

Universidad de Huelva

Departamento de Ciencias Agroforestales



Diseño de un sistema integral de monitorización de la actividad pesquera

Memoria para optar al grado de doctora
presentada por:

María del Mar Galotto Tébar

Fecha de lectura: 1 de julio de 2022

Bajo la dirección de los doctores:

Juan Carlos Gutiérrez Estrada

Ivone Alejandra Czerwinski

Huelva, 2022





DISEÑO DE UN SISTEMA INTEGRAL DE MONITORIZACIÓN DE LA ACTIVIDAD PESQUERA

MEMORIA DE TESIS DOCTORAL

María del Mar Galotto Tébar

DIRIGIDA POR

Dr. Juan Carlos Gutiérrez Estrada y Dra. Ivone Alejandra Czerwinski

PROGRAMA DE DOCTORADO

Ciencia y Tecnología Industrial y Ambiental

UNIVERSIDAD DE HUELVA

Huelva, febrero de 2022



Los doctores Juan Carlos Gutiérrez Estrada, Profesor Titular de Universidad del Departamento de Ciencias Agroforestales de la Universidad de Huelva, e Ivone Alejandra Czerwinski, Titulada Superior del Instituto Español de Oceanografía - CSIC

HACEN CONSTAR

Que la presente memoria titulada “DISEÑO DE UN SISTEMA INTEGRAL DE MONITORIZACIÓN DE LA ACTIVIDAD PESQUERA”, presentada por María del Mar Galotto Tébar para optar al grado de doctor, ha sido realizada bajo su tutorización y dirección, respectivamente, y, dada la idoneidad del trabajo, cumple las condiciones exigidas para su presentación, por lo que autorizan su lectura y defensa.

Para que así conste a los efectos oportunos, firman la presente en Huelva a 1 de febrero de 2022.

Fdo.: Dr. Juan Carlos Gutiérrez Estrada

Fdo.: Dra. Ivone Alejandra Czerwinski

*Dpto. Ciencias Agroforestales, Escuela
Técnica Superior de Ingeniería, Universidad
de Huelva*

Titulada Superior del IEO-CSIC

A mi incondicional familia

Agradecimientos

Mi más sincero agradecimiento a:

Instituto Español de Oceanografía de Murcia y en especial a Antonio Esteban Acón, técnico I+D+I del IEO de Murcia y Jefe de Campaña del programa MEDITS-GSA6. Así mismo, deseo extender mi agradecimiento al Capitán y tripulación del buque Miguel Oliver, así como a los investigadores que participaron en la campaña MEDITS-GSA6, porque gracias a la colaboración y disposición de todos ellos, fue posible la realización de experimentos y recopilación de la información fundamental para la elaboración de esta tesis.

La empresa BQ por proporcionar gratuitamente algunos de los dispositivos móviles con los que se realizaron los experimentos, en especial a la Srta. Marta Vergel del departamento de marketing de BQ.

La Universidad de Huelva, por permitirme acceder a su Programa Oficial de Doctorado en Ciencia y Tecnología Industrial y Ambiental, en la línea de investigación Ciencia y tecnología del medio ambiente y a los coordinadores del programa doctorado.

La Universidad Miguel Hernández donde desarrollo mi labor docente y realicé los estudios previos que me permitieron acceder al programa de doctorado.

A mi tutor y director de Tesis Dr. Juan Carlos Gutiérrez Estrada, profesor Titular de Universidad del Departamento de Ciencias Agroforestales de la Universidad de Huelva, tengo que agradecerle de manera especial por aceptarme para realizar esta tesis doctoral bajo su dirección.

A mi directora de Tesis Dra. Ivone Alejandra Czerwinski Kulikowska, Titulada Superior del IEO_CSIC, le agradezco sinceramente su aportación del tema de investigación, su ánimo y empuje para que iniciara este trabajo y su apoyo durante el largo tiempo transcurrido desde entonces.

A ambos directores de Tesis tengo que agradecer profundamente y de manera especial su paciencia, disponibilidad y generosidad para compartir su experiencia y amplio conocimiento sobre el sector pesquero. Por sus siempre oportunas e inteligentes observaciones que han contribuido a dar forma a este proyecto. Su apoyo y confianza ha sido en las ocasiones de mayor desánimo las que me han impulsado a seguir adelante. Y, sobre todo, por estar siempre disponibles en la distancia, fuera de horarios

y en festivos, por todo ello, gracias por ser grandes profesionales y sobre todo por ser grandes personas.

Al profesor Dr. Alejandro Pomares Padilla, Catedrático de Escuela Universitaria del Departamento Ingeniería de Computadores de la Universidad Miguel Hernández, por su dedicación y apoyo incondicional durante la elaboración de la presente investigación. Ha sido un estímulo constante en mi trabajo y ejemplo de disciplina y esfuerzo.

Al Dr. César Fernández Peris y a la Dra. M^a Asunción Vicente Ripoll, profesores Titulares de Universidad de la Universidad Miguel Hernández, por su orientación y consejos en el desarrollo de la aplicación móvil.

A mi familia, por su infinita paciencia, por el tiempo que no les he dedicado y por darme ánimo durante este proceso.

Finalmente, a todos los colegas y amigos por su amistad, consejos, apoyo, ánimo, tiempo e información para el logro de mis objetivos.

Resumen

La gestión de la pesca se basa generalmente en la regulación del esfuerzo pesquero, limitando la capacidad y la actividad pesquera. Conocer la actividad de los buques pesqueros con precisión y en tiempo real supone un salto de calidad en la gestión de la actividad pesquera. La capacidad de pesca puede cuantificarse objetivamente, sin embargo, el cálculo de la actividad pesquera requiere conocer el tiempo efectivo de pesca, para lo cual es imprescindible el seguimiento de la actividad de los buques.

Los actuales sistemas de seguimiento de buques no proporcionan información suficiente para determinar con precisión la actividad pesquera, tienen un coste tan alto que dejan fuera de control casi la totalidad de la flota artesanal, proporcionan únicamente la información que registra el GPS instalado en la embarcación, la frecuencia de envío de información es muy baja y no envía información entre muestras.

Esta tesis presenta el desarrollo de un nuevo sistema integral de seguimiento de la actividad pesquera que puede complementar y superar las limitaciones de los actuales sistemas de seguimiento. Para evaluar con mayor precisión la actividad pesquera, se propone incorporar nuevos sensores en los buques que aporten información adicional. El sistema propuesto está desarrollado sobre la base de un dispositivo móvil de bajo coste con los sensores GPS, acelerómetro, giroscopio y sensor de campo magnético; además dispone de capacidad de procesamiento que permite incorporar algoritmos de inteligencia artificial para identificar en tiempo real cuándo el barco está pescando.

En este trabajo se evalúa la capacidad de los sensores integrados en los dispositivos móviles actuales de bajo coste para detectar cambios significativos en el comportamiento estático y dinámico de la embarcación durante la actividad pesquera de arrastre. Se analiza la capacidad de diferentes métodos estadísticos y heurísticos para clasificar e identificar la fase del lance en la que se encuentra el buque durante su actividad pesquera. Así mismo, se identifica la combinación de sensores y la estructura del lance que ofrece mejor respuesta en la tarea de clasificación de las máquinas de aprendizaje.

Los resultados obtenidos indican que, en general, las técnicas heurísticas tienen un alto grado de discriminación de cada una de las fases de la operación de pesca y que, en particular, el perceptrón multicapa (MLP) es capaz de identificar correctamente el 96,3% de las muestras de la fase de arrastre utilizando únicamente los sensores GPS y giróscopo.

Abstract

Fisheries management is generally based on the regulation of fishing effort, limiting fishing capacity and fishing activity. Knowing the activity of fishing vessels accurately and in real time represents a leap in quality in the management of fishing activity. Fishing capacity can be objectively quantified, however, the calculation of fishing activity requires knowledge of the effective fishing time, for which the monitoring of vessel activity is essential.

Current vessel monitoring systems do not provide sufficient information to accurately determine fishing activity, they are so expensive that they leave almost the entire artisanal fleet out of control, they only provide the information recorded by the GPS installed on the vessel, the frequency of sending information is very low and they do not send information between samples.

This thesis presents the development of a new integrated system for monitoring fishing activity that can complement and overcome the limitations of current monitoring systems. In order to assess fishing activity more accurately, it is proposed to incorporate new sensors on vessels to provide additional information. The proposed system is developed on the basis of a low-cost mobile device with GPS, accelerometer, gyroscope and magnetic field sensors; it also has processing capacity that allows the incorporation of artificial intelligence algorithms to identify in real time when the vessel is fishing.

This work evaluates the capacity of sensors integrated in current low-cost mobile devices to detect significant changes in the static and dynamic behaviour of the vessel during trawling activity. The ability of different statistical and heuristic methods to classify and identify the phase of the haul in which the vessel is during its fishing activity is analysed. It also identifies the combination of sensors and the set structure that provides the best response in the classification task of the learning machines.

The results obtained indicate that, in general, the heuristic techniques have a high degree of discrimination of each of the phases of the fishing operation and that, in particular, the multilayer perceptron (MLP) is capable of correctly identifying 96.3% of the trawl phase samples using only the GPS and gyroscope sensors.

Índice general

1	Introducción	15
1.1	Antecedentes	15
1.1.1	Sistema de seguimiento de buques (VMS)	18
1.1.2	Sistema de localización y seguimiento de embarcaciones pesqueras andaluzas (SLSEPA)	22
1.1.3	Sistema de identificación automática (AIS)	23
1.1.4	Otras iniciativas	25
1.1.5	La actividad pesquera actual	26
1.2	Hipótesis	29
1.3	Objetivos	29
2	Material y método	30
2.1	El dispositivo móvil y los sensores	30
2.2	La aplicación móvil de recogida de datos	32
2.3	El paquete estadístico R	35
2.4	Caso de estudio	36
2.5	Procesado de datos y creación de las tablas de datos	40
2.6	Análisis de datos	41
2.6.1	Análisis descriptivo de los datos	41
2.6.2	Análisis multivariante	43
2.7	Técnicas de aprendizaje automático	47
2.7.1	Análisis Discriminante Lineal (LDA)	47
2.7.2	Redes neuronales artificiales	49
2.7.3	Máquina de Soporte Vectorial (SVM)	60
2.8	Métricas de evaluación	62
2.9	Calibración del modelo	63
2.10	Evaluación del modelo	65
2.11	Simplificación del modelo	65
3	Resultados	67
3.1	Análisis de datos	67
3.1.1	Análisis descriptivo	67
3.1.2	Análisis multivariante	75
3.2	Técnicas de aprendizaje automático	77
3.2.1	Calibración del modelo	77
3.2.2	Evaluación del modelo	80

3.2.3 Simplificación del modelo	83
4 Discusión.....	86
4.1 Dinámica del buque	87
4.1.1 Escora longitudinal.....	87
4.1.2 Cabeceo y balance	89
4.1.3 Guiñada	91
4.1.4 Avance	92
4.2 Máquinas de aprendizaje	93
5 Conclusiones	98
6 Referencias	100
7 Anexos	113
7.1 Códigos fuente.....	113
7.1.1 La aplicación móvil de recogida de datos en Android	113
7.1.2 Procesado de datos en R	122
7.1.3 Análisis descriptivo de los datos en R.....	124
7.1.4 Análisis multivariante.....	127
7.1.5 Máquinas de aprendizaje.....	128
7.1.6 Calibración del modelo	130
7.1.7 Métricas de evaluación	131
7.1.8 Validación cruzada	132
7.1.9 Evaluación del modelo.....	133
7.1.10 Simplificación del modelo.....	134
7.2 Artículos	137

Índice de figuras

Figura 1. Ejes de coordenadas del dispositivo móvil utilizado por los sensores para registrar la información del vector aceleración, velocidad de giro alrededor de cada eje y vector de campo magnético.	31
Figura 2. Interfaz de usuario de la aplicación móvil y flujo de información entre el dispositivo móvil, la base de datos local SQLite y la base de datos MySQL del servidor, y entre RStudio y la base de datos MySQL del servidor.	33
Figura 3. Estructura de la base de datos relacional formada por las tablas acelerómetro, giróscopo, campo_magnético y gps, que almacenan los datos registrados por los sensores; la tabla lance que almacena los lances realizados; la tabla actividad con las fases en que se divide el lance, la tabla barco con la información de las embarcaciones; la tabla dispositivo con la información de los dispositivos móviles; la tabla arte_pesca con los aparejos utilizados y la tabla evento que relaciona cada dato registrado por los sensores con el lance y la fase del lance.	34
Figura 4. Buque oceanográfico Miguel Oliver	36
Figura 5. Arte de pesca de arrastre GOC73 utilizada en la campaña MEDITS. (Carpentieri et al., 2020).....	37
Figura 6. Maniobra de largado de arte de pesca de arrastre en el buque Miguel Oliver durante la campaña MEDITS del año 2016.....	37
Figura 7. Ubicación del dispositivo móvil en el puente del buque oceanográfico. Las flechas de color rojo muestran los ejes de referencia de los datos vectoriales proporcionados por los sensores del dispositivo.....	38
Figura 8. Distribución espacial de los 22 lances registrados a lo largo de las costas de Castellón, Tarragona, Barcelona y Gerona (España)	39
Figura 9. Grados de libertad de movimiento del buque alrededor y a lo largo de su eje longitudinal (balance y avance), transversal (cabeceo y deriva) y vertical (guiñada y arfada).....	41
Figura 10. Representación gráfica de las muestras originales en el plano X_1 - X_2 y de las componentes principales Z_1 y Z_2	45
Figura 11. Neurona artificial. x_i representan la i -ésima entrada de la neurona, w_i representa el peso asociado a la entrada i -ésima, Σ representa el sumatorio del	

producto de entradas y pesos, θ representa el sesgo de la neurona y f la función de activación que proporciona la salida de la neurona.....50

Figura 12. Arquitectura del perceptrón multicapa formada por tipos de capas: la capa de entrada que recibe la información de entrada del perceptrón y con una neurona por cada atributo del patrón de entrada i , las capas ocultas que procesan la información con un número variable de capas y neuronas en cada capa, y la capa de salida que proporciona la respuesta del perceptrón con un número de neuronas igual al de atributos del vector de salida o51

Figura 13. Funciones de activación.53

Figura 14. Arquitectura de la red neuronal probabilística formada por cuatro capas: capa de entrada con un número de neuronas igual al de atributos de entrada, una primera capa oculta de patrones con un número de neuronas igual al de muestras de entrenamiento, una segunda capa oculta sumatoria con un número de neuronas igual al número de clases y una capa de salida con una neurona que proporciona el resultado de la clasificación.58

Figura 15. Hiperplano de separación óptimo H y margen de separación $H_1 \triangleleft H_2$ delimitado por los vectores soporte de cada clase.....60

Figura 16. Margen de separación $H_1 \triangleleft H_2$ suavizado con datos mal clasificados marcados en rojo y su distancia al hiperplano de separación H indicada con flecha roja.....61

Figura 17. Procedimiento de validación cruzada con $K=5$ iteraciones. Los datos se dividen en dos bloques: entrenamiento y test. Los datos de entrenamiento se dividen en cinco bloques, cada uno ellos se utiliza para validar el modelo entrenado con los otros cuatro bloques.64

Figura 18. Gráfico de cajas y densidad por actividad de la variable $AceY_m$ 69

Figura 19. Gráfico de cajas y densidad por actividad de la variable $AceZ_m$ 69

Figura 20. Gráfico de cajas y densidad por actividad de la variable $AceX_s$ 70

Figura 21. Gráfico de cajas y densidad por actividad de la variable $AceZ_s$ 70

Figura 22. Gráfico de cajas y densidad por actividad de la variable $GirY_s$ 71

Figura 23. Gráfico de cajas y densidad por actividad de la variable $MagX_s$. Se ha utilizado escala logarítmica en el eje OX72

Figura 24. Gráfico de cajas y densidad por actividad de la variable MagY _s . Se ha utilizado escala logarítmica en el eje OX.....	72
Figura 25. Gráfico de cajas y densidad por actividad de la variable GirZ _s . Se ha utilizado escala logarítmica en el eje OX.....	73
Figura 26. Gráfico de cajas y densidad por actividad de la variable GpsR _s . Se ha utilizado escala logarítmica en el eje OX.....	73
Figura 27. Gráfico de cajas y densidad por actividad de la variable GpsV _m	74
Figura 28. Gráfico de cajas y densidad por actividad de la variable GpsV _s . Se ha utilizado escala logarítmica en el eje OX.....	74
Figura 29. Porcentaje de varianza explicada por cada componente principal.	76
Figura 30. Error cuadrático por iteración en modelo MLP con 1 capa oculta de 22 neuronas.	78
Figura 31. Error cuadrático por iteración en modelo MLP con 2 capas ocultas de 22 neuronas cada una.	79
Figura 32. Media de la métrica F1 del modelo SVM para diferentes valores de kernel y cost.....	80
Figura 33. Escora longitudinal, centro de carena y centro de gravedad del buque. Fuerzas que actúan sobre el buque durante la actividad pesquera de arrastre.	88
Figura 34. Movimiento de cabeceo del buque de pesca y su efecto sobre la fuerza que ejercen los cables de arrastre sobre el buque.....	89
Figura 35. Movimiento de balance del buque de pesca y par adrizante que se genera por efecto de los cables de arrastre.....	90
Figura 36. Movimiento de guiñada del buque de pesca y su efecto sobre las fuerzas de arrastre.	91
Figura 37. Velocidad del buque durante el lance 4 frente a la costa de Castellón (España).	92
Figura 38. Representación gráfica de la predicción de 4 lances de los modelos LDA, MLP, PNN y SVM. La zona naranja representa los lances reales, los puntos azules la predicción NAVEGAR y los puntos rojos la predicción ARRASTRAR.	97

Índice de tablas

Tabla 1. Principales características de los sensores utilizados (acelerómetro, giróscopo, campo magnético y GPS), descripción del parámetro físico registrado, intervalo de generación de datos de cada sensor (período de muestreo) y variables extraídas de cada parámetro.....	30
Tabla 2. Resumen estadístico de las 22 variables cualitativas de la tabla de datos. La variable con subíndice m representa la media de los datos generados por el sensor en periodos de 10 segundos. La variable con subíndice s representa la desviación estándar de los datos generados por el sensor en periodos de 10 segundos. Para cada variable, la tabla incluye el número de muestras (N), la media, la mediana, la desviación estándar (SD), el coeficiente de variación de Pearson (CV_Pearson), la desviación absoluta de la mediana (MAD), el coeficiente de variación de la desviación absoluta de la mediana (CV_MAD), el valor mínimo (Min), el valor máximo (Max), el rango de valores, el grado de simetría (Skewness), la curtosis y el p-val del test de Kruskal-Wallis.....	68
Tabla 3. Matriz de correlación de las 22 variables normalizadas de la tabla de datos. En la parte inferior de la diagonal muestra el coeficiente de correlación de Pearson y en la parte superior su p-valor.....	75
Tabla 4. Valores propios, porcentaje de varianza y porcentaje de varianza acumulada de los componentes principales.....	76
Tabla 5. Exactitud, precisión, sensibilidad y valor F1 (expresada en porcentajes) en la prueba final de las máquinas de aprendizaje LDA, MLP, PNN y SVM. Este análisis incluyó las muestras registradas durante todas las categorías de actividad con todos sus atributos.....	81
Tabla 6. Puntuación de F1 (expresada como porcentaje) del test final de las máquinas de aprendizaje LDA, MLP, PNN y SVM. Las muestras contienen todos los atributos. 1-vs-1 incluye las muestras de dos categorías. 1-vs-RESTO incluye todas las muestras manteniendo la categoría de la izquierda y agrupando las muestras restantes en la categoría RESTO.	82
Tabla 7. Puntuación de F1 (expresada como porcentaje) para la categoría ARRASTRAR en la prueba final de las máquinas de aprendizaje LDA, MLP, PNN y SVM con todos los atributos y después de reducir el número de categorías. Muestra los resultados con 4 (sin reducción), 3 y 2 categorías. El símbolo > escrito entre dos	

categorías indica que la categoría de la izquierda se fusiona con la categoría de la derecha, asumiendo la nueva categoría la etiqueta de la derecha.83

Tabla 8. Exactitud, precisión, sensibilidad y valor F1 (expresados como porcentaje) de la categoría ARASTRAR del test final de las máquinas de aprendizaje LDA, MLP, PNN y SVM, con todas las categorías y con reducción de atributos agrupados por sensores. Se muestran los resultados sin reducción (Ace-Gir-Mag-GPS) y el conjunto de sensores que ofrece mejor métrica F1 en la fase ARASTRAR. Los atributos agrupados por sensores son: Ace = (AceXm, AceXs, AceYm, AceYs, AceZm, AceZs), Gir = (GirXm, GirXs, GirYm, GirYs, GirZm, GirZs), Mag = (MagXm, MagXs, MagYm, MagYs, MagZm, MagZs) y GPS = (GpsRm, GpsRs, GpsVm, GpsVs).84

Tabla 9. Exactitud, precisión, sensibilidad y valor F1 (expresados como porcentaje) de la categoría ARRASTRAR del test final de las máquinas de aprendizaje LDA, MLP, PNN y SVM. Se muestran los resultados sin reducción (Todas actividades/Todos sensores) y con la reducción de categorías y atributos agrupados por sensores que ofrece mejor valor F1 en la fase ARRASTRAR.85

1 Introducción

1.1 Antecedentes

La pesca es una fuente importante de alimentos para la humanidad, proporciona sustento a más de 500 millones de personas, principalmente en los países en desarrollo. El crecimiento exponencial de la población mundial desde finales de la Segunda Guerra Mundial ha favorecido un crecimiento continuo de la industria alimenticia y el sector pesquero no ha sido una excepción. Las flotas pesqueras de los diferentes países se han modernizado paulatinamente en los últimos años, permitiendo una mejora de las técnicas de pesca y un aumento significativo de las capturas, lo que ha llevado a gran parte de los stocks a una situación de colapso biológico (Pontecorvo y Schrank, 2014, 2012), pudiendo llegar a un punto de no retorno donde resultará casi imposible su recuperación (Dai *et al.*, 2012).

Desde los años 70 se han detectado signos de estrés en muchas de las poblaciones de peces con las consiguientes modificaciones de los ecosistemas (Goñi, 1998), considerables pérdidas económicas para el sector y conflictos internacionales sobre la ordenación y el comercio de productos pesqueros (Charles, 1992).

Cabe destacar el colapso de la pesquería de la Anchoveta (*Engraulis ringens*) en la costa peruana ocurrido en la década de los años 70. Después de varios años de sobrepesca, una temporada larga del fenómeno meteorológico “El Niño” y la poca regulación existente, ocasionaron una gran reducción de la población de anchoas disminuyendo su captura de 12 a 4 millones de toneladas (Csirke *et al.*, 1996). Simultáneamente a la crisis económica¹, que supuso una gran pérdida para la economía de Perú, sobrevino la ecológica, afectando a diferentes especies en cuya

¹ Al respecto de las repercusiones económicas de explotación y ordenación (o no ordenación) de los recursos pesqueros, la Organización de las Naciones Unidas para la Agricultura y la Alimentación (FAO) señala: “Existe una interacción estrecha entre las variables social y económica y, con toda probabilidad, cualquier medida de ordenación repercutirá, por ejemplo, en la distribución de los ingresos y la riqueza, el volumen y el tipo de empleo, la asignación de los derechos de uso, y la composición y cohesión de los grupos y subgrupos de interés. Con carácter más general, las decisiones y medidas adoptadas en materia de ordenación influirán en las actitudes, positivas y negativas, de los grupos interesados. Las medidas de ordenación pesquera pueden condicionar también la contribución de la pesca en aspectos de política primordiales como la seguridad alimentaria, los ingresos netos en divisas, las subvenciones y otros costos y beneficios.” (FAO Fishery Resources Division, 1999).

cadena alimenticia participa directa o indirectamente la anchoveta, y perturbando la sostenibilidad del ecosistema marino.

Otro acontecimiento a destacar ocurrió en la década de los años 90 en las costas de Terranova, donde la sobrepesca del bacalao provocó su colapso y el cierre de sus caladeros en 1992. Veinte años después, la población de bacalao en esa costa seguía sin recuperarse afectando dramáticamente a la economía y la vida de la región. Se espera que se recupere a niveles históricos y sostenibles hacia el año 2030 (Ferguson-Cradler, 2018). Esta crisis generalizada fue motivada fundamentalmente por un exceso de esfuerzo pesquero². Rose y Kulka (1999) realizaron un análisis espacial de la captura por unidad de esfuerzo³ (CPUE) y de las densidades y abundancia del bacalao, obteniendo diferentes relaciones entre la CPUE y la biomasa⁴ en las distintas

² Definición de esfuerzo de pesca: 1) La cantidad de artes de pesca de un tipo concreto utilizadas en los caladeros durante una unidad de tiempo dada, por ejemplo, horas de arrastre por día, número de anzuelos lanzados por día o número de caladas de un arte de playa por día. 2) Cantidad general de actividad pesquera (normalmente por unidad de tiempo) expresada en unidades como: días-embarcación en el caladero, número de nasas o redes de arrastre, o (longitud de almadraba x tiempo de inmersión), etc. El esfuerzo puede ser nominal, que refleja el total simple de las unidades de esfuerzo ejercidas sobre una población en un período de tiempo dado). También puede ser normal o efectivo cuando se corrige para tener en cuenta las diferencias entre la capacidad y la eficiencia pesquera y asegurar la proporcionalidad directa con la mortalidad debida a la pesca). Normalmente se relaciona con una pesquería y un arte específico. Si se considera más de un arte, se requiere una normalización en relación con una de ellas. Para los biólogos, una buena medida del esfuerzo de pesca debería ser proporcional a la mortalidad debida a la pesca. Para los economistas, debería ser proporcional al costo de la pesca. (FAO, 2001)

³ Definición de captura por unidad de esfuerzo (CPUE): La cantidad de capturas que se logran por unidad de arte de pesca; por ejemplo, el número de peces por anzuelo de palangre-mes es una forma de expresar la CPUE. La CPUE puede utilizarse como medida de la eficiencia económica de un tipo de arte, pero normalmente se utiliza como índice de abundancia, es decir, se espera que una variación proporcional en la CPUE represente la misma variación proporcional en la abundancia. La CPUE nominal es simplemente la medida de la CPUE de la pesquería. Sin embargo, se sabe que existen muchos factores (incluidos factores económicos, distribuciones geográficas) que pueden afectar a la CPUE pero que no representan variaciones de abundancia. Por tanto, las CPUE suelen “normalizarse” utilizando varias técnicas estadísticas para eliminar los efectos de dichos factores que se sabe que no están relacionados con la abundancia. Por ello, la utilización de la CPUE normalizada resultará más apropiada para un índice de abundancia (FAO, 2001).

⁴ Definición de biomasa: 1) Peso total de los organismos vivos de que se trate, tanto en un sistema, una población o una parte de una población: por ejemplo, biomasa de plancton en una zona, biomasa de desovadores o de peces recientemente reclutados. 2) Peso total de un recurso, una población o un componente de dicha población. Ejemplos: la biomasa de todos los peces bentónicos de Georges Bank; la biomasa de la población de bacalao; la biomasa de desovadores (es decir, el peso de las hembras maduras) (también ictiomasa) (FAO, 2001).

zonas observadas, indicando probablemente diferencias espaciales en la agregación de los peces. Estos autores concluyeron que la incapacidad de reconocer los cambios en la abundancia real debido a la insensibilidad relativa de la CPUE local llevó al colapso del stock.

El crecimiento de la pesca basado en una optimización exclusivamente económica puso de manifiesto que los recursos acuáticos vivos no son infinitos y, por tanto, era necesario establecer algún tipo de regulación que limite y controle la actividad pesquera.

A principios de la década de los 80, la Convención de las Naciones Unidas sobre el Derecho del Mar (United Nations, 1982) proporcionó un nuevo marco jurídico sobre el mar estableciendo zonas de jurisdicción nacional. Esto hizo posible que los países ribereños tuvieran herramientas de gestión para una ordenación de los recursos marinos ajustada a las características técnicas de sus flotas, ejerciendo de esta forma derechos de explotación de los recursos pesqueros dentro de sus aguas territoriales (Barkin *et al.*, 2018).

Con objeto de facilitar el flujo de información entre países sobre el estado real del sector pesquero, en el año 2000, la Organización de las Naciones Unidas para la Alimentación y la Agricultura (FAO, 2000) propone utilizar en todos los países unos indicadores comunes destinados a analizar la evolución de la actividad pesquera y sus consecuencias sobre el ecosistema marino. Estos indicadores reflejarían la situación del sector a nivel ecológico, económico, social y de gobierno. En el caso de los indicadores ecológicos, proporcionarían información sobre la estructura de la captura, calidad de los hábitats y presión de pesca.

En Septiembre de 2015, las Naciones Unidas pusieron en marcha la *Agenda 2030 para el Desarrollo Sostenible* (ONU, 2015), un plan ambicioso encaminado a tomar medidas para reconducir el mundo por el camino de la sostenibilidad y la resiliencia. En referencia a la vida submarina se plantearon objetivos destinados a mantener la salud de los océanos y a gestionar sus recursos con responsabilidad minimizando el impacto perjudicial sobre las pesquerías; concluyendo que es necesaria una “gestión cuidadosa” del entorno marino, como elemento clave de un futuro sostenible.

En Europa, la pesca se rige por una política común (PPC), con normas comunes adoptadas a nivel de la Unión que se aplican en todos los estados miembros. Los objetivos iniciales de la PPC, incluida en la política agraria común (PAC), eran preservar las poblaciones de peces, proteger el medio ambiente marino, garantizar la viabilidad económica de las flotas de la Unión y proporcionar a los consumidores

alimentos de calidad, en definitiva, una explotación sostenible de los recursos acuáticos vivos de modo equilibrado y desde los puntos de vista medioambiental, económico y social.

En 1992, el reglamento del Consejo de la Comunidad Económica Europea que reguló la política pesquera hasta 2002 (CEE, 1992) introdujo el concepto de esfuerzo pesquero a fin de restablecer y mantener el equilibrio entre los recursos disponibles y las actividades pesqueras. Así mismo, sometió el acceso a los recursos a un sistema de concesión de licencias, pero éste no fue capaz de evitar la sobrepesca y el deterioro de muchas poblaciones de peces. Desde entonces se han sucedido diferentes reformas de la PPC con el objetivo de asegurar un futuro sostenible para el sector pesquero, además de la creación de la Agencia Europea de Control de la Pesca (Vigo, España).

En 2013, la Unión Europea alcanzó un acuerdo para el nuevo régimen de pesca basado en tres pilares: (i) la nueva PPC (Unión Europea, 2013a); (ii) la nueva organización común de mercados en el sector de los productos de la pesca y de la acuicultura (Unión Europea, 2013b) y; (iii) el nuevo Fondo Europeo Marítimo y de Pesca (Unión Europea, 2014). En este acuerdo y posteriores resoluciones se plantea la conservación de los recursos pesqueros mediante el ajuste de la capacidad de pesca a las posibilidades de pesca, gestionando las poblaciones de peces de acuerdo con el principio de rendimiento máximo sostenible fijando un porcentaje de capturas que permitan a las poblaciones de peces reproducirse para garantizar la sostenibilidad de la explotación. Así mismo, se destaca la importancia de la pesca costera y se establecen nuevas obligaciones a los Estados miembros para reforzar el aspecto científico, incrementar la recogida de datos y la puesta en común de información sobre poblaciones, flotas e impacto de la actividad pesquera. La PPC basa sus decisiones en el mejor asesoramiento científico disponible y aplica el principio de cautela, en virtud del cual la insuficiencia de datos científicos no puede justificar el aplazamiento ni la falta de adopción de medidas para conservar las especies.

1.1.1 Sistema de seguimiento de buques (VMS)

En 1995 la Unión Europea puso en marcha un proyecto piloto con la finalidad de evaluar las funcionalidades y costes de diferentes sistemas satelitales que permitieran el control permanente del posicionamiento y la actividad de los barcos de pesca. El resultado de este programa piloto fue la introducción en el año 2000 del Sistema de Seguimiento de Buques (*Vessel Monitoring System* VMS) con el objetivo de mejorar la capacidad de supervisión, así como la gestión de la actividad pesquera. De esta

forma, desde el 1 de enero de 2000 todos los buques con una eslora superior a 24 metros debían instalar obligatoriamente el sistema VMS, norma que se extendió a los buques de eslora superior a 15 metros en 2003 y posteriormente a los de eslora superior a 12 metros en 2012 (European Union, 2009). Los datos proporcionados por el sistema VMS (posición geográfica, rumbo y velocidad del buque) son transmitidos cada hora a un Centro Estatal de Seguimiento de Pesca (CESP) donde la información es almacenada y cruzada en tiempo real con diferentes destinos internacionales en cuyas aguas faena la flota europea.

La Unión Europea, en el marco de la Política Pesquera Común, inicia en el año 2007 un debate sobre cómo limitar la actividad pesquera cuando sea necesario para la conservación de las poblaciones de peces y cómo medir la capacidad pesquera y la presión que ejerce la flota pesquera sobre el ecosistema marino. En concreto, propuso utilizar unos indicadores comunes para medir la capacidad pesquera y el esfuerzo de pesca de cada buque de la flota (European Commission, 2007). La capacidad pesquera puede cuantificarse a partir de las características técnicas del buque y del arte de pesca. Sin embargo, el cálculo del esfuerzo pesquero requiere conocer el tiempo durante el cual se halla en funcionamiento efectivo su capacidad pesquera.

Durante los últimos años, numerosos investigadores han propuesto diferentes técnicas de análisis de los datos VMS para completar la información que ofrecen sobre la actividad de la flota pesquera. Mills *et al.* (2007) obtuvieron estimaciones en alta resolución del esfuerzo de pesca ejercido por la flota de arrastre. Para ello, combinaron reglas de velocidad y direccionalidad para identificar la actividad de los buques de arrastre en cada posición VMS y estimar posibles trayectorias del buque entre señales. Según los autores, a mínima velocidad el buque trazará una trayectoria rectilínea entre señales y a máxima velocidad podrá alcanzar su máxima desviación. A cada posible posición del buque entre muestras se asocia un nivel de incertidumbre creciente con la distancia a la recta que une las dos señales VMS, que utilizarán posteriormente para estimar el esfuerzo de pesca en dicha posición.

Bertrand *et al.* (2008) combinaron la información proporcionada por los observadores de a bordo con el sistema VMS para estimar la posición de las operaciones de pesca de la flota peruana de cerco de anchoa. Utilizaron una red neuronal (perceptrón multicapa) para identificar las posiciones o puntos de pesca, entrenada con 5 atributos (velocidad, hora, cambio absoluto de rumbo entre el movimiento actual y anterior, variación de velocidad entre el movimiento actual y anterior, variación de velocidad entre la actual y siguiente) etiquetados con la información de observadores de a bordo. Estos autores afirmaron identificar el 83% de las muestras positivas. Así mismo, Lee *et al.* (2010) propusieron un método para estimar el esfuerzo pesquero a partir de los

datos VMS. El método consiste en eliminar registros VMS duplicados y registros cercanos a los puertos, calcular el intervalo de tiempo entre registros sucesivos para identificar períodos de actividad, vincular cada registro a un tipo de embarcación y aparejo, diferenciar la actividad pesquera y no pesquera, y sumar los registros de pesca en tiempo y espacio para estimar el esfuerzo pesquero por área o cuadrícula espacial.

Por otro lado, Gerritsen *et al.* (2012) utilizaron datos VMS de buques arrastreros demersales irlandeses para identificar regiones con composiciones de especies homogéneas en el espacio y el tiempo. Proporcionaron una herramienta para explorar diferentes escenarios de desplazamiento de esfuerzo, evaluar posibles efectos de cierres de pesca de especies o restricciones en ciertas áreas. En este trabajo se identificó la actividad pesquera a través de la velocidad de los buques (entre 1.5 y 4.5 nudos) y se evaluó el esfuerzo pesquero a través del tiempo pescando (número de señales catalogadas como pesca) en cada cuadrícula espacial.

Otros investigadores complementan la información VMS con los diarios de pesca y el conocimiento aportado por los pescadores. Tal es el caso de Bastardie *et al.* (2010) quienes fusionaron VMS y el diario de pesca para identificar con mayor precisión los periodos y zonas de navegación y pesca. Presentaron un método genérico y preciso para vincular los libros de registro y los datos VMS, utilizando inicialmente los libros de registro para dividir la flota en conjuntos de buques homogéneos que pescan la misma especie en la misma zona y/o temporada (unidades de *métier*⁵), procesando los datos VMS para identificar viajes de pesca (salida, lances de pesca, llegada) y fusionando finalmente los datos VMS con el diario de pesca, resolviendo desajustes y errores. En la misma línea, Gerritsen y Lordan (2011) exploraron y validaron el método más apropiado y práctico para integrar los datos VMS con los datos del libro de registro. Su trabajo se centró en buques arrastreros y muestran un método sencillo: (i) aplicar criterios de velocidad para identificar registros VMS con actividad pesquera; (ii) estimar el esfuerzo para cada registro VMS como el intervalo de tiempo al registro anterior; (iii) repartir la captura por igual en todos los registros VMS de la jornada de pesca y; (iv) asignar a cada cuadrícula las capturas de los puntos VMS que agrupa. Finalmente, Szostek *et al.* (2017) evaluaron la precisión y la validez de los datos de esfuerzo de pesca (extensión espacial y esfuerzo relativo) obtenidos de los

⁵ "*métier*" : grupo de operaciones de pesca que tienen por objeto la captura de especies (o conjuntos de especies) similares con artes similares durante el mismo período del año y/o en la misma zona, y se caracterizan por modelos de explotación similares (Unión Europea, 2008)

datos de conocimiento local de los pescadores en comparación con los derivados de los datos del sistema de monitoreo de barcos (VMS).

En relación a los patrones de comportamiento de los pescadores, Joo *et al.* (2015) estudiaron los patrones espaciales exhibidos por las trayectorias de los viajes de pesca de cerco de anchoveta peruana. Iniciaron el trabajo con la información de los observadores de a bordo para establecer 3 tipos de actividad durante el viaje: pesca, búsqueda y navegación. Utilizaron un clasificador semi-Márkov oculto supervisado para inferir la actividad, entrenado y validado con datos de velocidad y variación de velocidad entre señales VMS etiquetadas con la información de los observadores a bordo de los buques. A continuación, de los datos VMS y de la actividad inferida, extrajeron métricas que describen el viaje, utilizaron el análisis de componentes principales (PCA) para identificar las más relevantes (duración, distancia máxima a la costa, extensión geográfica y tiempo dedicado a pescar, buscar y navegar) e identificaron con ellas los patrones espaciales de viaje de pesca asociados a las regiones visitadas, segmentos de flota y comportamiento de los patrones de los buques. Por otro lado, Shepperson *et al.* (2016) utilizaron los datos VMS y libros de registro para confirmar y explicar los patrones de comportamiento de los pescadores de vieiras recogidos directamente mediante encuestas y entrevistas personales. Establecieron parámetros ambientales (altura de ola, distancia a puerto, etc.) e identificaron tres estrategias de comportamiento para la elección de la zona de pesca: maximizar cantidad, maximizar calidad y pesca eficiente. Los autores afirmaron que su enfoque proporciona datos fiables y relevantes para la parametrización de modelos predictivos más robustos y realistas sobre el comportamiento de los pescadores.

Otra línea de investigación interesante se centra en el reconocimiento del tipo de arte de pesca utilizado por los buques pesqueros a partir de la información proporcionada por el VMS. Cabe destacar la propuesta de Marzuki *et al.* (2015) quienes propusieron detectar usos anormales de artes de pesca no declarados a partir de trayectorias VMS. Su enfoque combinaba técnicas de minería de datos supervisadas y no supervisadas, resumidas en tres pasos: (i) entrenamiento de un modelo de mezcla gaussiana (GMM) para la identificación no supervisada de las características discriminatorias de cada pesquería (velocidad y ángulo de giro) extraídas de los datos VMS; (ii) entrenamiento supervisado de modelos de reconocimiento (random forest -RF- y máquina de soporte vectorial -SVM-) y; (iii) aplicación de los modelos de reconocimiento a nuevos datos VMS para verificar la coherencia entre las artes de pesca reconocidas y declaradas.

En cuanto al desarrollo de herramientas que faciliten el trabajo con la información proporcionada por el sistema VMS, resaltar el trabajo de Hintzen *et al.* (2012) desarrollando en R el paquete de software de código abierto *VMStools* para el

procesamiento, análisis y visualización de desembarques (diario de pesca) y datos de ubicación de buques pesqueros VMS. Esta herramienta permitía detectar y eliminar puntos de datos erróneos, identificar grupos de *métiers*, vincular el diario de pesca con los datos proporcionados por VMS, identificar la pesca frente a otras actividades mediante el filtro de velocidad, proporcionar mapas de alta resolución del esfuerzo de pesca y desembarques, interpolar rutas, calcular indicadores de impacto de la pesca espacio-temporal, transformar datos a otros formatos, etc. Así mismo, Hery *et al.* (2020) ofrecieron una página web para mostrar la ubicación de la captura de atún en las aguas territoriales de Indonesia. En esta página se utilizaban datos VMS y métodos de aprendizaje automático (*Naives Bayes* y SVM) para identificar los puntos de pesca. Sus resultados mostraban precisiones de verdaderos positivos superiores al 94.2%.

Finalmente, cabe destacar la propuesta de Pedreschi *et al.* (2021) donde utilizaban el sistema VMS para implantar un sistema de incentivos en tiempo real para los pescadores por realizar acciones deseables en el marco de la política de gestión de la actividad pesquera.

1.1.2 Sistema de localización y seguimiento de embarcaciones pesqueras andaluzas (SLSEPA)

El Sistema de Localización y Seguimiento de Embarcaciones Pesqueras Andaluzas (SLSEPA), conocido como sistema de “cajas verdes” desarrollado por el gobierno regional de la Junta de Andalucía (España) (Junta de Andalucía, 2004) utiliza la red de telefonía móvil GPRS para transmitir al centro de control, la posición, rumbo y velocidad de los barcos de la flota de pesca artesanal a intervalos de 3 minutos. El aumento de la frecuencia de muestreo de SLSEPA junto con los bajos costes de transmisión de datos permite mejorar la precisión de la estimación del esfuerzo de pesca y la exactitud de la distribución espacial de la flota (Cojan y Burgos, 2015).

Varios trabajos de investigación examinan la utilidad del sistema SLSEPA, entre los que destaca el trabajo realizado por Burgos *et al.* (2013) donde analizaban los datos SLSEPA de alta resolución espacial y temporal para identificar las zonas de pesca preferidas, las rutas de navegación y la secuencia de operaciones de pesca de la flota voracera artesanal en pequeña escala de la zona del Estrecho de Gibraltar. Los datos SLSEPA les permitían trabajar a una escala espacial más fina que con el sistema VMS, pudiendo identificar con gran precisión la actividad de la flota costera. Otro trabajo interesante es el propuesto por Cojan y Burgos (2015) donde analizaban la información proporcionada por SLSEPA de la flota que explota la chirila (*Chamelea*

gallina) en el Golfo de Cádiz con objeto de mejorar el conocimiento de esta pesquería. Los autores evaluaron el esfuerzo pesquero considerando el tiempo efectivo de pesca (desde el inicio del primer lance hasta el final del último arrastre), el número de lances y la duración de cada uno de ellos. Estimaron una duración máxima de lance de 20 minutos en base a la información obtenida por observadores de a bordo e identificaron los lances apoyándose en la velocidad del barco.

1.1.3 Sistema de identificación automática (AIS)

El sistema de identificación automática (*Automatic Identification System* AIS) fue presentado en 1974 por la Organización Marítima Internacional (International Maritime Organization, 1974) dentro del convenio internacional para la seguridad de la vida humana en el mar (Convenio SOLAS) con objeto de mejorar la seguridad marítima y evitar colisiones entre buques. Originalmente se diseñó para buques de más de 300 toneladas de arqueo y buques de pasaje, pero con la entrada en vigor de la Directiva Europea 2011/15 del Parlamento Europeo (European Parliament, 2011) los buques pesqueros también tienen la obligación de instalar el dispositivo AIS (actualmente todos los buques pesqueros de más de 15 m de eslora).

A nivel mundial, el número de buques pesqueros dotados con AIS aumenta anualmente entre un 10 y un 30 por ciento. En el año 2018, la *Global Fishing Watch* (GFW) publicó la primera base de datos mundial (Global Fishing Watch, 2018) de las operaciones pesqueras utilizando la información proporcionada por los AIS. De este modo, los datos de la actividad de más de 60000 buques pesqueros sirvieron para conocer la situación de la pesca en el mundo. En noviembre de 2019, la FAO presentó en Roma el nuevo atlas mundial de la actividad pesquera basado en los datos proporcionados por el AIS, donde se realiza un seguimiento detallado de las rutas de decenas de miles de buques de pesca industrial, ofreciendo una visión casi completa de la actividad pesquera en algunas regiones, como por ejemplo con los buques de más de 15 metros de eslora en el Atlántico norte (FAO, 2019). Sin embargo, en otras regiones estos datos solamente proporcionan información sobre una pequeña parte de la actividad pesquera, como ocurre en el caso del Océano Índico. Esto se debe en parte al gran número de embarcaciones artesanales o de pequeño tamaño en muchas regiones centrales y meridionales, y al menor uso de AIS por parte de buques más grandes en regiones como el sudeste asiático.

A diferencia del VMS, que generalmente se basa en comunicaciones satelitales punto a punto entre el barco y los centros terrestres, los barcos emiten mensajes AIS

omnidireccionalmente a través de la banda de ondas métricas del servicio móvil marítimo y pueden ser recibidos por otros barcos que se encuentren en las proximidades, por receptores terrestres y por satélites.

Existen dos tipos de sistemas AIS, el de Clase A se utiliza en embarcaciones de altura, trabaja en la banda marítima de VHF en una gama de frecuencias comprendida entre 156.025 MHz y 162.025 MHz, envía información continuamente, su potencia de emisión es de 12.5 W y su alcance puede rondar las 50 millas. El sistema AIS Clase B dispone de menores prestaciones y requisitos tecnológicos, trabaja entre las frecuencias de 161.500 MHz y 162.025 MHz, envía información cada 3 minutos, tiene una potencia de emisión de 2 W y un alcance aproximado de 12 millas.

El objetivo fundamental del sistema AIS es evitar colisiones, para ello ofrece la posibilidad a los barcos de intercambiar información estática del buque (número de identificación IMO, nombre, eslora, manga, tipo de barco, etc.), información dinámica cada 2-12 segundos (posición del barco, hora UTC, derrotero sobre el fondo COG, velocidad de desplazamiento sobre el fondo SOG, rumbo, índice de giro, estado de navegación, etc.), información sobre la travesía cada 6 minutos (calado del barco, cargamento peligroso, destino y hora prevista de llegada, etc.) y mensajes de seguridad cuando sean necesarios (International Telecommunication Union, 2014). En consecuencia, el sistema AIS permite ver en una pantalla la posición de los buques del entorno y obtener información siempre y cuando los otros buques también transmitan información vía AIS.

