

ÁRBOLES DE DECISIÓN COMO TÉCNICA DE FUSIÓN DE SENSORES PARA MEDICIÓN DE DISTANCIAS

1 Bernardo Cornelio Pozadas, cornelio.p.bernardo@gmail.com

2 Ángel Vergara Betancourt, angel.vb@zacapoaxtla.tecnm.mx

3 Ignacio Algreto Badillo, ialgreto@conacyt.mx

RESUMEN

La robótica móvil tiene múltiples aplicaciones que benefician y mejoran la calidad de vida de las personas. Gracias a los diferentes sensores con los que cuenta un robot móvil terrestre, es posible adquirir una gran cantidad de datos e información relativa a su entorno. Con esta información y después de un procesamiento adecuado, un robot puede ser capaz de realizar tareas específicas, tales como percepción del entorno, navegación, exploración, planificación de trayectorias, entre otros. Sin embargo, los sensores son susceptibles a errores, por lo cual es necesario el uso de técnicas de fusión de sensores para obtener datos precisos. El propósito de este trabajo consistió en determinar y predecir las distancias entre el robot y un obstáculo, con un coeficiente de determinación superior al 90%. Para llevar a cabo esta tarea se empleó la técnica de aprendizaje automático supervisado de árboles de decisión para regresión, con la finalidad de fusionar diversos sensores. Para ello se utilizaron dos sensores ultrasónicos, un sensor LIDAR y un encoder, gestionado con una Raspberry Pi 3B+ utilizando Python. Se recolectaron datos en una ruta lineal en laboratorio, se analizó la dispersión de éstos. Como resultado se obtuvieron modelos con un coeficiente superior al 90%, además, se consideró el error cuadrático medio para un mejor análisis, si bien, los modelos tienen un coeficiente esperado, las predicciones no son las óptimas en los árboles con baja profundidad, lo anterior es indicado por el error cuadrático medio. Para esta aplicación, árboles de decisión es una buena técnica para fusionar los sensores, pero se debe cuidar el ajuste del árbol, ya que es susceptible al sobre entrenamiento.

ABSTRACT

Mobile robotics has multiple applications that benefit and improve people's quality of life. Thanks to the different sensors that a terrestrial mobile robot has, it is possible to acquire a large amount of data and information related to its environment. With this information and after proper processing, a robot may be able to perform specific tasks, such as environmental perception, navigation, exploration, trajectory planning, among others. However, sensors are susceptible to errors, which is why it is necessary to use sensor fusion techniques to obtain accurate data. The purpose of this work was to determine and predict the distances between the robot and an obstacle, with a coefficient of determination greater than 90%. To carry out this task, the technique of supervised machine learning of decision trees for regression was used, in order to merge various sensors. For this, two ultrasonic sensors, a LIDAR sensor and an encoder were used, managed with a Raspberry Pi 3B + using Python. Data were collected in a linear route in the laboratory, their dispersion was analyzed. As a result, models with a coefficient greater than 90% were obtained, in addition, the mean square error was considered for a better analysis, although the models have an expected coefficient, the predictions are not optimal in trees with low depth, the above is indicated by the mean square error. For this application, decision trees is a good technique to fuse the sensors, but the adjustment of the tree must be taken care of, since it is susceptible to overfitting.

PALABRAS CLAVE

Árboles de decisión
Machine learning
Fusión de sensores
Robot móvil

KEYWORDS

Trees of decision
Machine learning
Sensor fusion
Mobile robot

¹ TecNM - Instituto Tecnológico Superior de Zacapoaxtla, estudiante de Ingeniería Mecatrónica.

² TecNM - Instituto Tecnológico Superior de Zacapoaxtla, docente-investigador de la división de Ingeniería Mecatrónica.

³ Instituto Nacional de Astrofísica, Óptica y Electrónica, investigador CONACYT-INAOE.

