Key words: Machine learning, underwater positioning, acoustic waves, passive sonar, underwater drones.

Introducción

Antecedentes

Existen estudios que han permitido realizar modelos de propagación acústica para estimar la presencia de objetos con su ubicación. Estas investigaciones llevan muchas décadas, pero no es hasta la segunda guerra mundial que empieza a llamar la atención debido a la necesidad de detectar submarinos enemigos. Posteriormente y gracias a la aparición de tecnologías como VLSI y la fabricación de DSPs, se produce un verdadero auge en el desarrollo de aplicaciones relacionada al procesamiento de señales acústicas (Aparicio Sosa, 2014). En un principio el método de detección de señales se basaba en el criterio de Neyman Pearson (Nielsen , 1991), donde se compara la señal recibida con un valor umbral para detectar si existe un objetivo en ese momento, siendo una de sus limitantes la posibilidad de obtener falsas alarmas. Posteriormente se incluyeron umbrales adaptativos que utilizaban la información tanto en el dominio del tiempo como de la frecuencia para mitigar estos problemas (Komari & Farsi, 2018).

En la actualidad, las técnicas de *machine learning* han mostrado tener un gran desempeño en el procesamiento digital de señales y se muestran junto a *deep learning* como reemplazo de los métodos convencionales (Yalta, Nakadai, & Ogata, 2017). Por lo expuesto anteriormente, existen estudios acerca de la aplicación de *machine learning* en el procesamiento de señales sonoras, pudiendo citar por ejemplo el estudio de Shin y Kil (1996) en la cual se analizaron varias características de las señales acústicas y se aplicaron algoritmos de aprendizaje automático para comprobar que su técnica era mejor que los métodos tradicionales. Otro ejemplo que utiliza estas técnicas es el clasificador de ruidos marinos propuesto por Hemminger y Pao (1994) para diferenciar entre seis tipos diferentes de sonidos utilizando como base la transformada de Fourier de tiempo reducido de la señal observada y se la compara con la ventana de tiempo que la precede, esta información se usa como entrada de una red neuronal.

Justificación

La propagación de ondas sonoras subacuáticas es un campo en el cual se estudia el comportamiento del sonido en este medio, dando lugar a múltiples aplicaciones como la comunicación submarina, la detección de objetos, la exploración de recursos marinos, la cartografía de su entorno, etc. (Yang, Lee,, Kim, & Choo, 2020). Debido a las limitaciones que se presentan en este medio, frecuentemente se utilizan vehículos remotamente operados (ROV), vehículos subacuáticos autónomos (AUV) y los vehículos autónomos para intervención I-AUV, los cuales generalmente pueden ser manipulados por medio de un cordón umbilical, teniendo autonomía energética y/o equipados con equipos de maniobra (Bejarano Cedeño & Vera Cedeño, 2019).

Para llevar a cabo una tarea exitosa con estos vehículos es necesario conocer la ubicación de los mismos en todo momento; por lo tanto, el posicionamiento ha cobrado gran importancia debido al interés de implementar nuevas tecnologías en este entorno (Vilarnau, 2014). Uno de los sistemas de posicionamiento más conocidos y empleados es GPS (*Global Positioning System*), sin embargo, este sistema tiene problemas en entornos subacuáticos, ya que las señales electromagnéticas presentan una gran atenuación en el agua. Es por ello que se hace necesario estudiar otros tipos de señales que tengan una mejor propagación como es el caso de las ondas sonoras (Aparicio Sosa, 2014).

Alcance del proyecto

Por lo expuesto anteriormente, este trabajo tiene como finalidad aportar en el desarrollo de nuevas técnicas que permitan determinar el posicionamiento de un vehículo acuático mediante un arreglo de hidrófonos, enfocado en las diversas técnicas de *Machine Learning* y cómo la variación de parámetros como el tamaño del *frame*, dominio de la señal, número de componentes, número de muestras, etc. influyen en el rendimiento de los modelos dejando un precedente para futuras investigaciones en este campo.

Objetivos

General

Diseñar un sistema de posicionamiento subacuático mediante señales acústicas y técnicas de *machine learning*.

Específicos

- Investigar las principales técnicas de posicionamiento subacuático y los algoritmos más populares de *machine learning*.
- Crear una base de datos con información recopilada por hidrófonos y asociada a diferentes posiciones del dron subacuático.
- Identificar las principales características que permitan determinar el posicionamiento de los drones subacuáticos.
- Entrenar un sistema basado en *machine learning* para realizar el posicionamiento de un dron subacuático real.
- Realizar las pruebas correspondientes y determinar la efectividad del sistema.

Marco teórico

Posicionamiento acuático

En las últimas décadas se han logrado grandes avances tecnológicos respecto a los vehículos submarinos que conllevan a que estos sean vistos como una opción interesante para múltiples aplicaciones subacuáticas (Castillo, Castro, & Vogulys, 2020). De los diferentes tipos de vehículos, los que mayor interés presentan son los vehículos submarinos autónomos (AUV) debido a libertad de movimiento ya que no se encuentran atados a un cordón (Palacios García, 2007). Esto los convierte en vehículos con un gran potencial en aplicaciones militares, científicas y comerciales (Carrera Erazo & Paredes, 2019). No obstante, una parte fundamental es el sistema de navegación que, debido a problemas ya mencionados anteriormente, no es fácil de implementar sistemas de posicionamiento tradicionales que se usan en tierra. Por lo tanto, las tareas que estos vehículos realizan se ven limitados por la precisión de sus sistemas de navegación. De forma general, existen 3 tipos principales de navegación que son: navegación estimada e inercial, sistemas acústicos externos y algoritmos de mapeo del terreno y visión (Leonard & Bahr, 2016). Cada uno presenta diferentes características en el costo, tamaño y consumo energético.

Navegación estimada e inercial

El sistema de navegación estimada necesita los datos del posicionamiento y velocidad inicial del vehículo, ya que gracias a estos datos más los datos obtenidos de los sensores internos se puede estimar la nueva posición del vehículo. La principal desventaja de estos sistemas es la acumulación del error ya que conforme pasa el tiempo, el error se va acumulando haciendo que cada estimación de la posición sea menos precisa (LaPointe, 2006).

Los sistemas inerciales se caracterizan por utilizar un punto de referencia para determinar su posición relativa a ese punto sin la necesidad de emplear un sistema externo. Para ello y al igual que en la navegación estimada es indispensable conocer las condiciones iniciales como es la posición y la velocidad inicial. Con estos datos más los obtenidos mediante

la unidad de medición inercial IMU se puede estimar la posición y velocidades posteriores (Sánchez Hernández, 2016). De forma general, estos IMUs cuentan con acelerómetros y giroscopios ortogonales que les permite tener una medición tridimensional de las aceleraciones lineales y angulares (Aldana Afanador & Esteban Villegas, 2018).

Navegación geofísica

En la navegación geofísica se utilizan las características físicas del terreno, por lo tanto, es necesario contar con un mapa previo del entorno o uno generado durante la operación. De esta forma se realiza un mapeo del sitio y de forma simultánea se la compara con la información del mapa y se establece una posición estimada (Arango Rivera, 2015).

Actualmente las propuestas más populares para este tipo de navegación son por sonares y sensores ópticos, los primeros tienen la desventaja de no poseer una buena resolución y que su utilidad es dependiente de las características del terreno. Mientras que los ópticos se los complementa con la obtención de imágenes, algo común en sonares (Vilarnau, 2014). Si bien esta técnica presenta un gran potencial, en la actualidad todavía no se ha desarrollado lo suficiente para ser un sistema confiable por lo cual no se encuentra ampliamente implementado en sistemas de navegación (LaPointe, 2006).

Navegación acústica

Actualmente el sistema más implementado es el basado en señales acústicas. Estos utilizan una red de transmisores y receptores de sonido que gracias al principio de triangulación logran estimar la posición del vehículo (Carrera Erazo, Mena, Arciniega, Padilla, & Paredes, 2020). Existen diferentes tipos de dispositivos que tienen la tarea de ser receptores, transmisores e incluso ambos (Leonard & Bahr, 2016). A continuación, se realiza una breve descripción de cada uno.

- Interrogador: Este elemento funciona como receptor y a la vez como transmisor, su funcionamiento básicamente se resume en enviar una señal acústica a una determinada frecuencia y recibir una señal de respuesta en una frecuencia diferente.

- **Transponder:** Se lo implementa en el lecho marino o en el vehículo y también cumple con las funciones de receptor y transmisor, recibiendo la señal de un interrogador o de un transductor para posteriormente enviar una señal de respuesta.
- **Hidrófono:** Este elemento se instala en la plataforma de control y su principal característica es que solo puede recibir señales acústicas.
- **Transductor:** Es la antena del transpondedor o del interrogador que se conecta al sistema de control para recibir y transmitir señales.

Considerando estos elementos la navegación acústica se puede clasificar en sonares activos y pasivos (Carrera Erazo & Guevara Bustillos, 2022), los sonares pasivos solo está formados por hidrófonos y reciben solo el sonido generado por el vehículo subacuático, mientras que los activos utilizan los elementos ya mencionados anteriormente que generan una señal específica para estimar el posicionamiento (Perez, Gómez, & Franco, 2017). Basado en la distancia entre elementos se tiene las siguientes clasificaciones.

Long base line

Este sistema emplea entre 3 a 12 transpondedores que se instalan en el lecho marino y la longitud de la línea base varía entre los 100 y 6000 metros, el vehículo posee un transductor o interrogador que permite la comunicación con esta red de transpondedores. Como primer paso para determinar la posición es necesario un proceso de calibración en la cual se estima el posicionamiento de cada transpondedor gracias a un interrogador en la embarcación de control, con ello se obtiene un posicionamiento referencial a la embarcación, y esta a su vez tiene un posicionamiento GPS. De esta forma se puede establecer la posición global de los transpondedores.

Una vez realizada esta calibración se puede establecer el posicionamiento del vehículo el cual emite una señal a una determinada frecuencia que es receptada por la red de transpondedores los cuales responden en una frecuencia diferente, la posición se estima

gracias al tiempo transcurrido desde que se emite la señal del vehículo hasta que recibe una señal de respuesta (Arango Rivera, 2015).

Estos sistemas son los que poseen la mejor precisión debido a la distancia entre cada uno de los transpondedores de la red y gracias que se encuentran fijos en el lecho marino, no es necesario compensar errores por el movimiento de la red. Sin embargo, estas mismas características son las que limitan al sistema en determinadas aplicaciones, ya que el número de componentes empleados son mayores a otros sistemas, se debe tener un proceso de calibración previo, la complejidad de la instalación y de retiro es mayor (Leonard & Bahr, 2016).

Short base line

Estos son los sistemas más antiguos ya que utilizan a la propia embarcación como una red de transpondedores e hidrófonos que se colocan en la parte inferior de la misma, la distancia entre estos dependerá del tamaño de la embarcación, pero varía entre los 10 a 20 metros y generalmente las formaciones empleadas son triangulares y cuadradas, el vehículo cuenta con un transpondedor para comunicarse con esta red. Para estimar la posición primero los transpondedores de la red envían una señal que es recibida por el vehículo, este a su vez envía una respuesta que es captada por la red de transductores e hidrófonos, y gracias al tiempo de llegada a cada uno de estos elementos se estima su posicionamiento gracias a la triangulación. Si bien la instalación y complejidad de este sistema es menor a LBL, también se tiene menor precisión ya que está directamente relacionado a la distancia entre los componentes de la red de transpondedores e hidrófonos (Arango Rivera, 2015).

Ultra short baseline

En este sistema se tiene como principal característica que la distancia entre los elementos de la red de transpondedores es tan pequeña que no sobrepasa los 10 cm y caben en un solo dispositivo que se instala en la embarcación de control, la forma de determinar el posicionamiento también es por triangulación y para estimar la orientación del vehículo se

utiliza el desfase de la señal emitida por el transductor del vehículo. Este sistema al ser el más compacto es el más fácil de emplear e instalar (Leonard & Bahr, 2016).

Preprocesamiento de señales

Uno de los elementos más importantes en el aprendizaje automático es el preprocesamiento de señales, esto se debe a que el aprendizaje automático busca determinar patrones o reglas que permitan cumplir con un objetivo propuesto, cuando se incluyen datos sin procesar estos pueden ser “impuros” como tener presencia de ruido, datos inconsistentes o incompletos, lo que conlleva un riesgo ya que puede generar patrones o reglas poco relevantes (Ghanbari, Naghavi, Ghaderi, & Sabaghian, 2009).

Así que una preparación y organización previa de la información ayuda a disminuir el tamaño de datos original manteniendo las características relevantes y, por lo tanto, mejorando la eficiencia en todo el proceso de entrenamiento y evaluación del modelo debido a que el proceso de aprendizaje obtendrá patrones y reglas relevantes para la aplicación.

Procesamiento en frames

La segmentación de una señal en pequeñas ventanas es un método común para el procesamiento de señales. La idea es segmentar la señal de entrada en bloques de una determinada longitud la cual dependerá de las características de la señal y de los requerimientos de la aplicación. Esto permite utilizar algoritmos basados en *frames* ya que estos se caracterizan por tener una reducción en la utilización de recursos, el coste computacional y obtener vectores de características (Aguirre Martín, 2017).

Para aumentar el número de muestras se puede incluir un solapamiento lo que significa que una parte del siguiente *frame* comparte información del *frame* anterior, el porcentaje de información compartida entre ambos representa el porcentaje de solapamiento. Esto se realiza con la finalidad de aumentar el tamaño de muestras y tener información más completa de todos los posibles escenarios (Martínez, 2020).

Transformada rápida de Fourier

La transformada rápida de Fourier o FFT es un algoritmo optimizado de la transformada discreta de Fourier al separar los componentes pares e impares. Esta transformada permite ver el comportamiento de una señal en el dominio del tiempo al dominio de la frecuencia (Madisetti & Williams, 2010), algo fundamental en el análisis de señales ya que los componentes frecuenciales de una señal poseen numerosas aplicaciones.