El uso del sistema AIS con fines científicos y de gestión ha crecido rápidamente, principalmente debido a su alta resolución temporal, al gran volumen de información que envía y a la facilidad de acceso a dicha información. Cabe destacar algunas publicaciones científicas recientes como la propuesta por de Souza *et al.* (2016) donde proponían métodos automatizados para detectar el comportamiento de pesca potencial de diferentes tipos de artes (arrastre, palangre y cerco) basados en datos de seguimiento AIS. Debido a que la actividad pesquera para cada tipo de arte se ajusta a un determinado patrón (trayectoria, velocidad y tiempo de pesca), propusieron tres enfoques distintos de aprendizaje automático, extracción de datos y filtrado para detectar posibles actividades pesqueras. En el caso de la pesca de arrastre, propusieron utilizar modelos bayesianos ocultos de Márkov (HMM) utilizando la velocidad del buque como parámetro crítico. Para la detección de la actividad pesquera de palangre propusieron un enfoque de minería de datos basado en el movimiento de animales y selección de hábitat (Bertrand *et al.*, 2005), y para la pesca de cerco propusieron utilizar el algoritmo de DM basado en una estrategia de filtrado similar al utilizado por McCauley *et al.* (2016).

Por otro lado, Natale *et al.* (2015) evaluaron la adopción del AIS en la flota pesquera de la UE y su viabilidad para elaborar un mapa del esfuerzo pesquero con alta resolución espacial y temporal a escala europea. Para ello, identificaron la actividad pesquera de arrastre analizando los perfiles de velocidad de los buques y validaron sus resultados con los cuadernos de bitácora. Así mismo, Russo *et al.* (2020) utilizaron datos AIS para reconstruir patrones espaciales y temporales de la actividad de pesca de arrastre en el Mar Adriático Norte y Central, una de las áreas más productivas del Mar Mediterráneo. En este trabajo, relacionaron el esfuerzo de pesca con la distancia a puerto, la estación del año, las zonas prohibidas, caladeros, etc. Estos autores concluyeron que el sistema AIS es una herramienta esencial para analizar con precisión la actividad espacial y temporal de la flota pesquera y establecer estrategias de gestión que permitan una explotación sostenible de los recursos acuáticos.

Finalmente, Chuaysi y Kiattisin (2020) trabajaron en el examen del comportamiento de los buques pesqueros para combatir la pesca ilegal no declarada y no reglamentada. Su trabajo se centró en la pesca del marisco y el uso de técnicas de aprendizaje profundo para identificar cuándo un buque está pescando, navegando o en transbordo. Su metodología la aplicaron a cerqueros, arrastreros, palangreros y buques congeladores. En concreto, utilizaron datos procedentes de los sistemas VMS y AIS para identificar la actividad pesquera. Propusieron utilizar las estancias en puerto para segmentar la actividad y obtener rutas únicas, identificar puntos de interés, extraer patrones que determinen el tipo de pesquería y extraer datos procesados relacionados con la velocidad, distancia y tiempo entre señales, sinuosidad de la trayectoria, etc., que permitan entrenar una red neuronal MLP (perceptrón multicapa) para que identifique cuándo el buque está pescando. Finalmente, destacaron las dificultades encontradas para identificar algunas pesquerías únicamente con datos VMS debido al alto periodo entre señales.

A pesar de que el sistema AIS se diseñó originalmente para reforzar la seguridad marítima, la FAO reconoció la utilidad de los datos AIS en la gestión de la actividad pesquera (FAO, 2019). En la actualidad, el AIS puede rastrear a la mayoría de los grandes buques, aunque su nivel de implantación varía en función de la zona de pesca.

1.1.4 Otras iniciativas

Behivoke *et al.* (2021) propusieron utilizar pequeños rastreadores GPS con una frecuencia de muestreo mayor que el VMS en la flota costera dedicada a la pesquería de arrecifes a pequeña escala en Madagascar. Con la información proporcionada por

estos dispositivos investigaron el rendimiento de los métodos del umbral de velocidad y el algoritmo de aprendizaje *Random Forest* para estimar y mapear el esfuerzo pesquero. Monitorizando viajes de varios tipos de pesquerías con una frecuencia de muestreo de 45 segundos, consiguieron buenos resultados con ambos métodos (>89%) en el reconocimiento de la actividad en la red de arrastre, red de enmalle, línea de mano y arpón.

Por otro lado, Susanto y Wibisono (2019) propusieron incorporar telemetría y utilizar los datos VMS para detectar irregularidades en el consumo de combustible de los buques pesqueros. Emplearon el método de clasificación de *Naive Bayes* y regresión logística para la detección de irregularidades obteniendo una precisión del 92% y 96% respectivamente. En su trabajo concluyeron que la incorporación de telemetría al sistema VMS permitiría monitorizar en tiempo real las condiciones del motor y uso de combustible, aportando información relevante para mejorar el control de la actividad de la flota pesquera.

1.1.5 La actividad pesquera actual

La FAO presentó en el año 2020 un informe sobre el estado mundial de la pesca y la acuicultura donde muestra la situación actual del sector (FAO, 2020). Afirma que, en 2018, la producción mundial de la pesca de captura alcanzó la cifra récord de 96.4 millones de toneladas, siendo China, Indonesia, Perú, India, Rusia, Estados Unidos y Vietnam los siete principales países productores de la pesca de captura con casi el 50% de la producción mundial. Las especies más capturadas fueron la Anchoqueta (*Engraulis ringens*) con 7 millones de toneladas, el Colín de Alaska (*Theragra chalcogramma*) con 3.4 millones de toneladas y el Listado (*Katsuwonus pelamis*) con 3.2 millones de toneladas.

Se estima que, en 2018, el total mundial de embarcaciones pesqueras ascendía a 4.56 millones, de los cuales, el 37% corresponden a embarcaciones pequeñas sin cubierta y sin motor, el 52% a embarcaciones con motor menores de 12 metros, el 9.5% a embarcaciones con motor entre 12 y 24 metros y apenas el 1.5% eran embarcaciones con motor de eslora superior a 24 metros. Estos datos son aproximados, pues existen gran cantidad de embarcaciones pequeñas que no están sujetas a requisitos de licencia y registro.

A pesar de la gran envergadura de la flota costera, los pescadores a pequeña escala rara vez se consideran explícitamente en la gestión regional de la pesca y su contribución al suministro de pescado tiende a ser muy subestimada en comparación

con la pesca industrial (Teh y Pauly, 2018). La publicación en el año 2018 de las Directrices de la FAO para asegurar la pesca sostenible a pequeña escala proporciona un punto de partida básico para que los gobiernos comiencen a abordar la pesca a pequeña escala en sus países (FAO, 2018). La FAO, en su informe del año 2021 sobre la pesca a pequeña escala y el derecho humano a una alimentación adecuada, se reafirma en el problema (FAO, 2021). La pesca a pequeña escala emplea a más del 90% de los pescadores de captura y trabajadores de la pesca del mundo. Además, el 90% de la pesca a pequeña escala opera en países en desarrollo. Entre el 90 y el 95% de todos los desembarcos a pequeña escala se destinan al consumo local, proporcionando una fuente de alimentos nutritivos a comunidades que se enfrentan a muchas formas de vulnerabilidad y marginación.

Reconociendo el indudable avance que ha supuesto la implantación del sistema VMS, existen ciertos aspectos que limitan la fiabilidad y precisión de las estimaciones deducidas a partir de la información proporcionada por este sistema. En primer lugar, la instalación de transpondedores sólo es obligatoria para buques con una eslora superior a 12 metros, por lo que la información procedente de una gran parte de la flota artesanal no está disponible. Por otro lado, los dispositivos VMS envían datos con una frecuencia muy baja (intervalos de 1 a 2 horas) lo que limita la precisión en la predicción de trayectorias de los buques, número y recorrido de lances, mapas de pesca espaciales y temporales, presión sobre caladeros, etc. En este sentido, Deng *et al.* (2005) afirmaron que los errores en la predicción en la pesca de arrastre aumentan con el intervalo de tiempo entre los registros VMS y que la pérdida de precisión es bastante importante con intervalos de muestreo superiores a 30 minutos. Además, el transpondedor VMS envía únicamente la posición, rumbo y velocidad del buque que le proporciona su GPS. Mills *et al.* (2007) indicaron que la velocidad del buque por sí sola no es un criterio adecuado para identificar la actividad de arrastre y confirmaron que la frecuencia de muestreo del sistema VMS es demasiado baja para permitir caracterizar las huellas de las redes de arrastre, ya que la mayoría de los lances están representados por un solo registro y algunos no se detectan en absoluto al ejecutarse entre dos muestras. En la misma línea, en el caso de la anchoveta peruana, Bertrand *et al.* (2008) afirmaron que el parámetro velocidad proporcionado por VMS conduce a una sobreestimación del número de eventos de pesca cercano al 182%.

Por otro lado, el sistema SLSEPA diseñado específicamente para la flota costera de la Comunidad de Andalucía (España) mejora en resolución y coste al VMS, pero sigue teniendo ciertas deficiencias por la escasa información que proporciona al mantener el GPS como único dispositivo de registro, además de seguir utilizando el método de

discriminación basado en el umbral de velocidad para identificar la actividad pesquera (Burgos *et al.*, 2013; Lee *et al.*, 2010).

En la misma línea, el sistema AIS reduce el periodo entre muestras en un rango que oscila entre 2 y 12 minutos y sigue basando su funcionamiento en la información que proporciona su GPS. Además, el sistema no es tan fiable como el VMS o SLSEPA frente a la pesca ilegal porque es más vulnerable al sabotaje, pudiendo transmitir información incorrecta sobre su identidad, falsificando datos o simplemente, desconectando el dispositivo a voluntad.

En la actualidad, el seguimiento de la actividad pesquera deja fuera de control casi la totalidad de la flota de pequeña escala. De las embarcaciones monitorizadas por algún sistema de seguimiento, solo se dispone de la información periódica que se recibe del GPS instalado en la embarcación. El alto coste de los actuales dispositivos de seguimiento dificulta su implantación en la flota artesanal, otro aspecto negativo es el hecho de que la información que registra su GPS no se procesa por el dispositivo y no se envía en su totalidad al centro de control, y tampoco dispone de otros sensores que pudieran ayudar a identificar con más exactitud qué tarea realiza el buque en cada momento.

En definitiva, cualquier avance sustancial en la cobertura del sistema de seguimiento de la flota pesquera y en la mejora de la información recopilada y recibida de cada embarcación redundaría en una mejor gestión de los recursos pesqueros.

1.2 Hipótesis

Considerando que la presencia de un arte de pesca activo afecta al comportamiento dinámico del buque y que las maniobras del buque pueden estar relacionadas con su actividad pesquera, se plantea la hipótesis de que un nuevo dispositivo equipado con sensores de posición y movimiento podría registrar y ofrecer la información necesaria que permita identificar con precisión dónde y cuándo el buque está pescando.

Así mismo, si este dispositivo ofreciera la capacidad de procesamiento suficiente, podría ejecutar algoritmos de inteligencia artificial que sean capaces de clasificar los datos registrados por los sensores e identificar en tiempo real la actividad que realiza el buque de pesca.

Además, si el coste de estos dispositivos fuera lo suficientemente asequible, el nuevo sistema podría ser instalado en la totalidad de la flota pesquera.

1.3 Objetivos

Se propone el desarrollo de un nuevo Sistema Integral de Monitorización de la Actividad Pesquera (FAMIS) sobre la base de un dispositivo móvil de bajo coste con los sensores que habitualmente incorpora. En este trabajo se plantean los siguientes objetivos:

- Identificar una serie de parámetros detectables con sensores electrónicos que sirvan para distinguir el momento de la maniobra de pesca.
- Evaluar la capacidad de los sensores integrados en los dispositivos móviles actuales de bajo coste (GPS, acelerómetro, giroscopio y campo magnético) para detectar cambios significativos en el comportamiento estático y dinámico de la embarcación durante la actividad pesquera de arrastre.
- Evaluar la capacidad de diferentes máquinas de aprendizaje para clasificar e identificar la fase del lance en la que se encuentra el buque durante su actividad pesquera.
- Identificar la combinación de sensores con la que ofrecen mejor respuesta las máquinas de aprendizaje.
- Identificar la estructura del lance (número de fases e inicio y final de cada fase) que más favorece la tarea de clasificación de las máquinas de aprendizaje.

2 Material y métodos

2.1 El dispositivo móvil y los sensores

Hoy en día, la mayoría de los dispositivos móviles y tabletas integran los sensores GPS⁶, Acelerómetro, Giróscopo y Campo Magnético. En este trabajo se utilizó una tableta Samsung SM-P600 y 6 dispositivos de respaldo: 3 tabletas LG V-400, un dispositivo móvil BQ Aquaris M5, un dispositivo móvil BQ Aquaris E5 y un dispositivo móvil BQ Aquaris E4.5. Los datos utilizados en este trabajo proceden de la tableta Samsung SM-P600. Ésta dispone de un sensor inercial de 6 ejes (Bosch Sensortec BMI055), un acelerómetro digital triaxial de 12 bits, un giróscopo digital triaxial de 16 bits, un sensor de campo magnético (compás) digital triaxial de 16 bit (Asahi Kasei Microdevices AK8963C) y un receptor GPS.

A partir de estos sensores se pueden obtener datos para un total de 11 variables con un periodo de muestreo que oscila entre 50 ms y 1 s (Tabla 1).

Tabla 1. Principales características de los sensores utilizados (acelerómetro, giróscopo, campo magnético y GPS), descripción del parámetro físico registrado, intervalo de generación de datos de cada sensor (período de muestreo) y variables extraídas de cada parámetro.

Sensor	Parámetro medido	Periodo muestreo	Variable
Acelerómetro			<i>AceX</i>
Rango = $\pm 19.6133 \text{ m/s}^2$	Vector aceleración lineal	50 ms	<i>AceY</i>
Sensibilidad = 0.009570 m/s^2			<i>AceZ</i>
Giróscopo			<i>GirX</i>
Rango = $\pm 8.7266 \text{ rad/s}$	Velocidad rotación alrededor de los 3 ejes	50 ms	<i>GirY</i>
Sensibilidad = 0.0002661 rad/s			<i>GirZ</i>
Campo magnético			<i>MagX</i>
Rango = $\pm 4900 \text{ } \mu\text{T}$	Vector campo magnético terrestre	50 ms	<i>MagY</i>
Sensibilidad = $0.6 \text{ } \mu\text{T}$			<i>MagZ</i>
GPS	Rumbo	1 s	<i>GpsR</i>
	Velocidad		<i>GpsV</i>

⁶ Sistema de Posicionamiento Global

Los sensores del dispositivo móvil utilizan el eje de coordenadas asociado a la pantalla del dispositivo (Figura 1). El dispositivo se instaló en el buque de pesca haciendo coincidir sus ejes con los ejes longitudinal, transversal y vertical del buque, y en su instalación se aseguró que estuviera firmemente sujeto al mobiliario fijo del barco para permitirle moverse de forma solidaria al buque y registrar todos sus movimientos durante la actividad pesquera.

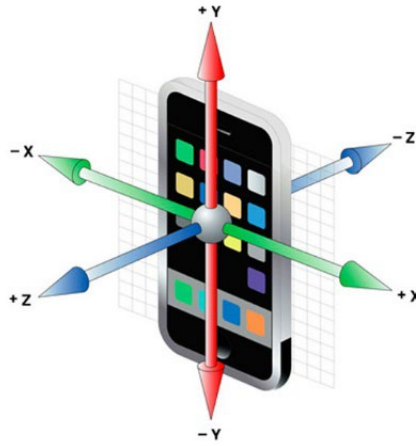


Figura 1. Ejes de coordenadas del dispositivo móvil utilizado por los sensores para registrar la información del vector aceleración, velocidad de giro alrededor de cada eje y vector de campo magnético.

El acelerómetro mide en metros por segundo al cuadrado (m/s^2) la aceleración aplicada al dispositivo ($\overrightarrow{Ace}$). Conceptualmente, mide la fuerza $\overrightarrow{F}$ aplicada a la masa del sensor. En este caso particular, la fuerza de la gravedad $\overrightarrow{G}$ influye también en la medida de la aceleración (Ecuación 1). Al inmovilizar el dispositivo de medida en el buque, éste se mueve de forma solidaria con el barco, por lo que la masa M de la ecuación corresponde a la suma de la masa del dispositivo más la del buque, pudiéndose despreciar este término y considerar que el acelerómetro va a registrar las componentes del vector gravedad $\overrightarrow{G}$ en cada instante.

$$\overrightarrow{Ace} = -\overrightarrow{G} - \frac{\sum \overrightarrow{F}}{M} \approx -\overrightarrow{G} \quad (1)$$

El giróscopo mide en radianes por segundo (rad/s) la velocidad de rotación alrededor de los ejes OX, OY y OZ, siendo positiva cuando el giro alrededor del eje se realiza en el sentido contrario a las agujas del reloj visto por un observador situado en una ubicación positiva del eje.

Por su parte, el sensor de campo magnético mide en micro-Teslas (μT) el campo magnético en el entorno del dispositivo. El sensor proporciona las componentes del vector resultante de la suma vectorial del campo magnético terrestre y el campo magnético ambiental generado por los objetos del entorno (motores, instalación eléctrica, etc.) (Ecuación 2).

$$\overrightarrow{\text{Mag}} = \overrightarrow{\text{Mag}}_{\text{Tierra}} + \overrightarrow{\text{Mag}}_{\text{Entorno}} \quad (2)$$

Finalmente, el sensor GPS es un dispositivo capaz de registrar información sobre su localización geográfica tridimensional en forma de latitud, longitud y altura apoyándose en una red de satélites. Asimismo, registra una marca temporal de la medida y genera información adicional procesada como rumbo y velocidad instantánea, velocidad media y precisión. En este trabajo sólo se ha considerado necesario almacenar el rumbo ($GpsR$) y la velocidad instantánea ($GpsV$) cada segundo durante la actividad del buque.

Aunque solo se utilice coloquialmente las siglas GPS para referirse a los sistemas de posicionamiento, en la actualidad existen varias redes de satélites: el sistema GPS con 24 satélites instalado por el departamento de defensa de Estados Unidos (United States Space Force, 2021), el sistema Galileo con 28 satélites financiado por la Unión Europea (European Space Agency, 2021), el sistema Glonass con 31 satélites creado por la antigua Unión Soviética (Russian Federal Space Agency, 2021) y el sistema Beidou con 30 satélites creado por China (China, 2021). La precisión de localización que alcanzan los receptores varía entre varios metros en los sistemas GPS y Glonass, hasta 1 cm en el sistema Galileo.

2.2 La aplicación móvil de recogida de datos

Con el objetivo de registrar el movimiento del buque durante su actividad pesquera se ha desarrollado una aplicación móvil (APP) fácilmente instalable en cualquier dispositivo portátil con sistema operativo Android. Se utilizó el entorno de desarrollo integrado (IDE) Android Studio para implementar la aplicación en la plataforma Android. La interfaz de usuario de la APP permite iniciar la grabación y seleccionar la fase del lance con los botones GRABAR, LARGAR, ARRASTRAR y VIRAR disponibles en la parte inferior de la pantalla del dispositivo. Una vez que la grabación ha comenzado, la interfaz muestra numérica y gráficamente los datos grabados por los

sensores. Además, incorpora un menú de configuración con información sobre el barco, el dispositivo móvil y el arte de pesca (Figura 2).



Figura 2. Interfaz de usuario de la aplicación móvil y flujo de información entre el dispositivo móvil, la base de datos local SQLite y la base de datos MySQL del servidor, y entre RStudio y la base de datos MySQL del servidor.

Durante la grabación, los datos proporcionados por los sensores vinculados a la fase del lance, la información del barco, del arte de pesca y del dispositivo móvil, se almacenan temporalmente en una base de datos local (SQLite) (Figura 3). La base de datos relacional está formada por las tablas *acelerómetro*, *giroscopo*, *campo_magnético* y *gps*, que almacenan los datos registrados por los sensores; la tabla *lance* que almacena los lances realizados; la tabla *actividad* con las fases en que se divide el lance, la tabla *barco* con la información de las embarcaciones; la tabla *dispositivo* con la información de los dispositivos móviles; la tabla *arte_pesca* con los aparejos utilizados y la tabla *evento* que relaciona cada dato registrado por los sensores con el lance y la fase del lance.

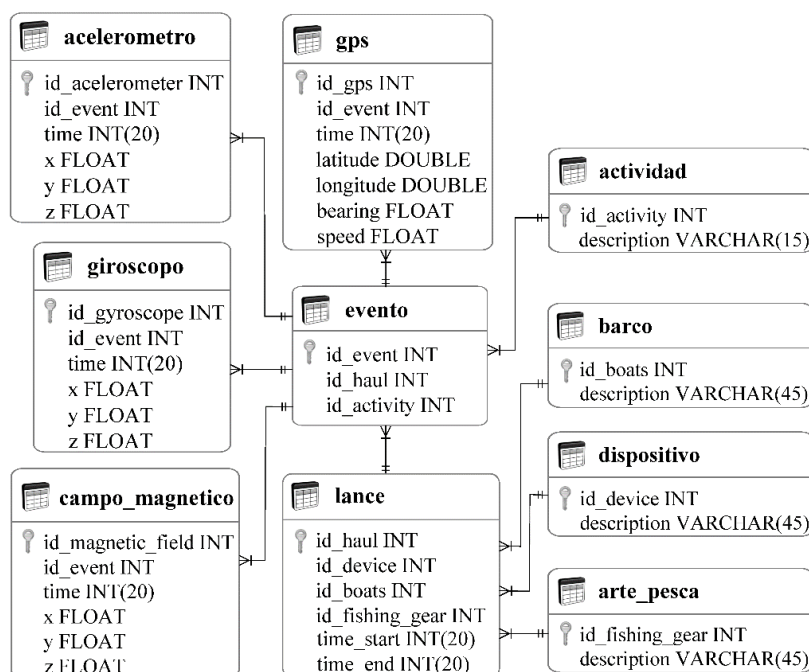


Figura 3. Estructura de la base de datos relacional formada por las tablas acelerómetro, giróscopo, campo_magnético y gps, que almacenan los datos registrados por los sensores; la tabla lance que almacena los lances realizados; la tabla actividad con las fases en que se divide el lance, la tabla barco con la información de las embarcaciones; la tabla dispositivo con la información de los dispositivos móviles; la tabla arte_pesca con los aparejos utilizados y la tabla evento que relaciona cada dato registrado por los sensores con el lance y la fase del lance.

En el anexo 7.1.1 se incluye parte del código fuente de la aplicación móvil, con los métodos de la clase *MainActivity* (anexo 7.1.1.1) que lanzan y detienen los servicios que gestionan los sensores y las clases que trabajan con los sensores Acelerómetro (anexo 7.1.1.2), Giróscopo (anexo 7.1.1.3), Campo Magnético (anexo 7.1.1.4) y GPS (anexo 7.1.1.5). Al iniciar la grabación pulsando el botón GRABAR de la interfaz de usuario (Figura 2), se lanza el método *iniciarServicios* de la clase *MainActivity* (anexo 7.1.1.1). Este método crea instancias de las clases *Acelerómetro*, *CampoMagnético*, *Giróscopo* y *Gps*, y lanza los servicios para su ejecución en segundo plano. Al crear cada servicio lanza el método *onStartCommand* que crea instancias de varios objetos que permitirán administrar y acceder a la base de datos, y administrar y recibir la información que envíen los sensores. Durante la ejecución en segundo plano, los servicios quedarán a la espera hasta recibir nuevos datos de los sensores, momento en el que se lanza el método del objeto receptor denominado *onSensorChanged* encargado de almacenar en la base de datos local la información recibida. Finalmente, cuando se vuelve a pulsar el botón GRABAR se detienen los servicios. En el anexo

7.1.1.6 se adjunta el fichero *AndroidManifest* con información de la APP, componentes, permisos, etc.

La comunicación del dispositivo móvil con el servidor se realiza a través de un servicio API REST, protocolo HTTP y método POST. Cuando el dispositivo móvil tiene una conexión con el servidor, envía la información almacenada temporalmente en su base de datos local y el servidor la almacena en una base de datos MySQL. Así mismo, el servidor permite el acceso a la información a través del servicio API REST para los clientes que desean descargar los datos en su computadora local o a través del servicio WEB utilizando la aplicación RStudio (Figura 2).

2.3 El paquete estadístico R

La herramienta utilizada para el procesamiento y análisis estadístico de los datos, y su posterior utilización en los diferentes algoritmos de aprendizaje automático fue el paquete estadístico R, un entorno de trabajo que se extiende más allá del propio lenguaje de programación R permitiendo integrar con facilidad multitud de paquetes o librerías de funciones creadas por la comunidad.

R es un lenguaje de programación interpretado orientado a objetos creado por Ross Ihaka y Robert Gentleman en 1996 (Ihaka y Gentleman, 1996). Es un lenguaje gratuito de código abierto bajo licencia GNU GPL⁷ muy utilizado por la comunidad científica, dispone de gran cantidad de herramientas matemáticas, estadísticas y gráficas y es capaz de conectarse a las bases de datos más utilizadas.

⁷ Licencia Pública General

2.4 Caso de estudio

La recogida de datos se realizó en el buque pesquero y de investigación oceanográfica Miguel Oliver (Figura 4) de la Secretaría General de Pesca del Ministerio de Agricultura, Pesca y Alimentación de España, durante la campaña de arrastre de fondo MEDITS GSA-6 que realiza anualmente el Instituto Español de Oceanografía (IEO) para evaluar los recursos demersales a lo largo de la plataforma continental y talud de la costa mediterránea española. En concreto, la recogida de datos se realizó desde el 28 de mayo al 1 de junio de 2016 en las costas de Castellón, Tarragona, Barcelona y Gerona.



Figura 4. Buque oceanográfico Miguel Oliver

El buque Miguel Oliver tiene una eslora de 70 m, una manga de 14.4 m, un desplazamiento de 1981 Tn y un sistema de propulsión diésel-eléctrico de 2000 kW. El arte de pesca utilizado en los lances de muestreo se diseñó específicamente para el proyecto MEDITS (G.O.C.-73) (Bertrand *et al.*, 2002; Carpentieri *et al.*, 2020). Es una red de arrastre de dimensiones reducidas (16 m de abertura horizontal y 2.7-3.2 m de abertura vertical) y 20 mm de malla de copo (Figura 5). Para conseguir la máxima eficacia, los lances se deben efectuar a una velocidad constante de 2.9-3.0 kn (Dremlère *et al.*, 1999; Fiorentini *et al.*, 1999).

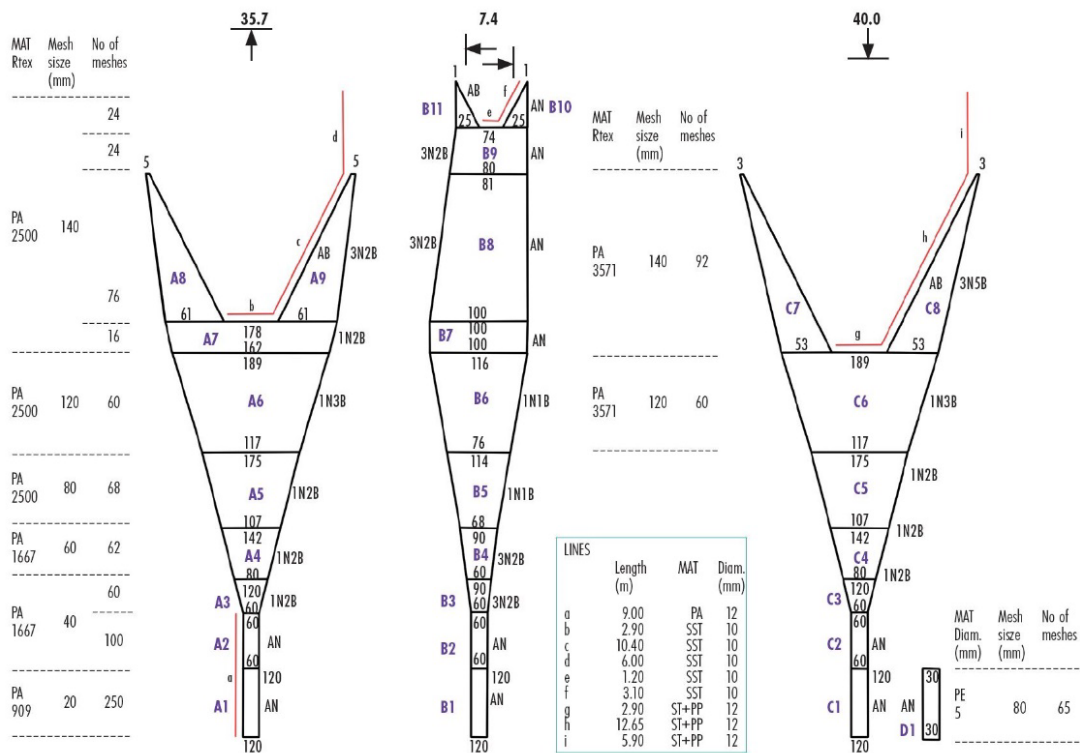


Figura 5. Arte de pesca de arrastre GOC73 utilizada en la campaña MEDITS. (Carpentieri et al., 2020)



Figura 6. Maniobra de largado de arte de pesca de arrastre en el buque Miguel Oliver durante la campaña MEDITS del año 2016.

Con el fin de registrar con la mayor precisión posible el movimiento del buque, el dispositivo móvil con la app FAMIS se colocó en el puente a unos 18 m sobre el centro de gravedad del buque, punto de cruce de los ejes de balanceo y cabeceo (Ibrahim y Grace, 2010), orientando el dispositivo con el eje OY paralelo al eje longitudinal (popa-proa), el eje OX paralelo al eje transversal (babor-estribor) y el eje OZ paralelo al eje vertical. La información vectorial registrada por los sensores hace referencia a estos ejes de coordenadas (Figura 7).

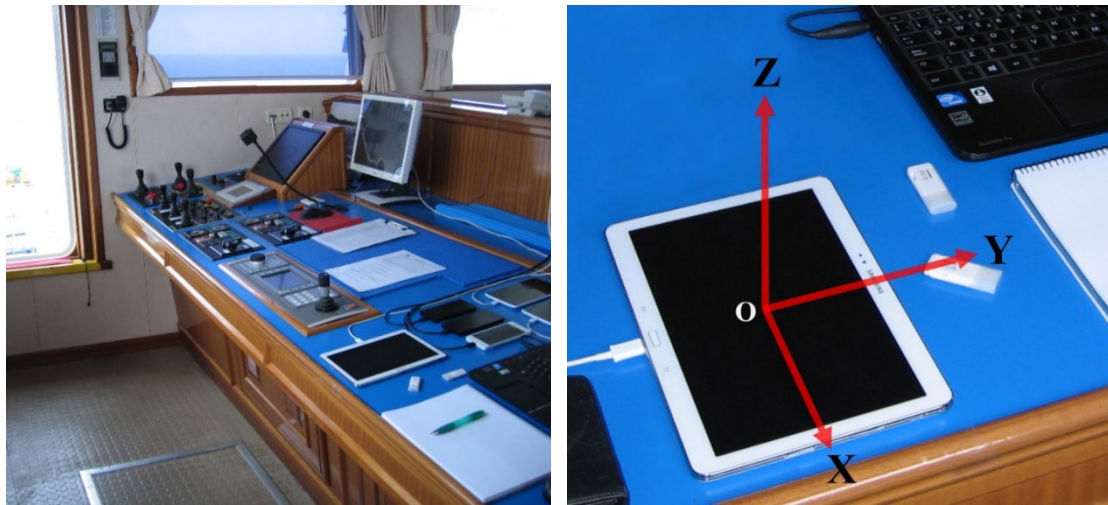


Figura 7. Ubicación del dispositivo móvil en el puente del buque oceanográfico. Las flechas de color rojo muestran los ejes de referencia de los datos vectoriales proporcionados por los sensores del dispositivo.

Se registró el movimiento del buque a lo largo de 22 lances de arrastre de muestreo a una velocidad de 3 kn, distribuidos en 20 lances de 30 min a profundidades de 50 a 200 m, y 2 lances de 60 min a profundidades mayores de 200 metros (Figura 8). Así mismo, como el arrastre de un arte de pesca afecta al movimiento del buque (Sun *et al.*, 2011), se dividió la actividad pesquera del buque en 4 fases: NAVEGAR, LARGAR, ARRASTRAR y VIRAR. Durante la fase NAVEGAR, el barco se desplaza sin ningún arte de pesca en el agua a una velocidad previsiblemente mayor que la velocidad de arrastre. A la llegada del barco al caladero se inicia el lance con la fase LARGAR, arriando⁸ por la popa el arte, maletas, puertas y cables de arrastre. En esta fase se requiere cierta velocidad del buque para que la resistencia que ofrece el agua

⁸ En Náutica: soltar o aflojar un cabo, un cable o una cadena

ayude a largar el arte y su correcto descenso hasta la profundidad de arrastre. En ese momento se inicia la fase **ARRASTRAR** a una velocidad constante de 3 kn aproximadamente, aumentando la resistencia al avance que ofrece el aparejo en paralelo al incremento de capturas. Cuando finaliza el arrastre se inicia la fase **VIRAR** recogiendo el aparejo y la captura en orden inverso al largado del arte, reduciendo la velocidad del barco para facilitar su ascenso y recogida a bordo.

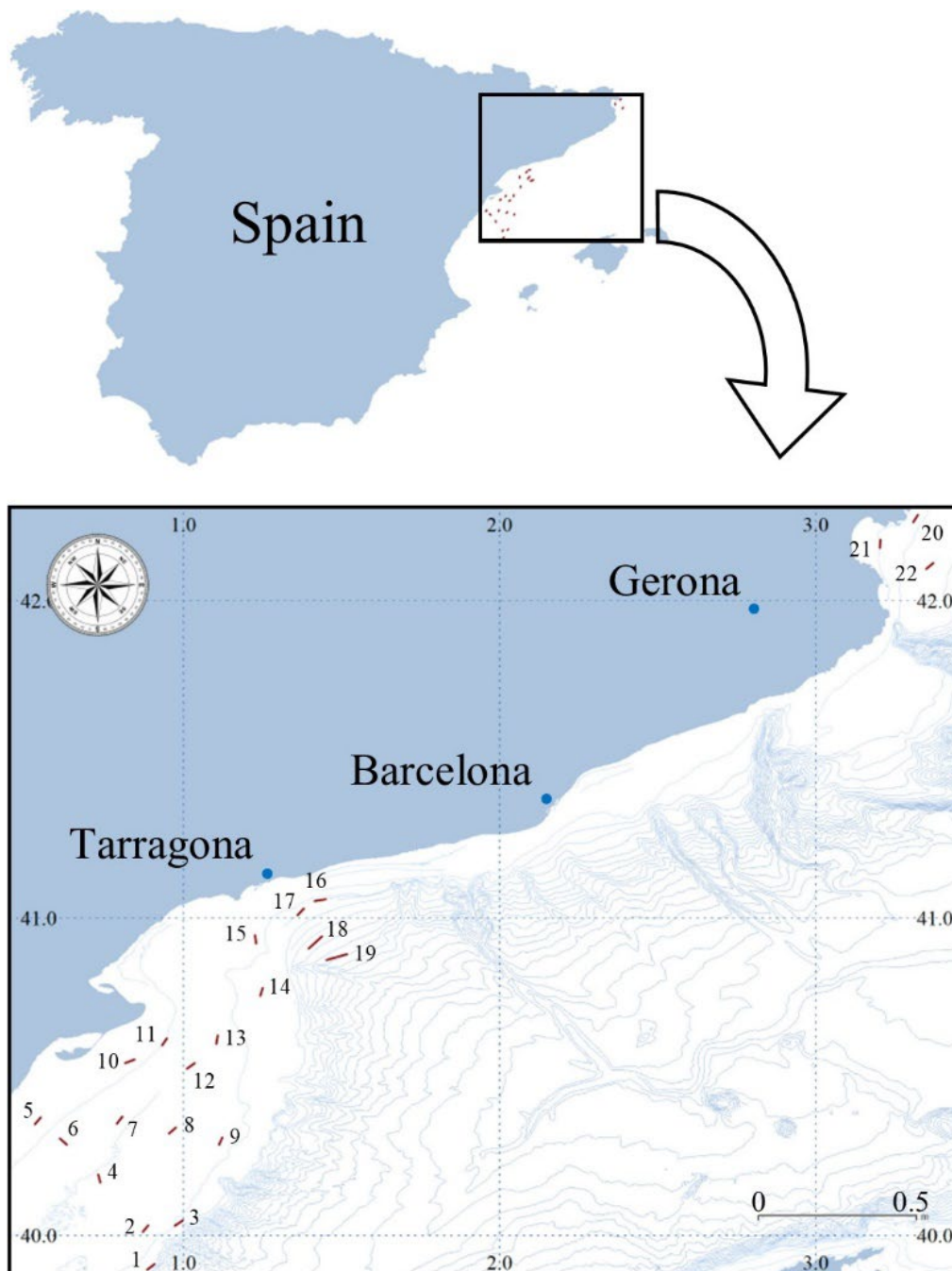


Figura 8. Distribución espacial de los 22 lances registrados a lo largo de las costas de Castellón, Tarragona, Barcelona y Gerona (España)

2.5 Procesado de datos y creación de las tablas de datos

El procesado de datos se inició filtrando los ceros y los valores atípicos por considerar que esos datos se registraron por error debido a alguna anomalía de los sensores. A continuación, se creó una primera tabla de datos para su análisis estadístico, con 22 variables continuas cuantitativas independientes (atributos) con la media (X_m) y la desviación estándar (X_s) de los datos registrados por los sensores en períodos de 1 minuto (2453 muestras). Esta aproximación tipo ventana y no tipo punto ha demostrado ser más efectiva en la identificación de lances pesqueros a partir de datos VMS (O'Farrell *et al.*, 2017). Así mismo, se añadió a la tabla una variable entera con el número de lance y una variable cualitativa nominal independiente (etiqueta) denominada *Actividad* para identificar la fase del lance de cada muestra con la información registrada y verificada con los datos proporcionados por el jefe de la campaña MEDITS.

La tabla de datos normalizada se creó fusionando los 22 atributos ($AceX_m$, $AceX_s$, $AceY_m$, $AceY_s$, $AceZ_m$, $AceZ_s$, $GirX_m$, $GirX_s$, $GirY_m$, $GirY_s$, $GirZ_m$, $GirZ_s$, $MagX_m$, $MagX_s$, $MagY_m$, $MagY_s$, $MagZ_m$, $MagZ_s$, $GpsR_m$, $GpsR_s$, $GpsV_m$ y $GpsV_s$), el número de lance y la etiqueta *Actividad*, eliminando los registros con algún atributo nulo y normalizando atributos mediante el método T de Student (Ecuación 3), siendo X el valor, $\bar{X}$ la media y S la desviación estándar.

$$N = \frac{X - \bar{X}}{S} \quad (3)$$

En el anexo 7.1.2.1 se incluye el código en lenguaje R que crea la tabla de datos a partir de los datos brutos registrados por los sensores.

Para entrenar y evaluar con mayor precisión la capacidad de las máquinas de aprendizaje fue necesario ampliar el número de muestras de cada fase del lance. Para obtener la nueva tabla de datos se redujo la ventana temporal a 10 segundos, tiempo suficiente para que la embarcación complete como mínimo medio ciclo de balanceo o cabeceo (Barrass y Derrett, 2012). Se repitió el proceso anterior de creación de tablas calculando la media y la desviación estándar los datos registrados en períodos de 10 segundos de las 22 variables cuantitativas independientes y normalizando los datos. Una vez creada la tabla normalizada, se balanceó la tabla a nivel de lance para igualar el número de muestras por actividad en cada lance conservando las muestras próximas a los cambios de actividad, obteniendo un total de 6184 muestras con 1546 muestras por actividad (anexo 7.1.2.2).

Posteriormente, para entrenar y evaluar la capacidad de clasificación de las máquinas de aprendizaje se dividió la tabla de datos balanceada en dos bloques: uno destinado a datos de entrenamiento con 3844 muestras (62.16%) de los 3 primeros días de campaña registrando un total de 15 lances y otro destinado a ser utilizado como datos de test con 2340 muestras (37.83%) de los 2 últimos días registrando un total de 7 lances. Este procedimiento permite entrenar y probar los modelos en diferentes condiciones meteorológicas y ambientales.

2.6 Análisis de datos

2.6.1 Análisis descriptivo de los datos

El análisis de los datos se centró en identificar los cambios en el comportamiento estático y dinámico del buque durante el lance, en cómo afecta la presencia de un arte de pesca activo en los ángulos de escora transversal y longitudinal, y movimiento del buque representado por sus 6 grados de libertad (Figura 9), en identificar los sensores capaces de medir esos cambios, en localizar las variables cuantitativas de la tabla de datos susceptibles de ser utilizadas para reconocer dichos cambios, y en analizar el comportamiento de dichas variables en las 4 fases de actividad.



Figura 9. Grados de libertad de movimiento del buque alrededor y a lo largo de su eje longitudinal (balance y avance), transversal (cabeceo y deriva) y vertical (guiñada y arfada).

Se analizó cada variable cuantitativa de la tabla de datos normalizada utilizando el coeficiente de curtosis de Fischer para identificar la concentración de los datos de la variable alrededor de la media respecto a una distribución normal, el coeficiente de asimetría de Fisher para evaluar la simetría de la distribución de la variable, el test de Shapiro-Wilk para contrastar la normalidad de los datos, el test de Levene para evaluar la igualdad de varianzas entre dos o más grupos de la variable, los diagramas de cajas y densidad por actividad para observar diferencias significativas en la variable durante los lances, la mediana y rango intercuartílico por actividad para detectar diferencias en su valor central y dispersión, su coeficiente de variación de la desviación absoluta de la mediana (MAD) para descartar variables poco representativas y el test no paramétrico de Kruskal-Wallis para comprobar que todas las muestras de la variable no pertenecen a la misma población y confirmar las diferencias apreciadas entre los valores de las variables en cada fase o actividad pesquera.

En el anexo 7.1.3 se incluye el código en R utilizado para obtener la descripción y representación gráfica de cada variable. A continuación, se incluyen las funciones que se utilizaron para el análisis descriptivo de los datos:

skewness: calcula la asimetría de los datos. Librería *moments* (Komsta and Novomestky, 2015).

skewness(x, na.rm = FALSE)

donde *x* es un vector numérico, matriz o un data.frame y *na.rm* indica si deben eliminarse los valores NA (Not Available).

kurtosis: calcula el estimador de la medida de curtosis de Pearson. Librería *moments* (Komsta y Novomestky, 2015).

kurtosis(x, na.rm = FALSE)

donde *x* es un vector numérico, matriz o un data.frame y *na.rm* indica si deben eliminarse los valores NA.

levene.test: prueba la igualdad de las *k* varianzas poblacionales. Librería *lawstat* (Gastwirth *et al.*, 2020).

*levene.test(y, group, location = c("median", "mean", "trim.mean"),
trim.alpha = 0.25, bootstrap = FALSE, num.bootstrap = 1000,
kruskal.test = FALSE, correction.method = c("none",
correction.factor", "zero.removal", "zero.correction"))*

donde *y* es un vector numérico de valores de datos, *group* el factor de los datos, *location* procedimiento utilizado (predeterminado "median"), *trim.alpha* la fracción (de 0 a 0.5) de observaciones que se recortarán de cada final de *x* antes de que se calcule la media, *bootstrap* un valor lógico que identifica si se debe implementar bootstrap, *num.bootstrap* el número de muestras de Bootstrap, *kruskal.test* el valor lógico que identifica si se debe utilizar Kruskal-Wallis en lugar de ANOVA, *correction.method* son procedimientos para hacer la prueba más robusta.

kruskal.test: realiza la prueba Kruskal-Wallis. Librería *stats* (R-proyect, 2022).

kruskal.test(x, formula, data, subset, na.action, ...)

donde *x* es un vector numérico de valores de datos, *formula* utilizada para describir los datos, *data* un data.frame de datos, *subset* un vector opcional que especifica un subconjunto de observaciones y *na.action* indica lo que debe hacer con los datos que contienen NA.

2.6.2 Análisis multivariante

El análisis multivariante engloba diferentes métodos estadísticos para analizar simultáneamente medidas múltiples de cada individuo u objeto sometido a investigación (Hair *et al.*, 1999). En este trabajo, se utilizaron la matriz de correlación de Pearson para medir el grado de relación lineal entre variables y el análisis de componentes principales (PCA) para intentar reducir su número.

2.6.2.1 Matriz de correlación de Pearson

La matriz de correlación muestra en una tabla de dos dimensiones el índice de correlación de Pearson entre las variables que intervienen en el estudio y el nivel de significancia de los índices (*p*-valor). Cada índice muestra la dependencia lineal entre dos variables cuantitativas continuas, mostrando valores comprendidos entre -1 y 1.

En este estudio se utilizó la matriz de correlación para identificar variables correlacionadas con objeto de reducir en una primera fase, el número de variables.

Para calcular en R la matriz de correlación (Anexo 7.1.4) se utilizó la función *rcorr* de la librería *Hmiscs* (Frank, 2021):

rcorr: calcula la matriz simétrica con los índices de correlación de Pearson o Spearman, el número de observaciones y la matriz simétrica con los *p*-valores.

$$rcorr(x, y, type=c("pearson", "spearman"))$$

donde *x* es una matriz numérica con al menos 5 filas y al menos 2 columnas, y un vector numérico o una matriz que se podrá concatenar con *x* y *type* especifica el tipo de correlación a calcular.

2.6.2.2 Análisis de componentes principales

El análisis de componentes principales (PCA) es una técnica estadística utilizada para analizar interrelaciones entre un gran número de variables intentando encontrar un modo de condensar la información en un número más pequeño de variables no correlacionadas con la mínima pérdida de información (Amat-Rodrigo, 2017; Garza-García *et al.*, 2013; Hair, *et al.*, 1999).

Las nuevas variables denominadas componentes se obtienen a partir de combinaciones lineales ortogonales de las variables originales siguiendo un orden de importancia en cuanto a la variabilidad total que recogen de las muestras.