I. INTRODUCCIÓN

A lo largo del tiempo, la robótica móvil ha presentado grandes avances y su crecimiento es tal que sus aplicaciones no sólo se incluyen en la industria, sino que también en la medicina, transporte, milicia, agricultura, entre otras; en todos los casos, su objetivo es mejorar la calidad de vida de las personas (Rubio et al., 2019). Una aplicación de lo antes mencionado es en el prototipo de un tractor desarrollado en Australia, cuyo objetivo principal es aumentar la producción con un bajo impacto ambiental, el vehículo es compacto y está diseñado para no atascarse en el terreno, causar el menor daño al suelo y ser robusto, cuenta con múltiples sensores para el tratamiento de las plantas, como es la detección de malezas; usa sensores de costo relativamente bajo para la detección de obstáculos y generación de trayectorias (Ball et al., 2014).

Como se puede observar, el uso de sensores es importante para el desarrollo de la robótica móvil (Faceli et al., 2001), sin embargo, la integración de estos dispositivos requiere el diseño de un sistema de captura, procesamiento e integración de información de distintas fuentes. Estos sistemas, en su mayoría, interactúan con el mundo real y necesitan información precisa y confiable. Con la información recopilada, el robot móvil puede modelar su entorno para realizar tareas de interés, por ejemplo, detección de obstáculos para la generación de rutas de navegación (Lázaro et al., 1999).

Uno de los retos en la robótica móvil es la medición de distancias precisas para describir su entorno, para ello pueden utilizarse sensores de diferente naturaleza, pero considerando que cada sensor tiene una técnica distinta para medir distancias. Dado que cada sensor puede presentar errores e imprecisiones en su medición, una alternativa es la combinación de sensores y la información obtenida de cada uno de ellos puede brindar distancias confiables y precisas. Este proceso se denomina fusión de sensores (Barreto-Cubero et al., 2022; Faceli et al., 2001). La fusión de sensores puede basarse en: enfoque probabilístico, filtro Kalman extendido, filtro de partículas, e Inteligencia Artificial (IA) (Barreto-Cubero et al., 2022). Dentro de la IA están las técnicas Aprendizaje Automático o Machine Learning (ML), que se definen como un conjunto de métodos capaces de detectar patrones en datos de forma automática. El concepto de aprendizaje automático existe al menos desde la década de los 50, aunque era exclusivamente del campo académico, actualmente están experimentando un auge en diversas áreas (Management Solutions, 2018).

Árboles de decisión para clasificación y regresión (CART) son parte de los algoritmos de Aprendizaje Automático Supervisado (SML), que se encargan de hacer predicciones de muestras en función de una serie de datos predictores, los árboles de decisión son modelos sencillos y de fácil interpretación, lo cual posibilita su aplicación en diversos proyectos, sin embargo, es fácil el sobreentrenamiento de modelos basados en este algoritmo si no se realiza un ajuste correcto en su uso (Management Solutions, 2018). Las métricas estadísticas más utilizadas en el análisis de modelos de SML para regresión son el coeficiente de determinación (R^2) y la raíz del error cuadrático medio (RMSE), los cuales nos indican cómo es la dispersión de los valores predichos respecto a los valores de referencia (Chicco et al., 2021; Pedregosa et al., 2011).

El objetivo de este trabajo es medir distancias en un robot móvil con un coeficiente de determinación (R^2) superior al 90% utilizando árboles de decisión para regresión (RT) de los métodos de SML, para lo cual se tiene que instrumentar un prototipo móvil con sensores que midan longitud, analizar las técnicas de aprendizaje automático supervisado para regresión y las métricas estadísticas de análisis, crear un conjunto de datos con los sensores, aplicar un método de SML para regresión y predecir distancias con los modelos de SML resultantes.

II. METODOLOGÍA

Sensores exteroceptivos y propioceptivos.