Muestreo

El número de muestras por segundo es una de las características más importantes para las señales digitales ya que esto representa la frecuencia de muestreo (F_s) y por lo tanto indica la frecuencia máxima que se puede representar en una señal, según el teorema de Nyquist (Pareja Contreras, 2017), se tiene que: $F_{max} = 2F_s$

Esto es necesario para evitar el efecto de *Aliasing*, el cual produce que dos señales continuas diferentes se vean iguales cuando se muestrean (Madisetti & Williams, 2010). Para esta aplicación, ya que no se conoce qué información será relevante, las especificaciones serán las más altas para tener la mejor representación de la señal con un formato sin pérdidas WAV y un muestreo de 48000 Hz.

Análisis de componentes principales PCA

Existen ciertas aplicaciones en las cuales se puede tener una base de datos con un gran número de variables. Sin embargo, al tener tantas variables se puede penalizar el tiempo de entrenamiento y la velocidad de predicción del sistema, por lo tanto, reducir el número de variables y que al mismo tiempo representen a la base de datos original lo más fiable posible es una de las áreas más importantes al momento entrenar modelos (George & Vidyapeetham, 2012).

La técnica más frecuente para la reducción de dimensiones es el Análisis de componentes principales PCA basada en la varianza (Hastie, Friedman, & Tibshirani, 2009). Si

bien la reducción de variables puede parecer que aumentará el error, se ha demostrado que incluso la reducción de componentes, disminuye el error de ciertos modelos ya que deja de considerar a variables que son irrelevantes, así que determinar el número ideal de componentes permitirá que el error en el modelo propuesto disminuya y también que el tiempo de entrenamiento y predicción sea menor gracias a un sistema menos complejo (García Álvarez & Fuente, 2011).

Machine learning

El aprendizaje de máquina o *machine learning* son un conjunto de técnicas que otorgan al ordenador la capacidad de aprender mediante el desarrollo de algoritmos y modelos que forman parte del proceso de aprendizaje. Con esto se busca que un ordenador tenga la posibilidad de aprender a detectar patrones en un conjunto de datos o realizar una estimación en base a datos desconocidos (Salvo Gutierrez, 2020). Estas técnicas de aprendizaje pueden dividirse en dos grupos: el aprendizaje supervisado y no supervisado.

Aprendizaje no supervisado

Se crea un modelo que se origina de un conjunto de datos que no presentan una respuesta o etiqueta, así que el sistema no es capaz de establecer si su respuesta es correcta o errónea a un determinado dato de entrada. Una de las técnicas más utilizadas en esta área es el *clustering* donde se agrupan conjuntos de datos en base a las características que comparten (Nieto Jeux, 2021). Esto no solo es útil para agrupar conjuntos de datos sino también para determinar cuáles son las características más relevantes, por lo tanto, también es empleado como parte del preprocesamiento de señales al determinar las características más relevantes que pueden ser útiles al momento de entrenar un sistema, un ejemplo de esto es el análisis de componentes principales que ya se mencionó anteriormente, o más recientemente el uso de *autoencoders*.

Agrupación jerárquica

Este tipo de agrupamiento genera varios subconjuntos que comparten características similares, los cuales son ordenados de forma jerárquica de acuerdo a sus características, este orden puede ser ascendente o descendente. Una de las formas más fáciles de representar estas agrupaciones es con el dendograma, este permite representar de forma gráfica cada una de los grupos y al mismo tiempo permite identificar el nivel de similitud entre estos grupos (Nieto Jeux, 2021).

Agrupación particional

Este agrupamiento segmenta en grupos diferentes, por lo tanto, todas las agrupaciones son mutuamente excluyentes. A diferencia de la agrupación jerárquica, se necesita de una inicialización con el número de agrupaciones deseadas, a este número por lo general se lo llama k (Nieto Jeux, 2021).

Aprendizaje supervisado

Desarrolla un modelo a partir de experiencias y patrones ya establecidos, esto se debe a que la información que usa para su entrenamiento tiene una “respuesta deseada” también llamada etiqueta. Así la red es capaz de variar sus parámetros para coincidir con la respuesta deseada o al menos acercarse al valor. De forma general, se puede clasificar en 2 grupos: Clasificación y regresión.

El aprendizaje por clasificación es similar a lo mencionado en el aprendizaje no supervisado, donde se busca crear agrupaciones de un conjunto de datos, pero la diferencia radica en que estos datos cuentan con una etiqueta que obliga al modelo a ajustarse a estas agrupaciones definidas por las etiquetas. La regresión en el aprendizaje automático hace posible que se pueda predecir un resultado de valores continuos a partir de una o más variables predictoras, mediante la utilización de diversos métodos matemáticos. Uno de los más conocidos es la regresión lineal por su facilidad de uso e interpretación (Faraway, 2004).

Técnicas de Aprendizaje Supervisado

Regresión lineal

Los modelos lineales permiten determinar la relación existente entre una variable dependiente (respuesta) y una o más variables independientes (predictores). De forma general la ecuación que representa un modelo lineal se presenta en la ecuación (1).

$$y = \beta_0 + \sum \beta_i X_i + \epsilon_i \quad (1)$$

En la cual, y es la variable dependiente, X_i es la variable independiente, β_i son las estimaciones de parámetros lineales a ser calculados y ϵ_i son los errores de cada término (Moral Peláez, 2016). Cuando la aplicación solo tiene un predictor se la llama regresión lineal simple, si posee más de un predictor se le llama regresión lineal múltiple y si la aplicación tiene más de una variable dependiente se la denomina regresión lineal multivariante (Aldaz Manzano & Uriel Jimenez, 2017).

Arboles de decisión

Este es uno de los métodos más sencillos ya que emplea árboles de decisión binaria y pueden ser usados tanto para clasificación como para regresión. Se empieza con un nodo inicial o también llamado nodo raíz, del cual se generan dos divisiones o ramas que divide a las muestras en dos subconjuntos, cada uno de estas divisiones se representa por un nodo, este proceso continúa hasta llegar a los nodos terminales (hojas) que son los que producen la predicción, por lo general cuando se tiene la estructura final los nodos terminales realizarán la predicción basada en la moda si es una aplicación de clasificación y la media si es de regresión (Díaz & Correa, 2013).

Máquina de vectores de soporte

Las máquinas de vectores de soporte o SVM son métodos estadísticos que fueron originalmente desarrolladas para tareas de clasificación mediante hiperplanos, con el paso del tiempo se desarrollaron para solventar tareas de regresión (Campo León, 2017). Este

algoritmo utiliza los vectores como un soporte que permite maximizar la distancia entre los datos y el hiperplano (Cristianini & Scholkopf, 2002).

Procesos gaussianos

Los procesos gaussianos se pueden resumir como una función en la cuál en cada punto entrega una media y varianza de una distribución normal, haciendo que pueda adaptarse a prácticamente cualquier función, y gracias a que no es un modelo paramétrico puede fácilmente adaptarse a nuevos datos ya que se caracteriza por ser altamente flexible (Rasmussen & Williams, 2006). Por lo general, en un proceso gaussiano la función de media se considera como nula, así que la única elección a modificar es la función de covarianza que se llama *kernel*, este será el responsable de establecer las propiedades del proceso gaussiano (Díaz García, 2020). De hecho, el principal desafío para el aprendizaje es encontrar un *kernel* que cumpla con las propiedades requeridas de la aplicación.

Bagging y Boosting

El *Bagging* utiliza las muestras de entrenamiento para entrenar un modelo que se usa para la predicción, al final se tiene tantos modelos como predicciones que a su vez son muestras de entrenamiento, la idea es promediar todas las predicciones ya mencionadas lo que reduce la varianza y simplifica la respuesta (Sánchez Berral, 2017). Una variante son los bosques aleatorios, estos agregan una aleatoriedad en las variables y observaciones para que todos los árboles sean diferentes, además son más eficientes con el costo computacional que *bagging*, ya que cada árbol se construye de forma más rápida al evaluar menos predictores (Nieto Jeux, 2021).

Boosting es una técnica lenta que usa muchos modelos poco predictivos y al combinarlos mejora su capacidad, un algoritmo común es *AdaBoost* que a diferencia del bosque aleatorio no utiliza árboles independientes entre sí, al contrario, genera árboles y cada uno depende de su predecesor. A diferencia de *Bagging* este puede sobrentrenarse ya que sus árboles son dependientes, y al ser lento son generalmente pequeños (Nieto Jeux, 2021).

Redes neuronales

Las redes neuronales tienen un funcionamiento similar a las neuronas biológicas, tienen elementos simples los cuales pueden conectarse entre sí formando múltiples interconexiones. Gracias a esta capacidad pueden adaptarse y cambiar su comportamiento según los datos entregados, esto las convierte en una técnica interesante en situaciones en la que la información no es suficiente o varía en el tiempo. Las redes neuronales son nodos conectados entre sí, cada uno tiene un peso y dependiendo del valor de entrada, este entregará un valor de salida al siguiente nodo (Ghanbari, Naghavi, Ghaderi, & Sabaghian, 2009). Considerando las diferentes características de este tipo de redes como el peso de los nodos, el número de nodos, el tipo de interconexiones y las capas de la red se puede tener un sinnúmero de posibles combinaciones (Salvo Gutierrez, 2020).

Evaluación de modelos

Para determinar la precisión de un modelo se debe tener parámetros y procedimientos que demuestren de forma cuantitativa si un modelo es mejor que otro. Esto se logra mediante el error obtenido en el modelo y la forma en la que se obtiene el mismo. A continuación, se muestra los tipos de error más comunes en regresiones y los procedimientos para estimar este error procurando evitar un sistema sesgado (Ochoa, 2019).

Error en regresión lineal

Cuando se habla de una regresión lineal, se desarrolla un modelo con la finalidad de predecir valores, donde estos no necesariamente serán exactos a los valores deseados, por ello se define como error a la distancia entre los datos predichos y los deseados. Actualmente existen parámetros que evalúan el rendimiento de un sistema de regresión, las principales son: el error cuadrático medio MSE, el error medio absoluto MAE y el R cuadrado (Sánchez Berral, 2017).

Error cuadrático medio MSE (*Mean Square Error*)

Este parámetro es el más utilizado para evaluar las tareas de regresión lineal, pues representa la distancia promedio al cuadrado entre el valor real y_i y el predicho \hat{y}_i , penalizando así las distancias más grandes de un conjunto de observaciones n , representadas en la ecuación (2).

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (2)$$

El error de raíz cuadrada media RMSE emplea una raíz cuadrada en el error cuadrático medio MSE. Este valor indica el ajuste absoluto de los datos con el modelo y determina la proximidad de los datos predichos con los reales. Esta puede considerarse como una desviación estándar de varianza inexplicada y es útil ya que posee las mismas unidades de los valores reales o de respuesta (Balaguer Beser & Ruiz Fernández, 2021). Cuanto menor sea su valor mejor será su ajuste a los valores deseados. En la ecuación (3), donde n es el número total de muestras, y_i son las predicciones y \hat{y}_i son los valores verdaderos.

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (3)$$

R cuadrado

Esta métrica indica la aptitud del modelo para explicar la variabilidad de sus variables independientes en relación con las variables dependientes. En la ecuación (4), su escala va de $-\infty$ a 1, valores negativos indica que esto modelo no logra la tarea propuesta, un valor de cero muestra que el modelo propuesto no presenta mejora alguna en relación a un modelo medio y 1 indica que el modelo es una representación perfecta (Balaguer Beser & Ruiz Fernández, 2021).

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y}_i)^2} \quad (4)$$

Una de las características de este parámetro es que tiende a mejorar conforme se aumentan predictores; no obstante, esta mejora puede ser una ilusión si dichos predictores no mejoran el ajuste del modelo. Por ello se tiene una variante llamada R cuadrado ajustado que disminuye cuando se le agregan predictores que no son relevantes (Balaguer Beser & Ruiz Fernández, 2021). En la ecuación (5), donde n representa el tamaño de la muestra y p es la cantidad de variables en el modelo.

$$R_{ajus}^2 = 1 - (1 - R^2) \left(\frac{n - 1}{n - p - 1} \right) \quad (5)$$

Error absoluto medio MAE (*Mean Absolute Error*)

La ecuación (6) indica el promedio de la diferencia absoluta entre el valor real y el predicho, a diferencia de MSE este es un valor lineal por lo que todas las diferencias se ponderan por igual en el promedio (Balaguer Beser & Ruiz Fernández, 2021).

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (6)$$

Métodos de validación

La validación en *machine learning* permite determinar si los datos que predice el modelo son aceptables y representan en parte a los datos deseados utilizando datos que no se han empleado en el entrenamiento, con esta información se puede determinar si el modelo necesita ser ajustado (Ochoa, 2019).

Validación Cruzada *k-fold*

La validación cruzada *k-fold* consiste en segmentar todo el conjunto de datos en k subconjuntos aproximadamente del mismo tamaño, haciendo que cada uno de estos segmentos sean considerados como un conjunto de pruebas y el resto como entrenamiento, este proceso se repite con cada uno de los subconjuntos hasta utilizar todos los subconjuntos

como datos de prueba, es decir k iteraciones. Al momento de terminar de realizar todas las iteraciones se promedia los errores de cada iteración.

El número k dependerá de la varianza de los datos, mientras menor varianza exista a lo largo de los datos originales se podrá utilizar un k más bajo ya que esto significa que los diferentes segmentos de los datos originales no difieren tanto y con menos iteraciones el costo computacional también se verá reducido, caso contrario es mejor usar un k mayor para compensar esta varianza penalizando el tiempo de entrenamiento. Por lo general se recomienda una validación cruzada de $k = 10$ para datos en los que se desconoce la confiabilidad de los datos de entrenamiento (Ochoa, 2019).