Si se representa por $(X_1, X_2, \dots, X_p)$ el grupo de variables originales, el primer componente principal sería:

$$Z_1 = \phi_{11}X_1 + \phi_{21}X_2 + \dots + \phi_{p1}X_p \quad (4)$$

Para que la combinación lineal sea normalizada, debe cumplirse:

$$\sum_{j=1}^p \phi_{j1}^2 = 1 \quad (5)$$

El término ϕ_{i1} representa el peso o importancia de la variable X_i en el componente Z_1 . A continuación, se calcula el segundo componente Z_2 repitiendo el proceso, pero añadiendo la condición de que el nuevo componente no puede estar correlacionado con el primero, lo que equivale a decir que son perpendiculares. El proceso se repite hasta calcular todos los componentes. El cálculo correcto de los componentes

principales requiere que no existan valores atípicos y que los datos estén estandarizados para que ninguna variable domine al resto.

Desde el punto de vista geométrico, si se representan las muestras en un espacio p -dimensional, el objetivo del proceso PCA sería encontrar un espacio de dimensión igual o inferior que mantenga la máxima información original posible y donde cada componente principal represente los nuevos ejes de coordenadas. Por ejemplo, si partimos con 2 variables originales (X_1, X_2) podríamos representar cada muestra como un punto en el espacio bidimensional (Figura 10). El vector que define el primer componente principal Z_1 muestra la dirección donde las observaciones tienen mayor variabilidad. La proyección de cada observación sobre esa dirección equivale al valor del primer componente de dicha observación. El segundo componente Z_2 muestra la dirección perpendicular a Z_1 con mayor varianza de los datos.

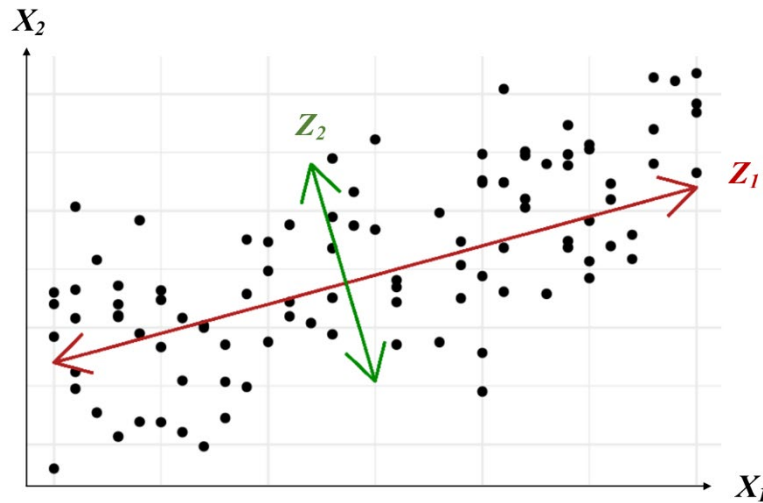


Figura 10. Representación gráfica de las muestras originales en el plano X_1 - X_2 y de las componentes principales Z_1 y Z_2 .

Para evaluar cuanta información del conjunto de datos original se pierde al proyectar las observaciones en un espacio de dimensión inferior se recurre a la proporción de varianza explicada por cada componente principal.

Con un conjunto de datos normalizado, la varianza total presente se define como:

$$\sum_{j=1}^p \text{Var}(X_j) = \sum_{j=1}^p \frac{1}{n} \sum_{i=1}^n x_{ij}^2 \quad (6)$$

La varianza explicada por el componente m se denomina valor propio (VP) y su valor es:

$$VP_m = \frac{1}{n} \sum_{i=1}^n z_{im}^2 = \frac{1}{n} \sum_{i=1}^n \left(\sum_{j=1}^p \phi_{jm} x_{ij} \right)^2 \quad (7)$$

El porcentaje de varianza explicada (VE) por el componente m viene dada por:

$$VE_m = \frac{\sum_{i=1}^n \left(\sum_{j=1}^p \phi_{jm} x_{ij} \right)^2}{\sum_{j=1}^p \frac{1}{n} \sum_{i=1}^n x_{ij}^2} \quad (8)$$

La cantidad de componentes principales seleccionados condicionará la varianza acumulada por todos ellos y, en consecuencia, la pérdida de información durante la reducción de dimensiones o número de variables de entrada. Habitualmente se selecciona el número de componentes principales a partir del cual, al añadir nuevas componentes no se produce un incremento sustancial en la varianza explicada acumulada.

La librería R utilizada (Anexo 7.1.4) para el cálculo de los componentes principales se denomina *FactoMineR* (Husson, F. *et al.*, 2022) con la función:

PCA: Realiza un análisis de componentes principales (PCA) con individuos suplementarios, variables cuantitativas suplementarias y variables categóricas suplementarias.

PCA(X , *scale.unit* = *TRUE*, *ncp* = 5, *ind.sup* = *NULL*, *quanti.sup* = *NULL*, *quali.sup* = *NULL*, *row.w* = *NULL*, *col.w* = *NULL*, *graph* = *TRUE*, *axes* = *c(1,2)*)

donde X es la tabla de datos con n filas (individuos) y p columnas (variables numéricas), *ncp* el número de dimensiones que se mantienen en los resultados (por defecto 5), *scale.unit* un booleano, si es *TRUE* (valor establecido por defecto) entonces los datos se escalan a la varianza unitaria, *ind.sup* un vector que indica los índices de los individuos suplementarios, *quanti.sup* un vector que indica los índices de las variables suplementarias cuantitativas, *quali.sup* un vector que indica los índices de las variables complementarias categóricas, *row.w* un vector opcional de pesos de fila (por defecto, vector de unos); los

pesos se dan sólo para los individuos activos, *col.w* una ponderación de columna opcional (por defecto, ponderación de columna uniforme); las ponderaciones se dan sólo para las variables activas, *graph* booleano, si es TRUE se muestra un gráfico y *axes* un vector de longitud 2 que especifica los componentes a representar.

2.7 Técnicas de aprendizaje automático

El aprendizaje automático abarca una serie de técnicas que permiten a los sistemas informáticos crear algoritmos capaces de analizar los datos y adquirir los conocimientos necesarios para realizar tareas como predecir, clasificar, ordenar o tomar decisiones, sin necesidad de definir reglas iniciales que faciliten dichas tareas.

En este trabajo se seleccionaron 4 técnicas de clasificación de aprendizaje automático supervisado con estrategias muy distintas: (i) la técnica estadística denominada análisis discriminante lineal (LDA) que clasifica muestras linealmente separables estimando la probabilidad de pertenencia a cada grupo; (ii) perceptrón multicapa (MLP) que es un aproximador universal de funciones no lineales; (iii) red neuronal probabilística (PNN) como clasificador bayesiano y; (iv) la máquina de soporte vectorial (SVM) que clasifica muestras no linealmente separables con un enfoque geométrico.

Así mismo, se utilizaron diferentes librerías del paquete estadístico R-Studio para implementar los cuatro algoritmos de aprendizaje seleccionados: *MASS* (Ripley *et al.*, 2020) para LDA, *RSNNS* (Bergmeir *et al.*, 2019) para MLP, *PNN* (Chasset, 2016) y *e1071* (Meyer, 2019) para SVM. En el anexo 7.1.5 se aporta la función en R que entrena y obtiene la predicción de los cuatro algoritmos.

2.7.1 Análisis Discriminante Lineal (LDA)

El análisis discriminante lineal (LDA) es un algoritmo de clasificación supervisado de variables cualitativas en el que dos o más grupos son conocidos *a priori* y nuevas observaciones se clasifican en uno de ellos en función de sus características (Amat-Rodrigo, 2016).

Para facilitar la tarea de clasificación, LDA utiliza previamente el método propuesto por Fisher para reducir el número de variables cualitativas aproximando el espacio n -dimensional actual (donde n es el número de variables originales) a un espacio de

menor dimensión formado por combinaciones lineales de las variables originales que mejor explican la separación de las clases (Fisher, 1936). El objetivo del método Fisher es encontrar esa transformación lineal que maximice la dispersión entre clases y minimice la dispersión dentro de cada clase.

Definido el nuevo espacio y transformados los datos de entrenamiento, LDA utiliza el teorema de Bayes para estimar la probabilidad de que una observación pertenezca a cada una de las clases de la variable cualitativa, asignando la observación a la clase más probable.

El teorema de Bayes establece que ante dos eventos A y B, la probabilidad de que ocurra B habiendo ocurrido A es igual a la probabilidad de que A y B ocurran al mismo tiempo dividida por la probabilidad de que ocurra A:

$$P(B|A) = \frac{P(AB)}{P(A)} \quad (9)$$

Suponiendo que se desea clasificar una nueva observación X con una sola variable en una de las K clases ($K \geq 2$) de la variable cualitativa Y , la probabilidad de que la observación x pertenezca a la clase k sería:

$$P(\text{valor } x \text{ observado}) = \frac{P(\text{pertenecer a la clase } k \text{ y observar } x)}{P(\text{observar } x)} \quad (10)$$

Formalmente, se puede expresar como:

$$P(X = x) = \frac{\pi_k P(Y = k)}{\sum_{j=1}^K \pi_j P(Y = j)} = \frac{\pi_k f_k(x)}{\sum_{j=1}^K \pi_j f_j(x)} \quad (11)$$

siendo π_k la probabilidad previa de que una observación aleatoria pertenezca a la clase k y $f_k(X)$ la función de densidad de probabilidad condicional de X para una observación que pertenece a la clase k .

Para que la clasificación basada en Bayes sea posible, se necesita conocer la probabilidad poblacional de que una observación cualquiera pertenezca a cada clase π_k y la probabilidad poblacional de que una observación que pertenece a la clase k adquiera el valor x en la variable de entrada $f_k(X)$. La probabilidad *a priori* π_k suele ser igual al número de observaciones de la clase k dividido por el número total de

observaciones. En cambio, la estimación de $f_k(X)$ requiere conocer cómo se distribuyen los datos de en la clase k , que en el caso de una distribución normal sería:

$$f_k(x) = \frac{1}{\sqrt{2\pi}\sigma_k} \cdot \exp\left(-\frac{1}{2\sigma_k^2}(x - \mu_k)^2\right) \quad (12)$$

siendo μ_k la media y σ_k^2 la varianza de la clase k .

En la práctica, a pesar de una certeza considerable de que los datos X se distribuyen de forma normal dentro de cada clase, los valores de las medias y varianza de cada clase deben ser estimadas a partir de las observaciones.

En el caso de observaciones X con más de una variable, el procedimiento puede generalizarse considerando que cada variable se distribuye de forma normal en cada una de las clases y que la varianza es igual en todas ellas.

Para implementar en R el algoritmo LDA (anexo 7.1.5) se utilizaron las funciones *lda* y *predict* de la librería *MASS* (Ripley *et al.*, 2020):

lda: crea el modelo LDA

lda(formula, data, prior, ...)

donde *formula* indica la descripción simbólica del modelo a ajustar, *data* los datos de entrenamiento y *prior* la probabilidad previa de cada clase.

predict: predice el porcentaje de pertenencia a cada categoría de un conjunto de observaciones.

predict(object, newdata, ...)

donde *object* es el modelo LDA y *newdata* el conjunto de nuevas observaciones.

2.7.2 Redes neuronales artificiales

Las redes neuronales artificiales son modelos matemáticos inspirados en la arquitectura neuronal de los sistemas biológicos (Rumelhart *et al.*, 1986). Para crear un sistema neuronal artificial se utilizan neuronas artificiales organizadas funcionalmente en capas. La información circula por las capas de la red neuronal a

través de conexiones unidireccionales que simulan las sinapsis entre neuronas. Cada conexión tiene un peso asociado w_i equivalente a la fuerza de la sinapsis, utilizado por la neurona receptora para ponderar la información que recibe x_i por esa conexión de entrada, pudiendo excitar a la neurona (sinapsis con peso positivo) o inhibirla (peso negativo). La neurona receptora calculará la suma ponderada de todas sus entradas, sumará un valor numérico denominado sesgo θ y aplicará al resultado una función de activación f para determinar su salida (Figura 11).

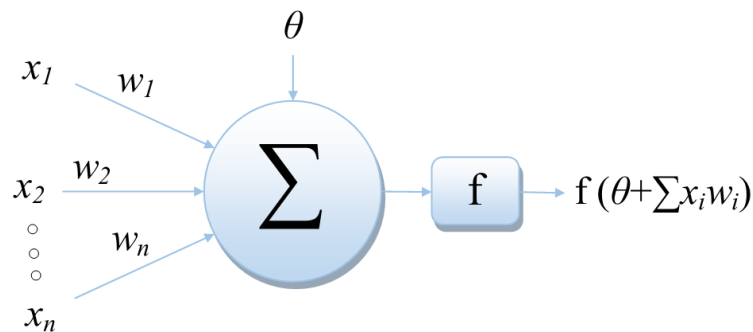


Figura 11. Neurona artificial. x_i representan la i -ésima entrada de la neurona, w_i representa el peso asociado a la entrada i -ésima, Σ representa el sumatorio del producto de entradas y pesos, θ representa el sesgo de la neurona y f la función de activación que proporciona la salida de la neurona.

2.7.2.1 Perceptrón Multicapa (MLP)

El perceptrón multicapa (MLP) es un modelo matemático capaz de aproximar o interpolar relaciones no lineales entre datos de entrada y salida. En este trabajo se explotará la capacidad del MLP para analizar conjuntos de datos complejos y realizar una clasificación no lineal en dos o más grupos, capacidad ampliamente utilizada en diferentes aplicaciones tecnológicas (Lek y Guegan, 1999; Gutiérrez-Estrada *et al.*, 2000; Dedeker *et al.*, 2005; Goethals *et al.*, 2007; Pulido-Calvo y Portela, 2007; Gutiérrez-Estrada *et al.*, 2008).

La arquitectura del MLP se caracteriza porque tiene agrupadas sus neuronas en capas de diferentes niveles. Se distinguen tres tipos de capas: (i) la capa de entrada; (ii) las capas ocultas y; (iii) la capa de salida. Las neuronas de la capa de entrada se encargan de recibir las señales del exterior y propagarlas a todas las neuronas de la siguiente capa o primera capa oculta. El número de neuronas que forman esta capa será igual al

número de atributos de los datos de entrada. Las neuronas de las capas ocultas realizan el procesamiento no lineal de los patrones recibidos. Puede contener un número variable de capas ocultas y neuronas en cada capa. La última capa actúa como salida de la red, proporcionando al exterior la respuesta de la red para cada uno de los patrones de entrada. El número de neuronas de la capa de salida coincidirá con el número de atributos del vector de salida de la red neuronal (Figura 12).

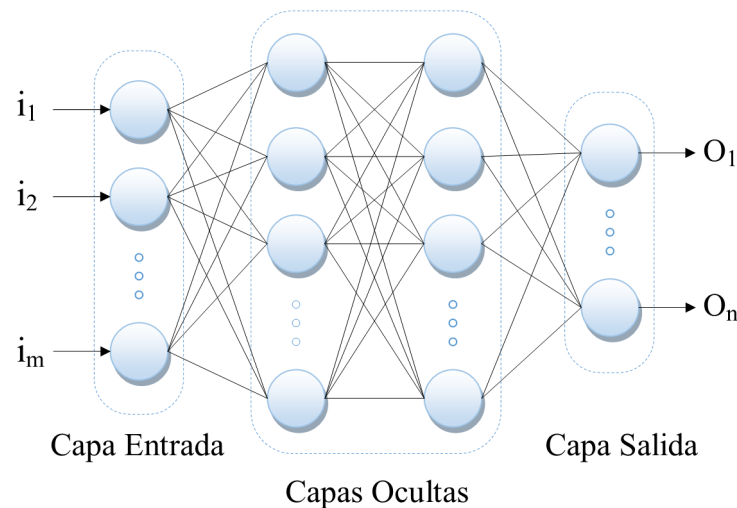


Figura 12. Arquitectura del perceptrón multicapa formada por tipos de capas: la capa de entrada que recibe la información de entrada del perceptrón y con una neurona por cada atributo del patrón de entrada $\vec{i}$, las capas ocultas que procesan la información con un número variable de capas y neuronas en cada capa, y la capa de salida que proporciona la respuesta del perceptrón con un número de neuronas igual al de atributos del vector de salida $\vec{o}$.

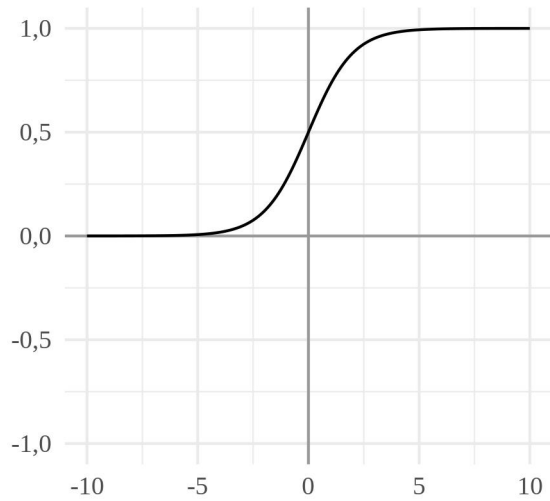
Las conexiones del MLP entre neuronas siempre van dirigidas hacia delante, es decir, las salidas de las neuronas de una capa se conectan con las entradas de las neuronas de la siguiente capa. El MLP es una red totalmente conectada, las neuronas de una capa están conectadas a todas las neuronas de la siguiente capa. Así mismo, en el MLP la información circula en una única dirección desde la capa de entrada hacia la capa de salida, de ahí que reciban el nombre de redes unidireccionales, alimentadas hacia adelante o redes *feedforward*.

El MLP define una relación entre las variables de entrada y las variables de salida de la red. Esta relación se obtiene propagando hacia adelante los valores de las variables de entrada. Cada neurona de la red procesa la información recibida por sus entradas y produce una respuesta (función de activación) que se propaga a través de las conexiones correspondientes hacia las neuronas de la siguiente capa. El diseñador deberá elegir las funciones de activación de cada neurona basándose en la tarea

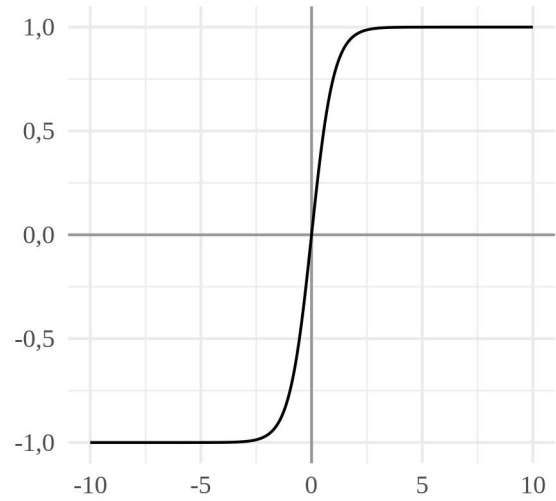
específica que realiza la capa a la que pertenece. Es habitual encontrar en el MLP funciones de activación lineales y no lineales sigmoide, ReLU, leaky ReLU, tangente hiperbólica, etc (Liew *et al.*, 2016; Mathias y Rech, 2012; Rynkiewicz, 2019) (Figura 13).

El número de neuronas en la capa de entrada y salida viene dado en ocasiones, por las variables que definen el problema y la información que deseamos que proporcione la red neuronal. En otros casos, se dispone de un gran número de variables de entrada de las que no se conoce su relevancia en la resolución del problema y su utilización implicaría arquitecturas de gran tamaño y complejidad. En estas situaciones, es conveniente realizar un análisis previo de las variables de entrada más relevantes al problema y descartar aquellas que no aportan información a la red, como por ejemplo, análisis de correlación, análisis de componentes principales, etc (Gutiérrez-Estrada *et al.*, 2009; Yáñez *et al.*, 2010).

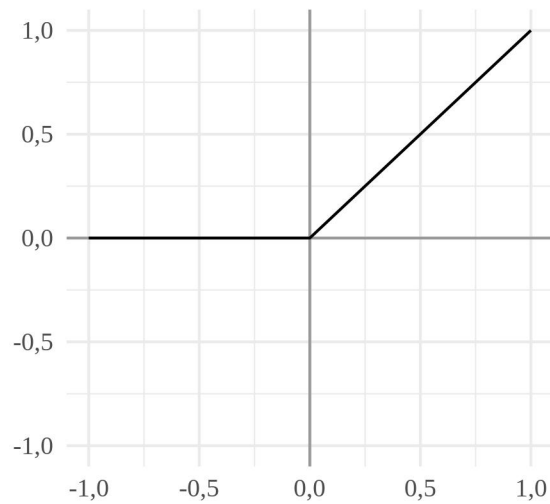
En cuanto al número de capas ocultas y de neuronas en cada una de ellas, no existe un método o regla que determine el número óptimo de capas y neuronas para resolver un problema dado. En la mayor parte de las aplicaciones, estos parámetros se resuelven mediante el método de prueba y error. En la actualidad existen líneas de investigación abiertas centradas en la resolución automática de este problema.



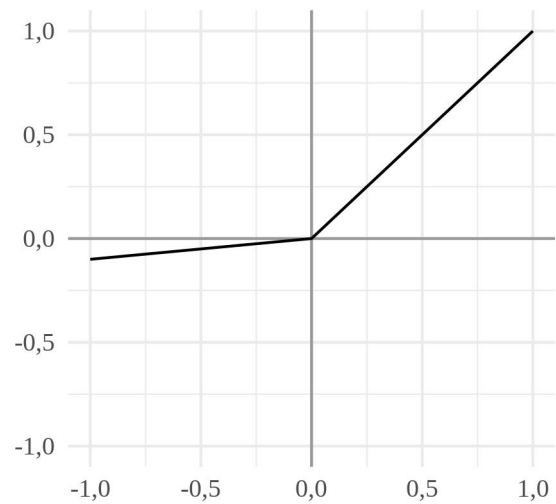
Sigmoide: $\sigma(x) = \frac{1}{1+e^{-x}}$



Tangente hiperbólica: $\tanh(x)$



ReLU: $\max(0, x)$



Leaky ReLU: $\max(0.1x, x)$

Figura 13. Funciones de activación.

Una vez definida la arquitectura del MLP se pasa a la fase de aprendizaje, donde se calculan el resto de parámetros de la red (pesos de las conexiones y umbrales asociados a cada neurona) para que la red neuronal realice una tarea específica. El MLP utiliza para ello un algoritmo de aprendizaje supervisado, donde los valores iniciales de los parámetros se van adaptando para que la salida de la red sea lo más próxima posible a la salida proporcionada por el supervisor o salida deseada. Para ello, los patrones de entrada utilizados para el aprendizaje del MLP denominados patrones o datos de entrenamiento, necesitan disponer de sus correspondientes patrones de salida deseada.

Puesto que el objetivo es que la salida de la red sea lo más próxima posible a la salida deseada, el aprendizaje de la red se reduce a un problema de minimización de la función de error entre salidas de la red y salidas deseadas. La presencia de funciones de activación no lineales hace que la respuesta de la red sea no lineal respecto al ajuste de los parámetros, por lo que el problema de minimización es un problema no lineal, y, como consecuencia, tienen que utilizarse técnicas de optimización no lineales para su resolución.

El algoritmo de aprendizaje que habitualmente se utiliza en el MLP se denomina retroprogramación o *backpropagation* (Werbos, 1974). En la primera fase del algoritmo, el patrón de entrada se presenta a la red y se propaga a través de las capas hasta llegar a la capa de salida, obteniéndose los valores de salida de la red. La segunda fase se inicia cuando se comparan los valores obtenidos con los valores de salida esperados para así obtener el error. El error se transmite hacia atrás desde la capa de salida hasta la capa de entrada, recibiendo cada neurona su contribución o aportación relativa al error total. Basándose en el valor del error recibido, se reajustan los pesos de conexión de cada neurona, de manera que la siguiente vez que se presente el mismo patrón, la salida esté más cercana a la deseada y, por tanto, disminuya el error. Este proceso se repite para todos los datos o patrones de entrenamiento. Para mejorar el ajuste, el algoritmo de aprendizaje se repite varias veces hasta que el error calculado de la salida es inferior al permitido o por alcanzar el número de iteraciones máximo establecido (Anzola, 2015).

La formulación matemática del algoritmo de *backpropagation* con una red MLP con una capa de entrada de N neuronas, una capa oculta de L neuronas y una capa de salida de M neuronas, se podría describir en los siguientes pasos (Anzola, 2015):

1. Se inicializan los pesos del MLP.
2. Mientras la condición de parada sea falsa se ejecutan los pasos 3 al 10.
3. Se aplica el vector de entradas a la red:

$$x_p = [x_{p1}, x_{p2}, \dots, x_{pi}, \dots, x_{pN}]^T \quad (13)$$

siendo x_p el patrón o vector de entrada, x_{pi} la entrada i -ésima del vector de entrada x_p y T la transpuesta del vector.

4. Se calcula la salida de las neuronas de la capa oculta:

$$i_{pj}^h = f_j^h \left(\sum_{i=1}^N w_{ji}^h x_{pi} + \theta_j^h \right) \quad (14)$$

siendo i_{pj}^h la salida de la j -ésima neurona de la capa oculta, f_j^h la función de activación de la j -ésima neurona de la capa oculta, N el número de neuronas de la capa de entrada, w_{ji}^h el peso de interconexión entre la neurona i -ésima de la entrada y la j -ésima de la capa oculta y θ_j^h el sesgo de la neurona j -ésima de la capa oculta.

5. Se calculan las salidas de la red:

$$y_{pk}^o = f_k^o \left(\sum_{j=1}^L w_{kj}^o i_{pj} + \theta_k^o \right) \quad (15)$$

siendo y_{pk}^o la salida de la k -ésima neurona de la capa de salida y f_k^o la función de activación de la k -ésima neurona de la capa de salida, L el número de neuronas de la capa oculta, w_{kj}^o el peso de interconexión entre la j -ésima neurona de la capa oculta y la k -ésima neurona de la capa de salida, i_{pj} la salida de la j -ésima neurona de la capa oculta y θ_k^o el sesgo de la k -ésima neurona de la capa de salida.

6. Se calculan los términos de error para las neuronas de la capa de salida:

$$\delta_{pk}^o = (d_{pk} - y_{pk}^o) \quad (16)$$

siendo δ_{pk}^o el término de error para la k -ésima neurona de la capa de salida y d_{pk} el valor de la salida deseado para la k -ésima neurona de la capa de salida.

7. Se estiman los términos de error para las neuronas de la capa oculta:

$$\delta_{pj}^h = \sum_{k=1}^M \delta_{pk}^o w_{kj}^o \quad (17)$$

siendo δ_{pj}^h el término de error para la j -ésima neurona de la capa oculta y M el número de neuronas de la capa de salida.

8. Se actualizan los nuevos pesos de las neuronas de la capa de salida:

$$w_{kj}^o(t+1) = w_{kj}^o(t) + \alpha \delta_{pk}^o i_{pj}^h \quad (18)$$

siendo α la tasa o velocidad de aprendizaje

9. Se actualizan los nuevos pesos de las neuronas de la capa oculta:

$$w_{ji}^h(t+1) = w_{ji}^h(t) + \alpha \delta_{pj}^h x_{pi} \quad (19)$$

10. Se verifica si el error global cumple con la condición de finalizar:

$$E_p = \frac{1}{2} \sum_{p=1}^P \sum_{k=1}^M (d_{pk} - y_{pk})^2 \quad (20)$$

siendo P el número de vectores de entrada utilizados para entrenar la red y M el número de neuronas de la capa de salida.

Para implementar en R el perceptrón multicapa se utilizaron las funciones *mlp* y *predict* de la librería *RSNNS* (Bergmeir *et al.*, 2019):

mlp: crea una nueva red neuronal MLP y la entrena.

```
mlp(x, y, size, maxit, initFunc, initFuncParams, learnFunc,
    learnFuncParams, updateFunc, updateFuncParams,
    hiddenActFunc, shufflePatterns, linOut, outputActFunc,
    inputsTest, targetsTest, pruneFunc, pruneFuncParams)
```

donde x es una matriz con entradas de entrenamiento para la red, y los valores de destino correspondientes, $size$ el número de neuronas en las capas ocultas, $maxit$ el máximo de iteraciones de aprendizaje, $initFunc$ la función de inicialización, $initFuncParams$ los parámetros de la función de inicialización, $learnFunc$ la función de aprendizaje, $learnFuncParams$ los parámetros de la función de aprendizaje, $updateFunc$ la función de actualización, $updateFuncParams$ los parámetros de la función de actualización, $hiddenActFunc$ la función de activación de todas las neuronas ocultas, $shufflePatterns$ indica si se barajan los patrones, $linOut$ tipo de salida de función de activación de las unidades de salida (lineal o lógica), $outputActFunc$ la función de activación de todas las unidades de salida, $inputsTest$ la matriz con entradas para probar la red, $targetsTest$ los objetivos correspondientes para la entrada de prueba, $pruneFunc$ la función de poda y $pruneFuncParams$ los parámetros de la función de poda.

predict: predice el porcentaje de pertenencia a cada categoría de un conjunto de observaciones.

predict(object, newdata, ...)

donde *object* es la red neuronal MLP y *newdata* el conjunto de nuevas observaciones.

2.7.2.2 Red Neuronal Probabilística (PNN)

La red neuronal probabilística (PNN) es un clasificador de la familia de las redes neuronales con función base radial (Specht, 1990). La clasificación se realiza a través del análisis de las similitudes que guarda el patrón de entrada con los patrones de entrenamiento, utilizando para ello la distancia euclídea entre patrones. Los clasificadores PNN son de gran utilidad en muchos sectores, incluido el sector pesquero (Gutiérrez-Estrada y Pulido-Calvo, 2015; Andayani *et al.*, 2019; Fontana *et al.*, 2017; Mustakim *et al.*, 2018; Oliva-Teles *et al.*, 2015; Robertson y Morison, 2006; Tu *et al.*, 2020).

La red PNN está formada por cuatro capas: una capa de entrada con un número de neuronas igual al de atributos de entrada, una primera capa oculta de patrones con un número de neuronas igual al de muestras de entrenamiento o calibración, una segunda capa oculta sumatoria con un número de neuronas igual al número de clases y una capa de salida con una neurona que proporciona el resultado de la clasificación (Figura

14). La red PNN no es una red totalmente conectada. Las salidas de las neuronas de la capa de entrada se conectan a las entradas de todas las neuronas de la capa patrón, en cambio, la salida de las neuronas de la capa patrón se conectan sólo a la entrada de la neurona sumatoria que representa la clase a la que pertenece el patrón de entrada vinculado a la neurona.

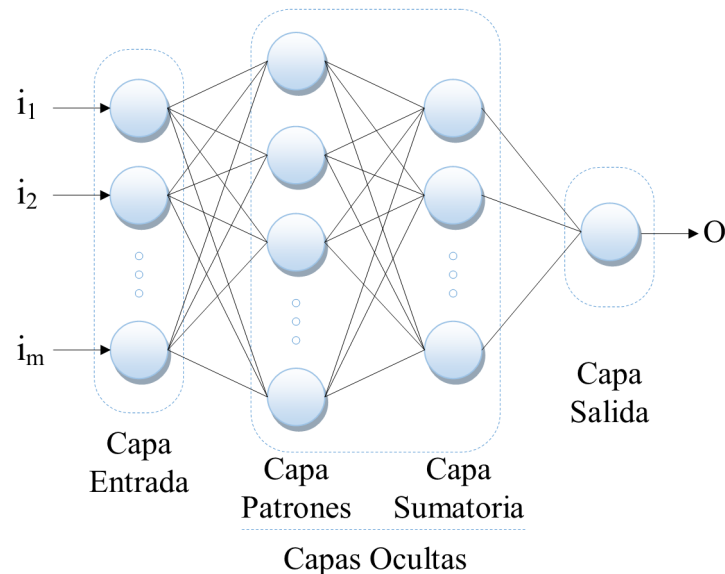


Figura 14. Arquitectura de la red neuronal probabilística formada por cuatro capas: capa de entrada con un número de neuronas igual al de atributos de entrada, una primera capa oculta de patrones con un número de neuronas igual al de muestras de entrenamiento, una segunda capa oculta sumatoria con un número de neuronas igual al número de clases y una capa de salida con una neurona que proporciona el resultado de la clasificación.

Conceptualmente, durante el entrenamiento cada vector de entrada x_p se almacena en una neurona de la capa de patrones, se identifica la neurona con la clase a la que pertenece el vector de entrada y se conecta su salida a la neurona de la capa sumatoria que representa dicha clase. A diferencia del proceso de entrenamiento que se lleva a cabo en la mayoría de las Redes Neuronales Artificiales para aproximar los valores de sus parámetros (pesos y sesgos), en las redes PNN sus parámetros se determinan de manera totalmente no supervisada sin ningún proceso de aproximación.

Durante el proceso de clasificación de nuevas entradas, la capa de entrada provee la información de las variables del vector de entrada a las neuronas de la capa patrón. Estas neuronas evalúan su contribución a la estimación de la función de densidad de probabilidad para la clase a la que pertenece, calculando cada una de ellas la distancia

euclídea $\|x - x_i\|$ entre el vector de entrada x y el vector de entrenamiento x_p al que representa la neurona patrón, y aplicando una función f_p de activación que habitualmente corresponde a una función Gaussiana.

$$f_p = \exp\left(\frac{\|x - x_p\|^2}{\sigma^2}\right) \quad (21)$$

siendo σ (desviación típica) el parámetro definido por el usuario que permite establecer cuán rápido decae con la distancia la influencia de una observación o patrón de entrenamiento. Finalmente, las neuronas de la capa sumatoria combinan la información de todas las neuronas patrón de la misma clase y la capa de salida elige la clase con mayor puntuación o probabilidad.

La librería R utilizada para implementar la red neuronal probabilística se denomina *pnn* (Chasset, 2016) con las funciones:

learn: Crea una nueva red neuronal probabilística con un nuevo conjunto de entrenamiento o actualiza una existente con nuevas observaciones conocidas.

$$\text{learn}(\text{set}, \text{nn}, \text{category.column} = 1)$$

donde *set* es el conjunto de datos de entrenamiento, *nn* la red neuronal probabilística con o sin entrenamiento y *category.column* el número de campo de la categoría (1 de forma predeterminada).

smooth: Establece el parámetro de suavizado. Si no se conoce el valor, un algoritmo genético busca el mejor valor.

$$\text{smooth}(\text{nn}, \text{sigma}, \text{limits} = c(0, 10))$$

donde *nn* es la red neuronal probabilística entrenada, *sigma* establece directamente el parámetro y no busca el mejor valor y *limits* un vector con el intervalo (mínimo, máximo) en el que la función tiene que buscar el mejor valor.

guess: Predice la categoría de una nueva observación.

$$\text{guess}(\text{nn}, x)$$

donde *nn* es la red neuronal probabilista entrenada y *x* un vector con la nueva observación.

2.7.3 Máquina de Soporte Vectorial (SVM)

La máquina de soporte vectorial (SVM) es un algoritmo de aprendizaje supervisado desarrollado por Vapnik (Vapnik, 1995) perteneciente a la familia de los clasificadores lineales utilizado en muchos campos científicos e industriales (Raj *et al.*, 2016; Robotham *et al.*, 2010; Yu, 2021).

En problemas de clasificación binaria, el algoritmo SVM busca el hiperplano óptimo que maximice el margen de separación entre las dos clases (Anzola, 2015). El problema se resuelve bajo programación cuadrática tratando de encontrar el hiperplano óptimo (H) y los dos hiperplanos paralelos (H_1 y H_2) definidos por los puntos de cada clase más próximos a H denominados vectores soporte. Se maximiza la distancia entre H_1 y H_2 (*margen*) y se restringe la presencia de ningún dato entre los dos hiperplanos (Figura 15).

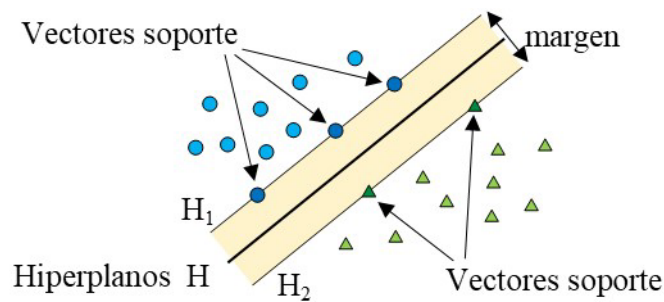


Figura 15. Hiperplano de separación óptimo H y margen de separación $H_1 \leftrightarrow H_2$ delimitado por los vectores soporte de cada clase.

Cuando el conjunto de datos de entrenamiento no es linealmente separable, SVM dispone de dos estrategias complementarias para clasificar estos datos: suavizar el margen e incrementar la dimensión del espacio euclídeo del problema. En el primer caso, al suavizar el margen se permite cierta cantidad de datos mal clasificados penalizando los puntos que generan la no linealidad (Figura 16). La penalización se calcula a partir de la distancia del punto al hiperplano H y de un parámetro definido por el usuario denominado coste, que establece el grado de penalización en función del tipo de datos a clasificar. El parámetro coste oscila entre 0 donde no se penalizan los datos mal clasificados, e infinito donde no se permite ningún dato mal clasificado (clasificación lineal).

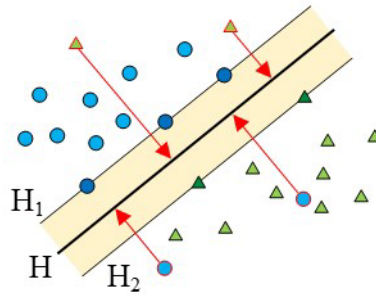


Figura 16. Margen de separación $H_1 \leftrightarrow H_2$ suavizado con datos mal clasificados marcados en rojo y su distancia al hiperplano de separación H indicada con flecha roja.

La segunda estrategia SVM consiste en transformar el conjunto de datos con una determinada dimensión hacia un espacio euclídeo de dimensión superior. Para ello, dispone de varias funciones denominadas kernel a elegir por el usuario en función del tipo de problema a clasificar. La función kernel transforma los datos de entrenamiento a un nuevo espacio euclídeo donde el problema pueda tratarse de forma lineal o mediante el suavizado del margen. Las funciones kernel más utilizadas son:

Kernel lineal	$K(x, x') = x \cdot x'$
---------------	-------------------------

Kernel radial/gaussiano	$K(x, x') = \exp(-\gamma \ x - x'\ ^2)$
-------------------------	---

Kernel polinomial	$K(x, x') = (x \cdot x' + c)^d$
-------------------	---------------------------------

Kernel sigmoide/neural	$K(x, x') = \tanh(ax_i * x_j + b)$
------------------------	------------------------------------

La extensión del algoritmo SVM a problemas con más de dos clases (K clases) se resuelve mediante varias estrategias. Las más conocidas son:

- *One-versus-one*: se generan $K(K-1)/2$ clasificadores SVM comparando todos los posibles pares de clases, se genera la predicción de cada uno de los clasificadores y se asigna la predicción final del algoritmo a la clase asignada con mayor frecuencia. Esta estrategia tiene dificultades cuando hay un número elevado de clases.
- *One-versus-all*: se generan K clasificadores SVM comparando cada clase con las $K-1$ restantes agrupadas, se genera la predicción de cada uno de los clasificadores y se asigna la predicción final del algoritmo a la clase vencedora frente al grupo de clases restantes. Esta estrategia no es consistente cuando hay más de una clase vencedora o cuando se entrena de forma no balanceada (mismo número de muestras por clase).

- *Directed Acyclic Graph SVM (DAGSVM)*: es una mejora del método *one-versus-one* para reducir su tiempo de ejecución eliminando comparaciones innecesarias.

La librería R utilizada para implementar la máquina de soporte vectorial SVM se denomina *e1071* (Meyer, 2019), con las funciones:

svm: crea la máquina de soporte vectorial y lo entrena:

svm(formula, data = NULL, kernel, cost, scale = TRUE)

donde *formula* indica la descripción simbólica del modelo a ajustar, *data* los datos de entrenamiento, *kernel* la función kernel elegida, *cost* el grado de penalización de los datos mal clasificados y *scale* indica si los datos están estandarizados.

predict: predice porcentaje de pertenencia a cada categoría de un conjunto de observaciones.

predict(object, newdata, ...)

donde *object* es el modelo SVM y *newdata* el conjunto de nuevas observaciones.

2.8 Métricas de evaluación

La capacidad de clasificación de cada máquina de aprendizaje está ligada al número de aciertos y fallos. En este trabajo se utilizaron 4 métricas específicas para valorar la predicción de los algoritmos de clasificación (Sun *et al.*, 2020; Paola *et al.*, 2021; Purwandari *et al.*, 2021):

Exactitud: Porcentaje de predicciones positivas y negativas correctas.

$$\text{Exactitud} = \frac{(TP + TN)}{(TP + TN + FP + FN)} \quad (22)$$

Precisión: Porcentaje de predicciones positivas correctas.

$$\text{Precision} = \frac{TP}{(TP + FP)} \quad (23)$$

Sensibilidad: Porcentaje de verdaderos positivos identificados.

$$\text{Sensibilidad} = \frac{TP}{(TP + FN)} \quad (24)$$

Medida F1: Media armónica de la precisión y sensibilidad.

$$\text{Medida F1} = 2 * \frac{\text{Precisión} * \text{Sensibilidad}}{(\text{Precisión} + \text{Sensibilidad})} \quad (25)$$

siendo: TP (Verdadero Positivo) el número de respuestas correctas de pertenencia a una clase, TN (Verdadero Negativo) el número de respuestas correctas de no pertenencia a una clase, FP (Falso Positivo) el número de respuestas incorrectas de pertenencia a una clase y FN (Falso Negativo) el número de respuestas incorrectas de no pertenencia a una clase. En el anexo 7.1.7 se incluye la función *Métrica* en R que calcula las 4 métricas comparando las predicciones del modelo con las salidas correctas.

2.9 Calibración del modelo

Los modelos de aprendizaje automático se parametrizan para que su comportamiento pueda ajustarse a un problema determinado. Estos modelos pueden tener muchos parámetros y encontrar la combinación de valores óptima requiere, en algunos casos, de un análisis más profundo que queda fuera del objeto de este trabajo. Se seleccionaron los valores de cada parámetro mediante el procedimiento denominado matriz de búsqueda de hiperparámetros (Liu *et al.*, 2006), estableciendo un rango discreto de valores de cada parámetro y creando una matriz con todas las combinaciones posibles de hiperparámetros para evaluar metódicamente todos modelos resultantes. La métrica utilizada para seleccionar el modelo de mejor calidad fue la medida F1.

Para evitar elegir los hiperparámetros del modelo que mejor se ajustan a los datos de test, el proceso de calibración utiliza datos distintos para entrenar y validar cada configuración, reservando los datos de test para la evaluación final del modelo. La validación de cada configuración del modelo se realizó mediante el procedimiento de remuestreo de validación cruzada con K iteraciones (Burman, 1989) (Figura 17).

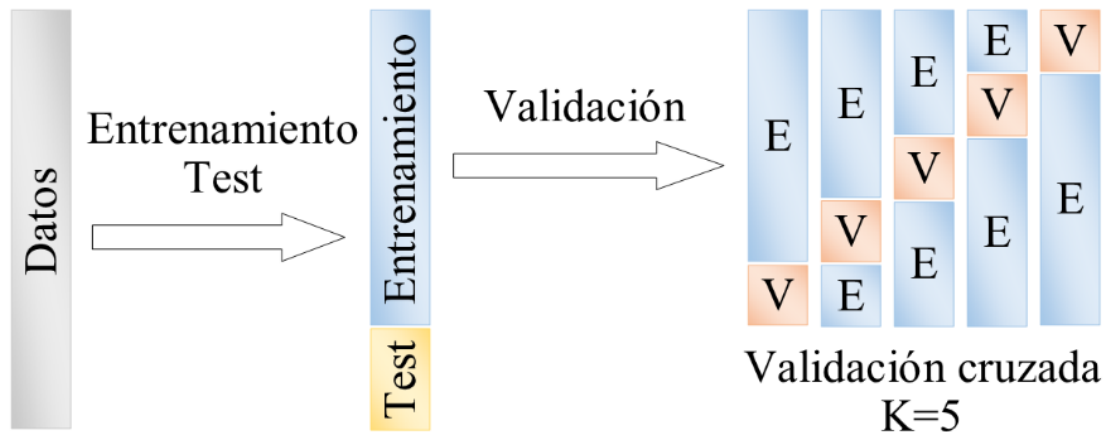


Figura 17. Procedimiento de validación cruzada con $K=5$ iteraciones. Los datos se dividen en dos bloques: entrenamiento y test. Los datos de entrenamiento se dividen en cinco bloques, cada uno de ellos se utiliza para validar el modelo entrenado con los otros cuatro bloques.

Las muestras de la tabla de datos se dividen en datos de entrenamiento a utilizar en el proceso de validación cruzada y datos de test a utilizar en la prueba final del modelo. El procedimiento de validación consiste en (anexo 7.1.8):

1. Dividir los datos de entrenamiento en K bloques.
2. Entrenar el modelo con $(K-1)$ bloques de datos y validar con el bloque restante.
3. Repetir K veces el punto 2 con bloque de validación distinto en cada iteración.
4. Calcular la media de las métricas obtenidas en las K validaciones.

El proceso se repite para cada combinación de hiperparámetros y se selecciona la combinación que ofrece la mayor medida F1.

2.10 Evaluación del modelo

Una vez seleccionados los parámetros que mejor resultado ofrece en el proceso de validación, se procede a entrenar el modelo con todos los datos de entrenamiento y se evalúa su capacidad de clasificación con los datos de test (anexo 7.1.9).

En un primer análisis obtenemos una visión general de las cuatro máquinas de aprendizaje y su capacidad (precisión, exactitud, sensibilidad y valor F1) de identificar muestras de las cuatro categorías o fases del lance.

A continuación, se procede a evaluar la capacidad de los cuatro modelos para separar muestras de dos clases. Para ello, se entrenan y validan los modelos con los datos registrados en dos fases del lance, obteniendo métricas de las 6 combinaciones posibles.

Posteriormente, analizamos la capacidad de identificar datos de una fase frente al resto de datos. En este caso, agrupamos los datos de las 3 fases restantes en una nueva categoría y equilibramos las muestras para obtener la tabla de datos con la que entrenar y validar los modelos.

2.11 Simplificación del modelo

Las muestras registradas durante la actividad del buque de pesca reflejan la influencia del arte de arrastre en el comportamiento dinámico del buque. Galotto *et al.* (2019) demostraron que los sensores de un dispositivo móvil son capaces de identificar 4 periodos distintos etiquetados como NAVEGAR, LARGAR, ARRASTRAR y VIRAR. La tarea de los modelos utilizados en este trabajo será clasificar esas muestras en una de estas 4 categorías.