Los sensores empleados en el desarrollo del prototipo son:

- Sensor ultrasónico HC-SR04: Mide la distancia a la que se encuentra un objeto justo frente a él mediante pulsos de ultrasonido, utilizando el tiempo de vuelo de éstos, emisión y recepción.
- Sensor LIDAR Lite V3HP: Mide la distancia a la que se encuentra un objeto justo frente a él mediante la emisión de una señal láser infrarroja, se calcula el tiempo de retraso entre la transmisión y recepción después de reflejarse en un objeto, utilizando la constante de la velocidad de la luz.
- Encoder de efecto Hall bidireccional DF Robots 28PA51G: Proporciona un número determinado de pulsos, producto de una conversión de la posición lineal o angular de un eje en una señal mediante un transductor, que con las especificaciones del fabricante se obtiene la posición, la dirección y la velocidad del sistema.

Programación de los sensores.

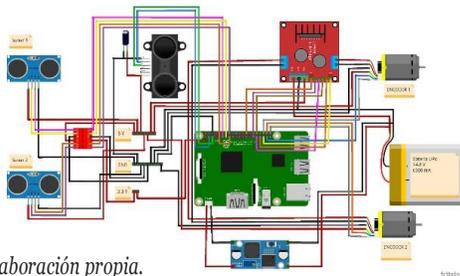
El lenguaje de programación empleado para el funcionamiento de los sensores es Python, para cada sensor se creó una función con la cual se mide la distancia entre el sensor y el objeto inmediato a éste. Se utilizó una Raspberry Pi 3B+

(RPi) para la gestión de los sensores, esto mediante el protocolo de comunicación Secure Shell (SSH), que facilita la comunicación segura entre dos sistemas (computadoras) de forma remota, esto con la finalidad de controlar el prototipo mediante una red WiFi.

Circuito electrónico.

El circuito electrónico, consta de dos sensores ultrasónicos, un sensor LIDAR y dos encoders, como se observa en la Figura 1. Se emplea un regulador de voltaje para la alimentación y protección de la RPi, así como un convertidor de nivel lógico bidireccional. La fuente de energía es una batería LiPo de 14.8 V 6000 mAh.

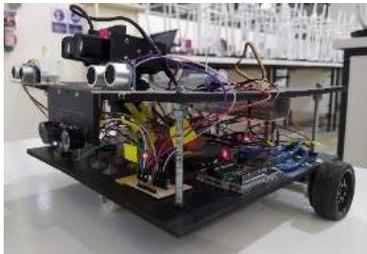
Figura 1
Radiografía Usuario "A"



Fuente: Elaboración propia.

El prototipo utilizado para la recolección de los datos de distancia hacia obstáculos se observa en la Figura 2.

Figura 2
Prototipo



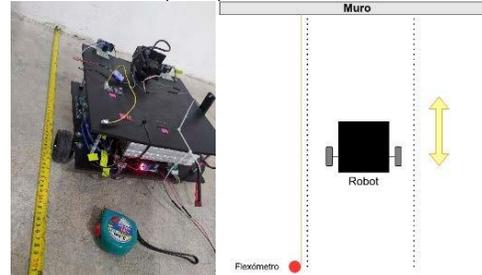
Fuente: Elaboración propia.

Recolección de datos.

Para la recolección de datos se utilizó el Laboratorio de Materiales del Instituto Tecnológico Superior de Zacapoaxtla (ITSZ), en el cual se colocó una cinta métrica flexible en la ruta lineal que iba a seguir el robot, como se observa en la Figura 3, para ir recolectando la distancia real entre el robot y uno de los muros del laboratorio. El robot cumplía una rutina en la cual retrocedía y se detenía para tomar lectura de sus 4 sensores, durante esta rutina el robot se detenía 27 veces, es decir, tomaba 27 medidas de cada sensor por cada rutina,

debido a que el intervalo de interés comprende desde los 0.3 m a 3.5 m. Los datos recolectados se almacenaron en un archivo CSV, ya que este tipo de archivos son los que se utilizan en los algoritmos de aprendizaje automático con Python, que se denomina conjunto de datos o dataset.

Figura 3
Recolección de datos con el prototipo.



Fuente: Elaboración propia.

Implementación de árboles de decisión para regresión.