Holdout

Este proceso segmenta los datos en dos subconjuntos, las proporciones de cada subconjunto son determinadas por el usuario, uno de estos segmentos se utilizará para entrenamiento y el otro para validación. Se realiza este proceso solo una vez por lo que el tiempo de entrenamiento y validación se reduce considerablemente en comparación con *k-fold*, esto la convierte en una propuesta interesante para conjunto de datos grandes. No obstante, la segmentación del conjunto puede generar resultados sesgados, una forma de mitigar este problema es que cada subconjunto se obtenga de forma aleatoria (Kelleher, Mac Namee, & D'Arcy, 2020).

Metodología

El procedimiento propuesto para la realización de este trabajo consiste en 4 etapas: Creación de una base de datos, preprocesamiento de señales, etiquetado de la base de datos y entrenamiento de modelos.

Creación de base de datos

Para este trabajo ya se contaba con un conjunto inicial de grabaciones, el cual está formado por una secuencia de audios tomados de cada hidrófono, se realizaron múltiples grabaciones de diferentes trayectorias para cada uno de los arreglos de hidrófonos. Se detalla a continuación la forma en la que se obtuvieron los datos.

Escenario

El primer punto es determinar un ambiente acuático controlado donde se pueda fácilmente realizar las pruebas y de ser necesario cambiar la distribución de los hidrófonos, el mejor lugar para ello es una piscina, en la Figura 1 se muestra el escenario planteado.

Figura 1

Piscina empleada como escenario de pruebas



Vehículo acuático

El vehículo utilizado para las pruebas es un AUVs *PowerVision PowerRay*, este modelo es de dimensiones pequeñas y permite una buena maniobrabilidad en una piscina, además se maneja a distancia.

Figura 2

Vehículo acuático PowerVision PowerRay



Hidrófonos

Se incluye 4 hidrófonos *DolphinEar DE200* que se conectan a un amplificador analógico para ingresar a una tarjeta de audio *Behringer U-phoria UMC404HD*. Mediante el software *Reaper V6* se graba las respectivas señales de audio de cada uno de los hidrófonos.

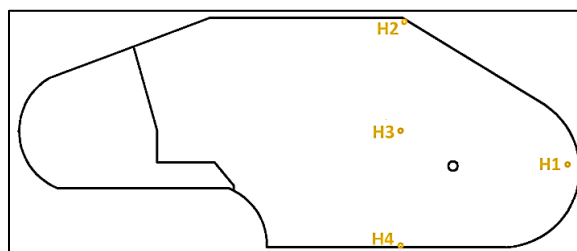
Como ya se mencionó en la navegación acústica SBL, los arreglos más utilizados son el triangular y cuadrado, por lo tanto, los arreglos propuestos son los mismos, en la Tabla 1 se muestra la posición de cada hidrófono para un arreglo triangular.

Tabla 1

Posición de los hidrófonos (formación triangular)

HIDRÓFONO	COORDENADA X	COORDENADA Y
H1	3.7	1.8
H2	0	4.9
H3	0	2.5
H4	0	0.1

Para una mejor visualización, en la Figura 3, se muestra su ubicación real en la piscina.

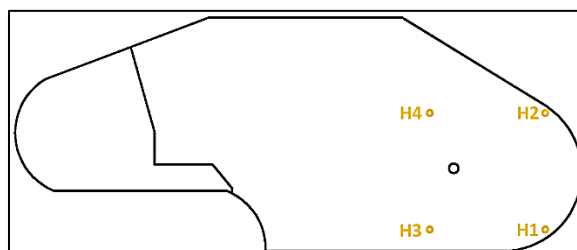
Figura 3*Arreglo de hidrófonos triangular*

Para el arreglo cuadrado se propone algo similar, la distribución de los hidrófonos solo ocupa una parte de la piscina ya que también es interesante observar si la precisión del modelo cambia si el vehículo está dentro o fuera del arreglo de hidrófonos. Tabla 2 muestra las coordenadas de cada hidrófono.

Tabla 2*Posición de los hidrófonos (formación cuadrada)*

HIDRÓFONO	COORDENADA X	COORDENADA Y
H1	3.5	0.5
H2	3.5	3.0
H3	1.0	0.5
H4	1.0	3.0

Igualmente, en la Figura 4 se muestra la ubicación real en la piscina.

Figura 4*Arreglo de hidrófonos cuadrado*

Grabación de audio

Una vez determinados los diferentes arreglos, se procede a realizar varias pruebas, que incluyen trayectorias dentro y fuera del arreglo de hidrófonos, pruebas entre determinados puntos y trayectorias libres. Cada prueba será grabada y contará con 4 archivos de audio correspondiente a cada hidrófono. Además, se incluye una grabación de video en la cual se observa el proceso de grabación de cada prueba, esto permite saber el momento en el que inicia la grabación y facilita el etiquetado de las muestras.

Con las pruebas realizadas se tiene una base de datos de 38 trayectorias entre ambos arreglos de hidrófonos, cada trayectoria posee 4 archivos de audio correspondientes a cada hidrófono. Con estos datos se crea en Matlab una estructura que contiene exclusivamente los archivos de audio, debidamente organizados, ya que estos serán sometidos a un preprocesamiento para obtener solamente los datos que sean relevantes para el entrenamiento.

Preprocesamiento

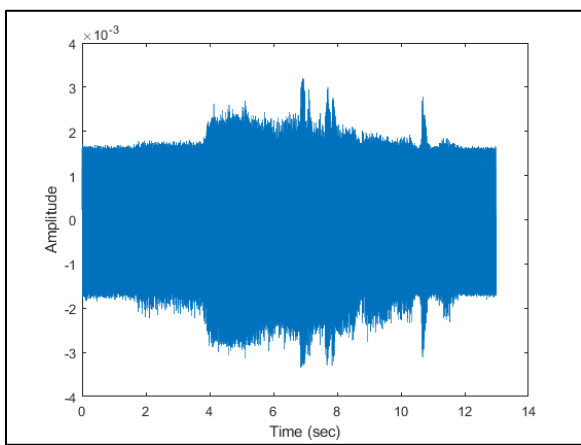
Recorte

La primera tarea de reducción de información radica en la longitud de los archivos de audio, ya que estos poseen momentos en los que el vehículo no tiene movimiento alguno o se lo está posicionando para iniciar con la trayectoria, es por ello que es necesario recortar el archivo para obtener una señal que contiene solamente momentos en el que el vehículo se mueve describiendo una trayectoria específica.

En la Figura 5 se observa una de las diferentes muestras tomadas sin procesar, a simple vista se observa algunos picos, pero es difícil determinar el momento en el que el vehículo inicia su trayectoria.

Figura 5

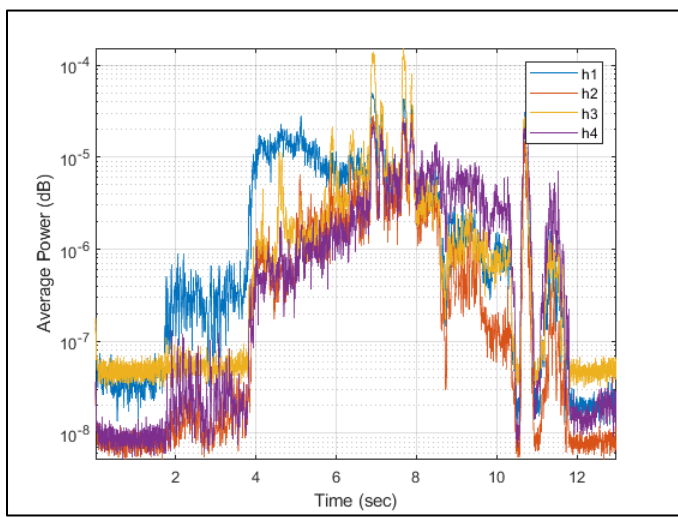
Señal original obtenida de un hidrófono



Así que para saber dónde es el momento exacto en donde empieza el movimiento del dron, se utilizará un análisis de la señal pero basado en la potencia media por tramos de 5 ms, esto se observa en la Figura 6, donde se tiene una mejor visualización de la señal y es más fácil estimar el punto en donde se mueve el dron, para complementar también se utilizó los videos de cada trayectoria y se estimó el tiempo el que el vehículo comenzó a describir la trayectoria respecto al tiempo en el que inició la grabación, de la misma forma se hizo con el final del recorrido.

Figura 6

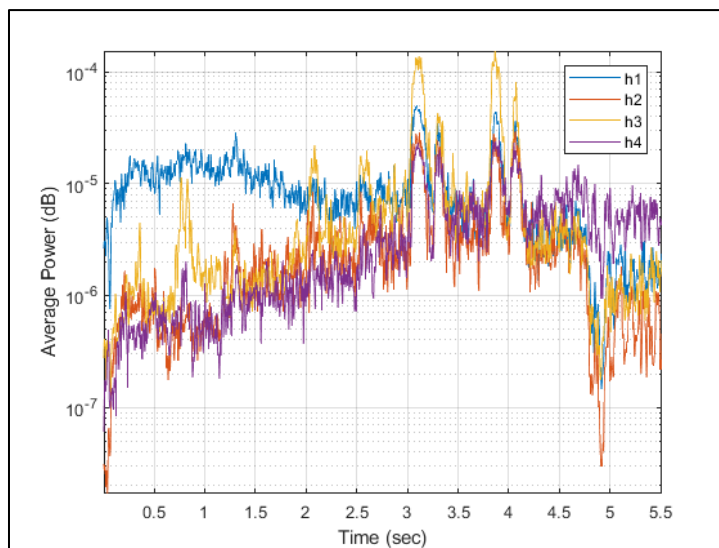
Potencia promedio (dB) de la señal original



Con el análisis de potencia y el tiempo estimado en cada video, se logró determinar el punto exacto en donde inicia el desplazamiento del vehículo y hasta donde termina. Con estos datos se procede a cortar el archivo de audio. En la Figura 7 se muestra la misma gráfica de potencia, pero solo con el segmento de audio que tiene al vehículo desplazándose por la piscina.

Figura 7

Potencia promedio (señal recortada)

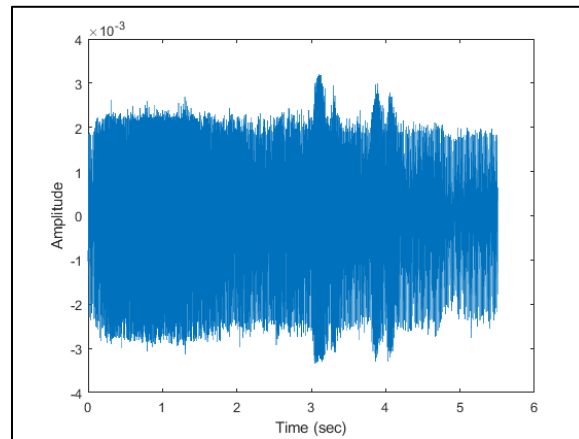


Por ello también es importante considerar los videos como referencia, ya que si bien en la Figura 6 parece que el vehículo comienza a moverse desde el segundo 2 y termina la trayectoria en el segundo 12, al final se observa que la señal útil de la Figura 7 inicia a los 4 segundos y llega hasta los 9.5 segundos aproximadamente.

La Figura 8 muestra la señal original de la Figura 5 recortada con la información que se considera útil para esta aplicación. Este procedimiento se realiza con cada una de las trayectorias planteadas tanto para el arreglo de hidrófonos triangular como también para el arreglo cuadrado.

Figura 8

Señal original recortada

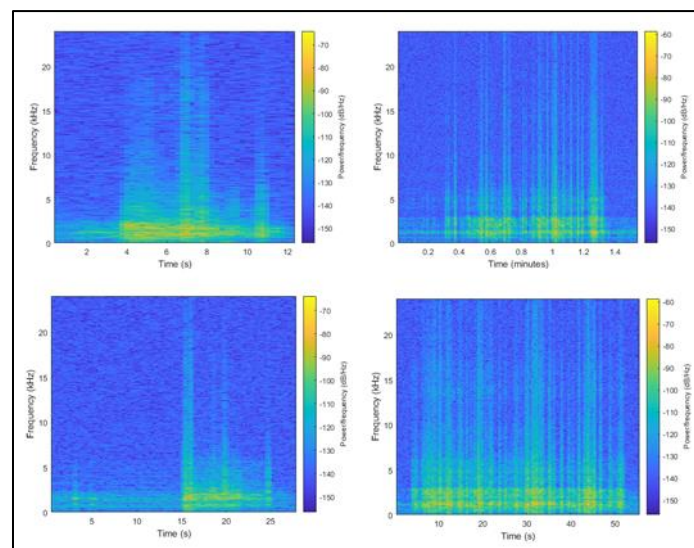


Filtrado

El modelo que se propone considera solamente el sonido generado por el vehículo, por lo que establecer dónde se encuentra la mayor parte de esas frecuencias generadas y eliminar las no deseadas ayudarán a que el sistema tenga información específica del vehículo. En la Figura 9 se utiliza un espectrograma en el cual se visualizará dónde se encuentra la mayor parte de la potencia respecto a la frecuencia.