Cabe señalar que el objetivo principal de este trabajo es identificar el instante de inicio y fin de la fase ARRASTRAR para obtener información del tiempo real de pesca, por lo que la eficacia en reconocer las muestras durante esta fase es fundamental para lograr este objetivo. Las fases restantes son susceptibles de fusionarse y reducirse en número si con ello se mejora la calidad de clasificación de la categoría ARRASTRAR. En este sentido, se validaron los modelos con 4, 3 y 2 categorías, manteniendo siempre la categoría ARRASTRAR (anexo 7.1.10.1).

Por otro lado, un exceso de atributos en las muestras puede provocar una pérdida de calidad en la tarea de clasificación de las máquinas de aprendizaje. Como cada atributo está ligado a un sensor y en ocasiones, a la orientación relativa del dispositivo móvil

respecto al buque, se agruparon los atributos por sensor y se analizó la respuesta de las cuatro máquinas de aprendizaje frente a todas las posibles reducciones del grupo de atributos (anexo 7.1.10.2).

Finalmente se evaluaron las cuatro máquinas de aprendizaje con la reducción de categorías y sensores más favorable para cada una de ellas (anexo 7.1.10.3).

3 Resultados

3.1 Análisis de datos

3.1.1 Análisis descriptivo

En este apartado se indican los resultados obtenidos a partir de las variables contenidas en la tabla de datos relacionadas con el equilibrio longitudinal del buque (escora longitudinal) y los 4 grados de libertad de movimiento afectados por el inicio del lance y susceptibles de ser registrados con facilidad por los sensores disponibles: cabeceo, balanceo, guiñada y avance.

3.1.1.1 Escora longitudinal

La presencia de un arte de pesca activo provoca cambios en la escora longitudinal del buque y en el ángulo que forma su eje longitudinal con el plano horizontal. El sensor capaz de registrar información sobre la orientación del buque es el acelerómetro. Las variables $AceX_m$, $AceY_m$ y $AceZ_m$ representan la media de las componentes del vector gravedad y aportan información sobre la orientación del buque respecto al plano horizontal. La proyección del vector gravedad sobre el plano de crujía, representada por las variables $AceY_m$ y $AceZ_m$, permite evaluar el ángulo de escora longitudinal del buque en cada momento.

La variable $AceY_m$ presentó un valor de curtosis ($\beta=4.27$) y un p -valor del test de Shapiro-Wilk ($p<0.001$) que sugería que los datos no se ajustaban a una distribución normal. El test de Kruskal-Wallis ($p<0.001$) mostró la existencia de diferencias significativas en su valor central y dispersión entre actividades. Durante el lance, su mediana se incrementó un 42% y su rango intercuartílico se redujo en un 58% (Tabla 2; Figura 18). Por su parte, $AceZ_m$ presentó un comportamiento claramente no normal ($\gamma=2.74$; $\beta=21.58$; Shapiro-Wilk test $p<0.001$) y su densidad por actividad mostró la ausencia de diferencias significativas en su valor central y dispersión (Kruskal-Wallis test $p=0.749$) (Tabla 2; Figura 19).

Tabla 2. Resumen estadístico de las 22 variables cualitativas de la tabla de datos. La variable con subíndice m representa la media de los datos generados por el sensor en periodos de 1 minuto. La variable con subíndice s representa la desviación estándar de los datos generados por el sensor en periodos de 10 segundos. Para cada variable, la tabla incluye el número de muestras (N), la media, la mediana, la desviación estándar (SD), el coeficiente de variación de Pearson (CV_Pearson), la desviación absoluta de la mediana (MAD), el coeficiente de variación de la desviación absoluta de la mediana (CV_MAD), el valor mínimo (Min), el valor máximo (Max), el rango de valores, el grado de simetría (Skewness), la curtosis y el p-val del test de Kruskal-Wallis.

	$AceX_m$	$AceY_m$	$AceZ_m$	$GirX_m$	$GirY_m$	$GirZ_m$	$MagX_m$	$MagY_m$	$MagZ_m$	$GpsR_m$	$GpsV_m$
N	2453	2453	2453	2453	2453	2453	2453	2453	2453	2453	2453
Media	-1.0302	0.0412	9.8941	-0.0001	0.0003	0.0000	-29.2530	60.0077	-15.2240	140.8623	4.9530
Mediana	-1.0420	0.0440	9.8940	0.0000	0.0000	0.0000	-33.5270	65.4720	-16.0900	114.2770	3.4580
SD	0.0692	0.0162	0.0121	0.0005	0.0005	0.0027	11.6915	15.1598	3.0334	103.6299	2.9446
CV_Pearson	0.0672	0.3932	0.0012	4.7825	1.4049	62.8112	0.3997	0.2526	0.1992	0.7357	0.5945
MAD	0.0667	0.0163	0.0074	0.0000	0.0000	0.0000	12.7593	17.3242	2.8807	116.2092	1.5879
CV_MAD	0.0640	0.3707	0.0007	-	-	-	0.3806	0.2646	0.1790	1.0169	0.4592
Min	-1.4660	-0.0420	9.8300	-0.0040	-0.0010	-0.0270	-50.0170	34.9760	-20.8220	2.6130	0.0580
Max	-0.6170	0.0870	9.9930	0.0020	0.0020	0.0240	-6.8830	81.2610	-7.5720	356.0770	10.3920
Rango	0.8490	0.1290	0.1630	0.0060	0.0030	0.0510	43.1340	46.2850	13.2500	353.4640	10.3340
Skewness	0.3780	-0.8532	2.7354	-1.7448	0.7071	0.0192	0.4113	-0.3437	0.6319	0.5513	0.6494
Curtosis	4.2573	4.2739	21.5804	14.2086	1.6542	36.7724	1.6587	1.5022	2.4319	2.0463	1.9805
Kr.Wallis p-val	0.1177	0.0000	0.7489	0.0158	0.9760	0.0001	0.0000	0.0000	0.0000	0.0000	0.0000

	$AceX_s$	$AceY_s$	$AceZ_s$	$GirX_s$	$GirY_s$	$GirZ_s$	$MagX_s$	$MagY_s$	$MagZ_s$	$GpsR_s$	$GpsV_s$
N	2453	2453	2453	2453	2453	2453	2453	2453	2453	2453	2453
Media	0.0872	0.0680	0.0645	0.0039	0.0032	0.0019	0.4904	0.5637	0.4417	7.6063	0.2339
Mediana	0.0680	0.0540	0.0490	0.0030	0.0020	0.0020	0.3290	0.4030	0.3940	1.0430	0.0690
SD	0.0568	0.0395	0.0450	0.0019	0.0020	0.0013	0.7251	0.7283	0.3198	22.0062	0.4935
CV_Pearson	0.6516	0.5815	0.6973	0.4818	0.6135	0.6631	1.4785	1.2921	0.7240	2.8931	2.1097
MAD	0.0356	0.0297	0.0282	0.0015	0.0015	0.0015	0.0593	0.0771	0.1097	0.9563	0.0474
CV_MAD	0.5233	0.5491	0.5749	0.4942	0.7413	0.7413	0.1803	0.1913	0.2785	0.9169	0.6876
Min	0.0180	0.0090	0.0150	0.0020	0.0010	0.0010	0.2330	0.2820	0.2540	0.1040	0.0000
Max	0.4170	0.2610	0.3350	0.0130	0.0150	0.0140	16.2410	16.8470	11.6380	179.0620	4.3130
Rango	0.3990	0.2520	0.3200	0.0110	0.0140	0.0130	16.0080	16.5650	11.3840	178.9580	4.3130
Skewness	1.8170	1.2624	1.9776	1.0152	1.9266	3.5891	9.6756	9.3140	20.9884	4.8946	4.5567
Curtosis	6.9348	4.2777	7.8477	3.5099	7.5787	23.9303	140.6254	139.7816	653.1549	29.7713	28.2055
Kr.Wallis p-val	0.0000	0.1677	0.0000	0.1977	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

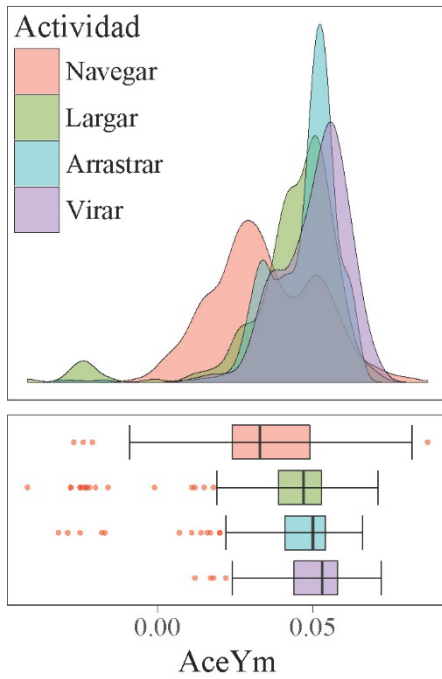


Figura 18. Gráfico de cajas y densidad por actividad de la variable $AceY_m$

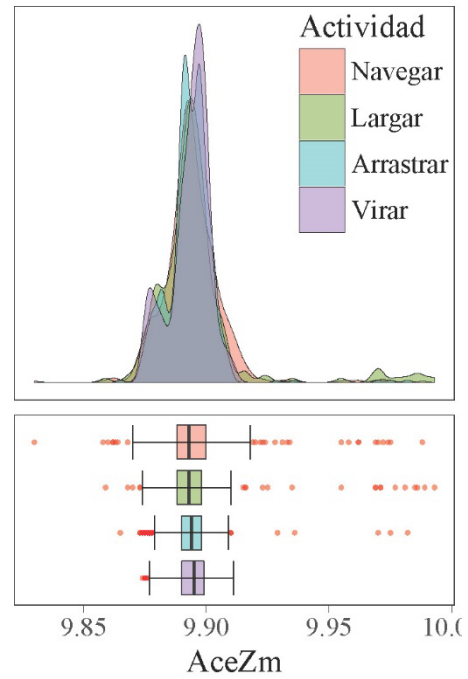


Figura 19. Gráfico de cajas y densidad por actividad de la variable $AceZ_m$

3.1.1.2 Cabeceo y balance

Los movimientos de cabeceo y balance se originan por efecto de fuerzas externas como el oleaje, viento, etc. En ambos casos, el buque describe un movimiento de rotación alrededor de los ejes transversal y longitudinal de la embarcación provocando la elevación y descenso alternativos de la proa y popa o de las bandas de babor y estribor. La presencia de un arte de pesca activo provoca el desplazamiento del centro de masas del buque (punto de cruce de los ejes transversal y longitudinal) y modifica el movimiento de cabeceo y balance del buque. Estos cambios son detectados por los sensores acelerómetro, giróscopo y campo magnético.

El acelerómetro presentó mayores valores del coeficiente de variación en $AceX_s$, $AceY_s$ y $AceZ_s$. Al igual que $AceZ_m$, las desviaciones asociadas a los tres ejes del acelerómetro presentaron comportamientos con distribuciones no normales. En cambio, sólo se encontraron densidades por actividad claramente diferentes del valor central y dispersión en $AceX_s$ y $AceZ_s$ (Kruskall-Wallis test $AceX_s$ $p < 0.001$; Kruskal-Wallis test $AceY_s$ $p = 0.168$; Kruskal-Wallis test $AceZ_s$ $p < 0.001$) (Tabla 2; Figura 20; Figura 21).

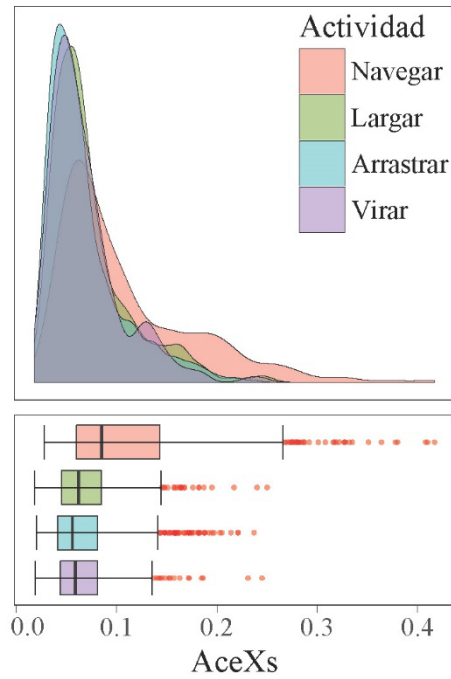


Figura 20. Gráfico de cajas y densidad por actividad de la variable $AceX_s$

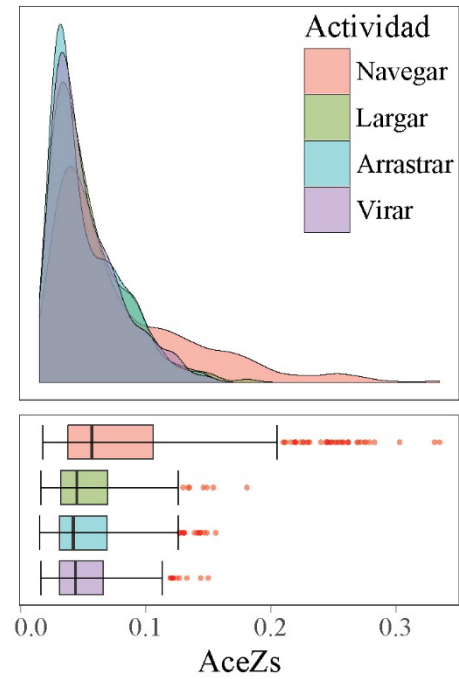


Figura 21. Gráfico de cajas y densidad por actividad de la variable $AceZ_s$

El giroscopio mide la velocidad de rotación alrededor de los ejes X, Y y Z. La orientación del sensor en el buque indica que la variable $GirX$ registra el movimiento de cabeceo y la variable $GirY$ el movimiento de balance. En concreto, las variables $GirX_s$ y $GirY_s$ de la tabla de datos deben reflejar cambios en la variabilidad o dispersión de la velocidad de rotación de cabeceo y balance. Las componentes $GirX_s$ y $GirY_s$ mostraron un comportamiento no normal y al menos una de ellas ($GirY_s$) apuntó la posibilidad de diferenciar la navegación de las actividades ARRASTRAR, LARGAR y VIRAR (Kruskall-Wallis test $GirY_s$ $p < 0.001$) (Tabla 2; Figura 22), en cambio, $GirX_s$ mostró ausencia de diferencias significativas respecto a su mediana en cada una de las 4 actividades (Kruskall-Wallis test $GirX_s$ $p = 0.198$).

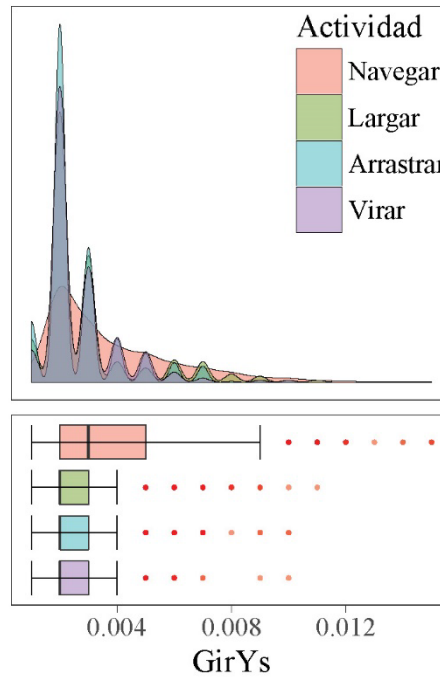


Figura 22. Gráfico de cajas y densidad por actividad de la variable $GirY_s$

Por otra parte, el sensor de campo magnético mide en μT el campo magnético en el entorno del dispositivo. Por tanto, todas las componentes del vector que registra el sensor ($MagX$, $MagY$ y $MagZ$) y particularmente las variables $MagX_s$, $MagY_s$ y $MagZ_s$ son candidatas a reflejar los cambios de cabeceo y balanceo en las diferentes fases del lance. Los test de Shapiro-Wilk indicaron que ninguna de las tres variables presentó un comportamiento normal. Sin embargo, una comparación de medianas demostró la presencia de diferencias significativas entre el valor central y dispersión entre la actividad NAVEGAR y ARRASTRAR en las variables $MagX_s$, $MagY_s$ (Kruskal-Wallis test $MagX_s$ y $MagY_s$ $p < 0.001$) (Tabla 2; Figura 23; Figura 24) y en menor medida en la variable $MagZ_s$.

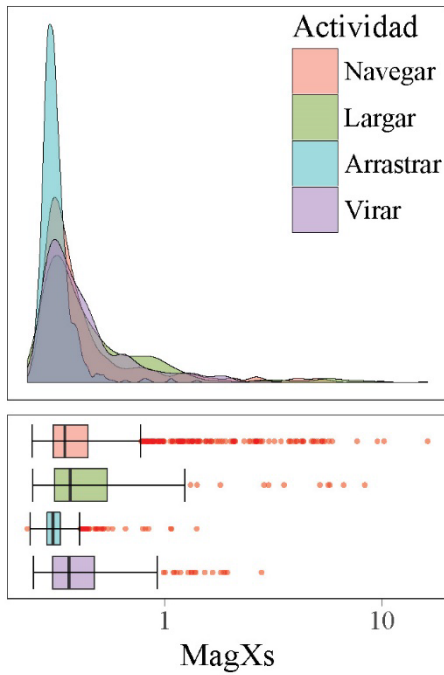


Figura 23. Gráfico de cajas y densidad por actividad de la variable $MagX_s$. Se ha utilizado escala logarítmica en el eje OX

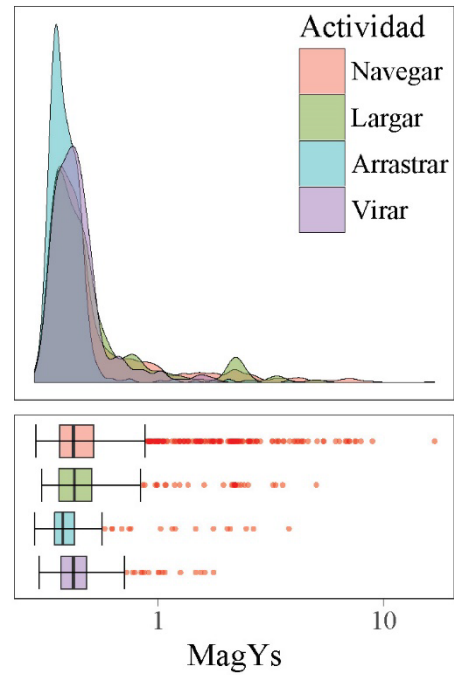


Figura 24. Gráfico de cajas y densidad por actividad de la variable $MagY_s$. Se ha utilizado escala logarítmica en el eje OX

3.1.1.3 Guiñada

El movimiento de guiñada es un movimiento oscilatorio de rotación alrededor del eje OZ del buque que provoca pequeños cambios de rumbo de la embarcación. Los sensores capaces de registrar el movimiento de guiñada son el giróscopo, el sensor de campo magnético y el GPS.

La componente Z del giróscopo mide la velocidad de giro alrededor del eje OZ y la variable $GirZ_s$ registra la magnitud de los cambios de rumbo durante la actividad pesquera. Los datos de $GirZ_s$ presentaron una distribución claramente asimétrica ($\gamma=3.59$) lo que provocó que el test de Shapiro-Wilk fuese altamente significativo ($p<0.001$). Así mismo, mostró una alta variabilidad alrededor de su mediana, que se reflejó en el nivel de significación del test Kruskal-Wallis ($p<0.001$) (Tabla 2; Figura 25).

A su vez, las componentes X e Y del sensor de campo magnético son las más sensibles a los movimientos de guiñada del buque y las variables descritas en el apartado anterior $MagX_s$ y $MagY_s$ reflejaron su magnitud.

El sensor GPS también aportó información sobre el rumbo y velocidad del buque. La variable derivada de este sensor relacionada con la guiñada es $GpsR_s$. Esta variable registró una alta variabilidad del 92% alrededor de su mediana (Tabla 2). El diagrama de densidad por actividad (Figura 26) reflejó diferencias significativas en su valor central y dispersión entre las actividades (Kruskall-Wallis test $p<0.001$). Su mediana descendió un 16% en la fase LARGAR, se incrementó un 53% en la fase ARRASTRAR y un 160% en la fase VIRAR.

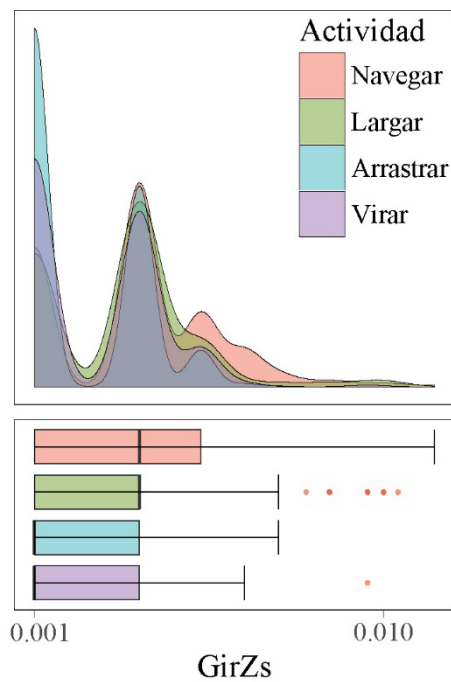


Figura 25. Gráfico de cajas y densidad por actividad de la variable $GirZ_s$. Se ha utilizado escala logarítmica en el eje OX

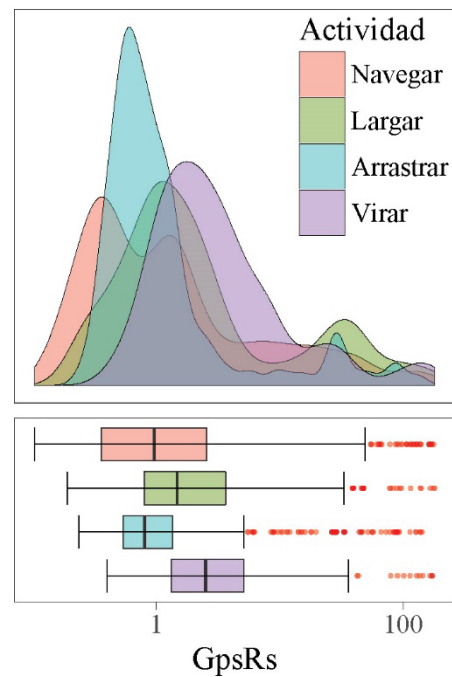


Figura 26. Gráfico de cajas y densidad por actividad de la variable $GpsR_s$. Se ha utilizado escala logarítmica en el eje OX

3.1.1.4 Avance

El estudio de la velocidad de avance del buque durante las diferentes fases de la actividad pesquera se centra en analizar la velocidad media y la estabilidad de esta velocidad frente a fuerzas variables externas. De esta forma, el sensor que mide la velocidad de avance del buque es el GPS y la variable que mejor refleja la evolución de la velocidad media es $GpsV_m$. Esta variable discrimina claramente la navegación

del resto de fases de actividad y eso se reflejó en diferencias significativas de su valor central y dispersión (Figura 27).

Por otra parte, la estabilidad en el avance queda caracterizada por la variable $GpsV_s$ (variabilidad de la velocidad). El análisis de esta variable concluyó que tanto los coeficientes de asimetría como de curtosis estuvieron fuera de rango ($\gamma=4.56$; $\beta=28.21$). El test de Shapiro-Wilk ($p<0.001$) confirmó que los datos no respondieron a una distribución normal. La variable mostró una alta variabilidad del 69% alrededor de su mediana (Tabla 2). La densidad por actividad (Figura 28) y el test de Kruskal-Wallis ($p<0.001$) reflejó diferencias significativas en su valor central y dispersión entre las actividades. Su mediana descendió un 24% en la fase ARRASTRAR y se incrementó un 125% en la fase LARGAR y un 62% en la fase VIRAR. Su rango intercuartílico descendió un 33% en la fase ARRASTRAR y se incrementó un 267% en la fase LARGAR y un 93% en la fase VIRAR.

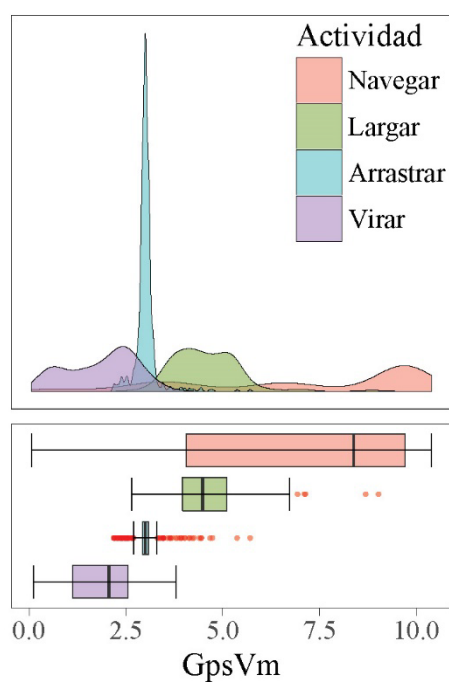


Figura 27. Gráfico de cajas y densidad por actividad de la variable $GpsV_m$

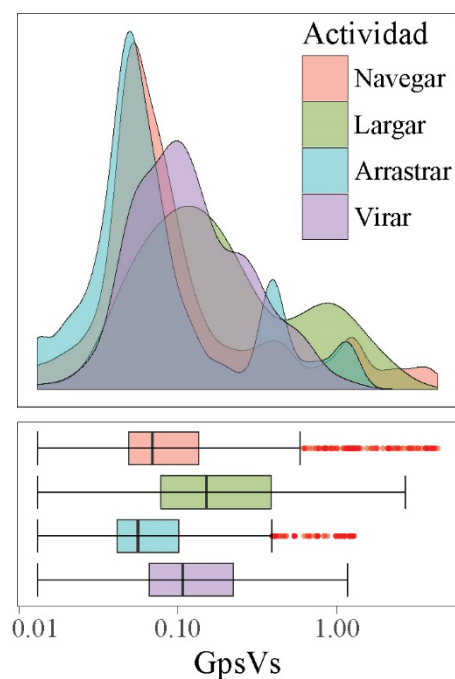


Figura 28. Gráfico de cajas y densidad por actividad de la variable $GpsV_s$. Se ha utilizado escala logarítmica en el eje OX

3.1.2 Análisis multivariante

La tabla 3 muestra la matriz de correlación de Pearson entre las 22 variables normalizadas de la tabla de datos. Se identificaron varias parejas de variables colineales con un índice de correlación superior a 0.7 y se eliminó una de las variables de la pareja. En concreto, se eliminaron *AceZm* (colineal con *AceXm*), *AceZs* (colineal con *AceXs* y *AceYs*), *AceXs* (colineal con *GirYs*), *AceYs* (colineal con *GirXs*), *MagXm* (colineal con *MagZm* y *GpsRm*), *MagZm* (colineal con *GpsRm*) y *MagZs* (colineal con *MagXs*).

Tabla 3. Matriz de correlación de las 22 variables normalizadas de la tabla de datos. En la parte inferior de la diagonal muestra el coeficiente de correlación de Pearson y en la parte superior su p-valor.

	<i>AceXm</i>	<i>AceXs</i>	<i>AceYm</i>	<i>AceYs</i>	<i>AceZm</i>	<i>AceZs</i>	<i>GirXm</i>	<i>GirXs</i>	<i>GirYm</i>	<i>GirYs</i>	<i>GirZm</i>	<i>GirZs</i>	<i>MagXm</i>	<i>MagXs</i>	<i>MagYm</i>	<i>MagYs</i>	<i>MagZm</i>	<i>MagZs</i>	<i>GpsRm</i>	<i>GpsRs</i>	<i>GpsVm</i>	<i>GpsVs</i>
<i>AceXm</i>		0.00	0.00	0.00	0.00	0.00	0.00	0.03	0.00	0.00	0.00	0.09	0.00	0.18	0.00	0.54	0.00	0.89	0.21	0.03	0.17	0.55
<i>AceXs</i>	0.10		0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.91	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.81	0.71	0.00	0.00
<i>AceYm</i>	-0.15	-0.23		0.14	0.00	0.00	0.00	0.44	0.00	0.00	0.47	0.00	0.00	1.00	0.00	0.01	0.00	0.00	0.01	0.00	0.00	0.01
<i>AceYs</i>	0.09	0.42	-0.03		0.00	0.00	0.00	0.00	0.00	0.00	0.20	0.00	0.00	0.62	0.00	0.00	0.00	0.12	0.00	0.06	0.00	0.94
<i>AceZm</i>	0.73	0.07	-0.38	0.08		0.00	0.00	0.03	0.00	0.00	0.00	0.24	0.00	0.02	0.00	0.00	0.04	0.00	0.03	0.01	0.02	0.43
<i>AceZs</i>	0.20	0.73	-0.19	0.75	0.16		0.00	0.00	0.00	0.00	0.14	0.00	0.00	0.02	0.00	0.00	0.00	0.00	0.20	0.08	0.00	0.01
<i>GirXm</i>	0.11	-0.07	0.31	-0.08	-0.17	-0.11		0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.04	0.00	0.00	0.84	0.49	0.82
<i>GirXs</i>	0.04	0.49	-0.02	0.95	0.04	0.69	-0.09		0.00	0.00	0.19	0.00	0.00	0.21	0.01	0.00	0.00	0.01	0.00	0.15	0.00	0.98
<i>GirYm</i>	0.19	0.17	-0.37	0.15	0.28	0.15	-0.25	0.18		0.00	0.12	0.00	0.00	0.01	0.00	0.00	0.09	0.00	0.00	0.08	0.99	0.24
<i>GirYs</i>	0.07	0.92	-0.26	0.32	0.15	0.54	-0.11	0.42	0.18		0.45	0.00	0.00	0.00	0.00	0.00	0.09	0.00	0.44	0.19	0.00	0.00
<i>GirZm</i>	-0.35	0.00	0.01	0.03	-0.23	0.03	-0.62	0.03	-0.03	-0.02		0.71	0.77	0.02	0.69	0.00	0.89	0.21	0.54	0.57	0.61	0.85
<i>GirZs</i>	0.03	0.55	-0.08	0.41	0.02	0.47	-0.08	0.44	0.15	0.41	0.01		0.00	0.00	0.02	0.00	0.00	0.00	0.02	0.00	0.00	0.00
<i>MagXm</i>	-0.13	-0.22	0.10	-0.28	-0.10	-0.29	0.11	-0.29	-0.21	-0.15	-0.01	-0.18		0.94	0.00	0.00	0.00	0.29	0.00	0.00	0.00	0.84
<i>MagXs</i>	0.03	0.14	0.00	0.01	0.05	0.05	-0.07	0.03	0.05	0.09	-0.05	0.40	0.00		0.05	0.00	0.32	0.00	0.00	0.00	0.79	0.01
<i>MagYm</i>	-0.29	0.11	-0.26	-0.11	-0.12	-0.14	-0.15	0.05	0.29	0.19	0.01	0.05	-0.26	0.04		0.19	0.29	0.00	0.00	0.39	0.00	0.59
<i>MagYs</i>	-0.01	0.17	-0.05	0.07	0.14	0.14	-0.23	0.08	0.09	0.13	0.09	0.37	-0.06	0.55	0.03		0.05	0.00	0.74	0.00	0.86	0.00
<i>MagZm</i>	-0.14	-0.10	-0.09	-0.18	-0.04	-0.19	0.04	-0.16	-0.03	-0.03	0.00	-0.12	0.90	0.02	-0.02	-0.04		0.00	0.00	0.00	0.28	0.87
<i>MagZs</i>	0.00	0.14	-0.18	0.03	0.08	0.08	-0.11	0.05	0.12	0.13	-0.03	0.18	0.02	0.70	0.13	0.56	0.09		0.00	0.00	0.00	0.37
<i>GpsRm</i>	-0.03	0.00	-0.05	-0.19	-0.04	-0.03	0.11	-0.23	-0.15	-0.02	-0.01	-0.05	0.77	0.06	-0.27	-0.01	0.72	0.07		0.00	0.00	0.27
<i>GpsRs</i>	-0.04	-0.01	0.10	-0.04	-0.05	-0.04	0.00	-0.03	-0.04	-0.03	-0.01	0.09	0.16	0.28	0.02	0.11	0.16	0.11	0.17		0.00	0.00
<i>GpsVm</i>	0.03	0.32	-0.64	-0.06	0.05	0.21	0.01	-0.10	0.00	0.29	-0.01	0.17	-0.06	0.01	0.06	0.00	-0.02	0.08	0.14	-0.11		0.00
<i>GpsVs</i>	-0.01	0.08	-0.06	0.00	-0.02	0.05	0.00	0.00	-0.02	0.06	0.00	0.13	0.00	0.05	-0.01	0.06	0.00	0.02	-0.02	0.45	0.09	

Con las 15 variables restantes (*AceXm*, *AceYm*, *GirXm*, *GirXs*, *GirYm*, *GirYs*, *GirZm*, *GirZs*, *MagXs*, *MagYm*, *MagYs*, *GpsRm*, *GpsRs*, *GpsVm* y *GpsVs*) se aplicó la técnica de análisis de componentes principales para reducir su número transformándolas en nuevas variables denominadas componentes principales (PC). La tabla 4 muestra los valores propios de los componentes principales y la figura 29 la representación gráfica del porcentaje de varianza explicada por cada componente.

Los primeros 2 componentes acumularon apenas el 32% de la varianza explicada, para alcanzar el 50% de varianza acumulada fueron necesarios 4 componentes, aplicando el criterio de selección de componentes de Kaiser (Kaiser, 1958) con valor propio superior a 1 se seleccionaron 7 componentes explicando solo el 78.55% de la varianza

y para alcanzar el 90% de varianza acumulada deberían seleccionarse 10 componentes principales.

Tabla 4. Valores propios, porcentaje de varianza y porcentaje de varianza acumulada de los componentes principales.

	Valor propio	% varianza	% varianza acumulada
CP1	2.81768990	18.7845994	18.78460
CP2	2.01638266	13.4425511	32.22715
CP3	1.75869487	11.7246325	43.95178
CP4	1.56829738	10.4553158	54.40710
CP5	1.28706571	8.5804381	62.98754
CP6	1.25436635	8.3624423	71.34998
CP7	1.08041601	7.2027734	78.55275
CP8	0.81226373	5.4150915	83.96784
CP9	0.51492605	3.4328403	87.40068
CP10	0.47560692	3.1707128	90.57140
CP11	0.43634090	2.9089393	93.48034
CP12	0.35227840	2.3485227	95.82886
CP13	0.31453270	2.0968847	97.92574
CP14	0.22940113	1.5293408	99.45508
CP15	0.08173729	0.5449153	100.00000

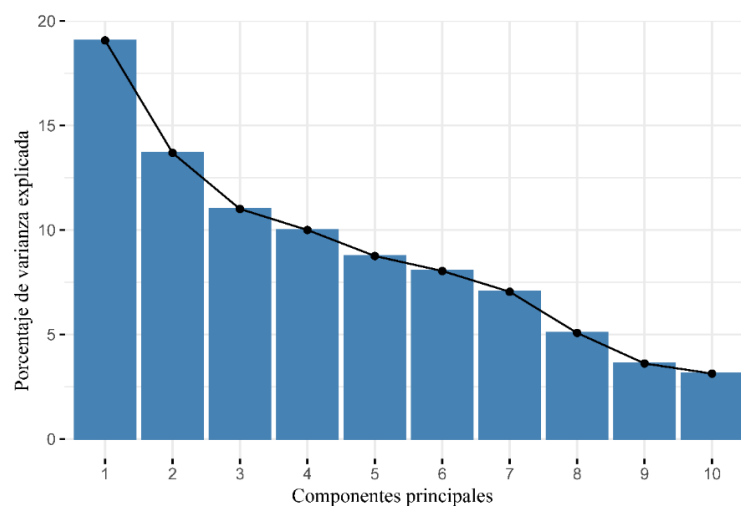


Figura 29. Porcentaje de varianza explicada por cada componente principal.

3.2 Técnicas de aprendizaje automático

3.2.1 Calibración del modelo

La construcción y evaluación de cada una de las máquinas de aprendizaje se inicia con el proceso de selección y validación de los parámetros que permitan adaptar su comportamiento al problema a resolver. En esta primera fase se utilizó el proceso de validación cruzada con un valor de $K=5$ para evitar conjuntos de datos de validación muy pequeño.

3.2.1.1 Análisis Discriminante Lineal (LDA)

El modelo LDA permite establecer la probabilidad inicial de pertenencia a cada clase o categoría, por lo que al utilizar datos de entrenamiento balanceados, es decir, con un número de muestras similar para cada clase, se estableció la misma probabilidad en cada una de ellas: $prior = 1 / n.categorias$.

3.2.1.2 Perceptrón Multicapa (MLP)

El modelo MLP es altamente configurable, permitiendo elegir el número de neuronas por capa, el número de capas, cómo inicializar la red, las funciones de activación, de aprendizaje, etc.

Se eligieron los parámetros que se consideraron más relevantes y se dejaron el resto con los valores por defecto asignados por la función *mlp* que crea la red neuronal. En concreto, en el proceso de calibración se configuraron los siguientes parámetros:

- Función de inicialización (*initFunc*)

Se seleccionó la función de inicialización de pesos y sesgos *Randomize_Weights* que asigna valores iniciales aleatorios distribuidos entre 0 y 1: *initFuncParams=c(0,1)*.

- Función de aprendizaje (*learnFunc*)

Se seleccionó la función de aprendizaje *Std_Backpropagation* con un error permitido nulo y el valor por defecto del ancho de paso de descenso de gradiente 0.2: *learnFuncParams=c(0.2,0)*.

- Función de actualización (*updateFunc*)

Se seleccionó la función de actualización más favorable para este tipo de redes *Topological_Order* donde las neuronas calculan su nueva activación en orden topológico. La primera capa procesada es la de entrada, continuando por la primera capa oculta, la siguiente capa oculta (si la hay) y así sucesivamente hasta la capa de salida.

- Funciones de activación (*hiddenActFunc* y *outputActFunc*)

La función de activación de las neuronas de las capas ocultas y de la capa de salida que ofreció mejores resultados en el proceso de validación ha sido la función sigmoidea: *hiddenActFunc* = "Act_Logistic" y *outputActFunc* = "Act_Logistic".

- Número máximo de iteraciones de aprendizaje (*maxit*)

El proceso de aprendizaje finaliza cuando el error calculado a la salida es inferior al permitido o cuando se alcance el número máximo de iteraciones. Con una capa oculta de 22 neuronas el error no disminuyó sustancialmente a partir de 30 iteraciones. Lo mismo ocurrió con dos capas ocultas de 22 neuronas cada una (Figura 30, Figura 31 y Anexo 7.1.6.1).

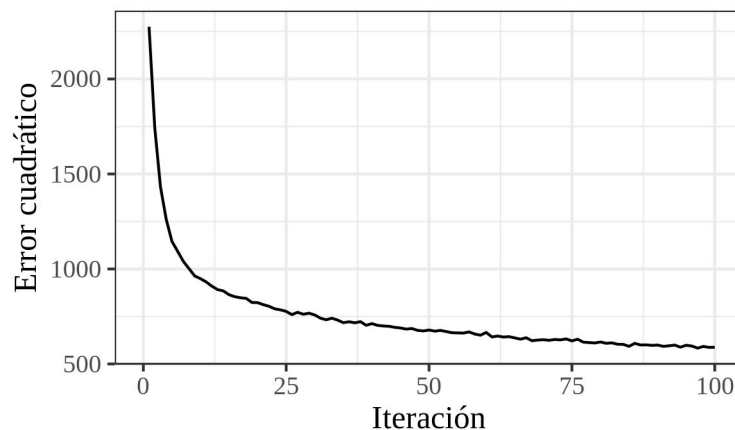


Figura 30. Error cuadrático por iteración en modelo MLP con 1 capa oculta de 22 neuronas.

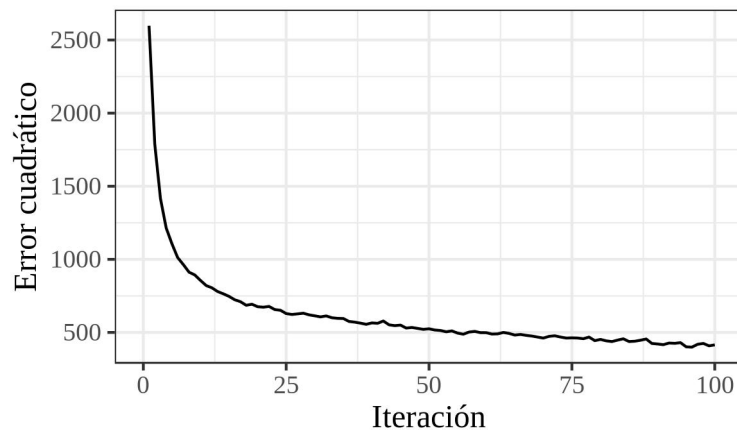


Figura 31. Error cuadrático por iteración en modelo MLP con 2 capas ocultas de 22 neuronas cada una.

- Número de capas ocultas y neuronas en cada capa (size)

Se validaron y compararon el valor medio de F1 de los modelos MLP con una y dos capas ocultas, con una cantidad de neuronas en cada capa entre 1 y 22 unidades. El modelo que ofreció mejor media del valor F1 fue el que contenía una capa oculta con 8 neuronas: $size=c(8)$ (Anexo 7.1.6.1).

3.2.1.3 Red Neuronal Probabilística (PNN)

La red neuronal probabilística permite configurar el parámetro de suavizado σ de la función gaussiana de activación de las neuronas de la capa patrón. Este parámetro establece cómo de rápido decae con la distancia la influencia de una observación o patrón de entrenamiento. La función *smooth* calcula el mejor valor de σ para cada modelo PNN entrenado. En este trabajo se configuró el modelo con el valor $\sigma=1.4$.

3.2.1.4 Máquina de Soporte Vectorial (SVM)

El modelo SVM clasifica muestras no linealmente separables transformándolas a una dimensión superior donde sea posible clasificarlas linealmente. La función que utiliza el modelo SVM para esta transformación se denomina *kernel*. Entre todos los *kernel* que ofrece la librería *e1071* de R para crear modelos SVM, el que mostró mejores resultados fue el *kernel Linear*. Así mismo, el modelo permite suavizar el margen de

separación entre muestras de cada clase admitiendo muestras mal clasificadas. El parámetro *cost* permite graduar la penalización de estas muestras. En este modelo se seleccionó un valor de *cost*=50. La figura 32 muestra las diferentes capacidades de clasificación que se obtuvieron del modelo SVM para diferentes *kernels* y valores de *cost* (Anexo 7.1.6.2).

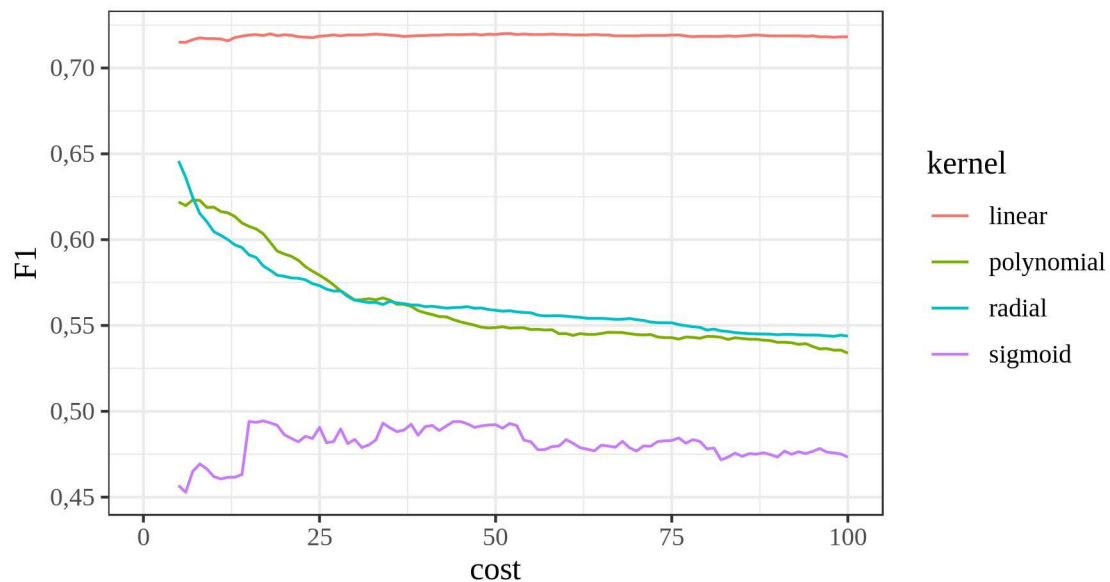


Figura 32. Media de la métrica F1 del modelo SVM para diferentes valores de kernel y cost.

3.2.2 Evaluación del modelo

Una vez seleccionados los parámetros que mejor resultado ofrecieron en el proceso de validación, se procedió a entrenar el modelo con todos los datos de entrenamiento y se evaluó su capacidad de clasificación con los datos de test.

En la tabla 5 se muestran los resultados obtenidos sobre la capacidad de clasificación en el test final de las cuatro máquinas de aprendizaje analizadas en este trabajo. En general, los modelos MLP y SVM exhibieron mejores resultados que los modelos LDA y PNN. La mejor media armónica entre precisión y sensibilidad (F1) la ofreció el modelo MLP alcanzando un valor medio entre categorías del 62.34% frente al 61.36% que ofreció el modelo SVM. Así mismo, SVM superó a MLP en precisión alcanzando un valor medio del 76.04% frente al 67.52%. En cambio, MLP mejoró sus resultados en exactitud y sensibilidad, obteniendo unos resultados medios del 81.75% y 63.50% frente al 80.92% y 61.84% del modelo SVM. Por otro lado, el modelo MLP mostró resultados más equilibrados que SVM en la clasificación de muestras

registradas durante la fase ARRASTRAR, alcanzando una exactitud del 87.24%, una precisión del 71.46% y valor de F1 del 76.26%. SVM solo superó a MLP en sensibilidad, identificando el 97.61% de las muestras de la actividad ARRASTRAR a costa de un número muy alto de falsos positivos. Cabe resaltar las dificultades que mostraron los modelos LDA, MLP y PNN en identificar la categoría LARGAR, reconociendo únicamente el 31.11%, 32.43% y 17.95% de las muestras registradas en esa fase.

Tabla 5. Exactitud, precisión, sensibilidad y valor F1 (expresada en porcentajes) en la prueba final de las máquinas de aprendizaje LDA, MLP, PNN y SVM. Este análisis incluyó las muestras registradas durante todas las categorías de actividad con todos sus atributos.