Para obtener los modelos de predicción basados en árboles de decisión se implementaron las siguientes librerías:

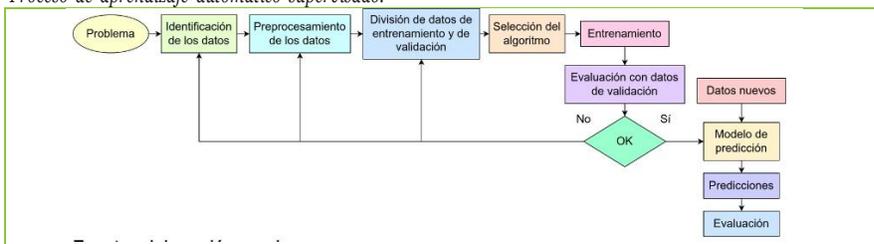
- NumPy para el manejo y adecuación del dataset.
- Pandas para importar y trabajar con el dataset.
- Matplotlib para la generación de gráficas.
- Scikit-Learn para la división del dataset, así como la implementación del modelo predictivo y su ajuste.
- Joblib para guardar y cargar los modelos de predicción.

En la Figura 4 se observa el diagrama general del proceso de aprendizaje automático supervisado (Ayodele, 2010), para implementar árboles de decisión para regresión se realizó lo siguiente:

1. Se importó DecisionTreeRegressor de Scikit-Learn.
2. Se dividió el conjunto de datos con la función `train_test_split()`, de la librería Scikit-Learn, indicando los nombres de los conjuntos de entrenamiento y prueba, así como el porcentaje de elementos a poner en dichos conjuntos, para este trabajo la proporción es de 80% para entrenamiento y 20% para prueba, y el uso de una semilla o `random_state` para tener reproducibilidad (Bonaccorso, 2017).
3. Se ajustaron los hiperparámetros del modelo, en este caso, la profundidad o `max_depth` es el hiperparámetro de interés.
4. Se entrenó el modelo con la función `fit()` y el conjunto acorde.
5. Se guardó el modelo con la función `dump()` de Joblib.

Con los modelos obtenidos, se realizan predicciones con un conjunto de datos nuevo o datos de campo, es decir, datos que no fueron utilizados en el entrenamiento, para evaluar el rendimiento de cada modelo se cuantificó el error de predicción, es decir, se calculan una serie de medidas de “parecido” entre datos predichos y datos reales (Bustamante et al., 2018).

Figura 4
Proceso de aprendizaje automático supervisado.



Fuente: Elaboración propia.

III. RESULTADOS O AVANCES

El conjunto de datos o dataset constó de 513 observaciones o mediciones. Hay cuatro columnas de medidas de distancia en centímetros, la quinta columna corresponde a la medida real entre el prototipo y el muro del laboratorio, ver Tabla 1, cabe destacar que los títulos de columna no se encuentran en el dataset, en este caso se escriben para una mejor comprensión, éstos deben indicarse en el código de programación para la lectura del dataset, recordar que cuando se les asignan títulos a las columnas de un dataset se le denomina dataframe.

Para visualizar el dataset se elaboró un código de programación para graficar los datos, Figura 5, en ésta se apreció la dispersión de los datos y cómo difieren entre ellos, es evidente que en todos los datos existe un error respecto a las distancias reales, que se describen en la Tabla 2, donde el error obtenido en las pruebas es mucho mayor al que el fabricante proporciona, como en el LIDAR, que su error real es 3 veces el valor del fabricante. Sin embargo, estos errores son de utilidad para obtener modelos de predicción con aprendizaje automático.

Para revisar la distribución de los datos se realizaron diagramas de cajas y bigotes de cada sensor, Figura 6, donde se aprecia que las medianas son diferentes para cada sensor: 185 cm en el LIDAR, 174.23 cm en el ultrasónico 1, 164.85 en el ultrasónico 2, 169.79 en el encoder; se observa que la distribución es diferente en cada sensor, algunos datos están más concentrados que otros, ya sea en la caja o bigote.