Figura 9

Espectrograma de señales grabadas

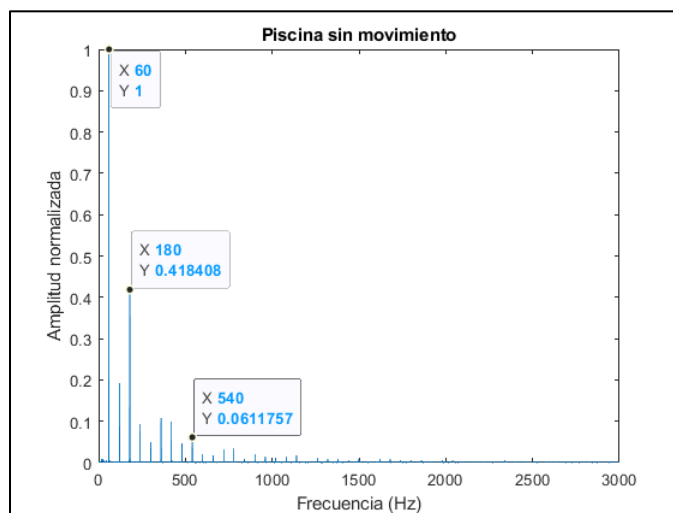


En la figura 9 se muestra algunas trayectorias tomadas, como se puede ver, la mayor parte de la potencia se concentra hasta los 3000 Hz. Las frecuencias bajas sin embargo también muestran que no son tan representativas, para ver de forma más minuciosa esas bajas frecuencias se utiliza la función FFT de Matlab para ver la potencia de la señal en frecuencia. Para ello se utiliza un audio especial grabado en el cuál no se tiene ningún desplazamiento del vehículo, solo graba a la piscina inmóvil, por lo que se puede observar el ruido generado.

En la Figura 10 se muestra a la piscina sin actividad alguna, se observa que tiene picos en bajas frecuencias, estas son múltiplos de 60 Hz y son considerables hasta los 540 Hz, por lo que se puede deducir que parte de la red eléctrica está afectando a los archivos de audio tomado por los hidrófonos ya que estos se conectan a una tarjeta de audio que se conecta a la red eléctrica, considerando que estos armónicos son considerables, se determina que el rango donde se encuentra mayor información del vehículo es entre los 560 Hz y los 3000 Hz.

Figura 10

Ruido presente en las grabaciones



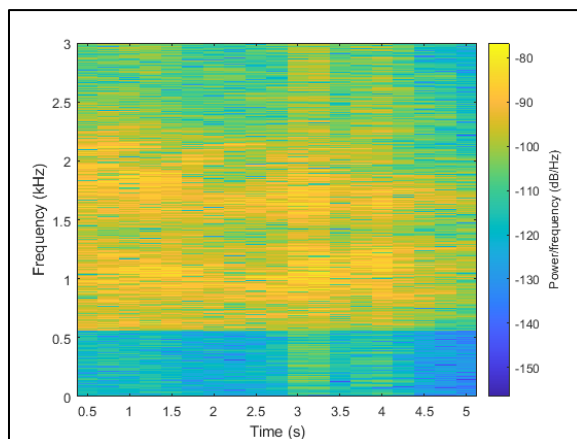
Para ello cada archivo de audio se somete a un filtro pasa alto de 560 Hz y luego de utiliza un remuestreo para bajar la frecuencia de 48000 muestras/s a 6000 muestras/s ya que según el teorema de Nyquist el muestreo necesario para recuperar una determinada frecuencia

debe ser el doble de esta. Además, esto reducirá el tamaño de las muestras que tendrá el sistema para entrenar haciéndolo más rápido tanto en el entrenamiento como en la predicción.

La Figura 11 muestra la señal resultante del filtrado y el remuestreo, gracias a ello el tamaño de la matriz de audio pasó de los 765 Mb a los 102 Mb. Conservando información relevante para el entrenamiento.

Figura 11

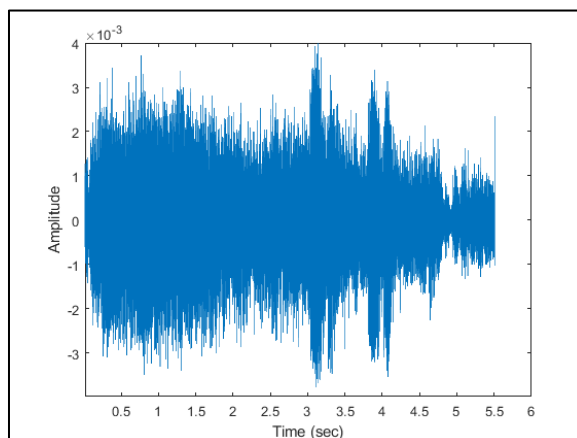
Espectrograma de señal filtrada y remuestreada



En la Figura 12 se muestra a la señal original preprocesada en el dominio del tiempo, esta presenta una gran diferencia respecto a la señal original de la Figura 5.

Figura 12

Señal original preprocesada (recorte, filtrado, remuestreo)



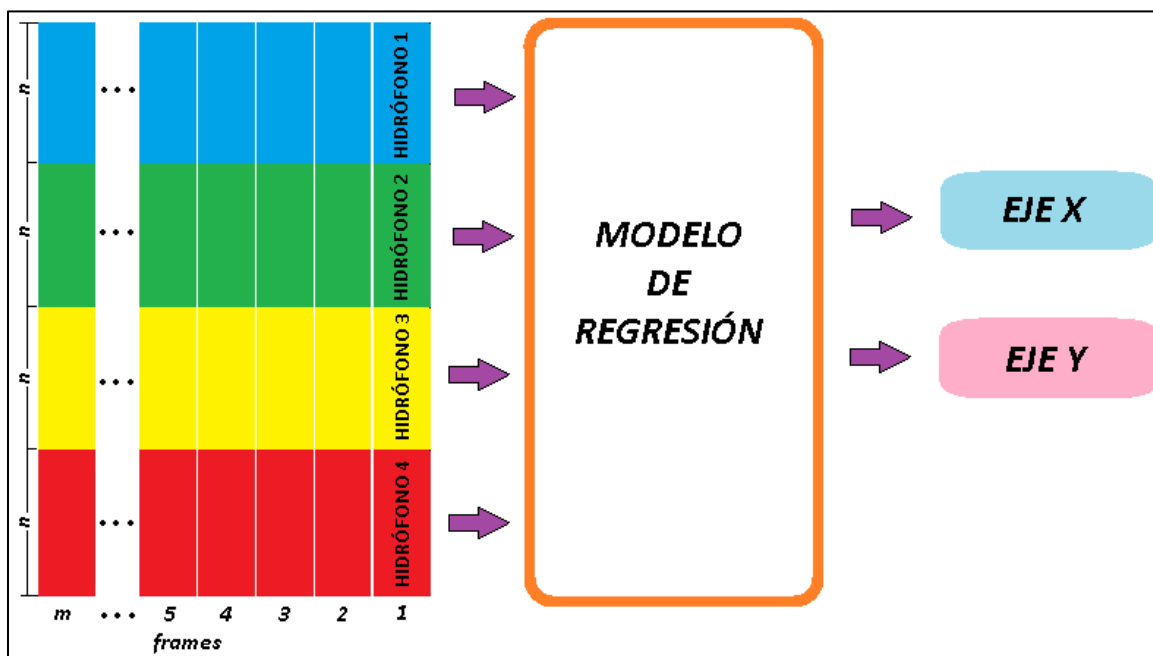
Análisis de la señal

Para este trabajo se propone hacer un análisis de señales obtenidas mediante *frames*, estos *frames* se cambiarán para observar el comportamiento de los modelos y cómo esto afecta a la precisión y al tiempo de entrenamiento. En cuanto a la base de datos se plantea un solapamiento del 80% para obtener un mayor número de muestras que representen mejor los posibles escenarios.

Ya que los sistemas de *machine learning* por lo general utilizan matrices para su entrenamiento, se utilizará los *frames* de cada hidrófono y se lo colocará de tal forma que se genere un vector de $1 \times 4n$ donde n representa la longitud de *frame* de cada hidrófono, este vector será una muestra y se realiza tanto para el dominio del tiempo como de la frecuencia. Este proceso se repite con todas las señales de audio hasta obtener una matriz de muestras de $m \times 4n$, donde m son todas las muestras obtenidas de la base de datos. La figura 13 representa la forma en la que se organiza la información para el entrenamiento.

Figura 13

Diagrama del proceso de entrenamiento

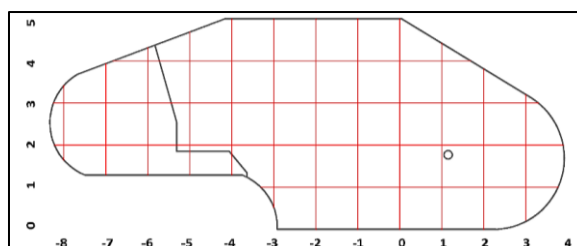


Etiquetado

Para el etiquetado es necesario tener una buena representación de las trayectorias, para lo cual se utilizará el software AutoCAD, ya que permite generar planos y múltiples herramientas de trazado. Lo primero es realizar una representación lo más realista posible de la piscina, esto se logra mediante un análisis de los videos para determinar las dimensiones exactas de la piscina y representarlas en un plano, la Figura 14 es el plano realizado en base a los videos, se encuentra segmentado en coordenadas (X, Y) y está dividida en metros.

Figura 14

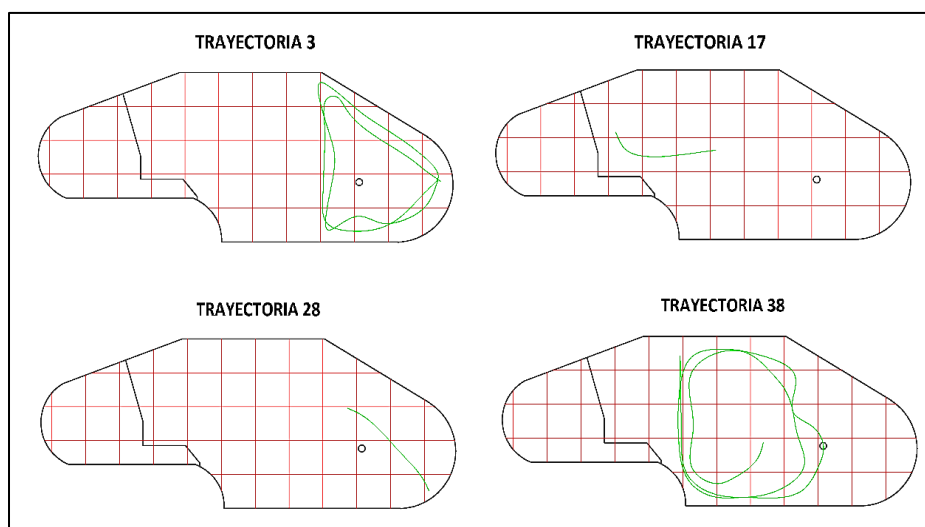
Representación del escenario



Con el plano finalizado, se procede a dibujar cada una de las trayectorias, para ello se utiliza las grabaciones de video como referencia. Obteniendo así las trayectorias de todas las pruebas. La Figura 15 muestra algunas de las trayectorias trazadas.

Figura 15

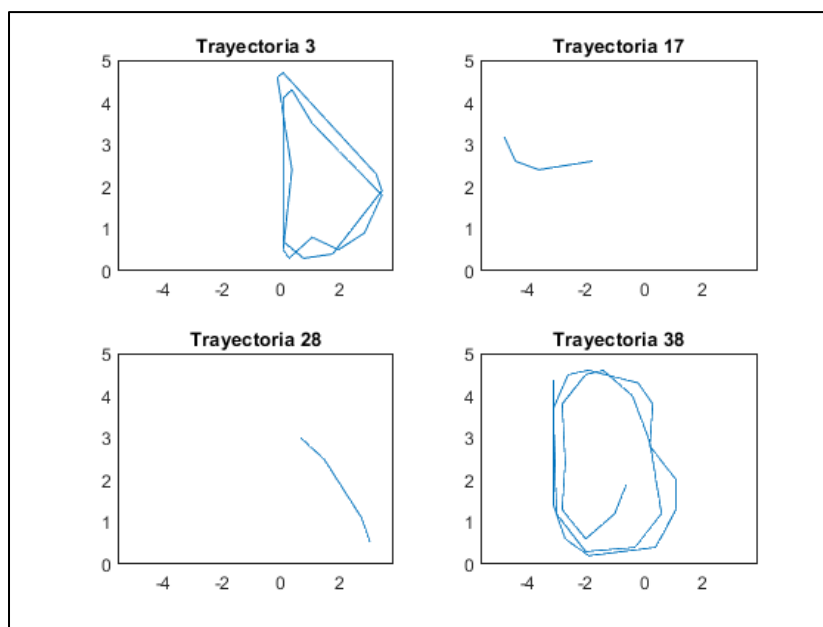
Trayectorias trazadas en AutoCAD



A estas trayectorias se las segmenta en secciones para posteriormente utilizarlo como etiqueta de las muestras, cabe destacar que en AutoCAD se utilizó también una resolución de 10 cm para una mejor resolución, por fines estéticos en la Figura 15 se muestra solo en metros. Una vez segmentada las trayectorias, se incluye esos segmentos en Matlab, la Figura 16 representa estas trayectorias por segmentos.

Figura 16

Trayectorias en Matlab



En Matlab se genera un código que permita crear estas etiquetas a partir de los segmentos, esto lo realiza en base al número de muestras que se tiene en cada trayectoria, este proceso debe ser automatizado ya que como se mencionó antes, se probará con diferentes tamaños de *frame*, lo que significa que el número de muestras y por lo tanto el número de etiquetas dependerán de esta variable.

Estas etiquetas se utilizan con la matriz de muestras de $m \times 4n$, quedando al final una matriz de $m \times (4n + 1)$ donde la última fila son los datos de respuesta que caracteriza al aprendizaje supervisado.

Entrenamiento

Para el entrenamiento se utilizará la herramienta de Matlab llamada *App Regression Learner*, la cual permite cargar una base de datos y utilizarla para entrenar diversas técnicas de *machine learning*, variando parámetros como en análisis de componentes principales, método de validación, *kernel*, entre otros dependiendo de la técnica empleada.