Actividad	Exactitud	Precisión	Sensibilidad	F1
LDA				
NAVEGAR	73.63	47.66	55.73	51.38
LARGAR	77.82	61.07	31.11	41.22
ARRASTRAR	80.30	62.76	52.14	56.96
VIRAR	78.76	55.05	82.05	65.89
MLP				
NAVEGAR	72.15	46.96	70.17	55.93
LARGAR	80.00	73.00	32.43	43.66
ARRASTRAR	87.24	71.46	81.97	76.26
VIRAR	87.60	78.66	69.42	73.51
PNN				
NAVEGAR	76.88	54.26	47.86	50.86
LARGAR	71.88	37.10	17.95	24.19
ARRASTRAR	68.08	40.24	57.09	47.21
VIRAR	71.45	44.16	53.68	48.46
SVM				
NAVEGAR	83.80	99.05	35.56	52.33
LARGAR	86.45	76.69	65.81	70.84
ARRASTRAR	68.59	44.20	97.61	60.84
VIRAR	84.83	84.23	48.38	61.45

La tabla 6 muestra la capacidad de las máquinas de aprendizaje para separar muestras de dos clases. En la primera parte (1-vs-1) se utilizaron muestras de dos fases del lance y en la segunda parte (1-vs-RESTO) se utilizaron todas las muestras fusionando 3 fases en una nueva clase denominada RESTO. La tabla contiene los resultados porcentuales de la métrica F1 del test final de los 4 modelos.

Tabla 6. Puntuación de F1 (expresada como porcentaje) del test final de las máquinas de aprendizaje LDA, MLP, PNN y SVM. Las muestras contienen todos los atributos. 1-vs-1 incluye las muestras de dos categorías. 1-vs-RESTO incluye todas las muestras manteniendo la categoría de la izquierda y agrupando las muestras restantes en la categoría RESTO.

Tipo de partición	LDA		MLP		PNN		SVM	
1-vs-1								
NAVEGAR-vs-LARGAR	59.37	35.58	69.48	44.82	34.46	68.11	57.65	73.77
NAVEGAR-vs-ARRASTRAR	74.34	77.12	84.94	85.04	51.52	75.39	63.56	78.87
NAVEGAR-vs-VIRAR	79.77	84.01	85.12	86.42	58.25	76.27	66.59	79.97
LARGAR-vs-ARRASTRAR	90.66	91.05	92.49	91.84	25.89	70.14	87.24	88.58
LARGAR-vs-VIRAR	95.21	95.22	94.14	94.47	81.59	83.44	93.81	93.17
ARRASTRAR-vs-VIRAR	81.23	80.28	66.37	75.58	75.88	54.79	84.83	79.27
1-vs- RESTO								
NAVEGAR-vs- RESTO	69.43	63.15	75.85	67.77	48.51	73.05	61.26	71.79
LARGAR-vs- RESTO	46.09	65.88	53.16	72.90	73.12	60.32	62.08	69.30
ARRASTRAR-vs- RESTO	60.54	61.67	88.72	88.54	69.60	22.46	71.03	34.95
VIRAR-vs- RESTO	80.09	70.55	81.84	79.71	74.73	64.62	77.48	61.32

Los resultados obtenidos mostraron debilidades de los modelos LDA, MLP y PNN en diferentes fases del lance. LDA y MLP tuvieron dificultad para diferenciar las muestras de las fases NAVEGAR y LARGAR, identificando menos del 33% de las muestras registradas en la fase LARGAR. Por otro lado, PNN tuvo dificultad para diferenciar las fases NAVEGAR de LARGAR, pero en este caso, los malos resultados correspondieron a las muestras de la fase NAVEGAR, identificando únicamente el 22.56% de las muestras de esta fase. Así mismo, PNN también mostró debilidad comparando las clases LARGAR de ARRASTRAR identificando únicamente el 14.87% de las muestras de la fase LARGAR.

3.2.3 Simplificación del modelo

A continuación, se exploró la posibilidad de mejorar la respuesta del proceso de clasificación reduciendo categorías, testeando los 4 modelos manteniendo la actividad ARRASTRAR y NAVEGAR, y eliminando la categoría LARGAR, VIRAR o ambas. Se puede observar en la tabla 7 que todos los modelos mejoraron sustancialmente su capacidad de clasificación (F1) de la categoría ARRASTRAR al eliminar las categorías LARGAR y VIRAR, y asignando las muestras tomadas en esas fases a NAVEGAR y ARRASTRAR respectivamente.

Tabla 7. Puntuación de F1 (expresada como porcentaje) para la categoría ARRASTRAR en la prueba final de las máquinas de aprendizaje LDA, MLP, PNN y SVM con todos los atributos y después de reducir el número de categorías. Muestra los resultados con 4 (sin reducción), 3 y 2 categorías. El símbolo > escrito entre dos categorías indica que la categoría de la izquierda se fusiona con la categoría de la derecha, asumiendo la nueva categoría la etiqueta de la derecha.

Actividad	LDA	MLP	PNN	SVM
NAVEGAR-LARGAR-ARRASTRAR-VIRAR	56.96	76.26	47.21	60.84
NAVEGAR-ARRASTRAR-VIRAR				
LARGAR > NAVEGAR	57.14	73.77	55.70	64.81
LARGAR > ARRASTRAR	55.16	68.13	58.70	67.06
NAVEGAR-LARGAR-ARRASTRAR				
VIRAR > NAVEGAR	41.64	71.15	55.06	18.88
VIRAR > ARRASTRAR	77.75	84.48	64.95	71.16
NAVEGAR-ARRASTRAR				
LARGAR > NAVEGAR / VIRAR > NAVEGAR	55.47	77.51	70.72	71.05
LARGAR > NAVEGAR / VIRAR > ARRASTRAR	85.38	87.34	76.18	79.42
LARGAR > ARRASTRAR / VIRAR > NAVEGAR	44.98	73.97	69.41	53.75
LARGAR > ARRASTRAR / VIRAR > ARRASTRAR	76.42	77.81	76.02	79.35

Así mismo, se examinó la respuesta de los 4 modelos frente a una reducción de atributos de entrada. Para ello, se agruparon los atributos por sensores y se testearon todas las posibilidades. En la tabla 8 se compara la métrica F1 de la fase ARRASTRAR de los 4 modelos con todos los atributos y la mejor respuesta a una posible reducción de atributos agrupados por sensor. Se puede observar la mejora sustancial del LDA cuando se eliminaron las variables aportadas por el sensor de campo magnético y del modelo PNN utilizando únicamente los atributos asociados al GPS. Así mismo, MLP mejoró un 3% con los atributos asociados al sensor giróscopo y el GPS, y SVM no mejoró su respuesta reduciendo atributos.

Tabla 8. Exactitud, precisión, sensibilidad y valor F1 (expresados como porcentaje) de la categoría ARRASTRAR del test final de las máquinas de aprendizaje LDA, MLP, PNN y SVM, con todas las categorías y con reducción de atributos agrupados por sensores. Se muestran los resultados sin reducción (Ace-Gir-Mag-GPS) y el conjunto de sensores que ofrece mejor métrica F1 en la fase ARRASTRAR. Los atributos agrupados por sensores son: Ace = (AceXm, AceXs, AceYm, AceYs, AceZm, AceZs), Gir = (GirXm, GirXs, GirYm, GirYs, GirZm, GirZs), Mag = (MagXm, MagXs, MagYm, MagYs, MagZm, MagZs) y GPS = (GpsRm, GpsRs, GpsVm, GpsVs).

Sensores	Exactitud	Precisión	Sensibilidad	F1
LDA				
Ace-Gir-Mag-GPS	80.30	62.76	52.14	56.96
Ace-Gir-GPS	82.35	60.29	86.15	70.94
MLP				
Ace-Gir-Mag-GPS	87.24	71.46	81.97	76.26
Gir-GPS	88.62	74.39	83.37	78.57
PNN				
Ace-Gir-Mag-GPS	68.08	40.24	57.09	47.21
GPS	66.67	42.55	95.21	58.82
SVM				
Ace-Gir-Mag-GPS	68.59	44.20	97.61	60.84
-	-	-	-	-

La tabla 9 muestra los resultados de la prueba final de los cuatro modelos considerando la reducción de atributos y categorías. El modelo MLP se comportó notablemente mejor que los demás, con una tasa de aciertos del 89.69% en sus predicciones positivas y negativas relativas a la fase ARRASTRAR, una tasa de predicción positiva del 85.11%, una tasa de identificación de muestras de la fase ARRASTRAR del 96.26%, y una media armónica de precisión y sensibilidad F1 del 90.33%.

Tabla 9. Exactitud, precisión, sensibilidad y valor F1 (expresados como porcentaje) de la categoría ARRASTRAR del test final de las máquinas de aprendizaje LDA, MLP, PNN y SVM. Se muestran los resultados sin reducción (Todas actividades/Todos sensores) y con la reducción de categorías y atributos agrupados por sensores que ofrece mejor valor F1 en la fase ARRASTRAR.

Actividades/Sensores	Exactitud	Precisión	Sensibilidad	F1
LDA				
Todas actividades / Todos sensores	80.30	62.76	52.14	56.96
NAVEGAR-ARRASTAR / Ace-Gir-GPS	84.02	76.78	97.52	85.92
MLP				
Todas actividades / Todos sensores	87.24	71.46	81.97	76.26
NAVEGAR-ARRASTAR / Gir-GPS	89.69	85.11	96.26	90.33
PNN				
Todas actividades / Todos sensores	68.08	40.24	57.09	47.21
NAVEGAR-ARRASTAR / GPS	74.19	66.45	97.69	79.10
SVM				
Todas actividades / Todos sensores	68.59	44.20	97.61	60.84
NAVEGAR-ARRASTAR / Todos sensores	70.04	62.66	99.23	76.81

4 Discusión

El procesamiento de los datos en información y de la información en conocimiento, a fin de optimizar el proceso de toma de decisiones, es uno de los grandes desafíos que enfrentan los órganos de gestión que planifican la explotación de los recursos pesqueros. En este contexto, el sistema VMS se ha convertido en una fuente relevante de datos, impulsando la investigación sobre el control y la planificación de la actividad pesquera (Gerritsen *et al.*, 2012; Jennings y Lee, 2012). A pesar de ello, este sistema solo se utiliza en buques de gran tamaño y la información que ofrece ha mostrado ciertas deficiencias para identificar con precisión la presión que se ejerce sobre los caladeros (Bertrand *et al.*, 2008; Deng *et al.*, 2005; Mills *et al.*, 2007). Otras alternativas como la utilización de los dispositivos AIS puede mejorar la frecuencia de recepción de datos de cada buque, pero sigue utilizando únicamente el GPS como fuente de información y tampoco está implantado en la gran mayoría de embarcaciones pesqueras de litoral. Lo mismo ocurre con el sistema SLSEPA promovido por la Junta de Andalucía para el control de su flota costera ya que utiliza únicamente el GPS como fuente de información.

En la actualidad, los dispositivos de seguimiento tienen un coste tan alto que deja fuera de control casi la totalidad de la flota artesanal, solo proporciona la información que registra el GPS instalado en la embarcación, la frecuencia de envío de información es muy baja (entre 30 minutos y varias horas) y no envía los datos GPS entre muestras. La carencia de otros sensores le impide ampliar la información de lo que ocurre en la embarcación durante la totalidad de su actividad pesquera para mejorar en fiabilidad y precisión las conclusiones derivadas de esos datos. Tal es el caso de la velocidad de la embarcación en superficie, su movimiento, la distribución de las capturas por cada lance, la trayectoria precisa de los lances, etc.

Ampliar el número de sensores y la frecuencia de envío de los datos registrados al centro de control supondría un incremento sustancial del tráfico de datos generado por los dispositivos de seguimiento de cada embarcación, y con ello, un incremento del coste de mantenimiento. Con objeto de evitar el incremento de costes, el dispositivo de seguimiento debería disponer de capacidad de procesamiento para analizar los datos registrados por sus sensores y reducir el volumen de información a enviar al centro de control.

Mejorar los actuales dispositivos de seguimiento con nuevos sensores, capacidad de procesamiento suficiente y a un menor coste parece una tarea difícil, pero la solución puede que la tengamos al alcance de nuestras manos. Los dispositivos móviles son más económicos que cualquier dispositivo actual de seguimiento de buques. Éstos

incorporan habitualmente además de un GPS, tres sensores que podrían registrar el movimiento de la embarcación a frecuencias superiores a 50 muestras por segundo: acelerómetro, giróscopo y sensor de campo magnético. Además, disponen de procesador y memoria suficiente para procesar y analizar los datos proporcionados por los sensores y reducir sustancialmente la información a enviar al centro de control.

En esta línea, se ha propuesto utilizar un dispositivo móvil de bajo coste para registrar con sus sensores el movimiento de la embarcación de pesca, almacenar los datos registrados y dar un paso más allá, incorporando algoritmos de inteligencia artificial que identifiquen cuándo el barco está pescando. En el caso de la pesca de arrastre, los sensores del nuevo dispositivo denominado FAMIS, son capaces de detectar cambios significativos en el movimiento de la embarcación durante las cuatro fases del lance (Galotto-Tebar *et al.*, 2019).

En los siguientes apartados se analizan los cambios registrados en el comportamiento dinámico del buque: escora longitudinal, cabeceo, balance, guiñada y avance. Así mismo, con la intención de incorporar una máquina de aprendizaje que identifique la fase del lance en la que se encuentra la embarcación, se discuten los resultados obtenidos con 4 técnicas de clasificación de aprendizaje automático supervisado: análisis discriminante lineal, perceptrón multicapa, red neuronal probabilística y la máquina de soporte vectorial.

4.1 Dinámica del buque

Durante el lance, el arte de pesca es arrastrado por dos cables situados en ambas bandas de la popa del buque (Figura 33). Cada cable ejerce una fuerza de tracción sobre la embarcación que causa cambios en su comportamiento estático y dinámico (Sun *et al.*, 2011). Esta fuerza aparece cuando se larga el arte, aumenta durante la fase ARRASTRAR y es máxima cuando la velocidad de arrastre aumenta durante la fase VIRAR. En los siguientes apartados se analizan los cambios en el comportamiento del buque durante el lance.

4.1.1 Escora longitudinal

Cuando un buque se encuentra en posición de equilibrio, el centro de carena (punto donde se aplica el empuje ascendente que ejerce el mar) y el centro de masas (punto donde se aplica la fuerza descendente que ejerce la gravedad) se encuentran en la

misma línea vertical. Un movimiento del centro de masas debido a cambios en la carga, o un movimiento del centro de carena por inclinación del buque debido a fuerzas externas, provoca la aparición de un par adrizante que intenta recuperar la posición de equilibrio perdida (Figura 33).

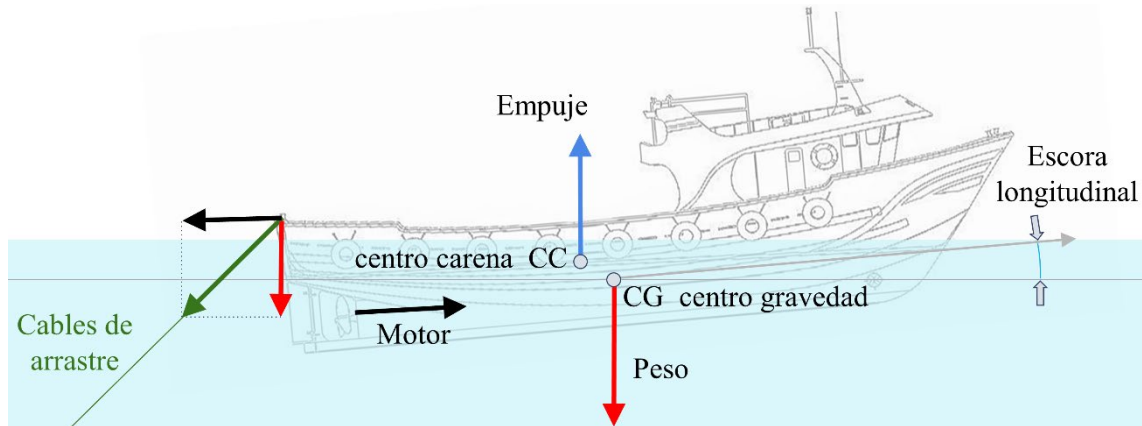


Figura 33. Escora longitudinal, centro de carena y centro de gravedad del buque. Fuerzas que actúan sobre el buque durante la actividad pesquera de arrastre.

Durante el lance, el arte de pesca es arrastrado por dos cables situados en ambas bandas de la popa del buque y ejercen cada uno de ellos una fuerza que se opone al avance del buque (componente horizontal) y modifica su desplazamiento (componente vertical). Su magnitud varía según la fase del lance en la que se encuentra el buque, pasando de ser nula cuando está NAVEGANDO, aparecer en el momento en que se empieza a LARGAR el arte, incrementarse cuando el buque se encuentra en la fase ARRASTRAR y ser máxima cuando se inicia la fase VIRAR o recogida del aparejo. Su dirección también varía en función del tipo de arte, profundidad de arrastre y velocidad del buque en cada fase. Como consecuencia de ello, a lo largo del lance se van produciendo cambios de comportamiento del buque en el mar, en su equilibrio estático y en su respuesta frente a fuerzas externas como el oleaje, corrientes, viento, etc.

Durante el lance, el incremento de las fuerzas verticales que generan los cables de arrastre provoca un desplazamiento progresivo hacia popa del centro de masas y un aumento del ángulo de escora longitudinal o asiento del buque. El análisis de las variables $AceY_m$ y $AceZ_m$ confirman que el acelerómetro fue capaz de registrar estos cambios en cada fase del lance.

4.1.2 Cabeceo y balance

Cuando un buque está en posición de equilibrio, el centro de flotabilidad (punto a través del cual la fuerza de flotabilidad que sostiene el buque actúa verticalmente hacia arriba) y el centro de gravedad (punto a través del cual todo el peso del recipiente actúa verticalmente hacia abajo) se encuentran en el mismo eje vertical.

Cuando una fuerza externa inclina la nave, se modifica la parte sumergida del casco, el centro de flotabilidad del casco se mueve y se genera un par adrizante que se opone a esta inclinación (Ibrahim y Grace, 2010; Romano y Marasco, 2014). La presencia de un arte de pesca activo provoca un desplazamiento del centro de masas que ayuda a incrementar el par adrizante y reducir el grado de cabeceo y balance de la embarcación (Sun *et al.*, 2011).

En el movimiento de cabeceo (Figura 34), el ascenso de la popa provoca un desplazamiento hacia proa del centro de carena, un incremento de la fuerza que generan ambos cables de arrastre, un desplazamiento hacia popa del centro de masas, un aumento de la distancia entre el centro de masas y el de carena y consecuentemente, un incremento del par adrizante. Un descenso de la popa provoca una reducción de la fuerza de arrastre, un desplazamiento hacia proa del centro de masas, un desplazamiento hacia popa del centro de carena y el mismo incremento del par adrizante. En ambos casos, el nivel de cabeceo se reduce con la presencia de un arte de pesca activo.

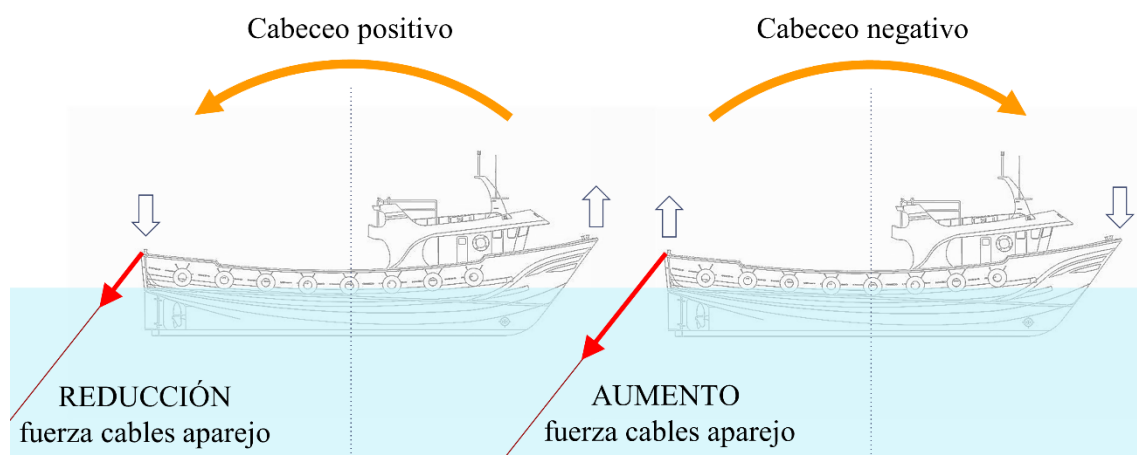


Figura 34. Movimiento de cabeceo del buque de pesca y su efecto sobre la fuerza que ejercen los cables de arrastre sobre el buque.

En el movimiento de balance (Figura 35) , el ascenso de la banda de estribor provoca un desplazamiento hacia babor del centro de carena, un aumento de la fuerza de arrastre del cable de estribor, una reducción de la fuerza de arrastre del cable de babor, un desplazamiento hacia estribor del centro de masas, un incremento de la distancia entre ambos centros y un incremento del par adrizante. El descenso de la banda de estribor provoca un efecto equivalente. En ambos casos, el nivel de balance se reduce con la presencia de un arte de pesca activo.

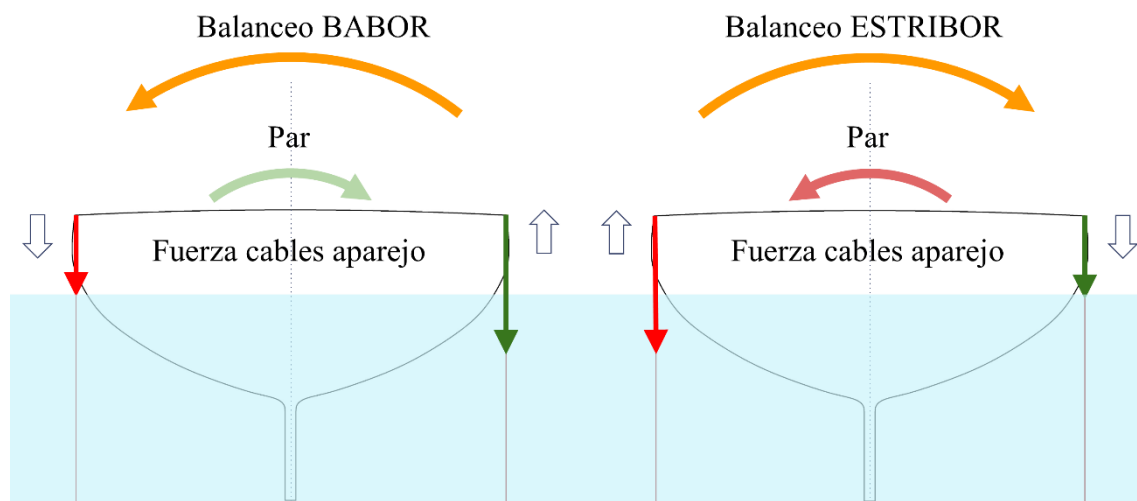


Figura 35. Movimiento de balance del buque de pesca y par adrizante que se genera por efecto de los cables de arrastre.

Los resultados obtenidos muestran que el acelerómetro, el giroscopio y el sensor de campo magnético fueron capaces de detectar una disminución en el nivel de cabeceo y balanceo durante la fase ARRASTRAR. En el caso del acelerómetro, las variables $AceX_s$ y $AceZ_s$ reflejaron una reducción significativa en su mediana y rango intercuartílico. Por otra parte, la variable procedente del giróscopo $GirY_s$ reflejó una reducción de su mediana y rango intercuartílico. Así mismo, las variables del sensor de campo magnético $MagX_s$ y $MagY_s$ registraron una reducción de su mediana y rango intercuartílico.

4.1.3 Guiñada

El movimiento de guiñada es un movimiento oscilatorio de rotación alrededor del eje OZ del barco que causa pequeños cambios en el rumbo de la nave (Figura 36). Un giro a estribor acelera la línea de arrastre de babor y ralentiza la línea de estribor, causando un desequilibrio en las fuerzas de arrastre que se oponen al giro del barco. Un giro a babor provoca el mismo efecto. Como resultado, durante el arrastre, el movimiento de guiñada se reduce y el rumbo del barco se mantiene más estable (Sun *et al.*, 2011).

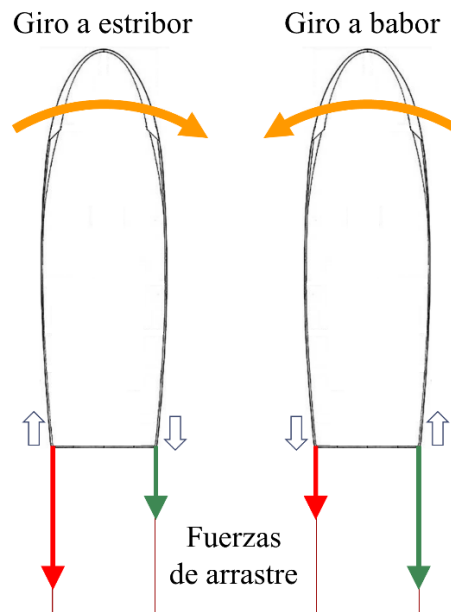


Figura 36. Movimiento de guiñada del buque de pesca y su efecto sobre las fuerzas de arrastre.

El giroscopio, el sensor de campo magnético y el GPS fueron capaces de detectar la reducción del ángulo de guiñada durante la fase ARRASTRAR. En el caso del giroscopio, la variable $GirZ_s$ mostró una reducción en la fase ARRASTRAR en su mediana y rango intercuartílico. Como se mencionó en la sección anterior, las variables $MagX_s$ y $MagY_s$ del sensor de campo magnético también mostraron una mayor estabilidad en el rumbo durante la fase ARRASTRAR. Por último, los resultados obtenidos con el GPS indican que la variable $GpsR_s$ mostró pequeñas reducciones de su mediana y una reducción significativa de su rango intercuartílico entre las fases NAVEGAR y ARRASTRAR. En las fases LARGAR y VIRAR se produce un aumento de la mediana de la variable $GpsR_s$ debido a que en estas fases suelen producirse cambios de rumbo para orientar el buque hacia el inicio de la fase ARRASTRAR o para dirigir el buque hacia la siguiente zona de pesca.

4.1.4 Avance

El estudio de la velocidad del barco durante las diferentes fases de la actividad pesquera se centra en el análisis de la velocidad media alcanzada voluntariamente actuando sobre el sistema de propulsión y observando la estabilidad de la velocidad deseada frente a fuerzas externas (oleaje, ráfagas de viento, etc.).

La variable $GpsV_m$ reflejó, en cada fase del lance, valores de velocidad similares a los encontrados en otros estudios relacionados con la pesca de arrastre (Burgos *et al.*, 2013; Lee *et al.*, 2010; Marzuki *et al.*, 2015), con medianas de 8.4 kn en la fase NAVEGAR, 4.5 kn en la fase LARGAR, 3 kn durante la fase ARRASTRAR y 2.1 kn en la fase VIRAR. Además, la variabilidad prevista de la velocidad se reflejó en sus rangos intercuartiles de 5.7 kn en la fase NAVEGAR, 1.2 kn en la fase LARGAR, 0.16 kn durante la fase ARRASTRAR y 1.4 kn en la fase VIRAR. Como muestra, en la figura 37 podemos observar la velocidad del buque oceanográfico Miguel Oliver a lo largo del lance número 4 frente a la costa de Castellón.

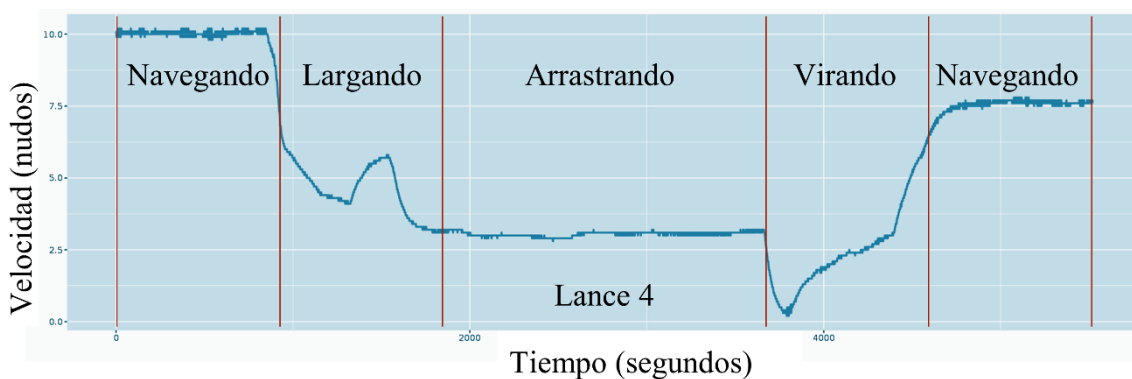


Figura 37. Velocidad del buque durante el lance 4 frente a la costa de Castellón (España).

Durante la fase ARRASTRAR, pequeñas variaciones en la velocidad causan cambios en la fuerza que ejercen los cables de arrastre que se opone al cambio de velocidad. El efecto amortiguador de la red de arrastre permite al buque mantener una velocidad de avance más estable (Sun *et al.*, 2011). El análisis de los datos de la variable $GpsV_s$ muestra que la variación de la velocidad es menor en la fase ARRASTRAR que en la fase NAVEGAR. Durante las fases LARGAR y VIRAR, la velocidad varía a voluntad del patrón, lo que justifica un aumento de la mediana y la dispersión de $GpsV_s$.

4.2 Máquinas de aprendizaje

Los resultados obtenidos en el análisis multivariante indicaron que gran parte de las variables originales se complementan y que el análisis de componentes principales no resultó útil para describir el comportamiento de las muestras y reducir su dimensión. El nivel tan bajo de varianza explicada por las componentes principales sugiere la presencia de una componente no lineal en las muestras y apunta la necesidad de utilizar técnicas más avanzadas capaces de identificar y clasificar dicho comportamiento.

Las cuatro máquinas de aprendizaje analizadas en este trabajo utilizan estrategias de clasificación muy distintas y, en consonancia con ello, su rendimiento de clasificación varía. Los resultados obtenidos con las muestras registradas en las cuatro fases del lance de arrastre y los atributos asociados a los cuatro sensores indicaron que los modelos SVM y MLP ofrecen mejores resultados que los modelos LDA y PNN. Esto se atribuye al hecho de que tanto MLPs como SVMs asocian funciones discriminantes altamente no lineales a los patrones de entrada, debido al gran número de variables en cada muestra, a las magnitudes físicas que representan y a la complejidad de la tarea de identificación que, en nuestro caso, se trata de la identificación de patrones de movimiento de una embarcación expuesta al oleaje, viento, motor, timón y aparejo de pesca.

Estos resultados son similares a los obtenidos por otros autores. Robotham *et al.* (2010) comprobaron que tanto MLPs como SVMs tuvieron una capacidad de clasificación significativamente mejores que los PNNs y LDAs, al intentar distinguir entre cuatro especies de peces pelágicos capturados frente a la costa norte de Chile basándose en los ecogramas de los bancos de peces. Además, Rodero *et al.* (2012) compararon los mismos cuatro tipos de modelos para discriminar razas bovinas andaluzas a partir de variables morfométricas y encontraron que los modelos MLPs y SVMs tenían un mejor rendimiento de clasificación.

La elevada no linealidad de los datos de entrada queda claramente reflejada en la incapacidad de los LDAs para encontrar una función discriminante que separe las muestras de diferentes clases. Así mismo, el modelo basado en PNN con un enfoque radial identificando nuevas muestras por su proximidad a las muestras de entrenamiento, proporcionó los resultados más pobres con un valor medio de la métrica F1 de solo el 42.68%. Por el contrario, el clasificador universal MLP con una sola capa oculta y un reducido número de neuronas alcanza un valor medio de F1 del 62.34%, siendo capaz de identificar casi el 82% de las muestras de la fase ARRASTRAR. Por último, la configuración más eficiente del modelo SVM con

núcleo lineal encuentra espacios con nuevas dimensiones en los que es posible separar diferentes clases con hiperplanos, proporcionando un aceptable valor medio de F1 del 61.36% y una excelente sensibilidad de la fase ARRASTRAR superior al 97%.

Estos resultados mejoran significativamente el rendimiento de la clasificación obtenido por los autores citados a partir de la información generada por VMS. Gerritsen y Lordan (2011) informaron que el uso de la velocidad del buque como criterio de clasificación de la actividad pesquera proporcionó una baja tasa de falsos negativos pero una tasa muy alta de falsos positivos en la clasificación (32%), y sólo el 68% de los casos se clasificaban correctamente. Estos autores destacaron que las posibles causas de estos resultados están relacionadas con el hecho de que una embarcación puede viajar a una velocidad similar a la de la pesca de arrastre cuando navega a baja velocidad mientras espera la marea correcta o como consecuencia del mal tiempo, entre otros factores.

En relación con esto, las cuatro fases de la pesca de arrastre identifican períodos de tiempo en los que el barco de pesca participa en diferentes actividades. En cada fase, el comportamiento dinámico del buque se ve influido por diversos elementos como la velocidad, el rumbo, las olas, el arte de pesca, la captura, etc. Estos elementos tienen un impacto diferente en cada fase. Por ejemplo, la fase NAVEGAR termina cuando el barco está maniobrando y frenando hasta que se posiciona en el lugar donde va a empezar a pescar. A continuación, la fase LARGAR comienza soltando la red, luego se colocan las puertas y se larga el cable de arrastre. En la propia fase ARRASTRAR, la resistencia que ejerce el arte de pesca sobre el buque crece a medida que aumenta el volumen de captura. A continuación, en la fase VIRAR, se recupera el cable de remolque, se levantan las puertas y, por último, se suben a bordo los aparejos de pesca con la captura. Al final del lance, el barco comienza una nueva fase de navegación, maniobrando y aumentando su velocidad para desplazarse al lugar del siguiente lance. Habida cuenta de todo ello, el efecto de las diferentes maniobras dentro de cada fase en el comportamiento del buque no puede ser detectado por los datos de velocidad proporcionados por el VMS, a menos que esta información se complemente con datos de sensores de movimiento como los registrados a través de FAMIS.

Por otra parte, un comportamiento homogéneo del buque dentro de una fase y marcados cambios entre fases deberían facilitar la tarea de clasificar las muestras e identificar el comienzo y el final de las fases del lance. El inicio de la fase LARGAR es el punto temporal más problemático. Esto se debe a que hasta que las puertas están en el agua, la resistencia de las redes es baja y el comportamiento del buque es similar al de la fase NAVEGAR. Por el contrario, la fase ARRASTRAR comienza cuando el arte de pesca, moviéndose a la velocidad de remolque, alcanza la profundidad

correcta, el cable deja de ser largado y la velocidad del buque se reduce a la velocidad de arrastre. Este cambio de velocidad permite a los modelos identificar las primeras muestras de la fase ARRASTRAR y, por lo tanto, el comienzo de la fase real de la pesca. Del mismo modo, la fase VIRAR es fácil de identificar, ya que comienza con una reducción de la velocidad del buque para facilitar la subida a bordo del arte de pesca y termina con el arte de pesca inactivo y un aumento de la velocidad del buque.

Los resultados de los modelos analizados son significativamente mejores cuando simplificamos la descripción de la actividad del buque pesquero a dos fases, un enfoque habitual de discriminar la actividad de un buque pesquero cuando los observadores a bordo registran si el buque está "pescando" o "no pescando" en función de si el arte de pesca está desplegado en el agua (Chang y Yuan, 2014).

Al fusionar las fases de NAVEGAR+LARGAR y ARRASTRAR+VIRAR, obtuvimos una puntuación F1 superior al 76% en todos los modelos. Estas fusiones propuestas evitan la dificultad de identificar las muestras al comienzo de la fase LARGAR y consideran que el buque está pescando desde que comienza a arrastrar el aparejo hasta que el arte de pesca sale del agua. Estos datos coinciden parcialmente con los resultados de Joo *et al.* (2015), quienes utilizaron los MLP para estimar los eventos de pesca en las pesquerías de anchoveta peruana, basados en los parámetros del VMS y validados *in situ* por observadores a bordo. En ese estudio, los observadores validaron dos segmentos de actividad en las series de datos de velocidad proporcionadas por el VMS, identificados como la aceleración entre el ritmo anterior y el actual (que coincidiría en parte con la fusión de NAVEGAR+LARGAR en marcha) y la aceleración entre el ritmo actual y el siguiente (aproximadamente ARRASTRAR+VIRAR).

En relación con los sistemas de sensores utilizados en este estudio, debemos indicar que los cuatro sensores proporcionaron información relevante sobre los cambios en el comportamiento del buque en las cuatro fases del lance (Galotto-Tebar *et al.*, 2019). La reducción del coste computacional asociado a la eliminación de las variables tuvo un impacto relativamente pequeño en el comportamiento de los modelos. El análisis de sensibilidad indicó que el sensor del GPS es sin duda el dispositivo que proporciona la información más importante, puesto que utilizando sólo este dispositivo los modelos evaluados lograron una puntuación media de F1 superior al 61%. Sin embargo, el uso de la información de los otros sensores mejoró el comportamiento de algunos modelos. Por ejemplo, el modelo MLP fue capaz de identificar correctamente el 96.26% de las muestras registradas por el GPS y el giróscopo en la fase ARRASTRAR.

Concretamente, para un lance estándar de 60 minutos, el perceptrón multicapa identificó el acto de pesca durante 57 minutos y 45 segundos. El error de predicción de 2 minutos y 45 segundos se traduce en el peor de los casos, con los falsos negativos concentrados al inicio o al final del arrastre, en la identificación de una operación de arrastre de menor duración y un adelanto o retraso en el comienzo o el final del arrastre. Por otro lado, debemos tener en cuenta que las características del barco en el que realizamos nuestro experimento (Miguel Oliver) hacen que tenga una relación entre el tamaño del barco y el arte de pesca más alta que en los barcos de pesca. Esto sugiere que la dinámica en un barco de pesca puede verse aún más afectada por el arte de pesca, por lo tanto, la diferencia entre las fases será más fácil de identificar utilizando este sistema.

Para finalizar, en la figura 38 se muestra gráficamente una parte de la prueba final de los modelos. En la imagen puede verse la predicción de cada modelo durante la jornada en la que se realizaron los lances 16 a 19 frente a la costa de Tarragona (Figura 8). Se ha seleccionado esta zona por la proximidad de los lances, los numerosos cambios de rumbo y la baja velocidad de navegación entre lances. En la imagen se ha resaltado la predicción del tramo final del lance 16 y parte de la navegación a baja velocidad hacia el inicio del lance 17, donde se pueden observar los falsos positivos de los 4 modelos analizados en este trabajo.

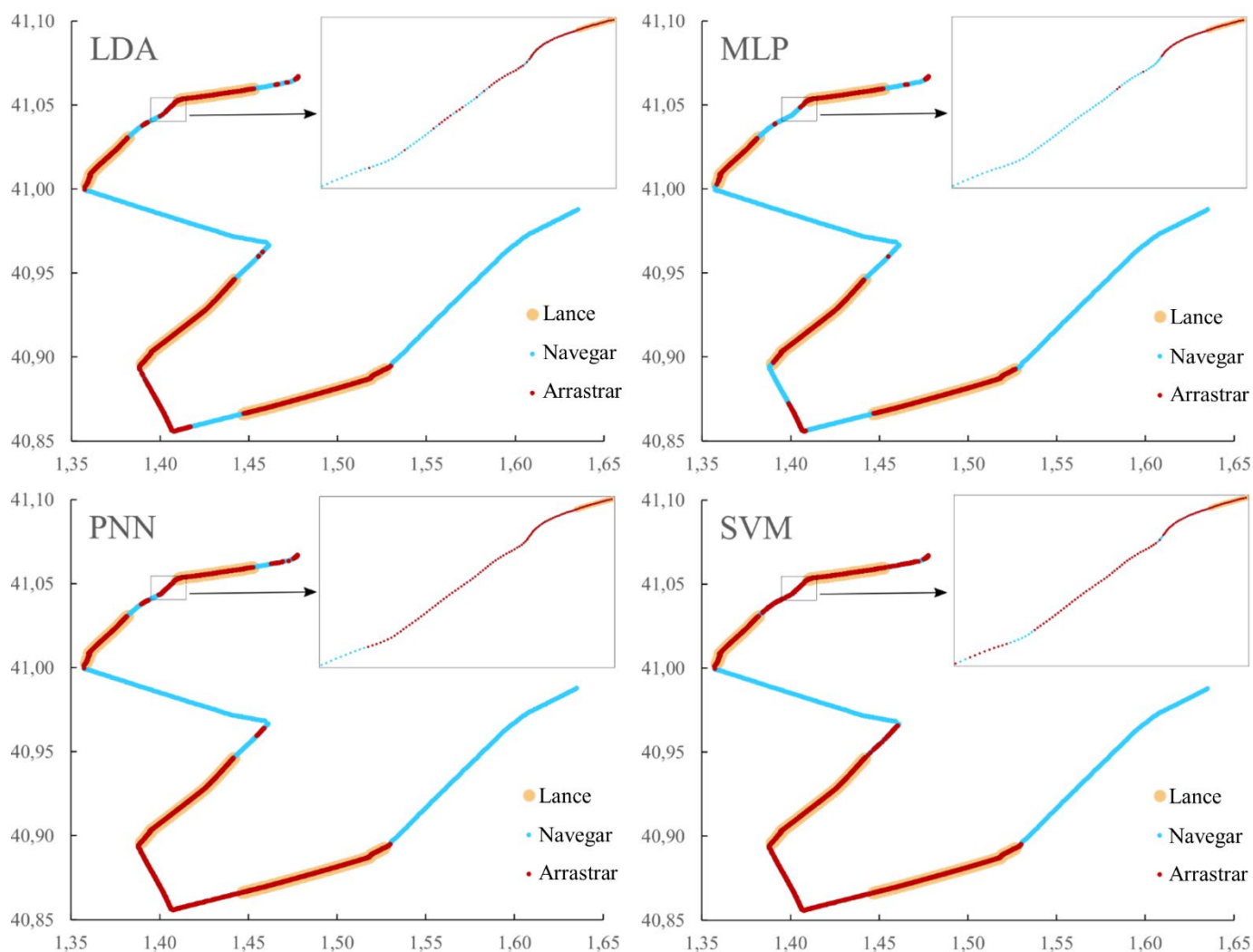


Figura 38. Representación gráfica de la predicción de 4 lances de los modelos LDA, MLP, PNN y SVM. La zona naranja representa los lances reales, los puntos azules la predicción NAVEGAR y los puntos rojos la predicción ARRASTRAR.

5 Conclusiones

Se propone incorporar en los buques de la flota pesquera un nuevo dispositivo denominado FAMIS capaz de identificar *in situ* y en tiempo real, la fase del lance de arrastre analizando el movimiento del buque registrado por sus sensores. Esta tesis representa los primeros pasos en el desarrollo de este nuevo dispositivo. Los resultados obtenidos nos permiten llegar a las siguientes conclusiones:

- Los sensores de un dispositivo móvil son capaces de detectar diferencias significativas en el comportamiento estático y dinámico del buque durante su actividad pesquera de arrastre:
 - El acelerómetro ha sido capaz de registrar un aumento progresivo de la escora longitudinal a lo largo de todo lance, durante las fases LARGAR, ARRASTRAR y VIRAR.
 - El acelerómetro, el giroscopio y el sensor de campo magnético han sido capaces de registrar una reducción en el nivel de cabeceo y balanceo durante la fase ARRASTRAR.
 - El giroscopio, el sensor de campo magnético y el GPS han sido capaces de detectar la reducción del ángulo de guiñada durante la fase ARRASTRAR.
 - El GPS ha sido capaz de registrar una velocidad más estable en la fase ARRASTRAR y más variable en las fases LARGAR y VIRAR.
- El análisis estadístico multivariante no ha resultado concluyente para reducir la dimensión de las muestras e identificar los sensores más representativos que permitan diferenciar las fases de la actividad pesquera.
- El resultado obtenido en el análisis estadístico multivariante sugiere la presencia de una componente no lineal en las muestras y apunta la necesidad de utilizar técnicas más avanzadas capaces de identificar y clasificar este comportamiento no lineal.
- Las 4 máquinas de aprendizaje analizadas muestran diferente capacidad de clasificación de la actividad pesquera de arrastre: los modelos SVM y MLP ofrecen mejores resultados que los modelos LDA y PNN.
- La reducción a 2 fases (NAVEGAR y ARRASTRAR) considerando que la fase ARRASTRAR comienza cuando el arte de pesca alcanza la profundidad y velocidad de arrastre correctos, y finaliza cuando el arte de pesca está sobre la cubierta del buque, facilita la tarea de clasificación y mejora sustancialmente los resultados de los 4 modelos.
- La reducción de los sensores que intervienen en cada modelo mejora la calidad de las predicciones de la mayoría de los modelos. En concreto, se obtienen los mejores resultados en el LDA con los sensores acelerómetro, giróscopo y GPS, en

el MLP con el gir6scopo y GPS, y en el PNN con el GPS. El modelo SVM no mejora sus resultados.

- La m1quina de aprendizaje que ofrece mejores resultados es el Perceptr6n Multicapa (MLP) identificando correctamente el 96.26% de las muestras de la fase ARRASTRAR utilizando 6nicamente el GPS y el Gir6scopo.

En resumen, los resultados de este trabajo indican que el nuevo sistema FAMIS es capaz de clasificar la informaci6n registrada por los sensores de un dispositivo m6vil durante la actividad pesquera e identificar en tiempo real la fase del lance en la que se encuentra el buque en cada momento.