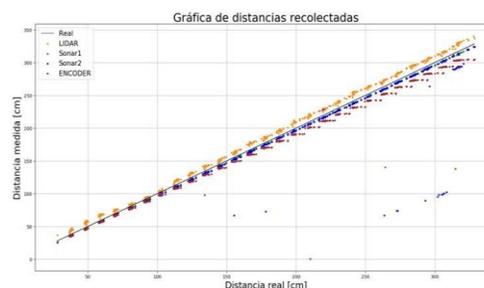
Tabla 1
Algunas medidas del conjunto de datos.

LIDAR	Sonar1	Sonar2	ENCODER	Real
45	35.9714031219482	34.9702835083008	35.4708433151245	37.5
58	47.3687648773193	45.8083152770996	45.7580192592323	48.4
68	56.25319480896	55.6046962738037	56.0143952753637	58
76	67.0385360717773	65.3929710388184	66.2399713635187	69
84	77.3334503173828	75.9959220886231	76.4963473796501	79.5
94	88.4957313537598	87.0285034179688	86.8143232517342	90.9
...

Fuente: Elaboración propia.

Figura 5

Gráfica de dispersión del conjunto de datos.



Fuente: Elaboración propia.

Tabla 2

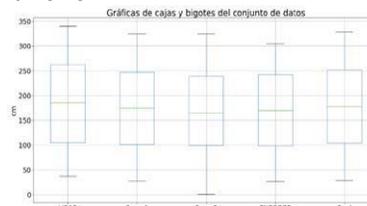
Error en los predictores.

Predictor	R ²	RMSE	Error según el fabricante
LIDAR	0.98975	8.52414	d < 5m: ±2.5 cm d ≥ 5m: ±10 cm
Sonar1	0.98567	10.07869	No hay un dato preciso.
Sonar2	0.84165	33.50382	
ENCODER	0.98772	9.32862	10%, con el robot usado se traduce a 2 cm

Fuente: Elaboración propia.

Los datos recolectados se cargaron a un código de programación para obtener un modelo de aprendizaje automático supervisado, el algoritmo elegido para este caso es el de árboles de decisión para regresión (RT). Se realizaron pruebas con diferentes profundidades en el árbol, esto para observar el ajuste de cada caso, ver si está sub entrenado o sobre entrenado. Para cada profundidad las métricas de interés cambiaron, como se muestra en la Tabla 3, esto debido a que existen mayor número de nodos terminales para predicción. Las unidades de RMSE son cm. y para R² es tanto por ciento.

Figura 6
Diagramas de cajas y bigotes.



Fuente: Elaboración propia.

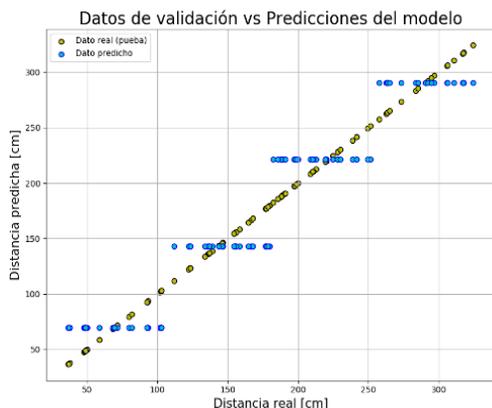
Tabla 3
Métricas para diferentes profundidades de árbol en entrenamiento y prueba.

Métrica	Profundidad				
	2	3	4	5	6
R ² en entrenamiento	0.940107	0.985323	0.996508	0.999646	0.999897
RMSE en entrenamiento	20.913042	10.352491	5.049258	1.606314	0.866125
R ² en prueba	0.923106	0.978192	0.994911	0.999144	0.999518
RMSE en prueba	21.897064	11.661286	5.632873	2.309891	1.733517

Fuente: Elaboración propia.