Para proceder con el entrenamiento se proponen las siguientes reglas:

- El entrenamiento se realizará con todos los modelos existentes en la aplicación, conforme se observe el rendimiento de cada modelo, se puede determinar qué modelos se ajustan mejor y en posteriores entrenamientos utilizar solo los que tienen mejores resultados.
- Se evaluará los modelos tanto en el dominio del tiempo como en frecuencia.
- Las pruebas se inician con un tamaño de *frame* de 50 ms, este valor se irá aumentando para observar si existe una mejora, en caso de ser así se seguirá incrementando los valores, caso contrario se irá disminuyendo hasta dar con el modelo que presente menor error.
- Una vez determinado el tamaño de *frame* con menor error para cada arreglo de hidrófonos, se procede a variar el número de componentes empleados en el entrenamiento con PCA hasta encontrar el que presente menor error.
- El error utilizado como referencia para los modelos será el error de raíz cuadrada media RMSE ya que se encuentra en las mismas unidades que las etiquetas.
- El solapamiento se mantendrá constante en 80%.
- El método de validación elegida es la validación cruzada *k-fold* con $k=10$.

Resultados

Análisis de resultados

En esta sección se presentan los resultados obtenidos en una serie de entrenamientos conforme las reglas presentadas en el apartado anterior. Para empezar, se realiza el entrenamiento con todos los modelos disponibles en *App Regression Learner* de Matlab R2021a, a continuación, se muestra un resumen de las técnicas usadas:

- Lineal: simple, robusta.
- Árboles de regresión: fino, mediano, robusto.
- SVM: lineal, cuadrático, cúbico, gaussiano fino, gaussiano medio y gaussiano robusto.
- *Boosted* y *bagged trees*.
- Procesos gaussianos: exponencial al cuadrado, *matern 5/2*, exponencial y cuadrático racional.
- Redes neuronales: estrecho, mediano, amplio, dos capas y tres capas.

Los escenarios planteados son para el arreglo cuadrado y triangular, al ser un posicionamiento con coordenadas rectangulares, cada una requiere de la regresión de dos coordenadas (X , Y)

Entrenamiento en el dominio del tiempo

Como punto de partida se entrena a los modelos en el dominio del tiempo, como se indica en las reglas propuestas, se inicia con un tamaño de frame de 50 ms, los resultados de los modelos se pueden ver a continuación:

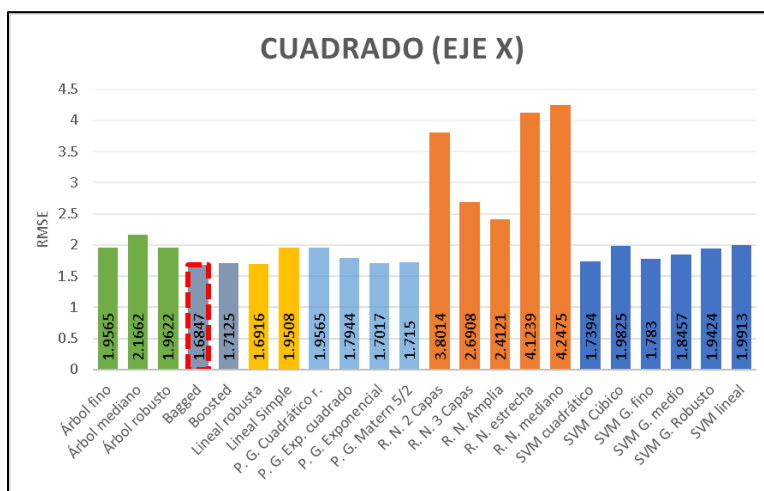
Entrenamiento inicial (dominio del tiempo)

Los resultados obtenidos se muestran en la Figura 17, donde se observa el error de raíz cuadrada media RMSE de cada técnica empleada, una de las características observadas es que las redes neuronales son las que presentan un error considerablemente más alto que otras

técnicas, mientras que el resto presentan mejores resultados siendo la mejor la técnica *bagged trees*, si bien estos datos no indican si el modelo cumple con los requerimientos, permite entender qué técnicas son las que tienen un mayor potencial de tener éxito.

Figura 17

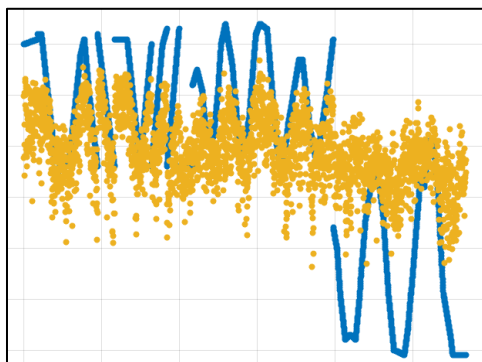
RMSE de técnicas empleadas (Cuadrado eje X) a 50 ms



Para una mejor observación se muestra en la Figura 18 un diagrama de dispersión de la técnica que tiene menor error RMSE, en este caso es la técnica *bagged trees*, en la gráfica se observa los puntos azules que representan los valores deseados mientras que los puntos amarillos indican los valores predichos. Se puede notar que el modelo no logra adaptarse completamente, una forma de confirmar esto es mediante el valor R cuadrado obtenido, que es de 0.3, un valor bajo que sugiere que el análisis en el dominio del tiempo es poco prometedor.

Figura 18

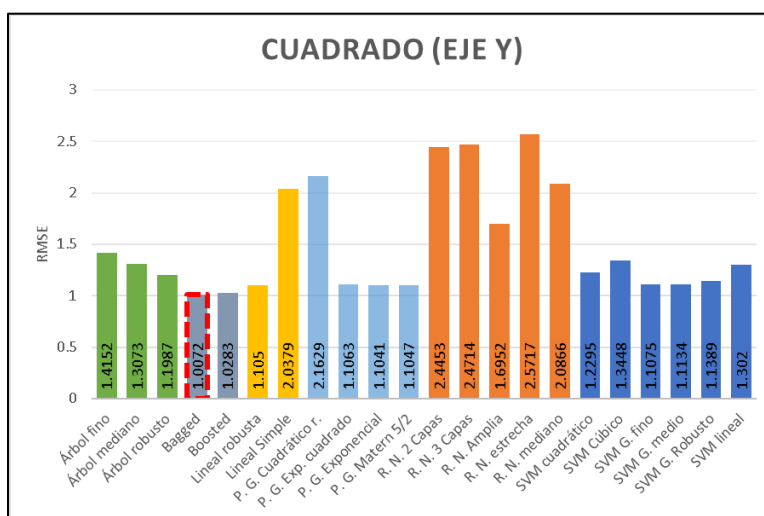
Modelo Cuadrado (eje X) entrenado con Bagged trees a 50 ms



El siguiente modelo a entrenar es con la misma configuración de hidrófonos en el eje Y, la Figura 19 muestra nuevamente que las redes neuronales no se adaptan bien para esta aplicación, mientras que la técnica *bagged trees* es la que obtiene mejores resultados.

Figura 19

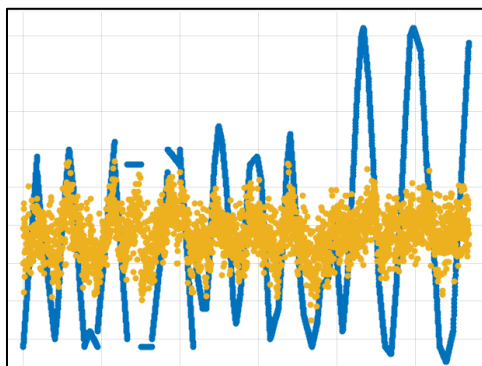
RMSE de técnicas empleadas (Cuadrado eje Y) a 50 ms



De igual forma se muestra en la Figura 20, el diagrama de dispersión de la técnica *bagged trees* que fue la que menor error RMSE obtuvo, al igual que en el eje X el modelo no logra adaptarse a los valores deseados y su valor R cuadrado es de 0.17 por lo que en general para el arreglo cuadrado ninguna de las técnicas parece ser adecuada.

Figura 20

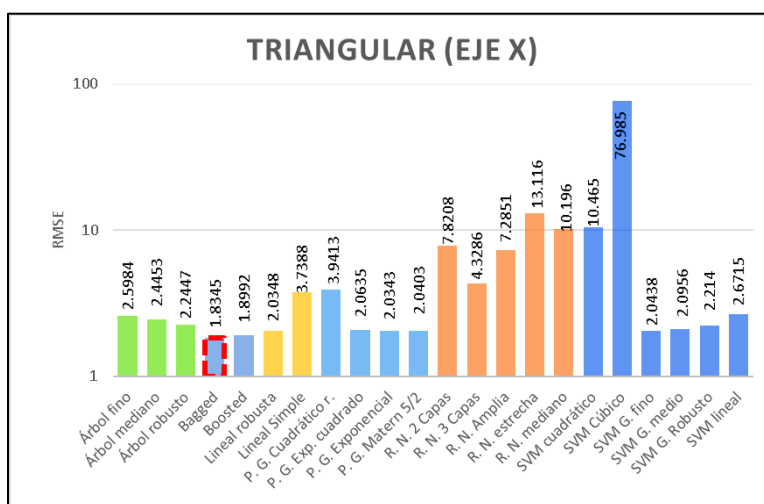
Modelo Cuadrado (eje Y) entrenado con Bagged trees a 50 ms



Para el arreglo triangular, en la Figura 21 se tiene el resumen de los datos obtenidos en el eje X, el comportamiento es parecido al arreglo cuadrado, las redes neuronales y SVM (cúbico) son las peores mientras que *bagged trees* tiene un mejor desempeño.

Figura 21

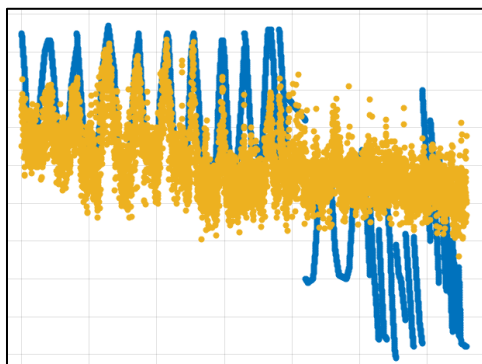
RMSE de técnicas empleadas (Triangular eje X) a 50 ms



En el diagrama de dispersión de la Figura 22 nuevamente los valores no son del todo contundentes, si bien, en ciertos segmentos se observa como el modelo pretende adaptarse a los datos deseados, se observa que en general tampoco logra buenos resultados, su valor R cuadrado es de 0.31, muy similar al arreglo cuadrado en el mismo eje.

Figura 22

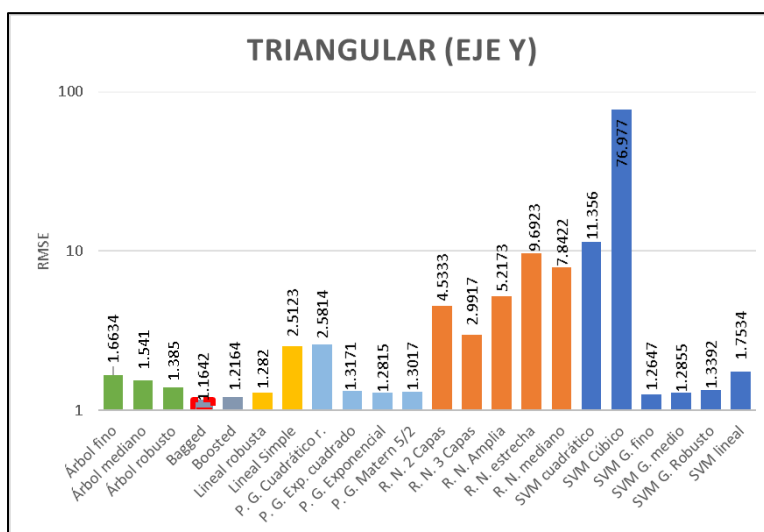
Modelo Triangular (eje X) entrenado con Bagged trees a 50 ms



El último modelo a entrenar es el triangular en el eje Y que se resume en la Figura 23, de la misma forma que en el eje X la técnica SVM (cúbico) presenta los peores resultados, al igual que las redes neuronales, mientras que *bagged trees* se ha convertido en la técnica que se ha mantenido constante como la mejor en cada uno de los modelos propuestos.

Figura 23

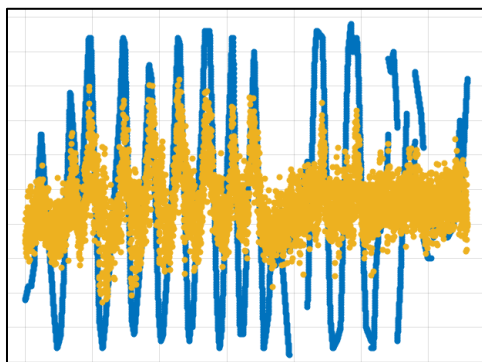
RMSE de técnicas empleadas (Triangular eje Y) a 50 ms



La Figura 24 indica que este modelo logra una mejor adaptación a los valores deseados, pero sigue sin ser prometedor, el valor R cuadrado es 0.24, un valor bajo que indica que el dominio del tiempo no es adecuado para ninguna de las técnicas propuestas.