6 Referencias

- Amat Rodrigo, J., 2017. Análisis de Componentes Principales (Principal Component Analysis, PCA) y t-SNE [WWW Document]. URL https://www.cienciadedatos.net/documentos/35_principal_component_analysis#C%C3%A1lculo_de_las_componentes_principales (accessed 12.30.21).
- Amat Rodrigo, J., 2016. Análisis discriminante lineal (LDA) y análisis discriminante cuadrático (QDA) [WWW Document]. URL https://www.cienciadedatos.net/documentos/28_linear_discriminant_analysis_lda_y_quadratic_discriminant_analysis_qda (accessed 2.22.21).
- Andayani, U., Wijaya, A., Rahmat, R.F., Siregar, B., Syahputra, M.F., 2019. Fish Species Classification Using Probabilistic Neural Network. Presented at the Journal of Physics: Conference Series. <https://doi.org/10.1088/1742-6596/1235/1/012094>
- Anzola, N.S., 2015. Máquinas de soporte vectorial y redes neuronales artificiales en la predicción del movimiento USD/COP spot intradiario [WWW Document]. ODEON. <https://doi.org/10.18601/17941113.n9.04>
- Barkin, J., DeSombre, E.R., Ishii, A., Sakaguchi, I., 2018. Domestic sources of international fisheries diplomacy: A framework for analysis. *Marine Policy* 94, 256–263. <https://doi.org/10.1016/j.marpol.2018.04.030>
- Barrass, C.B., Derrett, D.R., 2012. Ship Stability for Masters and Mates, Ship Stability for Masters and Mates. Elsevier Ltd. <https://doi.org/10.1016/C2010-0-68323-4>
- Bastardie, F., Nielsen, J.R., Ulrich, C., Egekvist, J., Degel, H., 2010. Detailed mapping of fishing effort and landings by coupling fishing logbooks with satellite-recorded vessel geo-location. *Fisheries Research* 106, 41–53. <https://doi.org/10.1016/j.fishres.2010.06.016>
- Behivoke, F., Etienne, M.-P., Guitton, J., Randriatsara, R.M., Ranaivoson, E., Léopold, M., 2021. Estimating fishing effort in small-scale fisheries using GPS tracking data and random forests. *Ecological Indicators* 123, 107321. <https://doi.org/10.1016/j.ecolind.2020.107321>
- Bergmeir, C., Benítez, J.M., team), A.Z. (Part of original S. development, team), N.M. (Part of original S. development, team), G.M. (Part of original S. development, team), M.V. (Part of original S. development, team), S.D. (Part of original S.

development, team), R.H. (Part of original S. development, team), K.-U.H. (Part of original S. development, team), Tobias Soye (Part of original SNNS development, team), M.S. (Part of original S. development, team), Tilman Sommer (Part of original SNNS development, team), A.H. (Part of original S. development, team), D.P. (Part of original S. development, team), Tobias Schreiner (Part of original SNNS development, team), B.K. (Part of original S. development, contributors), Martin Reczko (Part of original SNNS external, contributors), Martin Riedmiller (Part of original SNNS external, contributors), M.S. (Part of original S. external, contributors), Marcus Ritt (Part of original SNNS external, contributors), J.D. (Part of original S. external, contributors), J.B. (Part of original S. external, team), J.D. (Part of original S. development, team), C.W. (Part of original S. development, Version 4.3), P.K. (Contributors to S., Version 4.3), A.E.-A. (Contributors to S., 2019). RSNNS: Neural Networks using the Stuttgart Neural Network Simulator (SNNS).

Bertrand, J.A., Gil de Sola, L., Papaconstantinou, C., Relini, G., Souplet, A., 2002. The general specifications of the MEDITS surveys. *Scientia Marina* 66, 9–17. <https://doi.org/10.3989/scimar.2002.66s29>

Bertrand, S., Burgos, J.M., Gerlotto, F., Atiquipa, J., 2005. Lévy trajectories of Peruvian purse-seiners as an indicator of the spatial distribution of anchovy (*Engraulis ringens*). *ICES Journal of Marine Science* 62, 477–482. <https://doi.org/10.1016/j.icesjms.2004.12.002>

Bertrand, S., Díaz, E., Lengaigne, M., 2008. Patterns in the spatial distribution of Peruvian anchovy (*Engraulis ringens*) revealed by spatially explicit fishing data. *Progress in Oceanography* 79, 379–389. <https://doi.org/10.1016/j.pocean.2008.10.009>

Burgos, C., Gil, J., del Olmo, L.A., 2013. The Spanish blackspot seabream (*Pagellus bogaraveo*) fishery in the Strait of Gibraltar: spatial distribution and fishing effort derived from a small-scale GPRS/GSM based fisheries vessel monitoring system. *Aquatic Living Resources* 26, 399–407. <https://doi.org/10.1051/alr/2013068>

Burman, P., 1989. A comparative study of ordinary cross-validation, v-fold cross-validation and the repeated learning-testing methods. *Biometrika* 76, 503–514. <https://doi.org/10.1093/biomet/76.3.503>

- Carpentieri, P., Bonanno, A., Scarcella, G., 2020. Technical guidelines for scientific surveys in the Mediterranean and the Black Sea: Procedures and sampling for demersal (bottom and beam) trawl surveys and pelagic acoustic surveys, FAO Documento técnico de pesca y acuicultura. FAO, Rome, Italy. <https://doi.org/10.4060/ca8870en>
- CEE, 1992. Reglamento (CE) nº 3760/92 del Consejo por el que se establece un régimen comunitario de la pesca y la acuicultura, OJ L.
- Chang, S.-K., Yuan, T.-L., 2014. Deriving high-resolution spatiotemporal fishing effort of large-scale longline fishery from vessel monitoring system (VMS) data and validated by observer data. *Canadian Journal of Fisheries and Aquatic Sciences* 71, 1363–1370.
- Charles, A.T., 1992. Fishery conflicts: A unified framework. *Marine Policy* 16, 379–393. [https://doi.org/10.1016/0308-597X\(92\)90006-B](https://doi.org/10.1016/0308-597X(92)90006-B)
- Chasset, P.O., 2016. Package pnn [WWW Document]. URL <https://cran.r-project.org/web/packages/pnn/pnn.pdf>
- China, 2021. BeiDou Navigation Satellite System [WWW Document]. URL <http://en.beidou.gov.cn/> (accessed 12.6.21).
- Chuaysi, B., Kiattisin, S., 2020. Fishing Vessels Behavior Identification for Combating IUU Fishing: Enable Traceability at Sea. *Wireless Pers Commun* 1–23. <https://doi.org/10.1007/s11277-020-07200-w>
- Cojan, M., Burgos, C., 2015. Análisis de la información proporcionada por los sistemas de localización vía satélite de la flota que explota la chirla (*Chamelea gallina*) en el Golfo de Cádiz sistemas de localización vía satélite de la flota que explota la chirla (*Chamelea gallina*), in: *Teledetección: Humedales y Espacios Protegidos. XVI Congreso de La Asociación Española de Teledetección*. pp. 550–553.
- Csirke, J., Carrasco, R.G., Quintana, G.C., Carranza, M.Ñ., Montenegro, A.C., 1996. Situación de los recursos Anchoveta (*Engraulis Ringens*) y Sardina (*Sardinops sagax*) a principios de 1994 y perspectivas para la pesca en el Perú, con particular preferencia a las regiones norte y centro de la costa peruana. *Boletín Instituto del Mar del Perú* 15, 1–23.

- Dai, L., Vorselen, D., Korolev, K.S., Gore, J., 2012. Generic Indicators for Loss of Resilience Before a Tipping Point Leading to Population Collapse. *Science* 336, 1175–1177. <https://doi.org/10.1126/science.1219805>
- de Souza, E.N., Boerder, K., Matwin, S., Worm, B., 2016. Improving Fishing Pattern Detection from Satellite AIS Using Data Mining and Machine Learning. *PLOS ONE* 11, e0158248–e0158248. <https://doi.org/10.1371/journal.pone.0158248>
- Deng, R., Dichmont, C., Milton, D., Haywood, M., Vance, D., Hall, N., Die, D., 2005. Can vessel monitoring system data also be used to study trawling intensity and population depletion? The example of Australia's northern prawn fishery. *Canadian Journal of Fisheries and Aquatic Sciences* 62, 611–622. <https://doi.org/10.1139/f04-219>
- Dremière, P.-Y., Fiorentini, L., Cosimi, G., Leonori, I., Sala, A., Spagnolo, A., 1999. Escapement from the main body of the bottom trawl used for the Mediterranean international trawl survey (MEDITS). *Aquatic Living Resources* 12, 207–217. [https://doi.org/10.1016/S0990-7440\(00\)88471-5](https://doi.org/10.1016/S0990-7440(00)88471-5)
- European Commission, 2007. On improving fishing capacity and effort indicators under the common fisheries policy. Brussels.
- European Parliament, 2011. Commission Directive 2011/15/EU of 23 February 2011 amending Directive 2002/59/EC of the European Parliament and of the Council establishing a Community vessel traffic monitoring and information system.
- European Space Agency, 2021. Galileo [WWW Document]. URL <https://www.esa.int/> (accessed 12.6.21).
- European Union, 2009. Council Regulation (EC) No 1224/2009 of 20 November 2009 establishing a Community control system for ensuring compliance with the rules of the common fisheries policy, amending Regulations (EC) No 847/96, (EC) No 2371/2002, (EC) No 811/2004, (EC) No 768/2005, (EC) No 2115/2005, (EC) No 2166/2005, (EC) No 388/2006, (EC) No 509/2007, (EC) No 676/2007, (EC) No 1098/2007, (EC) No 1300/2008, (EC) No 1342/2008 and repealing Regulations (EEC) No 2847/93, (EC) No 1627/94 and (EC) No 1966/2006, OJ L.

- FAO, 2021. Small-scale fisheries and the human right to adequate food [Policy Support and Governance] Food and Agriculture Organization of the United Nations.
- FAO, 2020. El estado mundial de la pesca y la acuicultura 2020. Versión resumida. FAO. <https://doi.org/10.4060/ca9231es>
- FAO, 2019. New global atlas on using advanced technology to monitor fishing activity [WWW Document]. URL <http://www.fao.org/news/story/en/item/1251799/icode/> (accessed 1.24.21).
- FAO, 2018. Directrices voluntarias para lograr la sostenibilidad de la pesca en pequeña escala en el contexto de la seguridad alimentaria y la erradicación de la pobreza [WWW Document]. URL <http://www.fao.org/voluntary-guidelines-small-scale-fisheries/es/> (accessed 5.7.21).
- FAO, 2001. Directrices para la recopilación sistemática de datos relativos a la pesca de captura.
- FAO, 2000. Indicadores para el desarrollo sostenible de la pesca de captura marina, FAO Orientaciones Técnicas para la Pesca Responsable. FAO.
- FAO Fishery Resources Division, 1999. Indicators for sustainable development of marine capture fisheries (No. 92-5-104333-7), FAO Technical Guidelines for Responsible Fisheries. Rome.
- Ferguson-Cradler, G., 2018. Fisheries' collapse and the making of a global event, 1950s–1970s. *Journal of Global History* 13, 399–424. <https://doi.org/10.1017/S1740022818000219>
- Fiorentini, L., Dremière, P.-Y., Leonori, I., Sala, A., Palumbo, V., 1999. Efficiency of the bottom trawl used for the Mediterranean international trawl survey (MEDITS). *Aquatic Living Resources* 12, 187–205. [https://doi.org/10.1016/S0990-7440\(00\)88470-3](https://doi.org/10.1016/S0990-7440(00)88470-3)
- Fisher, R.A., 1936. The Use of Multiple Measurements in Taxonomic Problems. *Annals of Eugenics* 7, 179–188. <https://doi.org/10.1111/j.1469-1809.1936.tb02137.x>
- Fontana, I., Giacalone, G., Bonanno, A., Mazzola, S., Basilone, G., Genovese, S., Aronica, S., Pissis, S., Iliopoulos, C.S., Kundu, R., Fiannaca, A., Langiu, A., Lo Bosco, G., La Rosa, M., Rizzo, R., 2017. Pelagic species identification by

using a PNN neural network and echo-sounder data. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 10613 LNCS, 454–455.

Frank E., H., 2021. Hmisc: Harrell Miscellaneous.

Galotto-Tebar, M.M., Pomares-Padilla, A., Czerwinski, I.A., Gutiérrez-Estrada, J.C., 2019. Using mobile device's sensors to identify fishing activity. *Journal of Marine Science and Technology (Japan)* 1–12. <https://doi.org/10.1007/s00773-019-00694-5>

Garza García, J. de la., Morales Serrano, B.Nieves., González Cavazos, B.Adriana., 2013. *Análisis estadístico multivariante*. McGraw-Hill.

Gastwirth, J.L., Gel, Y.R., Hui, W.L.W., Lyubchich, V., Miao, W., Noguchi, K., 2020. Package 'lawstat.'

Gerritsen, H., Lordan, C., 2011. Integrating vessel monitoring systems (VMS) data with daily catch data from logbooks to explore the spatial distribution of catch and effort at high resolution. *ICES Journal of Marine Science* 68, 245–252. <https://doi.org/10.1093/icesjms/fsq137>

Gerritsen, H.D., Lordan, C., Minto, C., Kraak, S.B.M., 2012. Spatial patterns in the retained catch composition of Irish demersal otter trawlers: High-resolution fisheries data as a management tool. *Fisheries Research* 129–130, 127–136. <https://doi.org/10.1016/j.fishres.2012.06.019>

Global Fishing Watch, 2018. Global Fishing Watch [WWW Document]. URL <https://globalfishingwatch.org/> (accessed 1.24.21).

Goñi, R., 1998. Ecosystem effects of marine fisheries: an overview. *Ocean & Coastal Management* 40, 37–64. [https://doi.org/10.1016/S0964-5691\(98\)00037-4](https://doi.org/10.1016/S0964-5691(98)00037-4)

Gutiérrez-Estrada, J.C., Pulido-Calvo, I., 2015. Is the Atlantic surface temperature a good proxy for forecasting the recruitment of European eel in the Guadalquivir estuary? *Progress in Oceanography* 130, 112–124. <https://doi.org/10.1016/j.pocean.2014.10.007>

Gutiérrez-Estrada, J.C., Yáñez, E., Pulido-Calvo, I., Silva, C., Plaza, F., Bórquez, C., 2009. Pacific sardine (*Sardinops sagax*, Jenyns 1842) landings prediction. A neural network ecosystemic approach. *Fisheries Research* 100, 116–125. <https://doi.org/10.1016/j.fishres.2009.06.014>

- Hair, J., Anderson, R., Tatham, R., Black, W., 1999. *Análisis multivariante*, 5th ed. Prentice Hall Iberia.
- Hery, Lukas, S., Yugopuspito, P., Murwantara, I.M., Krisnadi, D., 2020. Website Design for Locating Tuna Fishing Spot Using Naïve Bayes and SVM Based on VMS Data on Indonesian Sea. 2020 3rd International Seminar on Research of Information Technology and Intelligent Systems (ISRITI), Research of Information Technology and Intelligent Systems (ISRITI), 2020 3rd International Seminar on 89–93. <https://doi.org/10.1109/ISRITI51436.2020.9315338>
- Hintzen, N.T., Bastardie, F., Beare, D., Piet, G.J., Ulrich, C., Deporte, N., Egekvist, J., Degel, H., 2012. VMStools: Open-source software for the processing, analysis and visualisation of fisheries logbook and VMS data. *Fisheries Research* 115–116, 31–43. <https://doi.org/10.1016/J.FISHRES.2011.11.007>
- Husson, F., Sébastien, J.J., Rennes, A., Mazet, J., 2022. FactoMineR: Exploratory Multivariate Data Analysis with R [WWW Document]. URL <http://factominer.free.fr/> (accessed 1.1.22).
- Ibrahim, R.A., Grace, I.M., 2010. Modeling of Ship Roll Dynamics and Its Coupling with Heave and Pitch. *Mathematical Problems in Engineering* 2010, 1–32. <https://doi.org/10.1155/2010/934714>
- Ihaka, R., Gentleman, R., 1996. R: A Language for Data Analysis and Graphics. *Journal of Computational and Graphical Statistics* 5, 299–314. <https://doi.org/10.1080/10618600.1996.10474713>
- International Maritime Organization, 1974. International Convention for the Safety of Life at Sea (SOLAS) [WWW Document]. URL [https://www.imo.org/en/About/Conventions/Pages/International-Convention-for-the-Safety-of-Life-at-Sea-\(SOLAS\)%2c-1974.aspx](https://www.imo.org/en/About/Conventions/Pages/International-Convention-for-the-Safety-of-Life-at-Sea-(SOLAS)%2c-1974.aspx) (accessed 1.19.21).
- International Telecommunication Union, 2014. M.1371 : Technical characteristics for an automatic identification system using time-division multiple access in the VHF maritime mobile band.
- Jennings, S., Lee, J., 2012. Defining fishing grounds with vessel monitoring system data. *ICES Journal of Marine Science* 69, 51–63. <https://doi.org/10.1093/icesjms/fsr173>

- Joo, R., Salcedo, O., Gutierrez, M., Fablet, R., Bertrand, S., 2015. Defining fishing spatial strategies from VMS data: Insights from the world's largest monospecific fishery. *Fisheries Research* 164, 223–230. <https://doi.org/10.1016/j.fishres.2014.12.004>
- Junta de Andalucía, 2004. Localización y seguimiento de embarcaciones pesqueras (SLSEPA) [WWW Document]. URL <http://www.juntadeandalucia.es/organismos/agriculturapescayderosarrollorural/areas/pesca-acuicultura/slsepa.html>
- Kaiser, H.F., 1958. The varimax criterion for analytic rotation in factor analysis. *Psychometrika* 23, 187–200. <https://doi.org/10.1007/BF02289233>
- Komsta, L., Novomestky, F., 2015. moments: Moments, cumulants, skewness, kurtosis and related tests.
- Lee, J., South, A.B., Jennings, S., 2010. Developing reliable, repeatable, and accessible methods to provide high-resolution estimates of fishing-effort distributions from vessel monitoring system (VMS) data. *ICES Journal of Marine Science* 67, 1260–1271. <https://doi.org/10.1093/icesjms/fsq010>
- Liew, S.S., Khalil-Hani, M., Bakhteri, R., 2016. Bounded activation functions for enhanced training stability of deep neural networks on visual pattern recognition problems. *Neurocomputing* 216, 718–734. <https://doi.org/10.1016/j.neucom.2016.08.037>
- Liu, R., Liu, E., Yang, J., Li, M., Wang, F., 2006. Optimizing the Hyper-parameters for SVM by Combining Evolution Strategies with a Grid Search, in: Huang, D.-S., Li, K., Irwin, G.W. (Eds.), *Intelligent Control and Automation: International Conference on Intelligent Computing, ICIC 2006 Kunming, China, August 16–19, 2006, Lecture Notes in Control and Information Sciences*. Springer, Berlin, Heidelberg, pp. 712–721. https://doi.org/10.1007/978-3-540-37256-1_87
- Marzuki, M.I., Garelo, R., Fablet, R., Kerbaol, V., Gaspar, P., 2015. Fishing gear recognition from VMS data to identify illegal fishing activities in Indonesia, in: *MTS/IEEE OCEANS 2015 - Genova*. Institute of Electrical and Electronics Engineers Inc., pp. 1–5. <https://doi.org/10.1109/OCEANS-Genova.2015.7271551>

- Mathias, A.C., Rech, P.C., 2012. Hopfield neural network: The hyperbolic tangent and the piecewise-linear activation functions. *Neural Networks* 34, 42–45. <https://doi.org/10.1016/j.neunet.2012.06.006>
- McCauley, D.J., Woods, P., Sullivan, B., Bergman, B., Jablonicky, C., Roan, A., Hirshfield, M., Boerder, K., Worm, B., 2016. Ending hide and seek at sea. *Science* 351, 1148–1150. <https://doi.org/10.1126/science.aad5686>
- Meyer, D., 2019. Package e1071 [WWW Document]. URL <https://cran.r-project.org/web/packages/e1071/e1071.pdf>
- Mills, C.M., Townsend, S.E., Jennings, S., Eastwood, P.D., Houghton, C.A., 2007. Estimating high resolution trawl fishing effort from satellite-based vessel monitoring system data. *ICES Journal of Marine Science* 64, 248–255. <https://doi.org/10.1093/icesjms/fsl026>
- Mustakim, Hidayat, A., Efendi, Z., Aszani, Novita, R., Lestari, E.T., 2018. Algorithm comparison of naive bayes classifier and probabilistic neural network for water area classification of fishing vessel in Indonesia. *Journal of Theoretical and Applied Information Technology* 96, 4114–4125.
- Natale, F., Gibin, M., Alessandrini, A., Vespe, M., Paulrud, A., 2015. Mapping fishing effort through AIS data. *PLoS ONE* 10. <https://doi.org/10.1371/journal.pone.0130746>
- O’Farrell, S., Sanchirico, J.N., Chollett, I., Cockrell, M., Murawski, S.A., Watson, J.T., Haynie, A., Strelcheck, A., Perruso, L., 2017. Improving detection of short-duration fishing behaviour in vessel tracks by feature engineering of training data. *ICES J Mar Sci* 74, 1428–1436. <https://doi.org/10.1093/icesjms/fsw244>
- Oliva Teles, L., Fernandes, M., Amorim, J., Vasconcelos, V., 2015. Video-tracking of zebrafish (*Danio rerio*) as a biological early warning system using two distinct artificial neural networks: Probabilistic neural network (PNN) and self-organizing map (SOM). *Aquatic Toxicology* 165, 241–248. <https://doi.org/10.1016/j.aquatox.2015.06.008>
- ONU, 2015. Agenda 2030 para el Desarrollo Sostenible [WWW Document]. Desarrollo Sostenible. URL <https://www.un.org/sustainabledevelopment/es/2015/09/la-asamblea-general-adapta-la-agenda-2030-para-el-desarrollo-sostenible/> (accessed 5.7.21).

- Paola, A.-C., Alvaro Agustín, O.-B., Eydy del Carmen, S.-B., Ana, O.-C., Miguel, U.T., Marlon, P.-M., Shariq Aziz, B., Collazos Morales, C.A., Ramón Enrique, R.G., 2021. Machine Learning approach applied to Human Activity Recognition – An application to the VanKasteren dataset. *Procedia Computer Science*, The 18th International Conference on Mobile Systems and Pervasive Computing (MobiSPC), The 16th International Conference on Future Networks and Communications (FNC), The 11th International Conference on Sustainable Energy Information Technology 191, 367–372. <https://doi.org/10.1016/j.procs.2021.07.070>
- Pedreschi, D., Vigier, A., Höffle, H., Kraak, S.B.M., Reid, D.G., 2021. Innovation through consultation: Stakeholder perceptions of a novel fisheries management system reveal flexible approach to solving fisheries challenges. *Marine Policy* 124, 104337. <https://doi.org/10.1016/j.marpol.2020.104337>
- Pontecorvo, G., Schrank, W.E., 2014. The continued decline in the world catch of marine fish. *Marine Policy* 44, 117–119. <https://doi.org/10.1016/j.marpol.2013.08.016>
- Pontecorvo, G., Schrank, W.E., 2012. The expansion, limit and decline of the global marine fish catch. *Marine Policy* 36, 1178–1181. <https://doi.org/10.1016/j.marpol.2012.03.005>
- Purwandari, K., Sigalingging, J.W.C., Cenggoro, T.W., Pardamean, B., 2021. Multi-class Weather Forecasting from Twitter Using Machine Learning Approaches. *Procedia Computer Science*, 5th International Conference on Computer Science and Computational Intelligence 2020 179, 47–54. <https://doi.org/10.1016/j.procs.2020.12.006>
- Raj, R.J.R., Sasipraba, T., Vasudev, M., Gupta, S., Rizwan, Md., srivastava, P., 2016. Predicting the Impact of Climate Change on Tidal Zone Fishes Using SVM Approach. *Procedia Computer Science*, 2nd International Conference on Intelligent Computing, Communication & Convergence, ICC3 2016, 24-25 January 2016, Bhubaneswar, Odisha, India 92, 237–243. <https://doi.org/10.1016/j.procs.2016.07.351>
- Ripley, B., Venables, B., Bates, D.M., Hornik, K., Gebhardt, A., Firth, D., 2020. Package mass [WWW Document]. URL <http://www.stats.ox.ac.uk/pub/MASS4/>

- Robertson, S.G., Morison, A.K., 2006. Age Estimation of Fish Using a Probabilistic Neural Network, in: Recknagel, F. (Ed.), *Ecological Informatics: Scope, Techniques and Applications*. Springer, Berlin, Heidelberg, pp. 445–458. https://doi.org/10.1007/3-540-28426-5_22
- Robotham, H., Bosch, P., Gutiérrez-Estrada, J.C., Castillo, J., Pulido-Calvo, I., 2010. Acoustic identification of small pelagic fish species in Chile using support vector machines and neural networks. *Fisheries Research* 102, 115–122. <https://doi.org/10.1016/j.fishres.2009.10.015>
- Rodero, E., González, A., Luque, M., Herrera, M., Gutiérrez-Estrada, J.C., 2012. Classification of Spanish autochthonous bovine breeds. Morphometric study using classical and heuristic techniques. *Livestock Science* 143, 226–232. <https://doi.org/10.1016/j.livsci.2011.09.022>
- Romano, A., Marasco, A., 2014. *Fluid Dynamics and Ship Motion*. Birkhäuser, New York, NY, pp. 429–461. https://doi.org/10.1007/978-1-4939-1604-7_13
- Rose, G., Kulka, D., 1999. Hyperaggregation of fish and fisheries: how catch-per-unit-effort increased as the northern cod (*Gadus morhua*) declined. *Can. J. Fish. Aquat. Sci.* 56, 118–127. <https://doi.org/10.1139/f99-207>
- R-proyect, 2022. R: The R Stats Package [WWW Document]. URL <https://stat.ethz.ch/R-manual/R-devel/library/stats/html/00Index.html> (accessed 1.9.22).
- Rumelhart, D.E., Hinton, G.E., Williams, R.J., 1986. Learning representations by backpropagation errors. *Nature* 323, 533–536.
- Russian Federal Space Agency, 2021. GLONASS navigation system [WWW Document]. URL <http://www.russianspaceweb.com/glonass.html> (accessed 12.6.21).
- Russo, E., Monti, M.A., Mangano, M.C., Raffaetà, A., Sarà, G., Silvestri, C., Pranovi, F., 2020. Temporal and spatial patterns of trawl fishing activities in the Adriatic Sea (Central Mediterranean Sea, GSA17). *Ocean & Coastal Management* 192, 105231. <https://doi.org/10.1016/j.ocecoaman.2020.105231>
- Rynkiewicz, J., 2019. Asymptotic statistics for multilayer perceptron with ReLU hidden units. *Neurocomputing, Advances in artificial neural networks, machine learning and computational intelligence* 342, 16–23. <https://doi.org/10.1016/j.neucom.2018.11.097>

- Shepperson, J., Murray, L.G., Mackinson, S., Bell, E., Kaiser, M.J., 2016. Use of a choice-based survey approach to characterise fishing behaviour in a scallop fishery. *Environmental Modelling & Software* 86, 116–130. <https://doi.org/10.1016/J.ENVSOFT.2016.09.013>
- Specht, D.F., 1990. Probabilistic Neural Networks. *Neural Networks* 3, 109–118. [https://doi.org/10.1016/0893-6080\(90\)90049-Q](https://doi.org/10.1016/0893-6080(90)90049-Q)
- Sun, Q., Liu, H., Wang, W., 2020. Evaluating the Quality of GANs Based on F1 Harmonic Mean. *Procedia Computer Science, Proceedings of the 3rd International Conference on Mechatronics and Intelligent Robotics (ICMIR-2019)* 166, 463–468. <https://doi.org/10.1016/j.procs.2020.02.064>
- Sun, X., Yin, Y., Jin, Y., Zhang, Xinyu, Zhang, Xiufeng, 2011. The modeling of single-boat, mid-water trawl systems for fishing simulation. *Fisheries Research* 109, 7–15. <https://doi.org/10.1016/j.fishres.2010.12.027>
- Susanto, H., Wibisono, G., 2019. Marine vessel telemetry data processing using machine learning, in: *International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)*. Institute of Advanced Engineering and Science, pp. 128–135. <https://doi.org/10.23919/EECSI48112.2019.8976923>
- Szostek, C.L., Murray, L.G., Bell, E., Kaiser, M.J., 2017. Filling the gap: Using fishers' knowledge to map the extent and intensity of fishing activity. *Marine Environmental Research* 129, 329–346. <https://doi.org/10.1016/j.marenvres.2017.06.012>
- Teh, L.C.L., Pauly, D., 2018. Who Brings in the Fish? The Relative Contribution of Small-Scale and Industrial Fisheries to Food Security in Southeast Asia. *Front. Mar. Sci.* 5. <https://doi.org/10.3389/fmars.2018.00044>
- Tu, Q., Yang, Y., Huang, H., Li, L., Xiong, S., Zhao, S., 2020. Quantity prediction of mixed freshwater fish based on passive underwater acoustic signals. *Transactions of the ASABE* 63, 1805–1811. <https://doi.org/10.13031/TRANS.13495>
- Unión Europea, 2014. Reglamento (UE) n ° 508/2014 del Parlamento Europeo y del Consejo, de 15 de mayo de 2014 , relativo al Fondo Europeo Marítimo y de Pesca, y por el que se derogan los Reglamentos (CE) n ° 2328/2003, (CE) n

- ° 861/2006, (CE) n ° 1198/2006 y (CE) n ° 791/2007 del Consejo, y el Reglamento (UE) n ° 1255/2011 del Parlamento Europeo y del Consejo, OJ L.
- Unión Europea, 2013a. Reglamento (UE) n ° 1380/2013 del Parlamento Europeo y del Consejo, de 11 de diciembre de 2013 , sobre la política pesquera común, por el que se modifican los Reglamentos (CE) n ° 1954/2003 y (CE) n ° 1224/2009 del Consejo, y se derogan los Reglamentos (CE) n ° 2371/2002 y (CE) n ° 639/2004 del Consejo y la Decisión 2004/585/CE del Consejo, OJ L.
- Unión Europea, 2013b. Reglamento (UE) n ° 1379/2013 del Parlamento Europeo y del Consejo, de 11 de diciembre de 2013 , por el que se establece la organización común de mercados en el sector de los productos de la pesca y de la acuicultura, se modifican los Reglamentos (CE) n ° 1184/2006 y (CE) n ° 1224/2009 del Consejo y se deroga el Reglamento (CE) n ° 104/2000 del Consejo, OJ L.
- Unión Europea, 2008. Diario Oficial L 346/2008.
- United Nations, 1982. Convention on the Law of the Sea.
- United States Space Force, 2021. Global Positioning System [WWW Document]. United States Space Force. URL <http://www.buckley.spaceforce.mil/Activity-Feed/Friends/Article/2197765/global-positioning-system> (accessed 12.6.21).
- Vapnik, V.N., 1995. The Nature of Statistical Learning Theory, The Nature of Statistical Learning Theory. Springer New York. <https://doi.org/10.1007/978-1-4757-2440-0>
- Werbos, P.J., 1974. Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences. Harvard.
- Yáñez, E., Plaza, F., Gutiérrez-Estrada, J.C., Rodríguez, N., Barbieri, M.A., Pulido-Calvo, I., Bórquez, C., 2010. Anchovy (*Engraulis ringens*) and sardine (*Sardinops sagax*) abundance forecast off northern Chile: A multivariate ecosystemic neural network approach. Progress in Oceanography, 3rd GLOBEC OSM: From ecosystem function to ecosystem prediction 87, 242–250. <https://doi.org/10.1016/j.pocean.2010.09.015>
- Yu, X., 2021. Support vector machine-based model for toxicity of organic compounds against fish. Regulatory Toxicology and Pharmacology 123, 104942. <https://doi.org/10.1016/j.yrtph.2021.104942>

7 Anexos

7.1 Códigos fuente

7.1.1 La aplicación móvil de recogida de datos en Android

7.1.1.1 Clase MainActivity reducida

```
public class MainActivity extends AppCompatActivity {

    // Inicia los servicio que gestionan los sensores
    private void iniciarServicios() {
        startService(new Intent(this, Acelerometro.class));
        startService(new Intent(this, CampoMagnetico.class));
        startService(new Intent(this, Giroscopo.class));
        startService(new Intent(this, Gps.class));
    }
    // Detiene los servicios que gestionan los sensores
    private void detenerServicios() {
        stopService(new Intent(this, Acelerometro.class));
        stopService(new Intent(this, CampoMagnetico.class));
        stopService(new Intent(this, Giroscopo.class));
        stopService(new Intent(this, Gps.class));
    }
}
```

7.1.1.2 Clase Acelerometro

```
public class Acelerometro extends Service {

    SensorManager gestorSensores;
    Sensor mAcelerometro;
    // Receptor de los datos del sensor
    MiSensorListener mSensorListener = new MiSensorListener();
    int velocidadSensor= SensorManager.SENSOR_DELAY_UI;
    // Administrador de la base de datos
    AdminSQLiteOpenHelper admin;
    // Crea la base de datos
    SQLiteDatabase BaseDeDatos;

    public Acelerometro() {
    }
}
```

```

public IBinder onBind(Intent intent) {
    throw new UnsupportedOperationException("Not yet implemented");
}

public void onCreate() {
}

public int onStartCommand(Intent intent, int flags, int startId) {
    // Crea el administrador de la base de datos
    admin = new AdminSQLiteOpenHelper(this, "administracion", null, 1);
    // Permite abrir la base de datos en modo escritura y lectura
    BaseDeDatos = admin.getWritableDatabase();
    // Objeto que me permite acceder a los sensores
    gestorSensores= (SensorManager) getSystemService(Context.SENSOR_SERVICE);
    // Objeto acelerómetro
    mAcelerometro = gestorSensores.getDefaultSensor(Sensor.TYPE_ACCELEROMETER);
    activarSensor();
    return START_STICKY;
}

public void onDestroy() {
    detenerSensor();
    // Cierra la base de datos
    BaseDeDatos.close();
}

private class MiSensorListener implements SensorEventListener {
    // Recibe nuevos datos del sensor
    public void onSensorChanged(SensorEvent sensorEvent) {
        // Crea registro con datos recibidos del sensor (sensorEvent)
        ContentValues registro = new ContentValues();
        registro.put("id_event",evento);
        registro.put("time",sensorEvent.timestamp);
        registro.put("x",sensorEvent.values[0]);
        registro.put("y",sensorEvent.values[1]);
        registro.put("z",sensorEvent.values[2]);
        try {
            // Inserta nuevo registro en base de datos
            BaseDeDatos.insert("acelerometro", null, registro);
        } catch (IOException e) { e.printStackTrace(); }
    }
    public void onAccuracyChanged(Sensor sensor, int i) {
    }
}

```

```

private void activarSensor() {
    // Registra mSensorListener como receptor de datos de mAcelerometro
    gestorSensores.registerListener(mSensorListener, mAcelerometro,
                                    velocidadSensor);
}

private void detenerSensor() {
    // Anula registro de mSensorListener
    gestorSensores.unregisterListener(mSensorListener, mAcelerometro);
}
}

```

7.1.1.3 Clase Gir6scopo

```

public class Giroscopo extends Service {

    SensorManager gestorSensores;
    Sensor mGiroscopo;
    // Receptor de los datos del sensor
    MiSensorListener mSensorListener = new MiSensorListener();
    int velocidadSensor= SensorManager.SENSOR_DELAY_UI;
    // Administrador de la base de datos
    AdminSQLiteOpenHelper admin;
    // Crea la base de datos
    SQLiteDatabase BaseDeDatos;

    public Giroscopo() {
    }

    public IBinder onBind(Intent intent) {
        throw new UnsupportedOperationException("Not yet implemented");
    }

    public void onCreate() {
    }

    public int onStartCommand(Intent intent, int flags, int startId) {
        // Crea el administrador de la base de datos
        admin = new AdminSQLiteOpenHelper(this, "administracion", null, 1);
        // Permite abrir la base de datos en modo escritura y lectura
        BaseDeDatos = admin.getWritableDatabase();
    }
}

```

```

    // Objeto que me permite acceder a los sensores
    gestorSensores = (SensorManager) getSystemService(Context.SENSOR_SERVICE);
    // Objeto Giroscopo
    mGiroscopo = gestorSensores.getDefaultSensor(Sensor.TYPE_GYROSCOPE);
    activarSensor();
    return START_STICKY;
}

public void onDestroy() {
    detenerSensor();
    // Cierra la base de datos
    BaseDeDatos.close();
}

private class MiSensorListener implements SensorEventListener {
    // Recibe nuevos datos del sensor
    public void onSensorChanged(SensorEvent sensorEvent) {
        // Crea registro con datos recibidos del sensor (sensorEvent)
        ContentValues registro = new ContentValues();
        registro.put("id_event", evento);
        registro.put("time", sensorEvent.timestamp);
        registro.put("x", sensorEvent.values[0]);
        registro.put("y", sensorEvent.values[1]);
        registro.put("z", sensorEvent.values[2]);
        try {
            // Inserta nuevo registro en base de datos
            BaseDeDatos.insert("giroscopo", null, registro);
        } catch (IOException e) { e.printStackTrace(); }
    }
    public void onAccuracyChanged(Sensor sensor, int i) {
    }
}

private void activarSensor() {
    // Registra mSensorListener como receptor de datos de mGiroscopo
    gestorSensores.registerListener(mSensorListener, mGiroscopo,
                                    velocidadSensor);
}

private void detenerSensor() {
    // Anula registro de mSensorListener
    gestorSensores.unregisterListener(mSensorListener, mGiroscopo);
}
}

```

7.1.1.4 Clase CampoMagnético

```
public class CampoMagnetico extends Service {

    SensorManager gestorSensores;
    Sensor mCampoMagnetico;
    // Receptor de los datos del sensor
    MiSensorListener mSensorListener = new MiSensorListener();
    int velocidadSensor= SensorManager.SENSOR_DELAY_UI;
    // Administrador de la base de datos
    AdminSQLiteOpenHelper admin;
    // Crea la base de datos
    SQLiteDatabase BaseDeDatos;

    public CampoMagnetico() {
    }

    public IBinder onBind(Intent intent) {
        throw new UnsupportedOperationException("Not yet implemented");
    }

    public void onCreate() {
    }

    public int onStartCommand(Intent intent, int flags, int startId) {
        // Crea el administrador de la base de datos
        admin = new AdminSQLiteOpenHelper(this, "BaseDeDatos", null, 1);
        // Permite abrir la base de datos en modo escritura y lectura
        BaseDeDatos = admin.getWritableDatabase();
        // Objeto que me permite acceder a los sensores
        gestorSensores = (SensorManager) getSystemService(Context.SENSOR_SERVICE);
        // Objeto CampoMagnetico
        mCampoMagnetico=
            gestorSensores.getDefaultSensor(Sensor.TYPE_MAGNETIC_FIELD);
        activarSensor();
        return START_STICKY;
    }

    public void onDestroy() {
        detenerSensor();
        // Cierra la base de datos
        BaseDeDatos.close();
    }
}
```

```

private class MiSensorListener implements SensorEventListener {
    // Recibe nuevos datos del sensor
    public void onSensorChanged(SensorEvent sensorEvent) {
        // Crea registro con datos recibidos del sensor (sensorEvent)
        ContentValues registro = new ContentValues();
        registro.put("id_event",evento);
        registro.put("time",sensorEvent.timestamp);
        registro.put("x",sensorEvent.values[0]);
        registro.put("y",sensorEvent.values[1]);
        registro.put("z",sensorEvent.values[2]);
        try {
            // Inserta nuevo registro en base de datos
            BaseDeDatos.insert("campo_magnetico", null, registro);
        } catch (IOException e) { e.printStackTrace(); }
    }
    public void onAccuracyChanged(Sensor sensor, int i) {
    }
}

private void activarSensor() {
    // Registra mSensorListener como receptor de datos de mCampoMagnetico
    gestorSensores.registerListener(mSensorListener, mCampoMagnetico,
                                    velocidadSensor);
}

private void detenerSensor() {
    // Anula registro de mSensorListener
    gestorSensores.unregisterListener(mSensorListener, mCampoMagnetico);
}
}

```

7.1.1.5 Clase Gps

```

public class Gps extends Service {

    LocationManager locManager;
    // Receptor de los datos del sensor
    LocationListener mLocListener = new MiLocationListener();
    // Administrador de la base de datos
    AdminSQLiteOpenHelper admin;
    // Crea la base de datos
    SQLiteDatabase BaseDeDatos;

```

```

public Gps() {
}

public IBinder onBind(Intent intent) {
    throw new UnsupportedOperationException("Not yet implemented");
}

public void onCreate() {
}

public int onStartCommand(Intent intent, int flags, int startId) {
    // Crea el administrador de la base de datos
    admin = new AdminSQLiteOpenHelper(this, "administracion", null, 1);
    // Permite abrir la base de datos en modo escritura y lectura
    BaseDeDatos = admin.getWritableDatabase();
    // Objeto que me permite acceder a los sensores
    locManager = (LocationManager) getSystemService(Context.LOCATION_SERVICE);
    activarSensor();
    return START_STICKY;
}

public void onDestroy() {
    detenerSensor();
    // Cierra la base de datos
    BaseDeDatos.close();
}

private class MiLocationListener implements LocationListener {

    public void onLocationChanged(Location location) {
        // Crea registro con datos recibidos del sensor (sensorEvent)
        ContentValues registro = new ContentValues();
        registro.put("id_event", evento);
        registro.put("time", location.getTime());
        registro.put("latitude", location.getLatitude());
        registro.put("longitude", location.getLongitude());
        registro.put("bearing", location.getBearing());
        registro.put("speed", location.getSpeed());
        try {
            // Inserta nuevo registro en base de datos
            BaseDeDatos.insert("gps", null, registro);
        } catch (IOException e) { e.printStackTrace(); }
    }
}

```



```

    public void onProviderDisabled(String provider) {
    }

    public void onProviderEnabled(String provider) {
    }

    public void onStatusChanged(String provider, int status, Bundle extras) {
    }
}

private void activarSensor() {
    if (ActivityCompat.checkSelfPermission(this,
        Manifest.permission.ACCESS_FINE_LOCATION) ==
        PackageManager.PERMISSION_GRANTED &&
        ActivityCompat.checkSelfPermission(this,
            Manifest.permission.ACCESS_COARSE_LOCATION) ==
            PackageManager.PERMISSION_GRANTED) {
        // Registra mLocListener como receptor de datos de Gps
        locationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER, 0, 0,
            mLocListener);
    }
}

private void detenerSensor() {
    if (ActivityCompat.checkSelfPermission(this,
        Manifest.permission.ACCESS_FINE_LOCATION) ==
        PackageManager.PERMISSION_GRANTED &&
        ActivityCompat.checkSelfPermission(this,
            Manifest.permission.ACCESS_COARSE_LOCATION) ==
            PackageManager.PERMISSION_GRANTED) {
        // Anula registro de mLocListener
        locationManager.removeUpdates(mLocListener);
    }
}
}

```

7.1.1.6 AndroidManifest

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.al3jandro.famisv01">

    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>

        <service
            android:name=".Acelerometro"
            android:enabled="true"
            android:exported="true" />
        <service
            android:name=".Giroscopo"
            android:enabled="true"
            android:exported="true" />
        <service
            android:name=".CampoMagnetico"
            android:enabled="true"
            android:exported="true" />
        <service
            android:name=".Gps"
            android:enabled="true"
            android:exported="true"/>
    </application>
</manifest>
```

7.1.2 Procesado de datos en R

7.1.2.1 Tabla de datos normalizada

```
variables = c("AceX1","AceY1","AceZ1","GirX1","GirY1","GirZ1","MagX1","MagY1",
"MagZ1", "GpsR1", "GpsV1")
tMin = 60*1000000000
# Actividades: 1=Navegar00, 2=Largar, 3=Arrastrar y 4=Virar
for (variable in variables) {
  # Obtener datos brutos de cada variable
  datos = read.csv(file=paste(variable,".csv",sep=""))
  colnames(datos) = c("tUnix","lance","act","dato")
  # Eliminar Valores Atípicos (Outliers)
  q1 = quantile(datos$dato,0.25)
  q3 = quantile(datos$dato,0.75)
  ri = IQR(datos$dato)
  datos = subset(datos,dato>=(q1-3*ri) & dato<=(q3+3*ri))
  # Eliminar ceros
  datos = subset(datos,dato!=0)
  # Calcula media y desviación estándar de cada periodo de 10s
  datos$tUnixMin = tMin*(datos$tUnix%%tMin)/6
  datosCadaMinuto = dplyr::summarize(datos,
    lance = mfv(lance),
    act = mfv(act),
    media = mean(dato),
    sd = sd(dato) )
  colnames(datosCadaMinuto) = c("tUnixMin","lance","act",
    paste(substr(variable,1,4),"m",sep=""),
    paste(substr(variable,1,4),"s",sep=""))
  if (variable == "AceX1") {
    tabla = datosCadaMinuto
  } else {
    tabla = merge(tabla,datosCadaMinuto[,c(1,4,5)],
      by.x='tUnixMin',by.y='tUnixMin',all.x = F)
  }
  # Elimina fila que contenga algún NA
  tabla = na.omit(tabla)
  # Normalizar
  for (col in 4:25) {
    tabla = cbind(tabla,(tabla[,col]-mean(tabla[,col]))/sd(tabla[,col]))
  }
  colnames(tabla)[26:47]=colnames(tabla)[4:25]
  tabla = tabla[,c(1:3,26:47)]
}
```

7.1.2.2 Tabla de datos balanceada

```
# Balanceado lances 138 a 159 (numeración campaña Medits)
tabla.b = Balancear.lances(tabla)

Balancear.lances = function(tabla) {
  for (lan in 138:159) {
    n.lance.ini = min(which(tabla$lance==lan & tabla$act==2))
    n.lance.fin = max(which(tabla$lance==lan & tabla$act==4))
    n.bal = min(table(tabla[tabla$lance==lan,"act"]))
    # Añadir actividad 1 (Navegar)
    antes = n.bal-trunc(n.bal/2)
    despues = trunc(n.bal/2)
    tabla[(n.lance.ini - antes):(n.lance.ini-1),"lance"] = lan
    tabla[(n.lance.fin+1):(n.lance.fin + despues),"lance"] = lan
    # Balancear actividades 2(largar), 3(arrastrar) y 4 (virar)
    for (a1 in 2:4) {
      filas = which(tabla$lance==lan & tabla$act==a1)
      med = trunc((max(filas)+min(filas))/2)
      num = length(filas)
      quitar = num - n.bal
      if (quitar > 0) {
        inicio = trunc(med-(quitar/2))
        tabla[inicio:(inicio+quitar-1),"lance"]=0
      }
    }
  }
  return(tabla[tabla$lance!=0,])
}
```

7.1.2.3 Datos de entrenamiento y test

```
# 3 días 28/5/16>30/5/16, 15 lances 138>152
tabla.entrena = tabla.b[tabla.b$lance<=152,c("act",variables)]
# 2 días 31/5/16>1/6/16 , 7 lances 153>159
tabla.prueba = tabla.b[tabla.b$lance>=153,c("act",variables)]
```

7.1.3 Análisis descriptivo de los datos en R