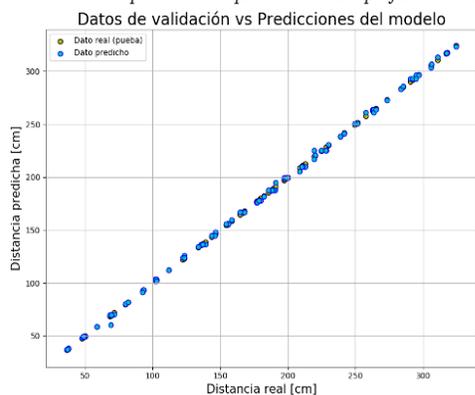
La raíz del error cuadrático medio (RMSE) disminuyó conforme aumentó la profundidad del árbol, esto significa que la dispersión de los valores predichos disminuye respecto a los valores reales, Figuras 7 y 8, cabe destacar que un RMSE mínimo no significa que el modelo funcionará correctamente, ya que esto puede indicar que el modelo está sobreentrenado, y al predecir valores que éste nunca “observó” puede resultar erróneo.

Figura 7
Dispersión de los datos predichos en prueba con una profundidad de 2.



Fuente: Elaboración propia.

Figura 8
Dispersión de los datos predichos en prueba con una profundidad de 6.



Fuente: Elaboración propia.

Se realizaron predicciones con datos que los modelos obtenidos no observaron en el entrenamiento y prueba, para ello se cargaron los modelos en un código de programación con los datos nuevos, donde las métricas obtienen valores diferentes, como se muestra en la Tabla 4, se observó que el rendimiento sigue siendo menor al obtenido en entrenamiento, aunque la diferencia entre los rendimientos de prueba y datos nuevos presentan una variación menor comparada con los rendimientos de entrenamiento, lo anterior respecto al RMSE. La dispersión de las predicciones se observa en las Figuras 9 y 10.

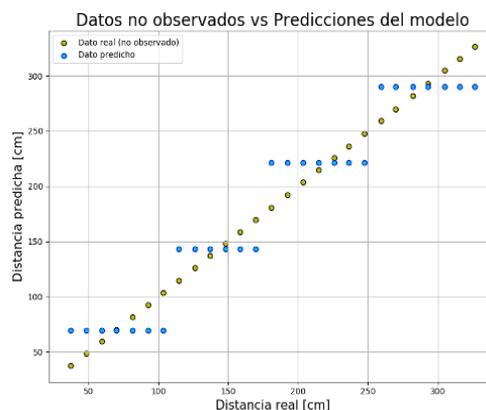
Tabla 4
Métricas para diferentes profundidades de árbol en predicciones con datos no observados.

Métrica	Profundidad				
	2	3	4	5	6
R ²	0.936188	0.983041	0.995677	0.998853	0.999599
RMSE	21.882204	11.280667	5.69547	2.933507	1.734273

Fuente: Elaboración propia.

Comparando los resultados de las Tablas 3 y 4, los modelos que presentaron mayor rendimiento son los que tienen profundidad de 5 y 6, ya que el RMSE tiene un valor cercano al rango de error proporcionado por el fabricante de los sensores. En todos los casos R² indica un rendimiento alto, pero RMSE nos proporciona que tan disperso puede estar una predicción. Como en el árbol con profundidad de 3, que nos proporciona un R² 98.30% pero que la predicción puede presentar una dispersión de hasta 11.28 cm, lo cual no resulta óptimo para la aplicación que se requiere.

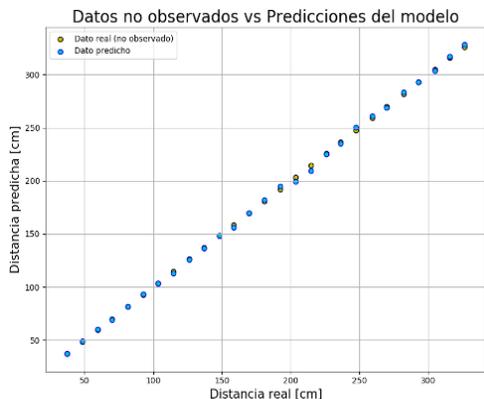
Figura 9
Dispersión de los datos predichos en prueba, con datos no observados, con un RT con profundidad de 2.



Fuente: Elaboración propia.

Figura 10

Dispersión de los datos predichos en prueba, con datos no observados, con un RT con profundidad de 6.



Fuente: Elaboración propia.