Figura 24

Modelo Triangular (eje Y) entrenado con Bagged tres a 50 ms

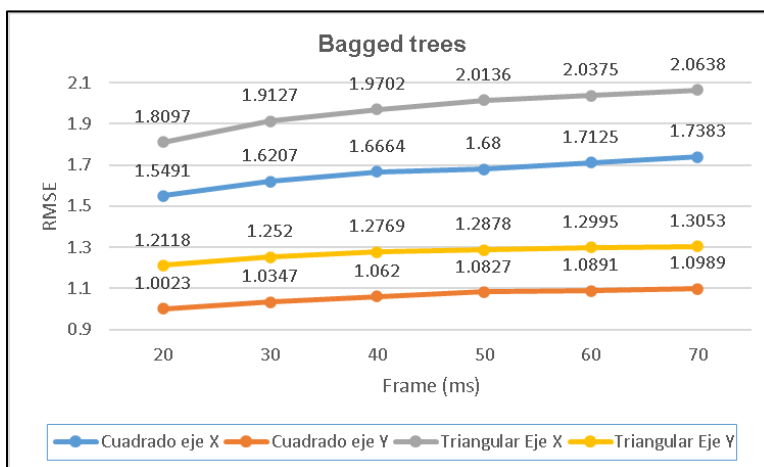


Entrenamientos posteriores (dominio del tiempo)

Con los datos obtenidos, se realizan pruebas con un tamaño de *frame* mayor, las mismas que muestran un error RMSE mayor, por lo cual se procede a disminuir el tamaño de *frame*, con lo cual se observa que el error disminuye, sin embargo, estos datos no son contundentes. En la Figura 25 se muestra este comportamiento para la técnica que demostró mejores resultados en el entrenamiento inicial.

Figura 25

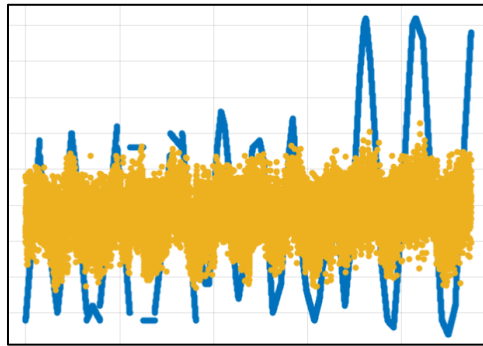
RMSE por variación de *frame* para Bagged tres



De los datos obtenidos, se observa que el modelo con menor error RMSE es el entrenado para el arreglo cuadrado en el eje Y, el diagrama de dispersión se muestra en la Figura 26.

Figura 26

Modelo Cuadrado (eje Y) entrenado con Bagged trees 20 ms



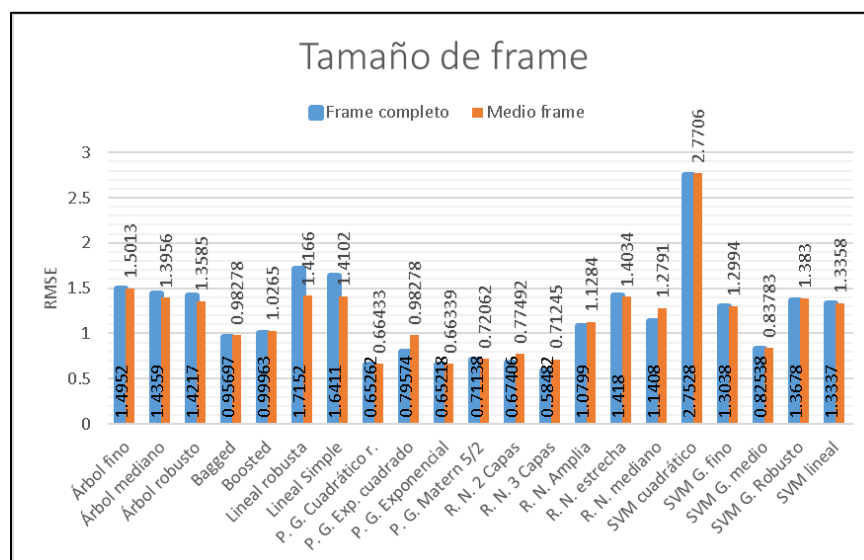
Esto indica que, aunque este modelo es el que presenta menor error al resto, todavía no es un buen candidato ya que no se ajusta a los datos deseados. El resto de técnicas realizadas en el dominio del tiempo presentan un comportamiento similar e indican que los resultados son poco prometedores para la navegación acústica, por ello no se muestran ya que no son relevantes para esta aplicación.

Entrenamiento en el dominio de la frecuencia

Para el entrenamiento en frecuencia primero se establece que el tamaño del *frame* será de la mitad del tamaño usado en el dominio del tiempo, esto se debe a que la FFT genera una imagen espejo del espectro de frecuencia, si esa imagen tiene los mismos valores que la primera mitad, entonces es una redundancia para el entrenamiento, esta hipótesis se comprueba con un preentrenamiento mostrado en la Figura 27.

Figura 27

Comparativa entre tamaños de frame



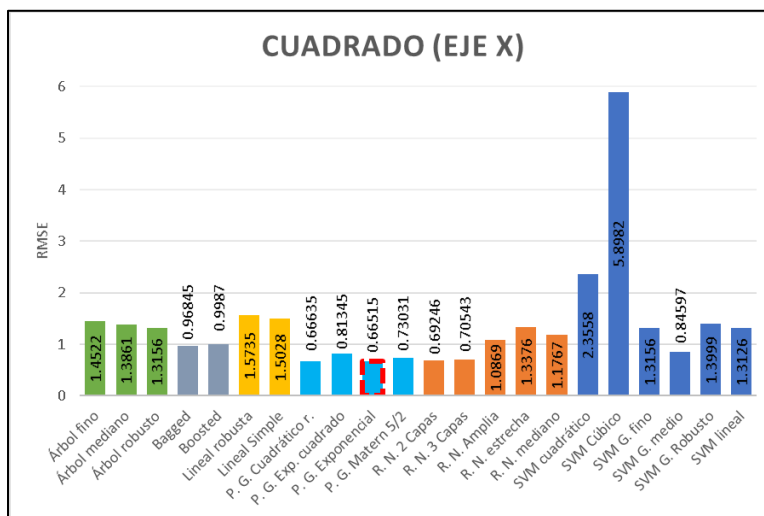
Se puede concluir que el tamaño de *frame* cambia el error RMSE, en la mayoría de técnicas se observa una pequeña disminución del error cuando se utiliza la mitad del *frame*, por lo tanto, eliminar la imagen espejo generada en una FFT es útil ya que no solo disminuye el error, sino que también ayudará al tiempo de entrenamiento, complejidad y tiempo de predicción del modelo. Cabe mencionar que la técnica SVM (cúbica) no se la incluyó ya que presenta un error RMSE de 7.1739 para el *frame* completo y 13.223 para el medio *frame*, ambos valores son elevados y son una anomalía en relación al resto de técnicas.

Entrenamiento inicial (dominio de la frecuencia)

Al igual que en el dominio del tiempo se inicia con un *frame* de 50 ms, al entrenar las diferentes técnicas se observa en la Figura 28 que el error RMSE obtenido en comparación con el mismo modelo en el dominio del tiempo es mejor en todos los modelos a excepción de SVM (cuadrático y cúbico).

Figura 28

RMSE de técnicas en frecuencia (Cuadrado eje X) a 50 ms

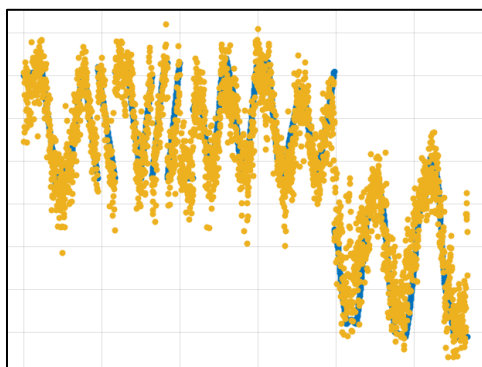


Las técnicas que más destacan son los procesos gaussianos en todas sus configuraciones siendo mejores las que usan el *kernel* exponencial y cuadrático racional.

A continuación, en la Figura 29, se muestra el diagrama de dispersión del modelo con menor error RMSE que es la técnica de procesos gaussianos (exponencial). A diferencia de lo visto en el dominio del tiempo, el modelo mejora considerablemente y se nota claramente que se adapta a los valores deseados, con un valor de R cuadrado de 0.88 mejora sustancialmente e indica que el dominio de la frecuencia es mucho mejor ya que ha conseguido un modelo que se adapta mejor y con un tamaño de *frame* de la mitad en comparación al dominio del tiempo.

Figura 29

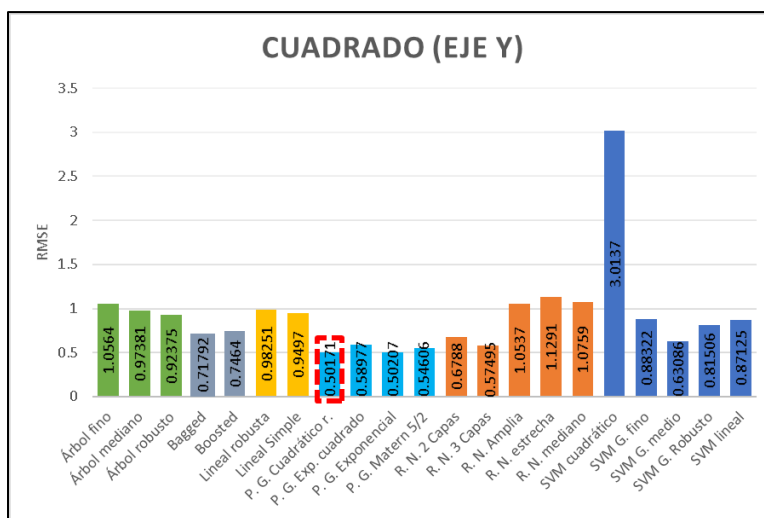
Modelo Cuadrado (eje X) entrenado en frecuencia con P.G. Exponencial a 50 ms



El siguiente entrenamiento se realiza para el eje Y del arreglo cuadrado, de la Figura 30 se nota que las redes neuronales y árboles de regresión no son tan buenos como los procesos gaussianos (exponencial y cuadrático racional) con la diferencia de que la mejor técnica es el *kernel* cuadrático racional, aunque la diferencia sea de solo 0.00036, podría considerarse como resultados iguales.

Figura 30

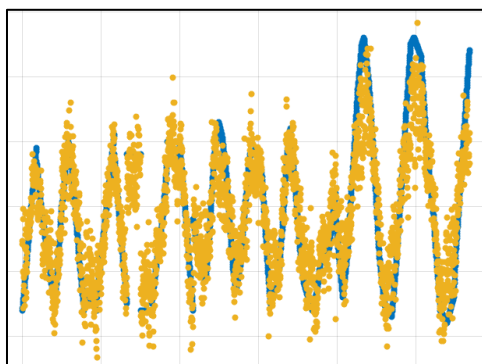
RMSE de técnicas en frecuencia (Cuadrado eje Y) a 50 ms



La Figura 31 indica que también para el eje Y, un análisis en frecuencia muestra mejores resultados, para este caso el valor R cuadrado es de 0.80 lo cual coincide con lo observado ya que el modelo logra adaptarse bien a lo datos requeridos.

Figura 31

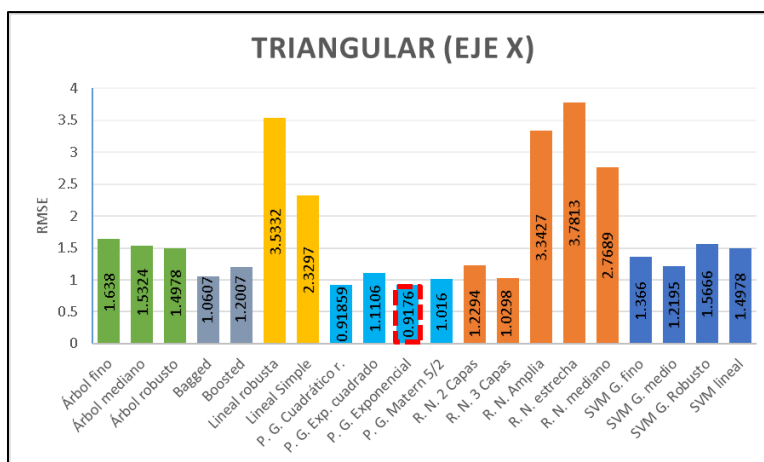
Modelo Cuadrado (eje Y) entrenado con P.G. Cuadrático Racional a 50 ms



A continuación, se realiza el entrenamiento para el arreglo triangular para el eje X, la Figura 32 muestra el mismo patrón que en el arreglo cuadrado, las mejores técnicas son los procesos gaussianos, mientras que las redes neuronales son poco prometedoras, además se observa que *bagged trees* también hace un buen trabajo a diferencia de lo visto en el arreglo cuadrado.

Figura 32

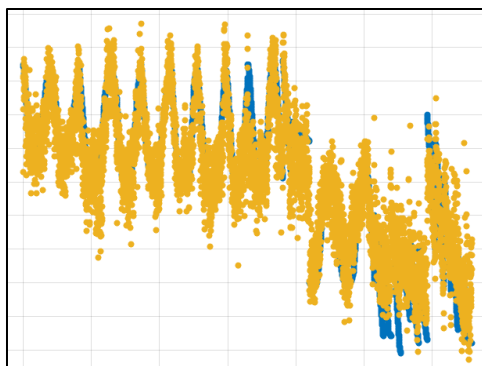
RMSE de técnicas en frecuencia (Triangular eje X) a 50 ms



Cuando se muestra el diagrama de dispersión en la Figura 33 es evidente que este modelo mejora considerablemente en relación al dominio del tiempo, sin embargo, se observa ciertas secciones donde no predice adecuadamente.