```
variables = c("AceXm","AceYm","AceZm","AceXs","AceYs","AceZs",
             "GirXm","GirYm","GirZm","GirXs","GirYs","GirZs",
             "MagXm","MagYm","MagZm","MagXs","MagYs","MagZs",
             "GpsRm","GpsVm","GpsRs","GpsVs")
for (variable in variables) {
  datos = tabla[,c("act",variable)]
  colnames(datos) = c("act","dato")
  summary(datos$dato)
  skewness(datos$dato)
  kurtosis(datos$dato)
  shapiro.test(datos$dato)
  levene.test(datos$dato,datos$act,location = "mean")
  levene.test(datos$dato,datos$act,location = "trim.mean")
  levene.test(datos$dato,datos$act,location = "median")
  kruskal.test(dato ~ act , data = datos)
  # Histograma
  ggplot(data=datos,aes(dato))+
    theme_bw()+
    theme(panel.grid.major = element_blank(),
          panel.grid.minor = element_blank(),
          axis.ticks.y = element_blank(),
          plot.title = element_text(hjust = 0.5))+
    scale_y_continuous(label=NULL)+
    labs(title="Histogram",x=variable,y="")+
    geom_histogram(fill="blue",color="blue4",bins=20) # Numero barras
  # Densidad por Actividad
  ggplot(data=datos,aes(dato,fill=factor(act)))+
    theme_bw()+
    theme(panel.grid.major = element_blank(),
          panel.grid.minor = element_blank(),
          axis.ticks.y = element_blank(),
          plot.title = element_text(hjust = 0.5))+
    scale_y_continuous(label=NULL)+
    labs(title="Density by activity",x=variable,y="",fill="Activity")+
    geom_density(size=0.1,alpha=0.5)
  # Grafico dispersión
  datos$aleatorio = sample(1:nrow(datos),nrow(datos),replace=F)
  ggplot(datos,aes(x=dato,y=aleatorio))+
    geom_point(size=1,colour="blue",alpha=0.3)+
    theme_bw()+
    theme(panel.grid.major = element_blank(),
          panel.grid.minor = element_blank(),
```

```

        axis.ticks.y = element_blank(),
        plot.title = element_text(hjust = 0.5))+
    scale_y_continuous(label=NULL)+
    labs(title="Scatter plot",x=variable,y="")
# Gráfico temporal
datos$orden = 1:nrow(datos)
ggplot(datos,aes(x=dato,y=orden))+
    geom_point(size=1,colour="blue",alpha=0.5)+
    theme_bw()+
    theme(panel.grid.major = element_blank(),
          panel.grid.minor = element_blank(),
          axis.ticks.y = element_blank(),
          plot.title = element_text(hjust = 0.5))+
    scale_y_continuous(label=NULL)+
    labs(title="Time plot",x=variable,y="")
# Gráfico de cajas
ggplot(datos,aes(x=orden,y=dato))+
    geom_boxplot(outlier.colour = "red",
                outlier.shape = 19,
                size=0.2,
                fill="blue", alpha=0.5)+
    stat_boxplot(geom = "errorbar",size=0.2)+
    labs(title="BoxPlot",x="",y=variable)+
    theme_bw()+
    theme(panel.grid.major = element_blank(),
          panel.grid.minor = element_blank(),
          plot.title = element_text(hjust = 0.5))+
    coord_flip()
# Grafico de cajas por actividad
ggplot(datos,aes(x=factor(act),y=dato))+
    geom_boxplot(outlier.colour = "red",
                outlier.shape = 19,
                size=0.2,
                fill="blue", alpha=0.5)+
    stat_boxplot(geom = "errorbar",size=0.2)+
    labs(title="BoxPlot Activity",x="",y=variable)+
    theme_bw()+
    theme(panel.grid.major = element_blank(),
          panel.grid.minor = element_blank(),
          plot.title = element_text(hjust = 0.5))+
    coord_flip()
# Gráfico múltiple cajas-densidad
xMax = max(datos$dato)
xMin = min(datos$dato)

```

```

ggplot(datos,aes(x=orden,y=dato))+
  geom_boxplot(outlier.colour = "red",
              outlier.shape = 19,
              size=0.2,
              fill="blue", alpha=0.5)+
  stat_boxplot(geom = "errorbar",size=0.2)+
  labs(title="BoxPlot",x="",y=variable)+
  theme_bw()+
  theme(panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        plot.title = element_text(hjust = 0.5))+
  coord_flip()
}

```

7.1.4 *Análisis multivariante*

```
variables = c("AceXm","AceYm","AceZm","AceXs","AceYs","AceZs",  
             "GirXm","GirYm","GirZm","GirXs","GirYs","GirZs",  
             "MagXm","MagYm","MagZm","MagXs","MagYs","MagZs",  
             "GpsRm","GpsVm","GpsRs","GpsVs")  
# Matriz de Correlacion  
Library(Hmiscs)  
t = tabla[,variables]  
mc1 = rcorr(as.matrix(t))  
mc = round(mc1$r,2)  
for (x in 1:nrow(mc1$r))  
  for (y in x:nrow(mc1$r)) {  
    mc[x,y] = round(mc1$P[x,y],2)  
  }  
# Eliminacion variables  
tr = subset(t,select=-c(AceZs,AceZm,AceXs,AceYs,MagXm,MagZm))  
# Análisis de componentes principales  
library(FactoMineR)  
pca = PCA(tr)
```


7.1.5 Máquinas de aprendizaje

```
Predict = function(maquina.aprendizaje,entrena,prueba) {
  switch (maquina.aprendizaje,
    "LDA" = {
      n.categorias = length(unique(entrena$act))
      # Crea modelo
      modelo = lda(formula=act~.,
                    data=entrena,
                    prior=(rep(1,n.categorias)/n.categorias))
      # Predicción modelo
      pred=predict(object=modelo,prueba[, -1,drop=FALSE])
      # Evalúa predicción
      metrica = Metrica(prueba$act,pred$class)
    },
    "MLP" = {
      datos.entrena = entrena[, -1,drop=FALSE]
      act.entrena.B = decodeClassLabels(entrena$act)
      datos.prueba = prueba[, -1,drop=FALSE]
      metrica = data.frame()
      for (indice in 1:10) {
        # Crea modelo
        modelo = mlp(datos.entrena,
                      act.entrena.B,
                      size=size,
                      maxit=maxit,
                      initFunc=initFunc,
                      initFuncParams=initFuncParams,
                      learnFunc=learnFunc,
                      learnFuncParams=learnFuncParams,
                      updateFunc=updateFunc,
                      updateFuncParams=updateFuncParams,
                      hiddenActFunc=hiddenActFunc,
                      shufflePatterns=shufflePatterns,
                      outputActFunc=outputActFunc,
                      inputsTest=inputsTest,
                      targetsTest=targetsTest,
                      pruneFunc=pruneFunc,
                      pruneFuncParams=pruneFuncParams )
        # Predicción modelo
        pred = predict(object=modelo,datos.prueba,type="class")
        pred.act = apply(pred,1,function(fila) which(fila==max(fila)) )
        metrica.1 = Metrica(prueba$act,pred.act)
        if (indice==1) metrica = metrica.1
      }
    }
  )
}
```

```

    else metrica = metrica + metrica.1
  }
  # Evalúa predicción
  metrica[,col.metrica] = metrica[,col.metrica]/10
},
"PNN" = {
  # Crea modelo
  modelo = learn(entrena)
  modelo = smooth(modelo, sigma=sigma)
  # Predicción modelo
  pred = apply(prueba[, -1, drop=FALSE], 1, function(dato)
                                                    guess(modelo, dato)[[1]])

  # Evalúa predicción
  metrica = Metrica(prueba$act, pred)
},
"SVM" = {
  # Crea modelo
  modelo = svm(formula=act~.,
                data=entrena,
                kernel=kernel,
                cost=cost,
                scale=TRUE)

  # Predicción modelo
  pred = round(predict(modelo, prueba[, -1, drop=FALSE]))
  pred[pred<1] = 1
  n.act = length(unique(prueba$act))
  pred[pred>n.act] = n.act
  # Evalúa predicción
  metrica = Metrica(prueba$act, pred)
}
)
metrica[,col.metrica] = round(metrica[,col.metrica], 2)
return(metrica)
}

```

7.1.6 Calibración del modelo

7.1.6.1 El perceptrón multicapa (MLP)

```
# Número máximo de iteraciones de entrenamiento (maxit)
datos.entrena = tabla.entrena[,-1,drop=FALSE]
act.entrena.B = decodeClassLabels(tabla.entrena$act)
mlp1 = mlp(datos.entrena,
            act.entrena.B ,
            size      = c(22) , # 1 capa oculta con 22 neuronas
            maxit     = 100 )
plotIterativeError(mlp1)
mlp2 = mlp(datos.entrena,
            act.entrena.B ,
            size      = c(22,22) , # 2 capas ocultas con 22 neuronas cada una
            maxit     = 100 )
plotIterativeError(mlp2)
# Evalúa modelo para 1-2 capas con 1-22 neuronas cada capa
sol = data.frame()
for (n1 in 1:22) {
  size = c(n1)
  sol = rbind(sol,data.frame(n1=n1,n2=0,
                             f1=Valida("MLP",tabla.entrena)["sm","f1"]))
  for (n2 in 1:22) {
    size = c(n1,n2)
    sol = rbind(sol,data.frame(n1=n1,n2=n2,
                               f1=Valida("MLP",tabla.entrena)["sm","f1"]))
  }
}
```

7.1.6.2 Máquina de soporte vectorial (SVM)

```
sol = data.table()
# Evalúa modelo para todos los kernel y valores de cost
for (kernel in c("linear","radial","polynomial","sigmoid"))
  for (cost in seq(5,100,1)) {
    cat(kernel,cost,"-")
    metrica = Valida("SVM",tabla.entrena)
    sol = rbind(sol,data.table(kernel = kernel,
                               cost = cost,
                               f1 = metrica[5,"f1"]))
  }
ggplot(sol,aes(x=cost,y=f1,col=kernel)) + geom_line()
```

7.1.7 Métricas de evaluación

```
Metrica = function(real,pred) { # real=salida correcta , pred=predicción
  real = Ventana(as.numeric(as.character(real)),ventana)
  pred = Ventana(as.numeric(as.character(pred)),ventana)
  n.act = length(unique(real))
  # Matriz de confusión
  mc = matrix(0,nrow=n.act,ncol=n.act)
  for (i in 1:length(real)) {
    mc[real[i],pred[i]] = mc[real[i],pred[i]] + 1
  }
  metrica = cbind(as.data.frame(mc),sum=rowSums(mc),exac=0,prec=0,sens=0)
  # Calcula exactitud, precisión, sensibilidad y valor F1
  for (n in 1:n.act) {
    VP = mc[n,n]
    FP = sum(mc[-n,n] )
    VN = sum(mc[-n,-n])
    FN = sum(mc[n,-n] )
    exac = (VP+VN)/(VP+VN+FP+FN)
    prec = VP/(VP+FP)
    sens = VP/(VP+FN)
    metrica[n,"exac"] = exac
    metrica[n,"prec"] = prec
    metrica[n,"sens"] = sens
  }
  metrica[is.na(metrica)] = 0
  colnames(metrica) = c(1:n.act,"sum","exac","prec","sens")
  metrica = rbind(metrica,
    c(colSums(metrica[metrica$sum>0,1:(n.act+1)]),
      colMeans(metrica[metrica$sum>0,c("exac","prec","sens")]))))
  metrica$f1 = 2*metrica$prec*metrica$sens/(metrica$prec+metrica$sens)
  metrica[is.na(metrica)] = 0
  row.names(metrica) = c(1:n.act,"sm")
  metrica[,col.metrica] = metrica[,col.metrica]*100
  return(metrica)
}
```

7.1.8 Validación cruzada

```
# K-fold Cross-Validation
Valida = function(maquina.aprendizaje,tabla) {
  n.filas = nrow(tabla)
  n.filas.k = trunc(n.filas/k)
  metrica = data.frame()
  for (indice in 1:k) {
    k.fin = indice * n.filas.k
    k.ini = k.fin - n.filas.k + 1
    tabla.entrena.k = tabla[-c(k.ini:k.fin),]
    tabla.valida.k = tabla[c(k.ini:k.fin),]
    pre = Predict(maquina.aprendizaje,tabla.entrena.k,tabla.valida.k)
    if (indice==1) metrica = pre
    else metrica = metrica + pre
  }
  metrica[,col.metrica] = metrica[,col.metrica]/k
  metrica$f1 = 2*metrica$prec*metrica$sens/(metrica$prec+metrica$sens)
  metrica[,col.metrica] = round(metrica[,col.metrica],2)
  return(metrica)
}
```

7.1.9 Evaluación del modelo

```
# General con todas las categorías y atributos
Predict.G = function(entrena,prueba) {
  sol = data.frame()
  for (maquina.aprendizaje in c("LDA","MLP","PNN","SVM")) {
    sol = rbind(sol,data.frame(ml=maquina.aprendizaje,
                              act=c(1:4,"sm"),
                              v=Valida(maquina.aprendizaje,tabla.entrena),
                              p=Predict(maquina.aprendizaje,tabla.entrena,tabla.prueba)))
  }
  return(sol)
}

# Tabla 1 & 1
Predict.1_1 = function(entrena,prueba) {
  sol = data.frame()
  for (a1 in 1:3)
    for (a2 in (a1+1):4) {
      entrena1 = entrena[entrena$act==a1 | entrena$act==a2,]
      prueba1 = prueba[prueba$act==a1 | prueba$act==a2,]
      for (maquina.aprendizaje in c("LDA","MLP","PNN","SVM"))
        sol = rbind(sol,data.frame(ml=maquina.aprendizaje,
                                    act=c(a1,a2,"sm"),
                                    v=Valida(maquina.aprendizaje,entrena1),
                                    p=Predict(maquina.aprendizaje,entrena1,prueba1)))
    }
  return(sol)
}

# Tabla 1 & Rest
Predict.1_R = function(entrena,prueba) {
  sol = data.frame()
  for (a1 in 1:4) {
    entrena1 = entrena
    entrena1[entrena1$act!=a1,"act"]=5
    entrena1 = Balancear(entrena1)
    prueba1 = prueba
    prueba1[prueba1$act!=a1,"act"]=5
    prueba1 = Balancear(prueba1)
    for (maquina.aprendizaje in c("LDA","MLP","PNN","SVM"))
      sol = rbind(sol,data.frame(ml=maquina.aprendizaje,
                                  act=c(a1,"Rest","sm"),
                                  v=Valida(maquina.aprendizaje,entrena1),
                                  p=Predict(maquina.aprendizaje,entrena1,prueba1))))
  }
  return(sol) }
```

7.1.10 Simplificación del modelo

7.1.10.1 Reducción de categorías

```
# Reducción a 3 categorías
Red.3C = function(entrena,prueba) {
  sol = data.frame()
  for (act.origen in c(2,4))
    for (act.destino in c(1,3)) {
      entrena1 = entrena
      entrena1[entrena1$act==act.origen,"act"] = act.destino
      entrena1 = Balancear(entrena1)
      prueba1 = prueba
      prueba1[prueba1$act==act.origen,"act"] = act.destino
      prueba1 = Balancear(prueba1)
      act.reducidas = sort(unique(entrena1$act))
      for (maquina.aprendizaje in c("LDA","MLP","PNN","SVM"))
        sol = rbind(sol,d=data.frame(ml=maquina.aprendizaje,
                                     red=paste(act.origen,">",act.destino,sep=""),
                                     act=c(act.reducidas,"sm"),
                                     v=Valida(maquina.aprendizaje,entrena1),
                                     p=Predict(maquina.aprendizaje,entrena1,prueba1)))
    }
  return(sol)
}

# Reducción a 2 categorías
Red.2C = function(entrena,prueba) {
  sol = data.frame()
  for (act.destino.1 in c(1,3))
    for (act.destino.2 in c(1,3)) {
      entrena1 = entrena
      entrena1[entrena1$act==2,"act"] = act.destino.1
      entrena1[entrena1$act==4,"act"] = act.destino.2
      entrena1 = Balancear(entrena1)
      prueba1 = prueba
      prueba1[prueba1$act==2,"act"] = act.destino.1
      prueba1[prueba1$act==4,"act"] = act.destino.2
      prueba1 = Balancear(prueba1)
      for (maquina.aprendizaje in c("LDA","MLP","PNN","SVM"))
        sol = rbind(sol,d=data.frame(ml=maquina.aprendizaje ,
                                     red=paste("2>",act.destino.1,"/",
                                               "4>",act.destino.2,sep=""),
                                     act=c(c(1,3),"sm"),
                                     v=Valida(maquina.aprendizaje,entrena1),
```

```

        p=Predict(maquina.aprendizaje,entrena1,prueba1)))
    }
    return(sol)
}

```

7.1.10.2 Reducción de sensores

```

Red.S = function(entrena,prueba) {
  sol=data.frame()
  var = list( c(variables[1:6]),
             c(variables[7:12]),
             c(variables[13:18]),
             c(variables[19:20]),
             c(variables[21:22]) )
  sensores = c("Ace","Gir","Mag","Rum","Vel")
  for (numero in (1:4)) {
    combinaciones = data.frame(combinations(5,numero))
    // Evalúa modelos para todas las combinaciones de sensores
    apply(combinaciones,1,
         function(fila) {
           col = c("act",unlist(var[filas]))
           entrena1 = entrena[,col]
           prueba1 = prueba[,col]
           for (maquina.aprendizaje in c("LDA","MLP","PNN","SVM"))
             sol <- rbind(sol,d=data.frame(ml=maquina.aprendizaje,
                                             sensores=paste(unlist(sensores[filas]),collapse = "-"),
                                             act=c(c(1:4),"sm"),
                                             v=Valida(maquina.aprendizaje,entrena1),
                                             p=Predict(maquina.aprendizaje,entrena1,prueba1)))
         })
  }
  return(sol)
}

```

7.1.10.3 Resultados finales con reducción de categorías y sensores

```

Final = function(entrena,prueba) {
  sol=data.frame()
  entrena1 = entrena
  entrena1[entrena1$act==2,"act"] = 1
  entrena1[entrena1$act==4,"act"] = 3

```