IV. CONCLUSIONES

Los resultados del rendimiento, considerando R^2 son óptimos en la mayoría de los árboles, con un rendimiento superior al 90 %, pero no es suficiente evaluar modelos de predicción con esta métrica, ya que la dispersión puede ser significativa, como lo indica RMSE.

Arboles de decisión es viable como modelo de predicción para fusión de sensores, ya que su aplicación brinda resultados confiables para la predicción de distancias con una precisión arriba del 90 %. Por otra parte, con esta técnica se abre un panorama general de cómo implementar otros métodos de aprendizaje automático de forma simple.

Este método puede usar datos numéricos y categóricos, ya que se pueden evaluar los resultados fácilmente con métricas estadísticas y tiene un desempeño que ofrece resultados confiables. Sin embargo, su desventaja es que es demasiado susceptible a sobreajuste, ya que, si la profundidad es alta, el árbol también aprende del ruido, puede ser inestable con variaciones mínimas en los datos y los árboles sesgados son comunes en los aprendices de ML.

Se sugiere probar otros métodos de Machine Learning, para comparar el rendimiento con árboles de decisión, y descartar que la causa de los errores en las predicciones es por el sensor y no por el modelo implementado. También se sugiere considerar el uso de técnicas de Deep Learning.

REFERENCIAS BIBLIOGRÁFICAS

- Ayodele, T. O. (2010). Types of Machine Learning Algorithms. *New advances in Machine Learning*, 3, 19-48.
- Ball, D., Ross, P., English, A., Patten, T., Upcroft, B., Fitch, R., Sukkarieh, S., Wyeth, G. y Corke, P. (2014). Robotics for Sustainable Broad-Acre Agriculture. En *Field and Service Robotics: Results of the 9th International Conference* (pp. 439-452). Springer International Publishing.
- Barreto-Cubero, A. J., Gómez-Espinosa, A., Escobedo Cabello, J. A., Cuan-Urquiza, E., y Cruz-Ramírez, S. (2022). Sensor Data Fusion for a Mobile Robot Using Neural Networks. *Sensors*, 22(1), 305.
- Bonaccorso, G. (2017). *Machine Learning Algorithms*. Birmingham, UK: Packt Publishing.
- Bustamante, Á. L., Padilla R., W., García Herrero, J., Molina López, J. M., Berlanga de Jesús, A., y Patricio Guisado, M. Á. (2018). *Ciencia de datos: técnicas analíticas y aprendizaje estadístico. Un enfoque práctico*. Bogota, Colombia: Alfaomega.
- Chicco, D., Warrens, M. J., y Jurman, G. (2021). The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation. *PeerJ Computer Science*, 7.
- Faceli, K., de Carvalho, A. C., y Rezende, S. O. (2001). Experiments on machine learning techniques for sensor fusion. *Proceedings Fourth International Conference on Computational Intelligence and Multimedia Applications. ICCIMA 2001*, 395-399.
- Lázaro, J. L., Gardel, A., Mataix, C., Rodriguez, F. J., y Martín, E. (1999). Adaptive workspace modeling, using regression methods, and path planning to the alternative guide of mobile robots in environments with obstacles. *1999 7th IEEE International Conference on Emerging Technologies and Factory Automation. Proceedings ETFA '99 (Cat. No.99TH8467)*, 1, 529-534. <https://ieeexplore.ieee.org/document/815400>
- Management Solutions. (2018). *Machine Learning, una pieza clave en la transformación de los modelos de negocio*. Madrid, España.: Management Solutions. <https://www.management-solutions.com/sites/default/files/publicaciones/esp/machine-learning.pdf>
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. y Duchesnay, É. (2011). 3.3. Metrics and scoring: quantifying the quality of predictions. https://scikit-learn.org/stable/modules/model_evaluation.html#regression-metrics
- Rubio, F., Valero, F., y Llopis-Albert, C. (Marzo de 2019). A review of mobile robots: Concepts, methods, theoretical framework, and applications. *International Journal of Advanced Robotic Systems*, 16(2), 1-22. <https://journals.sagepub.com/doi/10.1177/1729881419839596>