Figura 33

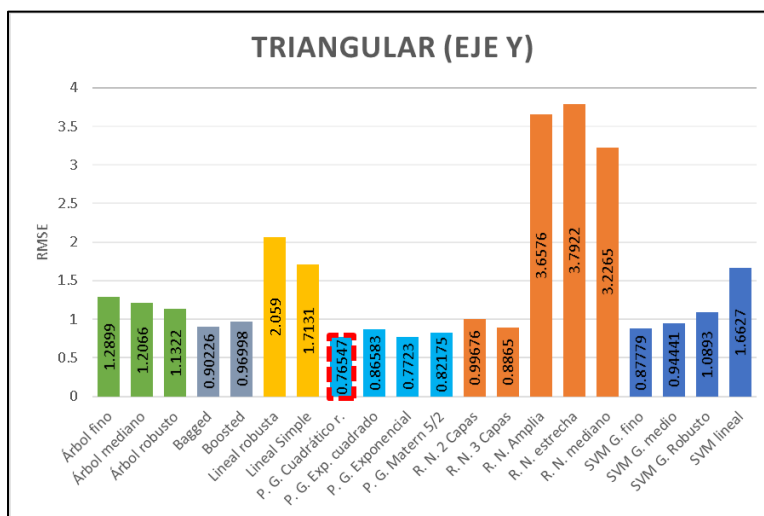
Modelo Triangular (eje X) entrenado con P.G. Exponencial a 50 ms



Finalmente, para comprobar que el dominio del tiempo presenta mejores resultados, se entrena el último modelo triangular en el eje Y, de la Figura 34 se tiene como la mejor técnica a los procesos gaussianos, lo que indica que tienen un potencial mayor para este tipo de aplicaciones.

Figura 34

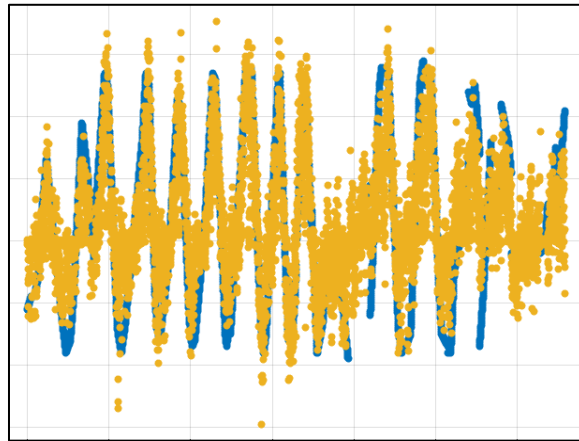
RMSE de técnicas en frecuencia (Triangular eje Y) a 50 ms



De la Figura 35 se tiene la representación de procesos gaussianos con *kernel* cuadrático racional, un valor de R cuadrado de 0.58 también indica mejoría y por lo tanto los modelos realizados en el dominio del tiempo son indicados para aplicaciones de posicionamiento subacuático.

Figura 35

Modelo Triangular (eje Y) entrenado con P.G. Cuadrático Racional a 50 ms



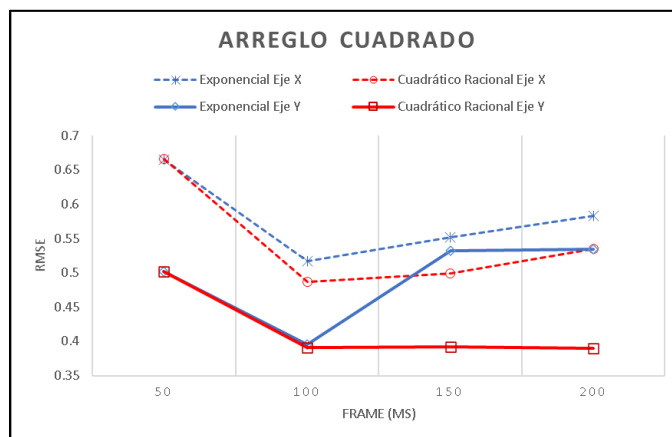
Entrenamientos posteriores (dominio de la frecuencia)

De forma resumida, el dominio de la frecuencia ha demostrado tener un mejor desempeño que el dominio del tiempo, además se observa que las técnicas con mejor rendimiento son los procesos gaussianos con *kernel* exponencial y cuadrático racional. Considerando que los procesos gaussianos son los que mejor resultados proveen, los posteriores entrenamientos se realizarán solo con estas técnicas ya que al no saber con qué configuración se obtendrá el punto óptimo para el modelo, se tendrá que hacer múltiples pruebas.

Para visualizar mejor el rendimiento de las técnicas con la variación del tamaño de *frame*, en la Figura 36 se muestra los datos obtenidos para el arreglo cuadrado. Para determinar cuál es el punto óptimo primero se establece que lo primordial es obtener un modelo con el menor error posible, por lo observado, se puede concluir que de forma general el arreglo cuadrado en el eje X presenta un mayor error que en el eje Y, por lo tanto, se determina que el eje X debe ser la base para determinar cuál es el punto idóneo, ya que la prioridad es mitigar el modelo con el mayor error.

Figura 36

RMSE variando el tamaño de frame (Arreglo cuadrado)

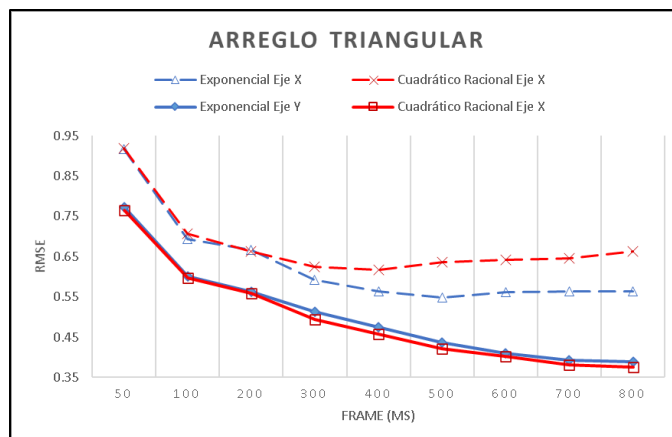


Con esta consideración se tiene que el punto ideal es con un tamaño de *frame* de 100 *ms* usando la técnica de procesos gaussianos (cuadrático racional), igualmente se evidencia que en el modelo (eje Y) los procesos gaussianos tienen un rendimiento similar en este punto.

La Figura 37 es la representación del arreglo triangular, al igual que para el arreglo cuadrado se utiliza el mismo criterio para determinar el punto ideal, siendo el eje X el que se considera como base para determinar el punto óptimo ya que presenta mayor error. En esta ocasión la técnica de procesos gaussianos (exponencial) se muestra como la mejor opción y el tamaño de *frame* se establece en 500 *ms* que es el punto donde el eje X presenta menor error.

Figura 37

RMSE variando el tamaño de frame (Arreglo triangular)

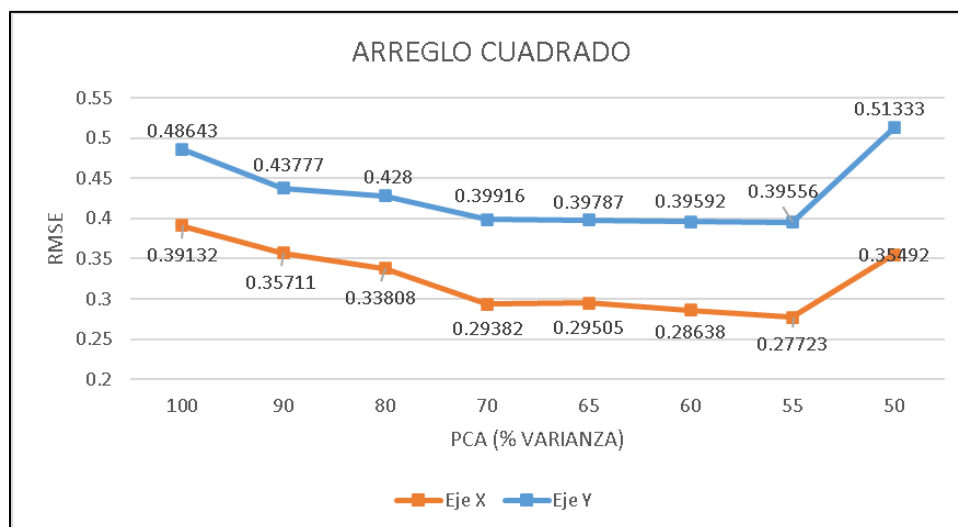


Una vez determinado los puntos ideales para cada arreglo, se procede a realizar una última serie de pruebas enfocadas en determinar si el análisis de componentes principales PCA es una herramienta que puede aportar en el mejoramiento de esta aplicación, para lo cual se realiza diversas pruebas con los puntos determinados como óptimos y se varía la cantidad de varianza explicada.

Al variar los parámetros de varianza explicada se comprueba que no necesariamente utilizar todas las variables ayuda a mejorar al modelo entrenado y que estas pueden afectar al rendimiento del modelo, para el arreglo cuadrado se tiene el resumen de las pruebas realizadas en la Figura 38, de lo observado se puede concluir que los resultados mejoraron hasta llegar a un PCA de 55% de varianza explicada, una vez pasado este umbral el modelo empeora considerablemente.

Figura 38

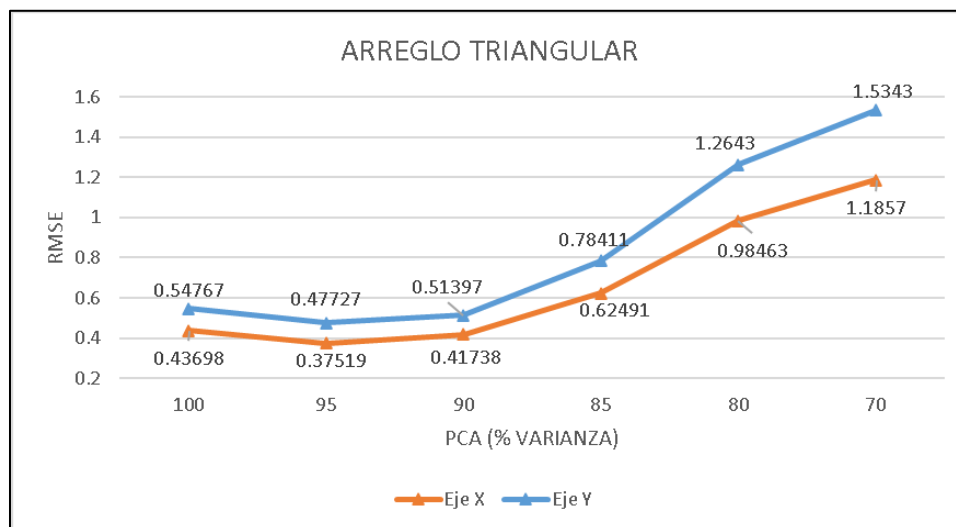
RMSE con variación de PCA (arreglo cuadrado)



A diferencia del arreglo cuadrado, el arreglo triangular que se muestra en la Figura 39 indica que también se tiene una mejora cuando se utiliza un PCA, no obstante, el porcentaje de varianza explicada ideal es de 95%.

Figura 39

RMSE con variación de PCA (arreglo triangular)



Con estos resultados se determina que una reducción de componentes con PCA puede ayudar a mejorar la precisión del sistema, no solo eso, sino que también reduce el tiempo de entrenamiento, su complejidad y el tiempo de predicción. En la Tabla 3 se indica los parámetros que se establecieron como los ideales para obtener el mejor modelo para cada uno de los arreglos de hidrófonos.

Tabla 3

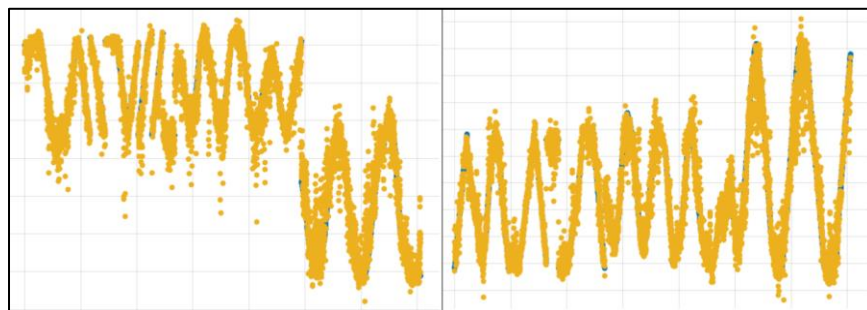
Configuración final para cada arreglo

Arreglo	Frame (ms)	Técnica	PCA (% varianza)
Cuadrado	100	Procesos gaussianos (cuadrático racional)	55
Triangular	500	Procesos gaussianos (exponencial)	95

Finalmente, para entender mejor el modelo final, se presenta en la Figura 40 el diagrama de dispersión del modelo cuadrado con la configuración final, se nota que para la gran mayoría de predicciones estas se acercan a los datos que se pretende obtener y de forma general se puede definir que este modelo si cumple con los requerimientos para esta aplicación.

Figura 40

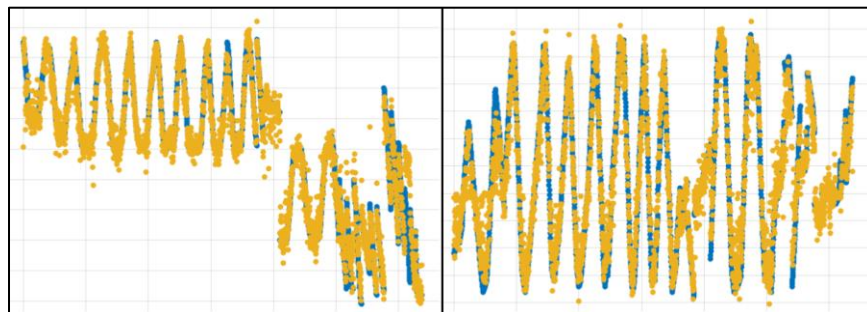
Diagrama de dispersión Arreglo Cuadrado eje X (izq) y eje Y (der)



Lo mismo ocurre con el arreglo triangular de la Figura 41, los valores se aproximan a las etiquetas definidas para el entrenamiento mostrando que puede ser útil para el posicionamiento acuático.