```

entrena1[entrena1$act==3,"act"] = 2
prueba1 = prueba
prueba1[prueba1$act==2,"act"] = 1
prueba1[prueba1$act==4,"act"] = 3
prueba1[prueba1$act==3,"act"] = 2
// Evalúa los modelos con categorías y sensores con mejores resultados
for (maquina.aprendizaje in c("LDA","MLP","PNN","SVM")) {
  switch (maquina.aprendizaje,
    "LDA" = {
      col = c(1,2:13,20:23)
      sen = "Ace-Gir-Gps"
    },
    "MLP" = {
      col = c(1,8:13,20:23)
      sen = "Gir-Gps"
    },
    "PNN" = {
      col = c(1,20:23)
      sen = "Gps"
    },
    "SVM" = {
      col = c(1,2:23)
      sen = "Ace-Gir-Mag-Gps"
    }
  )
  entrena2 = entrena1[,col]
  prueba2 = prueba1[,col]
  sol = rbind(sol,data.frame(ml= maquina.aprendizaje,
                             sensores=sen,
                             act=c(1,3,"sm"),
                             v=Valida(maquina.aprendizaje,entrena2),
                             p=Predict(maquina.aprendizaje,entrena2,prueba2)))
}
return(sol)
}

```

7.2 Artículos

Artículo 1

Título: Using mobile device's sensors to identify fishing activity

Autores: **M. M. Galotto-Tebar**, A. Pomares-Padilla, I. A. Czerwinski y
J. C. Gutiérrez-Estrada.

Revista: Journal of Marine Science and Technology

Volumen, número, páginas: 25(3), 978-989

Año: 2019

DOI: 10.1007/s00773-019-00694-5

Índice de impacto JCR (2019): 1.446

Recibido el 8 de febrero de 2019; aceptado el 12 de noviembre de 2019; publicado el 20 de noviembre de 2019; fecha de edición septiembre 2020.

Categoría de JCR	Clasificación en la categoría	Cuartil en la categoría
Engineering, Marine	7 de 14	Q2

Artículo 2

Título: Is the vessel fishing? Discrimination of fishing activity with low-cost intelligent mobile devices through traditional and heuristic approaches

Autores: **M. M. Galotto-Tebar**, A. Pomares-Padilla, I. A. Czerwinski y
J. C. Gutiérrez-Estrada.

Revista: Expert Systems With Applications

Índice de impacto JCR (2020): 6.954

Enviado el 12 de noviembre de 2020; primera revisión enviada el 26 de marzo de 2021; segunda revisión enviada el 20 de septiembre de 2021.

Categoría de JCR	Clasificación en la categoría	Cuartil en la categoría
Engineering, Electrical and Electronic	24 de 273	Q1-D1

ARTÍCULOS

El artículo “Using mobile device’s sensors to identify fishing activity. In Journal of Marine Science and Technology” que forma parte del apartado 7.2 Artículos, debido a restricciones relativas a derechos de autor, ha sido retirado de la tesis. En sustitución de éste ofrecemos la siguiente información: referencia bibliográfica, enlace a la revista y resumen.

Galotto-Tebar, M. M., Pomares-Padilla, A., Czerwinski, I. A., & Gutiérrez-Estrada, J. C. (2019). Using mobile device’s sensors to identify fishing activity. In Journal of Marine Science and Technology (Vol. 25, Issue 3, pp. 978–989). Springer Science and Business Media LLC. <https://doi.org/10.1007/s00773-019-00694-5>

Enlace al texto completo: <https://doi.org/10.1007/s00773-019-00694-5>

Resumen:

Fisheries management is generally based on the regulation of the fishing effort, by limiting fishing capacity and activity. The fishing capacity can be quantified objectively, however, the calculation of the fishing activity requires knowing the effective fishing time, for which it is essential to monitor the vessels activity. The European Vessel Monitoring System (VMS) only describes the geographical position, course and speed of the vessel at 2-h intervals, it is an expensive system used only in vessels over 12 m in length and there are no common criteria to infer the fishing activity from the VMS data. To evaluate more precisely the fishing activity, we propose to incorporate new sensors in the vessels that provide additional information. The sensors of mobile devices offer an economic solution that would allow their implementation throughout the fishing fleet. The objective of this work is to evaluate whether the most common sensors integrated in current mobile devices: GPS, accelerometer, gyroscope, and magnetic field, offer relevant information to identify the different phases of bottom trawling fishing activity. The results obtained indicate that these sensors detect, with very high precision, foreseeable changes in the movement of the vessel during the towing manoeuvre.

Using mobile device's sensors to identify fishing activity

M. M. Galotto-Tebar, A. Pomares-Padilla, I. A. Czerwinski & J. C. Gutiérrez-Estrada

Journal of Marine Science and Technology

Official Journal of the Japan Society of Naval Architects and Ocean Engineers (JASNAOE)

ISSN 0948-4280

J Mar Sci Technol
DOI 10.1007/s00773-019-00694-5



Expert Systems With Applications

Is the vessel fishing? Discrimination of fishing activity with low-cost intelligent mobile devices through traditional and heuristic approaches

Manuscript Number:	ESWA-D-20-06080R2
Article Type:	Full length article
Keywords:	Vessel monitoring system; mobile device; Sensor; trawl fishing; ship's behavior; Machine learning
Corresponding Author:	Maria del Mar Galotto-Tébar, PhD student Universidad Miguel Hernandez de Elche Elche, Alicante SPAIN
First Author:	Maria del Mar Galotto-Tébar, PhD student
Order of Authors:	Maria del Mar Galotto-Tébar, PhD student
	Alejandro Pomares Padilla, Ph.D
	Ivone Alejandra Czerwinski, Ph.D
	Juan Carlos Gutiérrez Estrada, Ph.D
Abstract:	Knowing the activity of fishing vessels accurately and in real time means a leap in quality in the management of fishing activity. This paper presents the development of a new comprehensive fisheries monitoring system (FAMIS) that can complement and overcome the limitations of current fishing vessel monitoring systems (VMS). FAMIS is developed on the basis of a low-cost mobile device with GPS sensors, accelerometer, gyroscope and magnetic field and integrates different statistical methods (discriminant functions) and heuristics (artificial neural networks and vectorial support machines) as techniques to classify the information recorded by the sensors of a mobile device during fishing activity. The results obtained with FAMIS indicate that, in general, heuristics have a high degree of discrimination of each of the phases of fishing operation and that, in particular, multilayer perceptrons (MLPs) are capable of correctly identifying 96.3% of towing phases using only GPS and gyro sensors.

Departamento de Ingeniería de Computadores
UNIVERSIDAD MIGUEL HERNÁNDEZ
Campus de Elche, Avenida del ferrocarril s/n, 03202 Alicante, SPAIN

Ms. Maria del Mar Galotto Tébar
Phone and fax: 34 966 65 83 96, E-mail: m.galotto@umh.es

March 26, 2021

Manuscript title: Is the vessel fishing? Discrimination of fishing activity with low-cost intelligent mobile devices through traditional and heuristic approaches.

Authors: Maria del Mar GALOTTO-TÉBAR, Alejandro POMARES-PADILLA, Ivone Alejandra CZERWINSKI, Juan Carlos GUTIERREZ-ESTRADA.

Recommend Editor
Dr. Binshan Lin
Editor-in-Chief, Expert Systems with Applications

Dear Dr. Lin:

Please find enclosed one copy of the revised manuscript "Is the vessel fishing? Discrimination of fishing activity with low-cost intelligent mobile devices through traditional and heuristic approaches " written by Maria del Mar GALOTTO-TÉBAR, Alejandro POMARES-PADILLA, Ivone Alejandra CZERWINSKI y Juan Carlos GUTIERREZ-ESTRADA.

I also attach a copy of the document "Detailed Response to Reviewers" with the response to the editors' and reviewers' comments.

Sincerely,

Ms. Maria del Mar Galotto-Tébar

A handwritten signature in black ink, appearing to be 'M. Galotto', written over the printed name.

Highlights:

- European fishing ships are controlled by a vessel monitoring system (VMS).
- FAMI system exceeds in reliability and accuracy the current VMS
- The mobile' sensors can be used by learning machines to predict fishing events.
- Multilayer perceptrons identified correctly 96.26% of the fishing events.

Table 1. Sensors used: main characteristics, data generation interval of each sensor (sampling period) and variables extracted from each sensor)

Sensor	Range	Sensitivity	Sampling period	Variable
Acceleration	$\pm 19.6133\text{ m/s}^2$	$0,009570\text{ m/s}^2$	50 ms	AceX AceY AceZ
Gyroscope	$\pm 8.7266\text{ rad/s}$	0.0002661 rad/s	50 ms	GirX GirY GirZ
Magnetometer	$\pm 4900\text{ }\mu T$	$0.6\text{ }\mu T$	50 ms	MagX MagY MagZ
GPS	-	-	1 s	GpsR GpsV

Table 2. *Accuracy, precision, recall and F1 score (expressed as percentages) in the final test of the LDA, MLP, PNN and SVM learning machines. This analysis included the samples recorded during all categories of activity with all their attributes.*

Activity	<i>Accuracy</i>	<i>Precision</i>	<i>Recall</i>	<i>F1 Score</i>
<i>LDA</i>				
Steaming	73.63	47.66	55.73	51.38
Setting	77.82	61.07	31.11	41.22
Towing	80.30	62.76	52.14	56.96
Hauling	78.76	55.05	82.05	65.89
<i>MLP</i>				
Steaming	72.15	46.96	70.17	55.93
Setting	80.00	73.00	32.43	43.66
Towing	87.24	71.46	81.97	76.26
Hauling	87.60	78.66	69.42	73.51
<i>PNN</i>				
Steaming	76.88	54.26	47.86	50.86
Setting	71.88	37.10	17.95	24.19
Towing	68.08	40.24	57.09	47.21
Hauling	71.45	44.16	53.68	48.46
<i>SVM</i>				
Steaming	83.80	99.05	35.56	52.33
Setting	86.45	76.69	65.81	70.84
Towing	68.59	44.20	97.61	60.84
Hauling	84.83	84.23	48.38	61.45

Table 3. *F1 score* (expressed as a percentage) in the final test of the LDA, MLP, PNN and SVM learning machines. The samples are included with all their attributes. *1-vs-1* includes samples from two categories. *1-vs-Rest* includes all the samples, maintaining the category on the left and grouping the other categories in the *Rest* category.

Type of partition	LDA		MLP		PNN		SVM	
1-vs-1								
Steaming-vs-Setting	59.37	35.58	69.48	44.82	34.46	68.11	57.65	73.77
Steaming-vs-Towing	74.34	77.12	84.94	85.04	51.52	75.39	63.56	78.87
Steaming-vs-Hauling	79.77	84.01	85.12	86.42	58.25	76.27	66.59	79.97
Setting-vs-Towing	90.66	91.05	92.49	91.84	25.89	70.14	87.24	88.58
Setting-vs-Hauling	95.21	95.22	94.14	94.47	81.59	83.44	93.81	93.17
Towing-vs-Hauling	81.23	80.28	66.37	75.58	75.88	54.79	84.83	79.27
1-vs-Rest								
Steaming-vs-Rest	69.43	63.15	75.85	67.77	48.51	73.05	61.26	71.79
Setting-vs-Rest	46.09	65.88	53.16	72.90	73.12	60.32	62.08	69.30
Towing-vs-Rest	60.54	61.67	88.72	88.54	69.60	22.46	71.03	34.95
Hauling-vs-Rest	80.09	70.55	81.84	79.71	74.73	64.62	77.48	61.32

Table 4. *F1 score* (expressed as a percentage) for the towing category in the final test of the LDA, MLP, PNN and SVM learning machines with all the attributes and after reducing the number of categories. The results without a reduction and with 3 and 2 categories. The symbol “>” written between two categories indicates that the category on the left is merged with that on the right, the label of the samples of the category on the left being changed to that of the one on the right.

Activity	LDA	MLP	PNN	SVM
<i>Steaming-Setting-Towing-Hauling</i>				
Unreduced	56.96	76.26	47.21	60.84
<i>Steaming-Towing-Hauling</i>				
Setting > Steaming	57.14	73.77	55.70	64.81
Setting > Towing	55.16	68.13	58.70	67.06
<i>Steaming-Setting-Towing</i>				
Hauling > Steaming	41.64	71.15	55.06	18.88
Hauling > Towing	77.75	84.48	64.95	71.16
<i>Steaming-Towing</i>				
Setting > Steaming / Hauling > Steaming	55.47	77.51	70.72	71.05
Setting > Steaming / Hauling > Towing	85.38	87.34	76.18	79.42
Setting > Towing / Hauling > Steaming	44.98	73.97	69.41	53.75
Setting > Towing / Hauling > Towing	76.42	77.81	76.02	79.35

Table 5. *Accuracy, precision, recall and F1 score* (expressed as percentages) for the towing category in the final tests of the LDA, MLP, PNN and SVM learning machines, with all the categories and after reducing attributes grouped by sensors. The results are shown before reduction (Ace-Gry-Mag-GPS) and after reduction using the set of sensors that provided the best *F1 score* for the *Towing* phase. The attributes were grouped by sensors as follows: Ace = (AceXm, AceXs, AceYm, AceYs, AceZm, AceZs), Gir = (GirXm, GirXs, GirYm, GirYs, GirZm, GirZs), Mag = (MagXm, MagXs, MagYm, MagYs, MagZm, MagZs) and Gps = (GpsRm, GpsRs, GpsVm, GpsVs).

Sensors	Accuracy	Precision	Recall	F1 Score
<i>LDA</i>				
Ace-Gir-Mag-Gps	80.30	62.76	52.14	56.96
Ace-Gir-Gps	82.35	60.29	86.15	70.94
<i>MLP</i>				
Ace-Gir-Mag-Gps	87.24	71.46	81.97	76.26
Gir-Gps	88.62	74.39	83.37	78.57
<i>PNN</i>				
Ace-Gir-Mag-Gps	68.08	40.24	57.09	47.21
Gps	66.67	42.55	95.21	58.82
<i>SVM</i>				
Ace-Gir-Mag-Gps	68.59	44.20	97.61	60.84
-	-	-	-	-

Table 6. *Accuracy, precision, recall, F1 score and Kappa index for the towing category in the final test of the LDA, MPL, PNN and SVM learning machines. The results are shown before reduction (all activities/all sensors) and after the reduction in categories and attributes grouped by sensors that provide the best F1 scores in the towing phase.*

Activities/Sensors	Accuracy	Precision	Recall	F1 Score	Kappa
<i>LDA</i>					
All activities / All sensors	80.30	62.76	52.14	56.96	0.60
Steaming-Towing / Ace-Gir-Gps	84.02	76.78	97.52	85.92	0.73
<i>MLP</i>					
All activities / All sensors	87.24	71.46	81.97	76.26	0.70
Steaming-Towing / Gir-Gps	89.69	85.11	96.26	90.33	0.83
<i>PNN</i>					
All activities / All sensors	68.08	40.24	57.09	47.21	0.34
Steaming-Towing / Gps	74.19	66.45	97.69	79.10	0.54
<i>SVM</i>					
All activities / All sensors	68.59	44.20	97.61	60.84	0.56
Steaming-Towing / All sensors	70.04	62.66	99.23	76.81	0.52

Figure 1



FIGURE 1

Figure 2

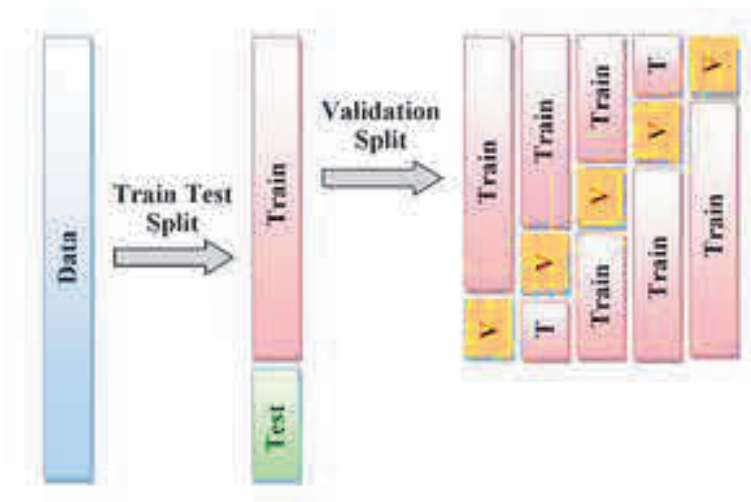


FIGURE 2

Figure 3

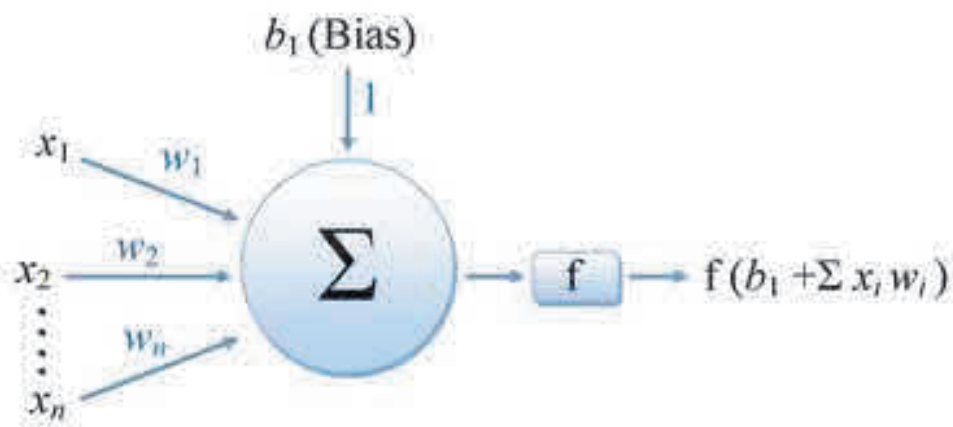


FIGURE 3

Figure 4

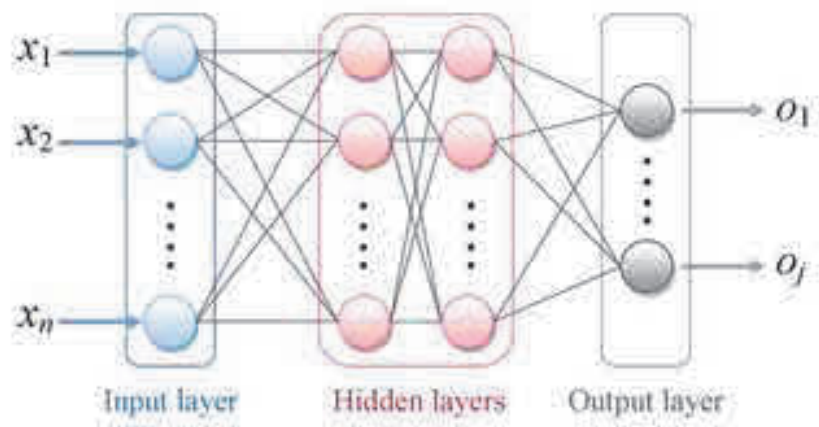


FIGURE 4

Figure 5

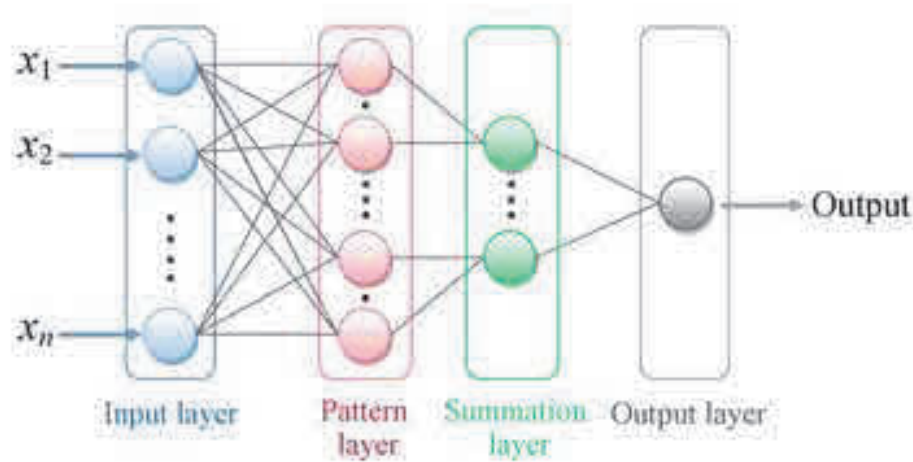


FIGURE 5

Figure 6

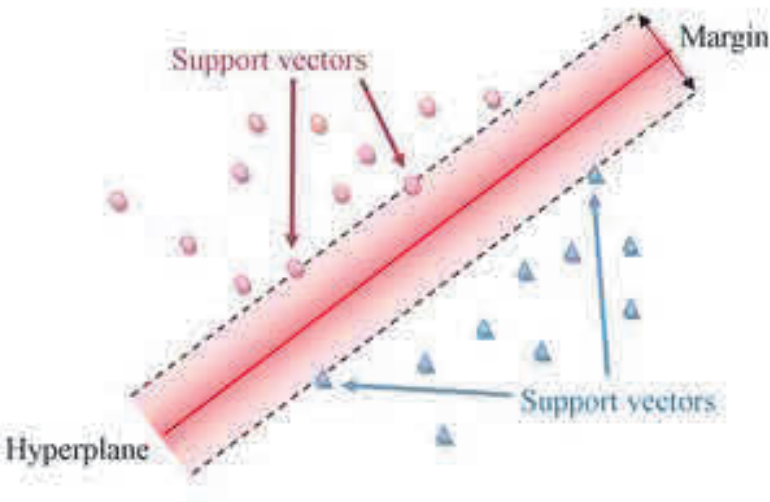


FIGURE 6

Figure Captions

Figure 1. FAMIS user interface installed in a smart phone and information flow.

Figure 2. K-fold cross-validation process. The samples from the table of data (Data) are split into training data (Train - T) to be used in the cross-validation process and test data (Test) to be used in the final model assessment. The 5-fold cross-validation process calculates the mean of the results from repeating the model training and validation process five times. Each process uses different data, ensuring that all the data have been used once in the validation phase.

Figure 3. Artificial neuron. It sums the inputs (x_i) multiplied by the weight associated with each one (w_i), adds the bias (b) and applies the activation function (f).

Figure 4. Architecture of the multi-layer perceptron neural network. Input layer with n neurons that receive input attributes (x_n). Hidden layers composed of a variable number of layers and neurons. Output layer that provides the response of the neural network (o_j). The lines represent the connections that transmit the data between layers.

Figure 5. Architecture of the probabilistic neural network. Input layer with n neurons that receive input attributes (x_n). Pattern layer with one neuron per sample. Summation layer with one neuron per class. Output layer with one neuron providing the output of the neural network.

Figure 6. Graphical representation of the hyperplane used by the support vector machine to discriminate samples of different classes. Support vectors show which samples are closest to the hyperplane and define the edge of the margin.

Figure 7. Graphical representation of the prediction of 4 hauls of the LDA, MLP, PNN and SVM models. The orange area represents the real hauls, the blue dots the *steaming* prediction and the red dots the *towing* prediction.

Figure 7

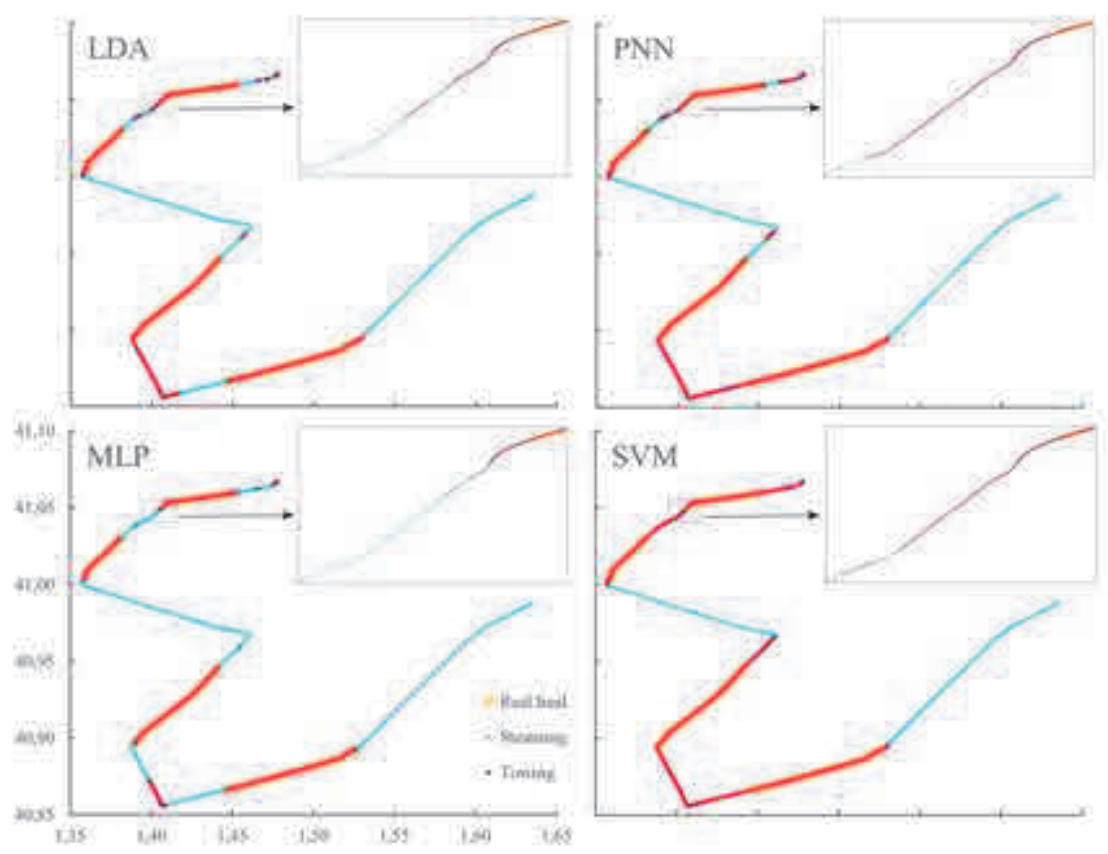


Figure 7

Maria del Mar Galotto Tébar

<https://orcid.org/0000-0001-6045-4563>

No public information available.

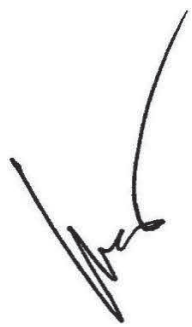
Record last modified Mar 25, 2021 8:58:28 PM

Credit Author Statement

Galotto-Tebar, M.M.: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Resources, Data Curation, Writing - Original Draft, Visualization; **Pomares-Padilla, A.:** Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Resources, Data Curation, Writing - Review & Editing, Visualization; **Czerwinski, I.A.:** Conceptualization, Methodology, Formal analysis, Investigation, Resources, Writing - Review & Editing, Visualization, Supervision; **Gutiérrez-Estrada, J.C.:** Conceptualization, Methodology, Formal analysis, Investigation, Resources, Writing - Review & Editing, Visualization, Supervision

Declaration of interests

- ☒ The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.
- ☐ The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

A handwritten signature in black ink, consisting of a series of loops and a long, sweeping upward stroke.

Is the vessel fishing? Discrimination of fishing activity with low-cost intelligent mobile devices through traditional and heuristic approaches

Galotto-Tebar, M.M.¹; Pomares-Padilla, A.¹; Czerwinski, I.A.²; Gutiérrez-Estrada, J.C.³

¹Dpto. de Ingeniería de Computadores, Universidad Miguel Hernández. Campus de Elche, Avenida del ferrocarril s/n, 03202 Elche, Alicante, Spain (m.galotto@umh.es; pomares@umh.es)

²Instituto Español de Oceanografía, Centro Oceanográfico de Cádiz, Puerto Pesquero, Muelle de Levante, s/n, 11006 Cádiz, Spain (ivone.czerwinski@ieo.es)

³Dpto. de Ciencias Agroforestales, Escuela Técnica Superior de Ingeniería, Campus de El Carmen, Universidad de Huelva, 21007 Huelva, Spain (juanc@uhu.es)

Abstract

Knowing the activity of fishing vessels accurately and in real time means a leap in quality in the management of fishing activity. This paper presents the development of a new fishing activity monitoring integral system (FAMIS) that can complement and overcome the limitations of current fishing vessel monitoring systems (VMS). FAMIS is developed on the basis of a low-cost mobile device with GPS sensors, accelerometer, gyroscope and magnetic field and integrates different statistical methods (discriminant functions) and heuristics (artificial neural networks and vectorial support machines) as techniques to classify the information recorded by the sensors of a mobile device during fishing activity. The results obtained with FAMIS indicate that, in general, heuristics have a high degree of discrimination of each of the phases of fishing operation and that, in particular, multilayer perceptrons (MLPs) are capable of correctly identifying 96.3% of towing phases using only GPS and gyro sensors.

Keywords

Vessel monitoring system; mobile device; sensor; trawl fishing; ship's behavior; machine learning

1. Introduction

For the exploitation of living aquatic resources to be sustainable, there is a need to limit the impact of fishing fleets on marine ecosystems. To assess the impact of a fishing vessel during its activity, we use the concept of fishing effort defined as fishing capacity multiplied by the duration of fishing activity (EC, 2007). Fishing capacity can be quantified from the vessel's technical characteristics and the fishing gear used. On the other hand, assessment of the duration of fishing activity requires knowledge of the time during which the fishing capacity of a vessel is effectively operated, a parameter which is highly variable and specific to each type of fisheries and fishing operation. Therefore, to reliably identify the duration of activity of a vessel, we need to have information, beyond just the time spent at sea, that allow us to track the activity of a fishing vessel.

In 1995, the European Commission undertook a pilot project to assess the functionality and costs of various satellite systems that might make it possible to continuously track the position and activity of fishing vessels. The result of this pilot was the implementation, in 2000, of a Vessel Monitoring System (VMS) with the goal of improving the management and monitoring of fishing activity. Specifically, all fishing vessels ≥ 24 m long were required to use the VMS, this requirement being extended to vessels ≥ 15 m in 2003 and those ≥ 12 m in 2012 (EC, 2009). Data provided by the VMS (the vessel's geographical position, course and speed) are transmitted every hour to the Spanish Fisheries Monitoring Centre where all this information is stored and shared internationally in real time with parties in whose waters European vessels operate.

Based on the information provided by the VMS, several authors have proposed actions and protocols to strengthen the monitoring and planning of fishing activity (Gerritsen and Lordan, 2011; Gerritsen et al., 2012; Jennings and Lee, 2012), a large proportion of published studies having focused on estimating fishing effort (Rijnsdorp et al, 1998; Deng et al., 2005; Murawski et al., 2005; Salthaug and Johannessen, 2006; Walter et al., 2007; Fock, 2008; Hintzen and Brunel, 2008). Nonetheless, despite the large amount of information provided by the VMS and the indisputable advantages of its analysis for monitoring the fishing fleet, the long sampling period, the limited information provided in each sample and the exclusion of the coastal fleet limit the reliability and accuracy of the estimations that can be obtained from this information (Russo et al, 2016).

To overcome these limitations, there is a need for new devices and/or monitoring procedures. For example, the Location and Track System for Andalusian Fishing Vessels (SLSEPA), also known as the “green box” system (Junta de Andalucía, 2004) developed by the regional government of Andalusia (Spain) uses the GPRS mobile network to transmit data on location, course and speed of the artisanal fisheries to a control centre at 3-min intervals. The higher sampling rate improves the accuracy of the estimation of fishing effort and spatial distribution of the fleet and the low data transmission costs make the system feasible and operative for vessels <12 m long of the artisanal fleets (Cojan and Burgos, 2015). Despite SLSEPA representing a significant improvement, however, this type of system still relies on assigning the type of activity based on the recorded speed alone (Lee et al., 2010; Burgos et al., 2013).

We intend to develop a new fisheries monitoring system that will provide better quality information and at a low cost to facilitate their deployment across the entire fishing fleet. Considering that fishing gear in operation has an effect on the dynamic behaviour of a vessel (Sun et al., 2011; Russo et al., 2011) and that the manoeuvres of a vessel may be related to its fishing activity, a device equipped with positioning and movement sensors could potentially provide the information necessary to accurately determine when a vessel is fishing.

Previous research has found that the sensors that are usually used in smart mobile devices such as mobile phones and tablets (GPS, accelerometer, gyroscope and compass) are capable of accurately detecting likely changes in the dynamic behaviour of a vessel during each trawl phase (Galotto-Tebar et al., 2020). On the other hand, the processing capacity of these devices enables us to include machine learning algorithms (artificial intelligence) that are able to learn about the dynamic behaviour of a vessel during the different phases of operation and identify the activity of a fishing vessel in real time, at any time.

Along these lines, some authors have succeeded in using various learning machines to classify a situation or recognise movements of objects or people based on data provided by movement sensors from smart mobile devices (Rodríguez-Martín et al., 2013; Tian et al., 2019; Cust et al., 2019). In this paper we have selected four supervised machine learning classification techniques: linear discriminant analysis (LDA) used to classify linearly separable samples, multilayer perceptron (MLP) and support vector machine

(SVM) used to classify non-linearly separable samples and Bayesian classifier or probabilistic neural network (PNN). The selected classifiers are commonly used in various fields such as livestock (Rodero et al., 2012), fisheries (Bertrand et al., 2008; Czerwinski et al., 2007; Gutiérrez-Estrada et al., 2000, 2007, 2008, 2010; Queirolo et al., 2012; Robotham et al., 2010), hydraulic management (Pulido-Calvo and Portela, 2007), economics (Pérez-Ramírez and Fernández-Castaño, 2007), quality control (Gutierrez and Vázquez, 2013), etc.

Given this, the aims of this study were first to assess the viability of a range of statistical and heuristic methods as tools to classify the data retrieved by sensors on a mobile device during fishing activity, and secondly, to test their ability to identify which trawl phase a fishing vessel is in at a given time.

2. Material and Methods

The models developed in this paper works with the Fishing Activity Monitoring Integral System (FAMIS) (Galotto-Tebar et al., 2020). FAMIS is a mobile application (APP) developed with Android Studio that records the vessel's movement during its fishing activity. During the recording, the data provide by the sensors of devices like smart phones and tablets are temporarily stored in a local database (SQLite) linked the fishing phase, ship's informations and fishing gear. When the mobile device connects to a server, it sends the information to a MySQL database. The server allows access to information through the REST API service which allows to analyse the data using RStudio application (Figure 1).

2.1. Mobile device and sensors

In this study, we used a Samsung SM-P600 tablet with movement sensors that include as standard: 6-axis inertial measurement unit (Bosch Sensortec BMI055) composed of a digital triaxial 12-bit acceleration sensor and a digital triaxial 16-bit gyroscope, a 16-bit 3-axis magnetometer (Asahi Kasei Microdevices AK8963C) and a GPS receptor. Using these sensors and sampling periods of 50 ms and 1 s, data were collected on 11 different variables (Table 1).

With the acceleration sensor, we measured the acceleration ($\overrightarrow{Ace}$) of the mobile device while it is moving, this movement being produced when there are external forces on the device, the value recorded being the force acting on the mass of the sensor. Once the mobile device is attached to the vessel's structure, we can say that the movement of the device reflects that of the vessel. One of the forces always involved in the movement and hence in the acceleration recorded by the sensor is gravity (G) (Equation 1). In this equation, the mass is the sum of the mass of the device and that of the vessel; this term can be ignored and we can consider that the accelerometer will record the components of the gravity vector ($\vec{G}$) at each point.

$$\overrightarrow{Ace} = -\vec{G} - \overrightarrow{\Sigma F} / Mass \approx -\vec{G} \quad (1)$$

With the gyroscope sensor, we measure the speed of rotation around the three main coordinate axes: OX, OY and OZ in radians per second (rad/s). If an observer placed on the positive side of an axis detects that the rotation around the axis is anticlockwise, the speed of rotation will be positive and when the rotation is clockwise, the speed of rotation will be negative.

With the magnetometer, we measure the Earth's magnetic field and the magnetic field around the device, in microTesla (μT). This sensor obtains the vector sum of the Earth's magnetic field and the magnetic fields generated by objects around the device such as engines, cables, etc., and provides the vector components resulting from the sum along OX, OY and OZ axes (Equation 2):

$$\overrightarrow{Mag} = \overrightarrow{Mag_{Earth}} + \overrightarrow{Mag_{Environ}} \quad (2)$$

Using a network of satellites, the global positioning system sensor measures the 3-D geographical position of the device, providing values for latitude, longitude and altitude. The sensor also records the timepoint of the measurement and is able to process these data and generate new information indicating the course, instantaneous and mean velocity, and accuracy. For this study, we have only considered it relevant to record the course (GpsR) and instantaneous speed (GpsV) every second during the activity of the vessel.

2.2. Data collection

Data were collected from the fishing and oceanographic research vessel Miguel Oliver of the Spanish General Secretariat for Fisheries, between 28 May and 1 June 1, 2016 in the waters off Castellón, Tarragona, Barcelona and Girona, during the Mediterranean International Trawl Survey (MEDITS)-Spain bottom trawl survey. This survey is undertaken annually by the Spanish National Institute of Oceanography (IEO), to assess demersal fish stock along the continental shelf and slope in the Spanish Mediterranean Sea (Bertrand et al., 2002).

Seeking to record the position of a vessel as accurately as possible, the mobile device was placed in the navigation bridge at around 18 m above the centre of gravity of the boat at the point where the pitch and roll axes intersect (Ibrahim and Grace, 2010), aligning the device with the OY axis parallel to the longitudinal axis (running from stern to prow), the OX axis parallel to the transverse axis (running from port to starboard) and the OZ axis parallel to the vertical axis. The vector information from the sensors is recorded with respect to these coordinate axes. In relation to this, we previously developed a mobile application (FAMIS) to allow users to activate sensors, select a vessel's phase of hauling at each point in time and store the information recorded in a database (Galotto et al., 2020).

We recorded the vessel's movements during 22 trawl hauls using a GOC 73 sampling gear with Morgère trawl doors at a speed of 3 knots, of which 20 were 30-minute hauls at depths of 50 to 200 m and 2 were 60-minute hauls at depths greater than 200 m. We divided the vessel's fishing activity into four phases: steaming, setting, towing and hauling, and recorded the start and end time of each phase of operation.

2.3. Automated learning techniques

Automated machine learning refers to a set of techniques that allow computer systems themselves to create algorithms that are able to analyse data and acquire the knowledge necessary to carry out tasks such as predicting, classifying, ranking and decision making, without the need for defining initial rules to facilitate these tasks. For this study, we

selected four supervised automated learning classification techniques with very different strategies: (i) linear discriminant analysis (LDA), a statistical technique that classifies samples that are linearly separable and estimates the probability of being in each group; (ii) support vector machines (SVMs) that classify nonlinearly separable samples using a geometric approach; (iii) multilayer perceptrons (MLPs) that are universal proxies of nonlinear functions; and (iv) Bayesian classifiers or probabilistic neural networks (PNNs).

2.4. Data processing

The first step in data processing was filtering, to eliminate zeros and outliers in the data recorded by the sensors. Next, we created 22 independent continuous quantitative variables (attributes) with a mean (X_m) and standard deviation (X_s) of the data in 10-s periods, allowing sufficient time for the boat to complete half a pitch/roll cycle (Barras and Derrett, 2012): For the Accelerometer, the mean and standard deviation were recorded for the three axes, resulting in variables $AceX_m$, $AceX_s$, $AceY_m$, $AceY_s$, $AceZ_m$, and $AceZ_s$. Similar variables were recorded for the gyroscope ($GirX_m$, $GirX_s$, $GirY_m$, $GirY_s$, $GirZ_m$, $GirZ_s$) and the Magnetometer ($MagX_m$, $MagX_s$, $MagY_m$, $MagY_s$, $MagZ_m$, $MagZ_s$). For the GPS the mean and standard deviation for the course and instantaneous speed were recorded: $GpsR_m$, $GpsR_s$, $GpsV_m$ and $GpsV_s$.

We then created an independent nominal qualitative variable (label) called Activity to identify the launch phase of each sample from the official MEDITS-2016 survey data. The labelling of this variable follows a window-type criterion similar to that used by O'Farrell with VMS samples (O'Farrell et al, 2017), assigning the most repeated value during the 10 seconds prior to the sample.

The table of labelled data was created by merging the 22 attributes and the label, removing the records with at least one null, normalising attributes and balancing the samples by activity. As a result, we obtained a total of 6184 samples (1546 samples per activity).

To evaluate the classification performance of each model, we divided the table of data into two blocks: a training data set consisting of 3844 samples (62.16%) from the first 3 days of the survey, covering a total of 15 hauls, and a test data set consisting of 2340

samples (37.83%) from the last 2 days of the survey, during which 7 hauls were completed. This procedure allows models to be trained and tested under different weather conditions.

2.5. Assessment metrics

The classification performance of each model is related to the number of hits and misses. In this study, we used five specific metrics to assess the predictive power of the classifiers:

Accuracy: Percentage of positive and negative predictions that are correct

$$Accuracy = \frac{(TP+TN)}{(TP+TN+FP+FN)} \times 100 \quad (3)$$

Precision: Percentage of positive predictions that are correct

$$Precision = \frac{TP}{(TP+FP)} \times 100 \quad (4)$$

Recall: Percentage of true positives retrieved

$$Recall = \frac{TP}{(TP+FN)} \times 100 \quad (5)$$

F1 Score: Harmonic mean of precision and recall

$$F1\ Score = \frac{2*Precision*Recall}{(Precision+Recall)} \quad (6)$$

where *TP* stands for *True Positive*, the number of fishing sets correctly predicted to be a member of a class, *TN* for *True Negative*, the number of fishing sets correctly predicted to not be a member of a class, *FP* for *False Positive*, the number of fishing set incorrectly predicted to be a member of a class, and *FN* for *False negative*, the number of fishing sets incorrectly predicted to not be a member of a class.

Kappa index: Measure of the degree of agreement between the classifier's prediction and the true classification (Cohen, 1960)

$$Kappa = \frac{p_o - p_e}{1 - p_e} \quad (7)$$

where p_o is the probability of success of the classifier and p_e is the probability of success of a random classifier.

2.6. Model hyperparameter tuning

Automated learning models are parameterised in order that their behaviour can be adjusted to fit a given problem. These models can have multiple parameters and finding the optimal combination of values sometimes requires in-depth analysis that is beyond the scope of this paper. In this study, we selected the values for each parameter using a grid search procedure for hyperparameters, establishing a discrete range of values for each parameter and creating a grid with all the possible combinations of hyperparameters to methodically assess all the resulting models. The best model was selected based on using the F1 score to compare the model's performance.

To avoid choosing hyperparameters of the model that best fit the test data set, the validation process uses different data to train and validate each configuration, saving the test data set for the final assessment of the model. The validation of each configuration of the model was carried out using k-fold cross-validation resampling (Burman, 1989) (Figure 2). To avoid overly small data sets, we used $K=5$. Having selected the parameters that provided the best results in the validation process, the model was trained with all the training data and then the classification performance is assessed with the test data.

2.7. Simplification of the model

The samples recorded during the vessel's fishing activity reflect the influence of trawling on the dynamic behaviour of the vessel. Galotto et al. (2020) demonstrated that the sensors of a mobile device were capable of identifying four distinct periods labelled: steaming, setting, towing and hauling. The task of the models used in this study is to classify the samples into one of these four categories. We should recall that the main

objective of this study is to identify the start and end of trawling to obtain real-time information on fishing, and hence, the efficacy in recognising the samples taken during the towing phase is key to achieving this objective. The remaining phases could potentially be merged, thereby reducing the number of phases, if, by doing so, we improve the performance of the model when classifying the towing category. In relation to this, we validated the models with 4, 3, and 2 categories, always keeping the towing category.

The inclusion of too many attributes of samples may lead to learning machines performing the classification task less well. Given that each attribute is linked to a sensor, and in some cases, to the orientation of the mobile device with respect to the vessel, we grouped the attributes by sensor, and analysed the response of the four learning machines to all of the potential reductions in the group of attributes.

2.8. Machine learning

2.8.1. Linear discriminant analysis (LDA)

Linear discriminant analysis (LDA) is a supervised learning algorithm used for data classification and dimension reduction. Based on a dependent qualitative variable and a set of independent quantitative variables, discriminant analysis allows samples to be classified into one of the groups established by the dependent variable.

This type of analysis provides classification procedures for new observations with an unknown origin into one of the groups analysed, by providing discriminant scores from which we can estimate the probability of being in each group. For this, the algorithm uses new variables known as discriminant variables that are able to characterise and differentiate between groups, described using discriminant functions which are linear combinations of the original variables. More detailed descriptions of the method can be found in Kim et al. (2007), Berstein et al. (2019) and Li et al. (2020).

2.8.2. Artificial neural networks: multilayer perceptrons (MLPs) and probabilistic neural networks (PNNs)

Artificial neural networks are mathematical models inspired by the neural architecture of the human brain (Rumelhart et al, 1986). To create an artificial neural system, we use

artificial neurons functionally organised in layers. The information flows through the layers of the neural network via one-way connections that simulate the synapses between neurons. Each connection has an associated weight (w_i) equivalent to the synapse strength, and each input (x_i) through this connection is multiplied by this weight. The receptor neurone calculates the weighted sum of all the inputs, adds a numerical value known as bias (b) and applies an activation function to the result (f) to determine the output (Figure 3).

In particular, multilayer perceptrons (MLPs) are one-way neural networks formed by an input layer with as many neurons as there are input attributes, one or several hidden layers with a variable number of neurons and output layer with the number of neurons required to show the output (Figure 4). The data flow in one direction from the input layer towards the output layer. The outputs from each neuron of the input layer and hidden layers are connected to the input neurons of the following layer. MPLs are able to analyse complex data sets and perform nonlinear classification into two or more groups, and given this, have been widely used in a range of technical applications (Lek and Guegan, 1999; Gutiérrez-Estrada et al., 2000; Dedeker et al., 2005; Goethals et al., 2007; Pulido-Calvo and Portela, 2007; Gutiérrez-Estrada et al., 2008).

On the other hand, probabilistic neural networks (PNNs) are Bayes-Parzen classifiers composed of four layers: an input layer with the same number of neurons as input attributes; a first hidden layer, which is a pattern layer with the same number of neurons as training or calibration samples; a second hidden layer, which is a summation layer with the same number of neurons as classes; and an output layer, with a neuron that provides the result of the classification (Figure 5) (Specht and Specht, 1990; Hajmeer and Basheer, 2002; Rodero et al., 2012).

When we supply an input to the PNN, the pattern layer assesses the distances from the input vector to the training vectors using a vector of distances, the elements of which indicate how close the input is to each training input. The summation layer (the second layer) sums the patterns of each class generating a vector of probabilities which is then used by a competitive transfer function at the output layer to select the most probable class (Pérez-Ramírez and Fernando-Castaño, 2007).

2.8.3. Support Vector Machine

The support vector machine (SVM) are statistical classifiers proposed by Vapnik (1995) that belong to a family of linear classifiers. SVMs seek to identify hyperplanes that divide the input feature space. At the algorithmic level, SVM learning is modelled as a quadratic optimisation problem with linear constraints, the size of the problem depending on the dimension of the feature space (Figure 6). While markedly less popular than the aforementioned models, they have been used in a range of applications and shown a discriminatory power similar to that of MLPs (Robotham et al., 2010; Rodero et al., 2012).

2.9. Calibration and validation procedures

The construction of learning machine models starts with the processes of validation (k-fold cross-validation) and parameter section allowing adaption of their behaviour to the problem to be solved. In the case of the LDA modelling, given the use of balanced samples, the initial probability of class membership is the same for all the classes. In MLP modelling, the following were selected: a hidden layer with 8 neurons, a maximum of 30 iterations of learning, `Randomize_Weights(0,1)` as the initialization function, `Std_Backpropagation(0.2,0)` as the learning function, `Topological_Order(0)` as the update function, and `Act_Logistic` as the activation function of all hidden units and all output units. As the initialisation function assigns random initial weights, producing small differences between the models, we calculated the mean of 10 training and testing runs of the MLP. In the PNN modelling, a value of 1.4 was set as the smoothing parameter for the pattern-layer activation function. Lastly, in the SVM modelling, we chose a linear kernel and set the constraint violation cost to 1.

For data processing, simulating the learning machines and obtaining the results, we used the R programming language (<https://www.r-project.org>) with specific libraries to work with learning machines such as Modern Applied Statistics with S (Ripley et al., 2020), R Neural Networks using the Stuttgart Neural Network Simulator (<https://github.com/cbergmeir/RSNNS>), probabilistic neural networks PNN (Chasset, 2016) and Misc Functions of the Department of Statistics Probability Theory Group (E1071; Meyer, 2019).

3. Results

Table 2 shows the results concerning the performance in the final test of the four learning machines analysed in this study.

In general, MLPs and SVMs provided better results than LDA or PNNs. The best harmonic mean between precision and recall (F1 scores) was obtained with the SVM model, with a mean among the categories of 68.21% compared to the 65.41% observed in the MLP model. Further, the SVM model was better than the MLP model in precision, with a mean of 76.04% compared to 67.52%. In contrast, the MLP model was better in terms of accuracy and sensitivity, with mean percentages of 81.75% and 63.50% compared to 80.92% and 61.84% in the SVM model. On the other hand, the MLP model provided more balanced results than the SVM one, with regards to classifying the samples recorded during the trawling phase, with an accuracy of 87.24%, a precision of 71.46% and an F1 score value of 76.26%. The SVM model only performed better than the MLP model in terms of sensitivity, being able to identify 97.61% of the trawling activity samples, at the expense of a very high number of false positives. We should highlight the difficulties of the LDA, MLP and PNN models in identifying the setting category, these only being able to recognise 31.11%, 32.43% and 17.95%, respectively, of the samples recorded in this phase.

Table 3 presents the ability of learning machines to distinguish samples from two fishing phases. The analysis from using samples from two trawl phases (1 versus 1) is reported in the top half of the table and from using all the samples, three phases being merged in a new class called “the rest” (1 versus the rest) in the bottom half. This table provides F1 scores as a percentage in the final test of the four models. The results obtained reveal the weaknesses of the LDA, MLP and PNN models in different trawl phases. The LDA and MLP models struggle to differentiate samples from the steaming and setting phases, being able to identify less than 33% of the samples recorded during setting. On the other hand, the PNN model did not differentiate well between steaming and setting, but in this case, the poor results correspond to the steaming phase, this model identifying only 22.56% of the samples obtained during steaming. Similarly, the PNN provided poor results for

differentiating between setting and towing, being able to identify only 14.87% of the samples from the setting phase.

Next, we explored the possibility of improving performance in the classification process by reducing the number of categories, testing all four models, maintaining the towing and steaming activities and removing the setting or hauling categories or both. Table 4 shows that all the models performed better in classifying towing (F1 score) when setting and hauling were eliminated and assigning the samples taken in these phases to steaming and towing, respectively.

We also explored the response of the four models to a reduction in the number of input attributes. For this purpose, we grouped the attributes by sensor and tested all the possible combinations. Table 5 shows the F1 score for towing with each of the four models with all the attributes and the best response to a potential reduction in attributes grouped by sensor. We can observe the substantial improvement in the LDA model after removing the variables based on data from the magnetometer while the PNN model provided better results using only the GPS-based attributes. Similarly, the MLP model improved 3% using only the gyroscope- and GPS-based attributes, while the performance of the SVM model did not improve by reducing attributes.

Table 6 shows the results in the final test of the four models considering the reduction of attributes and categories. The MLP model performs markedly better than the others, with a hit rate of 89.69% in their positive and negative predictions concerning the towing phase, a positive predictive rate of 85.11%, a towing phase sample identification rate of 96.26%, a harmonic mean precision and recall of 90.33%, and a Kappa index with a high degree of agreement of 0.83 ($p < 0.0001$). Figure 6 shows graphically a part of the final test of the models.

4. Discussion

The information provided by the VMS (position, course and speed) to identify fishing grounds and the activity of a fishing fleet can be improved using additional sensors as is suggested in this study. In trawl fishing, precision is key to assessing haul duration and track, the spatial and time distribution of the effort, and subsequently the catch per unit

effort in each fishing area. Deng (2005) stated that the errors in the prediction of trawl fishing increase with the time interval between VMS records and that the loss in precision is relatively high with sampling intervals longer than 30 minutes. On the other hand, Mills et al. (2007) indicated that a vessel's speed alone is not a suitable criterion for identifying trawling activity and confirmed that the sampling frequency of the VMS is too low to allow trawl tracks to be characterised, most trawls being represented by just one record and some not being detected at all, the haul occurring between two records. In addition, for Peruvian anchoveta, Bertrand et al. (2008) reported that speed provided by VMS data leads to the number of fishing events being overestimated by nearly 182%. These findings warrant the development and deployment of new low-cost monitoring systems, complementary to VMS, that are flexible and easily upgradeable and that allow accurate estimation of the activities of fishing vessels.

Since the deployment of the VMS in 2000, many researchers have studied how to process and complement the data provided by this system to improve the management and monitoring of fishing activity. For example, Szostek et al. (2017) included information extracted from interviews of fishermen about their experience fishing king scallops in the English Channel, while Bastardie et al. (2010) combined the data provided by the VMS with information extracted from logbooks through a linking process considering the degree of mismatch. In these ways, these authors were able to obtain disaggregated fishing effort data at a fine geographical scale. Overall, they indicate that the procedures they used significantly improve the delimitation of the catchment area as well as the assessment of fishing effort in time and space, but to achieve this, there was a need for offline processing of external data and its subsequent integration with data generated by the VMS. In relation to this, the data provided by FAMIS (Galotto et al., 2020), the system we propose, is at the same level as that from VMS, and hence, it is fully complementary and therefore easy to integrate.

The use of the FAMIS in fishing vessels would significantly improve on the use of the VMS alone, as it adds movement sensors on the vessel and processing capacity to allow the use of artificial intelligence algorithms to identify the activity of a vessel in situ. Galotto et al. (2020) showed that standard low-cost devices such as mobile phones and tablets do record different data in each trawl phase, and building on that, in this study, we

demonstrate that certain models, such as MLPs or SVMs, yield classifiers that are good tools for automated classification.

The four types of models analysed in this study use very different classification strategies, and in line with that, their classification performance varied. The results obtained with samples recorded in the four trawl phases and attributes associated with the four sensors indicated that SVMs and MLPs provide better results than LDA and PNNs. These results are similar to those obtained by other authors. Robotham et al. (2010) found that MLPs and SVMs performed significantly better than PNNs or LDA, when attempting to distinguish between four pelagic fish species caught off the northern coast of Chile based on echograms of fish shoals. Further, Rodero et al. (2012) compared the same four types of models for distinguishing between four Andalusian cattle breeds based on morphometric variables and found that MLPs and SVMs had better classification performance. This is attributable to the fact that both MLPs and SVMs associate highly nonlinear discriminant functions with the input patterns. This is a result of the large number of variables in each sample, the physical magnitudes that they represent and the complexity of the identification task, in our case, the identification of movement patterns of a vessel exposed to forces from the waves, wind, engine, rudder and fishing gear.

The high nonlinearity of the input data is clearly reflected in the inability of LDAs to find a discriminant function that distinguishes the samples from different classes. In relation to this, the PNN-based model, which takes a radial approach, identifying new samples based on similarities with training samples, provided the poorest results with a mean F1 score of just 44%. In contrast, the MLP as a universal classifier, with a single hidden layer and a small number of neurons, provided an F1 score of 65.41%, being able to identify nearly 82% of samples in the trawl phase. Finally, the most efficient configuration of the SVM model with a linear kernel finds spaces with new dimensions where it is possible to separate different classes with hyperplanes, providing an acceptable mean F1 score of 68.21% and excellent recall of the trawl phase of over 97%.

These results significantly improve on the classification performance found by previous authors based on VMS data. Gerritsen and Lordan (2011) reported that the use of vessel speed as a classification criterion for fishing activity provided a low rate of false negatives but a very high rate of false positives in the classification (32%), only 68% of cases being

correctly classified. These authors highlighted that the potential reasons for these results are related to the fact that a boat may travel at a speed similar to that when trawling when sail at a slow speed while waiting for the right tide or because of poor weather conditions, among other factors.

In relation to this, the four trawl phases identify periods in time when the fishing vessel is involved in different activities. In each phase, the dynamic behaviour of the vessel is influenced by various elements involved such as speed, course, waves, fishing gear, catch, etc. These elements have a different impact at each phase. For example, the steaming phase ends when the boat is manoeuvring and slowing down until it positions itself where it is going to start trawling. After that, the setting phase starts by shooting the net; then the doors are set and the towing cable paid out. During the towing phase, the drag on the vessel grows as the growing volume of the catch is added to the resistance of the fishing gear itself. Next, in the hauling phase, the towing cable is winched in, the doors are raised, and finally, the fishing gear and the catch are brought on board. At the end of the haul, the vessel starts a new steaming phase, manoeuvring and increasing its speed to move to the next fishing spot. Given all this, the effect of the different manoeuvres within each phase on the behaviour of the vessel cannot be detected by the speed data provided by the VMS unless this information is complemented with movement sensor data such as those recorded through FAMIS.

On the one hand, homogenous behaviour of the vessel within a phase and marked changes between phases should facilitate the task of classifying the samples and identifying the start and end of the trawl phases. The start of setting is the most problematic time point. This is because until the doors are in the water, the resistance of the nets is low and the behaviour of the vessel is similar to that in the steaming phase. In contrast, the towing phase starts when the fishing gear, moving at the towing speed, reaches the correct depth, the cable stops being paid out and the speed of the vessel is reduced to the towing speed. This change of speed enables the models to identify the first samples of the towing phase and therefore the start of the real phase of fishing. Similarly, the hauling phase is easy to identify, given that it starts with a reduction in the vessel's speed to make it easier to bring the fishing gear on board and finishes with the fishing gear inactive and an increase in the vessel's speed.

The results of the models analysed are significantly better when we simplify the description of the activity of the fishing vessel to two phases, a common approach for discriminating the activity in a fishing boat when onboard observers log whether the boat is “fishing” or “not fishing” as a function of whether the fishing gear is deployed in the water (Chang and Yuan, 2014).

By merging the phases steaming+setting and towing+hauling, we obtained F1 scores of over 76% in all the models. These proposed mergers avoid the difficulty of identifying the samples at the start of the setting phase and considers that the vessel is fishing from when it starts towing until the fishing gear is taken out of the water. This is partially consistent with the results of Joo et al. (2011) who used MLPs to estimate fishing events in Peruvian anchoveta fisheries, based on VMS parameters and validated in-situ by onboard observers. In that study, observers validated two segments of activity in the speed data series provided by VMS, identified as acceleration between the previous and current pace (which would coincide in part with the merging of steaming+setting) and the acceleration between the current and the following pace (approximately towing+hauling).

In relation to the sensor systems used in this study, we should indicate that all four sensors provided relevant information about changes in the vessel’s behaviour in the four trawl phases (Galotto et al., 2020). The reduction in computational costs associated with removing variables had a relatively small impact on the behaviour of the models. The sensitivity analysis indicated that the GPS sensor is undoubtedly the device that provides the most important information, given that using this device alone, the models assessed achieved a mean F1 score of over 61%. However, the use of information from the other sensors improved the behaviour of some models. For example, the MLP model was able to correctly identify 96.26% of the samples recorded by the GPS+gyroscope in the towing phase. Specifically, for a standard 60-minute haul, the model identified the act of fishing during 57 minutes and 45 seconds, in the worst-case scenario with the false negatives concentrated at the start or end of trawling. The prediction error of 2 minutes and 45 seconds translates to the identification of a shorter-duration hauling operation and an advance or delay in the start or end of the hauling. On the other hand, we should bear in mind that characteristics of the boat where we performed our experiment (Miguel Oliver) mean that it had a higher vessel size to fishing gear ratio than that in fishing boats. This suggests that the dynamics in a fishing boat may be even more affected by the fishing

gear, and hence, the difference between the trawl phases will be easier to identify using this system.

Conclusions

The results of this work indicate that the use of the proposed device (FAMIS) on the basis of a mobile device with GPS, accelerometer, gyroscope and magnetic field sensors significantly improves the accuracy of current VMS systems.

The 4 models analysed have shown different capacities to classify fishing activity. The results obtained with samples recorded in the 4 phases of the set (steaming, setting, towing and hauling) and attributes associated with the 4 sensors (Gps, accelerometer, gyroscope and magnetic field) indicate that the SVM and MLP models offer better results than the LDA and PNN models.

On the other hand, the reduction to 2 phases (steaming and towing) considering that the towing phase starts when the fishing gear reaches the correct depth and trawling speed, and ends when the fishing gear is on the deck of the vessel, facilitates the task of classification and substantially improves the results of the 4 models. Furthermore, the reduction of the sensors involved in each model also improves the quality of their predictions.

In summary, the results of this work indicate that the proposed system is capable of classifying the information recorded by the sensors of a mobile device during fishing activity and identifying the phase of the set in which the vessel is at any given time.

References

- Alfaro-Cortés, E. (2006). Combinación de clasificadores mediante el método boosting. Una aplicación a la predicción del fracaso empresarial en España. PhD Thesis, University of Castilla la Mancha (Spain).
- Barrass, C. B., & Derrett, D.R. (2012). Ship Stability for Masters and Mates. Elsevier Ltd.

643 Bastardie, F., Nielsen, J.R., Ulrich, C., Egekvist, J., & Degel, H. (2010). Detailed mapping
644 of fishing effort and landings by coupling fishing logbooks with satellite-recorded
645 vessel geo-location. *Fisheries Research*, 106(1), 41-53.

646 Bernstein, R., Osadchy, M., Keren, D., & Schuster, A. (2019). LDA classifier monitoring
647 in distributed streaming systems. *Journal of Parallel and Distributed Computing*, 123,
648 156-167.

649 Bertrand, J. A., Gil de Sola, L., Papaconstantinou, C., Relini, G., & Souplet, A. (2002).
650 The general specifications of the MEDITS surveys. *Scientia Marina*, 66(2), 9-17.

651 Bertrand, S., Díaz, E., & Lengaigne, M. (2008). Patterns in the spatial distribution of
652 Peruvian anchovy (*Engraulis ringens*) revealed by spatially explicit fishing data.
653 *Progress in Oceanography*, 79(2-4), 379-389.

654 Burman, P. (1989). A comparative study of ordinary cross-validation, v-fold cross-
655 validation and the repeated learning-testing methods. *Biometrika*, 76(3), 503-514.

656 Chang, S.K., & Yuan, T.L. (2014). Deriving high-resolution spatiotemporal fishing effort
657 of large-scale longline fishery from vessel monitoring system (VMS) data and
658 validated by observer data. *Canadian Journal of Fisheries and Aquatic Sciences*, 71,
659 1363–1370.

660 Chasset, P.O. (2016). Package pnn. <https://cran.r-project.org/web/packages/pnn/pnn.pdf>.

661 Cohen, J. (1960). A Coefficient of Agreement for Nominal Scales. *Educational and*
662 *Psychological Measurement*, 20, 1.

663 Cojan, M., & Burgos, C. (2015). Análisis de la información proporcionada por los
664 sistemas de localización vía satélite de la flota que explota la chirla (*Chamelea gallina*)
665 en el Golfo de Cádiz. In *Teledetección: Humedales y Espacios Protegidos. XVI*
666 *Congreso de la Asociación Española de Teledetección*, 550-553
667 <http://ocs.ebd.csic.es/index.php/AET/2015/schedConf/presentations>.

668 Council of the European Union. (2009). Establishing a Community control system for
669 ensuring compliance with the rules of the common fisheries policy, amending
670 Regulations. *Official Journal of the European Union*, L343, 1-50.

671 Cust, E. E., Sweeting, A. J., Ball, K., & Robertson, S. (2019). Machine and deep learning
672 for sport-specific movement recognition: a systematic review of model development
673 and performance. *Journal of Sports Sciences*, 37(5), 568-600.

674 Czerwinski, I.A., Gutiérrez-Estrada, J.C., & Hernando-Casal, J.A. (2007). Short-term
675 forecasting of halibut CPUE: Linear and non-linear univariate approaches. *Fisheries*
676 *Research*, 86(2-3), 120-128.

677 Dabiri, S., Marković, N., Heaslip, K., & Reddy, C.K. (2020). A deep convolutional neural
678 network based approach for vehicle classification using large-scale GPS trajectory
679 data. *Transportation Research Part C: Emerging Technologies*, 116, 102644-102644.

680 Deng, R., Dichmont, C., Milton, D., Haywood, M., Vance, D., Hall, N., & Die, D. (2005).
681 Can vessel monitoring system data also be used to study trawling intensity and
682 population depletion? The example of Australia's northern prawn fishery. *Canadian*
683 *Journal of Fisheries and Aquatic Sciences*, 62(3), 611-622.

684 European Commission. (2007). On improving fishing capacity and effort indicators under
685 the common fisheries policy. [http://eur-lex.europa.eu/legal-](http://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:52007DC0039&from=EN)
686 [content/EN/TXT/PDF/?uri=CELEX:52007DC0039&from=EN](http://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:52007DC0039&from=EN).

687 Galotto-Tebar, M.M., Pomares-Padilla, A., Czerwinski, I.A., & Gutiérrez-Estrada, J.C.
688 (2020). Using mobile device's sensors to identify fishing activity. *Journal of Marine*
689 *Science and Technology*, 25, 978-989.

690 Gerritsen, H., & Lordan, C. (2011). Integrating vessel monitoring systems (VMS) data
691 with daily catch data from logbooks to explore the spatial distribution of catch and
692 effort at high resolution. *ICES Journal of Marine Science*, 68(1), 245-252.

693 Gerritsen, H.D., Lordan, C., Minto, C., & Kraak, S.B.M. (2012). Spatial patterns in the
694 retained catch composition of Irish demersal otter trawlers: High-resolution fisheries
695 data as a management tool. *Fisheries Research*, 129-130, 127-136.

696 Gloaguen, P., Woillez, M., Mahévas, S., Vermard, Y., & Rivot, E. (2016). Is speed
697 through water a better proxy for fishing activities than speed over ground? *Aquatic*
698 *Living Resources*, 29 (210), 1-8.

699 Global Fishing Watch (2012). <https://globalfishingwatch.org>.

700 Gutierrez, P.T., & Vázquez, J.A., (2013). Aplicación de la red neuronal probabilística
701 para la clasificación de productos conforme a sus especificaciones. *Innovation in*
702 *engineering, technology and education for competitiveness and prosperity*, 1-10.

703 Gutiérrez-Estrada, J.C., Pulido-Calvo, I., & Prenda, J. (2000). Gonadosomatic index
704 estimates of an introduced pumpkinseed (*Lepomis gibbosus*) population in a

705 Mediterranean stream, using computational neural networks. *Aquatic Sciences*, 62(4),
706 350-363.

707 Gutiérrez-Estrada, J.C., Silva, C., Yáñez, E., Rodríguez, N., & Pulido-Calvo, I. (2007).
708 Monthly catch forecasting of anchovy (*Engraulis ringens*) in the north area of Chile:
709 Non-linear univariate approach. *Fisheries Research*, 86(2-3), 188-200.

710 Gutiérrez-Estrada, J. C., Vasconcelos, R., & Costa, M. J. (2008). Estimating fish
711 community diversity from environmental features in the Tagus estuary (Portugal):
712 Multiple Linear Regression and Artificial Neural Network approaches. *Journal of*
713 *Applied Ichthyology*, 24(2), 150-162.

714 Gutiérrez-Estrada, J.C., & Bilton, D. T. (2010). A heuristic approach to predicting water
715 beetle diversity in temporary and fluctuating waters. *Ecological Modelling*, 221(11),
716 1451-1462.

717 Hajmeer, M., & Basheer, I. (2002). A probabilistic neural network approach for modeling
718 and classification of bacterial growth/no-growth data. *Journal of Microbiological*
719 *Methods*, 51(2), 217-226.

720 Heyn, H.M., Blanke, M., & Skjetne, R. (2019). Ice Condition Assessment Using Onboard
721 Accelerometers and Statistical Change Detection. *IEEE Journal of Oceanic*
722 *Engineering*, 1-17.

723 Hintzen, N.T., Bastardie, F., Beare, D., Piet, G.J., Ulrich, C., Deporte, N., Egekvist, J., &
724 Degel, H. (2012). VMStools: Open-source software for the processing, analysis and
725 visualisation of fisheries logbook and VMS data. *Fisheries Research*, 115-116, 31-43.

726 Ibrahim, R. A., & Grace, I.M. (2010). Modeling of Ship Roll Dynamics and Its Coupling
727 with Heave and Pitch. *Mathematical Problems in Engineering*, 2010, 1-32.

728 Ichimura, S., & Zhao, Q. (2019). Route-Based Ship Classification. *IEEE 10th*
729 *International Conference on Awareness Science and Technology (iCAST)*, 1-6.

730 Jennings, S., & Lee, J. (2012). Defining fishing grounds with vessel monitoring system
731 data. *ICES Journal of Marine Science*, 69(1), 51-63.

732 Joo, R., Salcedo, O., Gutierrez, M., Fablet, R., & Bertrand, S. (2015). Defining fishing
733 spatial strategies from VMS data: Insights from the world's largest monospecific
734 fishery. *Fisheries Research*, 164, 223-230.

735 Junta de Andalucía (2004). Localización y seguimiento de embarcaciones pesqueras
736 (SLSEPA).<http://www.juntadeandalucia.es/organismos/agriculturapescaydesarrollorural/areas/pesca-acuicultura/slsepa.html>.
737

738 Kim, H., Drake, B.L., & Park, H. (2007). Multiclass classifiers based on dimension
739 reduction with generalized LDA. *Pattern Recognition*, 40(11), 2939-2945.

740 Lee, J., South, A.B., & Jennings, S. (2010). Developing reliable, repeatable, and
741 accessible methods to provide high-resolution estimates of fishing-effort distributions
742 from vessel monitoring system (VMS) data. *ICES Journal of Marine Science*, 67(6),
743 1260-1271.

744 Li, C.N., Shao, Y.H., Yin, W., & Liu, M.Z. (2020). Robust and Sparse Linear
745 Discriminant Analysis via an Alternating Direction Method of Multipliers. *IEEE*
746 *Transactions on Neural Networks and Learning Systems*, 31(3), 915-926.

747 Marzuki, M.I., Garello, R., Fablet, R., Kerbaol, V., & Gaspar, P. (2015). Fishing gear
748 recognition from VMS data to identify illegal fishing activities in Indonesia. In
749 *MTS/IEEE OCEANS 2015 - Genova*, 1-5.

750 Masters, T (1995). Advanced algorithms for neural networks : a C++ sourcebook. Wiley.

751 Meyer, D. (2019). Package e1071. [https://cran.r-](https://cran.r-project.org/web/packages/e1071/e1071.pdf)
752 [project.org/web/packages/e1071/e1071.pdf](https://cran.r-project.org/web/packages/e1071/e1071.pdf).

753 Mills, C.M., Townsend, S.E., Jennings, S., Eastwood, P.D., & Houghton, C.A. (2007).
754 Estimating high resolution trawl fishing effort from satellite-based vessel monitoring
755 system data. *ICES Journal of Marine Science*, 64(2), 248-255.

756 Neural Networks in R using the Stuttgart Neural Network Simulator.
757 <https://github.com/cbergmeir/RSNNS>.

758 O'Farrell, S., Sanchirico, J. N., Chollett, I., Cockrell, M., Murawski, S. A., Watson, J. T.,
759 Haynie, A., Strelcheck, A., & Perruso, L. (2017). Improving detection of short-
760 duration fishing behaviour in vessel tracks by feature engineering of training data.
761 *ICES Journal of Marine Science*, 74(5), 1428-1436.

762 Palmer, M.C. (2008). Calculation of distance traveled by fishing vessels using GPS
763 positional data: A theoretical evaluation of the sources of error. *Fisheries Research*,
764 89(1), 57-64.

765 Pérez-Ramírez, F.O., & Fernández-Castaño, H. (2007). Las redes neuronales y la
 766 evaluación del riesgo de crédito. *Revista Ingenierías Universidad de Medellín*, 6(10),
 767 77-91.

768 Politis, P.J., DeAlteris, J.T., Brown, R.W., & Morrison, A.T. (2012). Effects of sea-state
 769 on the physical performance of a survey bottom trawl. *Fisheries Research*, 123-124,
 770 26-36.

771 Pulido-Calvo, I., & Portela, M.M. (2007). Application of neural approaches to one-step
 772 daily flow forecasting in Portuguese watersheds. *Journal of Hydrology*, 332(1-2), 1-
 773 15.

774 Queirolo, D., Hurtado, C.F., Gaete, E., Soriguer, M.C., Erzini, K., & Gutiérrez-Estrada,
 775 J.C. (2012). Effects of environmental conditions and fishing operations on the
 776 performance of a bottom trawl. *ICES Journal of Marine Science*, 69(2), 293-302.

777 Ripley, B., Venables, B., Bates, D.M, Hornik, K., Gebhardt, A., & Firth, D. (2020).
 778 Package mass. <http://www.stats.ox.ac.uk/pub/MASS4/>.

779 Robotham, H., Bosch, P., Gutiérrez-Estrada, J. C., Castillo, J., & Pulido-Calvo, I. (2010).
 780 Acoustic identification of small pelagic fish species in Chile using support vector
 781 machines and neural networks. *Fisheries Research*, 102(1-2), 115-122.

782 Rodero, E., González, A., Luque, M., Herrera, M., & Gutiérrez-Estrada, J.C. (2012).
 783 Classification of Spanish autochthonous bovine breeds. Morphometric study using
 784 classical and heuristic techniques. *Livestock Science*, 143 (2-3), 226-232.

785 Rodriguez-Martin, D., Samà, A., Perez-Lopez, C., Català, A., Cabestany, J., &
 786 Rodriguez-Molinero, A. (2013). SVM-based posture identification with a single waist-
 787 located triaxial accelerometer. *Expert Systems with Applications*, 40(18), 7203-7211.

788 Rumelhart, D.E., Hinton, G.E., & Williams, R.J. (1986). Learning representations by
 789 backpropagation errors. *Nature*, 323, 533-536.

790 Russo, T., Parisi, A., Prorgi, M., Boccoli, F., Cignini, I., Tordoni, M., & Cataudella, S.
 791 (2011). When behaviour reveals activity: Assigning fishing effort to métiers based on
 792 VMS data using artificial neural networks. *Fisheries Research*, 111, 53-64.

793 Russo, T., D'Andrea, L., Parisi, A., Martinelli, M., Belardinelli, A., Boccoli, F., Cignini,
 794 I., Tordoni, M. & Cataudella, S. (2016). Assessing the fishing footprint using data

795 integrated from different tracking devices: Issues and opportunities. *Ecological*
796 *Indicators*, 69, 818-827.

797 Shepperson, J., Murray, L.G., Mackinson, S., Bell, E., & Kaiser, M.J. (2016). Use of a
798 choice-based survey approach to characterise fishing behaviour in a scallop fishery.
799 *Environmental Modelling & Software*, 86, 116-130.

800 Simeone, O. (2018). A Very Brief Introduction to Machine Learning With Applications
801 to Communication Systems. *IEEE Transactions on Cognitive Communications and*
802 *Networking*, 4(4), 648-664.

803 Specht D. F. (1990). Probabilistic Neural Networks. *Neural Networks*, 3, 109-118.

804 Sun, X., Yin, Y., Jin, Y., Zhang, X., & Zhang, X. (2011). The modeling of single-boat,
805 mid-water trawl systems for fishing simulation. *Fisheries Research*, 109(1), 7-15.

806 Szostek, C.L., Murray, L.G., Bell, E., & Kaiser, M. J. (2017). Filling the gap: Using
807 fishers' knowledge to map the extent and intensity of fishing activity. *Marine*
808 *Environmental Research*, 129, 329-346.

809 The R Project for Statistical Computing. <https://www.r-project.org>.

810 Tian, Y., Wang, X., Chen, W., Liu, Z., & Li, L. (2019). Adaptive multiple classifiers
811 fusion for inertial sensor based human activity recognition. *Cluster Computing*, 22(4),
812 8141-8154.

813 Vapnik, V.N. (1995). The Nature of Statistical Learning Theory. *Springer New York*.