Figura 41

Diagrama de dispersión Arreglo Triangular eje X (izq) y eje Y (der)

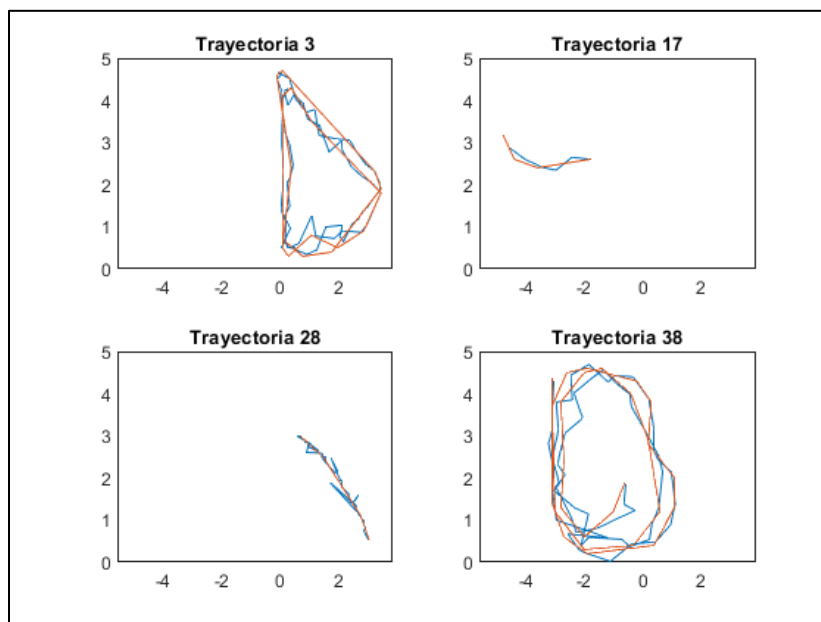


Comparación

Se propone utilizar estos modelos con datos que no han sido utilizados, para ello se utiliza la base de datos utilizada en el entrenamiento, con un solapamiento diferente que no sea múltiplo del utilizado en el entrenamiento, así los datos obtenidos serán ligeramente diferentes y por lo tanto pueden considerarse diferentes. En el entrenamiento se usó un solapamiento de 80%, es decir que la ventana avanza un 20% de su predecesora, para este caso se utiliza un solapamiento del 1%, es decir que la ventana avanza un 99%, se utiliza estos datos para ingresar en el modelo entrenado y obtener las coordenadas, esto se realiza con todas las trayectorias. En la Figura 42 se muestra algunas de las trayectorias obtenidas del modelo.

Figura 42

Trayectorias predichas por los modelos propuestos



De estas trayectorias obtenidas se puede notar que los modelos logran acercarse considerablemente a los valores usados para el entrenamiento, esto es similar para todas las trayectorias utilizadas, algo a resaltar es que las trayectorias más largas son las que presentan un mayor error, posiblemente por la segmentación en tramos, segmentar las trayectorias en tramos más pequeños puede ayudar a tener una representación más fiable del posicionamiento del vehículo.

Cada arreglo de hidrófono tiene un modelo para cada el eje X y otro para el eje Y, es decir, se presenta un error RMSE por cada eje, no obstante, es importante determinar el error bidimensional, para lo cual se utiliza el concepto de distancia euclidiana, mostrando la ecuación (7) para error de raíz cuadrada media.

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n ((x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2)} \quad (7)$$

Este error bidimensional se utiliza con los datos obtenidos por los modelos y comparados con las etiquetas, la Tabla 4 muestra el error obtenido en cada arreglo.

Tabla 4*Error cuadrático medio bidimensional*

Arreglo de hidrófonos	RMSE
Cuadrado	0.272
Triangular	0.4534

Como punto final se concluye que el modelo entrenado para el arreglo cuadrado genera mejores resultados que el arreglo triangular.

Conclusiones

Los resultados obtenidos a lo largo de esta serie de entrenamiento han logrado determinar que el dominio de la frecuencia presenta un mayor potencial para aplicaciones de posicionamiento acuático mediante *machine learning* en señales acústicas, en el dominio del tiempo ninguno de los modelos se adapta correctamente y ninguna de las técnicas ni tamaño de *frame* ha permitido encontrar un modelo que sea aceptable.

Gracias al preprocesamiento de señales, la cantidad de información utilizada para el entrenamiento se vio reducida considerablemente conservando la información más relevante que puede ser utilizada para encontrar un modelo que sea más eficiente. El análisis de componentes principales es una herramienta que permite mejorar significativamente ciertas aplicaciones de *Machine Learning*, para esta aplicación en específico, se pudo evidenciar que una reducción de componentes basado en el porcentaje de varianza explicada logró reducir no solo el error del modelo, sino también el tiempo de entrenamiento, complejidad del sistema y tiempo de predicción del modelo empleado.

De las diversas técnicas empleadas, las que mejores resultados entregaron son los procesos gaussianos en todas sus configuraciones, resaltando especialmente a los que utilizan *kernel* exponencial y cuadrático racional.

De forma general se ha desarrollado un sistema que es capaz de predecir el posicionamiento del vehículo en un entorno controlado que, para fines de investigación ha entregado resultados que son prometedores para futuras investigaciones que profundicen y optimicen los parámetros empleados.

Trabajos futuros

El presente estudio demostró que las técnicas basadas en *machine learning* en el dominio de la frecuencia son aplicaciones potenciales que logran predecir el posicionamiento del vehículo con buenos resultados, el siguiente paso es la implementación de un sistema con estas características en una aplicación real.

La aplicación de este sistema en entornos que tengan mayor presencia de ruido puede ayudar a desarrollar sistemas más robustos en contra de variaciones propias de entornos subacuáticos.

Otra línea de investigación es la inclusión de un sistema tridimensional que considere el eje Z o profundidad para lo cual es necesario un escenario con mayor profundidad en donde realizar las pruebas y buscar una forma de determinar de forma precisa el posicionamiento para la etiqueta de las muestras.

Como ya se mencionó en los tipos de navegación acústica, la distancia entre los elementos utilizados para el posicionamiento afecta directamente en la precisión de los sistemas, a mayor distancia mayor es su precisión, un buen punto de partida es determinar si esto también se aplica en sistemas entrenados con *machine learning*, establecer un arreglo y variar la distancia entre los hidrófonos permitirá establecer si este comportamiento también se cumple para este tipo de aplicaciones y si estas mejoras justifican un arreglo de hidrófonos más grande.

Referencias

- Aguirre Martín, F. (2017). *Desarrollo y análisis de clasificadores de señales de audio*. Gandia: Universidad Politécnica de Valencia.
- Aldana Afanador, A. F., & Esteban Villegas, H. S. (2018). *Diseño y validación de un sistema de control de velocidad para un vehículo subacuático remotamente operado*. Bucaramanga: Universidad Autónoma de Bucaramanga.
- Aldaz Manzano, J., & Uriel Jimenez, E. (2017). *Análisis multivariante aplicado con R* (Segunda ed.). Madrid: Ediciones Paraninfo.
- Aparicio Sosa, J. (2014). *Modelo de propagación acústica subacuática y su aplicación en sistemas de posicionamiento*. Alcalá de Henares: Universidad de Alcalá.
- Arango Rivera, V. (2015). *Sistemas de posicionamiento dinámico para vehículos subacuáticos operados remotamente*. Medellín: Universidad Pontificia Bolivariana.
- Balaguer Beser, A., & Ruiz Fernández, L. (2021). *Selección de un modelo de regresión lineal múltiple para el cálculo de la precipitación media en verano*. Valencia: Universidad Politécnica de Valencia.
- Bejarano Cedeño, S. F., & Vera Cedeño, J. M. (2019). *Construcción de un vehículo subacuático no tripulado autónomo para posicionamiento especial mediante GPS, usando cable umbilical y una boya para la comunicación*. Riobamba: Escuela Superior Politécnica de Chimborazo.
- Campo León, E. (2017). *Introducción a las máquinas de vector soporte (SVM) en aprendizaje supervisado*. Zaragoza: Universidad Zaragoza.
- Carrera Erazo, E. V., & Guevara Bustillos, D. E. (2022). *Passive positioning of autonomous underwater vehicles* (Vol. 980). Quito: Springer. doi:10.1007/978-3-031-18279-5

- Carrera Erazo, E. V., & Paredes, M. (2019). Analysis and evaluation of the positioning of autonomous underwater vehicles using acoustic signals. *Systems and Technologies (SIST series)*. 152, págs. 411-421. Springer. doi:10.1007/978-981-13-9155-2 33
- Carrera Erazo, E. V., Mena, E., Arciniega, P., Padilla, A., & Paredes, M. (2020). Analysis and Evaluation of sound-based positioning techniques for short distances. *In Proceedings of the Technical and Scientific Conference of the Andean Council of the IEEE (Andescon'20)*, (págs. 1-6). Quito. doi:10.1109/andescon50619.2020.9272155
- Castillo, A. R., Castro, F. C., & Vogulys, J. J. (2020). *Implementación y simulación de un algoritmo de posicionamiento articular para un robot planar continuo utilizando técnicas de Inteligencia Artificial*. Bogotá: Universidad Militar Nueva Granada.
- Cristianini, N., & Scholkopf, B. (2002). Support vector machines and kernel methods: the new generation of learning machines. *Ai Magazine*, 23(3), 31-31.
- Díaz García, P. J. (2020). *Planificación de movimiento de vehículos acuáticos de superficie basado en Optimización Bayesiana*. Sevilla: Universidad de Sevilla.
- Díaz, J. F., & Correa, J. C. (2013). *Comparación entre árboles de regresión CART y regresión lineal*. Medellín: Universidad Santo Tomás.
- Faraway, J. J. (2004). *Linear models with R*. New York: Chapman and Hall/CRC.
- García Álvarez, D., & Fuente, M. (2011). Estudio comparativo de técnicas de detección de fallos basadas en el Análisis de Componentes Principales (PCA). *Revista Iberoamericana de Automática e Informática industrial*, 182-195.
- George, A., & Vidyapeetham, A. V. (2012). Anomaly Detection based on Machine Learning: Dimensionality Reduction using PCA and Classification using SVM. *International Journal of Computer Applications*, 47(21), 5-8.

- Ghanbari, A., Naghavi, A., Ghaderi, S. F., & Sabaghian, M. (2009). *Artificial Neural Networks and Regression Approaches Comparison for Forecasting Iran's Annual Electricity Load*. Tehran: University of Tehran.
- Hastie, T., Friedman, J., & Tibshirani, R. (2009). *The elements of statistical learning: data mining, inference, and prediction* (Vol. II). New York: Springer.
- Kelleher, J., Mac Namee, B., & D'Arcy, A. (2020). *Fundamentals of machine learning for predictive data analytics: algorithms, worked examples, and case studies*. Cambridge: MIT press.
- Komari, H., & Farsi, H. (2018). Passive Sonar Target Detection Using Statistical Classifier and Adaptive Threshold. *Applied Sciences*, 61.
- LaPointe, C. E. (2006). *Virtual Long Baseline (VLBL) Autonomous Underwater Vehicle Navigation Using a Single Transponder*. Cambridge: Massachusetts Institute of Technology.
- Leonard, J. J., & Bahr, A. (2016). Autonomous Underwater Vehicle Navigation. *Springer Handbook of Ocean Engineering*, 341–358.
- Madisetti, V., & Williams, D. (2010). *The Digital Signal Processing Handbook* (Segunda ed.). Boca Raton: CRC press.
- Martínez, S. R. (2020). *Sistema en Matlab para la segmentación y clasificación automática de sonidos de ballenas*. Madrid: Universidad Politécnica de Madrid.
- Moral Peláez, I. (2016). Modelos de regresión: lineal simple y regresión logística. *Revista Seden*, 14, 195-214.
- Nielsen, R. (1991). *Sonar signal processing*. Boston: Artech House.

- Nieto Jeux, A. (2021). *Algoritmos de Aprendizaje Automático. Un Estudio de su Difusión y Utilización*. Madrid: Universidad Politécnica de Madrid.
- Ochoa, L. L. (2019). *Evaluación de Algoritmos de Clasificación utilizando Validación Cruzada*. Arequipa: Universidad Nacional de San Agustín de Arequipa.
- Palacios García, A. C. (2007). *Prototipo de robot acuático explorador*. Puebla de zaragoza: Benemérita Universidad Autónoma de Puebla.
- Pareja Contreras, J. M. (2017). *Eco-localización por medio del uso de sub-muestreo y señales multi-tono en frecuencias ultrasónicas*. Arequipa: Universidad Católica San Pablo.
- Perez, D., Gómez, L., & Franco, R. (2017). Algoritmo para la detección, clasificación y localización de cualquier tipo de embarcación utilizando un hidrófono. *Investigación Formativa en Ingeniería*, 192 - 199.
- Rasmussen, C. E., & Williams, C. (2006). *Gaussian Processes for Machine Learning*. Cambridge: Massachusetts Institute of Technology.
- Salvo Gutierrez, D. (2020). *Clasificador de sonidos urbanos mediante redes neuronales en dispositivos de bajo coste*. Valencia: Universidad Politécnica de Valencia.
- Sánchez Berral, R. (2017). *Análisis de Algoritmo de Regresión Multiobjetivo*. Madrid: Universidad Autónoma de Madrid.
- Sánchez Hernández, R. (2016). *Algoritmo de Navegación Integrada INS/GPS para vehículos autónomos subacuáticos*. La Habana: Universidad de las Ciencias Informáticas.
- Vilarnau, N. H. (2014). *Forward-looking sonar mosaicing for underwater environments*. Girona: Universidad de Girona.

Yalta, N., Nakadai, K., & Ogata, T. (2017). Sound Source Localization Using Deep Learning Models. *Journal of Robotics and Mechatronics*, 37-48.

Yang, H., Lee, K., Kim, K., & Choo, Y. (2020). Underwater Acoustic Research Trends with Machine Learning: Passive SONAR Applications. *Journal of Ocean Engineering and Technology*, 227-236.