



Modelo predictor de fallas de monitores de signos vitales de una unidad de cuidados intensivos

Juan Pablo Vargas Silva

Artículo de investigación presentado para optar al título de Especialista en Gerencia de Mantenimiento

Asesor

Noe Alejandro Mesa Quintero, Magíster (MSc) en Administración y gestión de empresas

Universidad de Antioquia
Facultad de Ingeniería
Especialización en Gerencia de Mantenimiento
Medellín, Antioquia, Colombia
2024

Cita	Vargas Silva [1]
<p>Referencia</p> <p>Estilo IEEE (2020)</p>	<p>[1] J. P. Vargas Silva, “Modelo predictor de fallas de monitores de signos vitales de una unidad de cuidados intensivos”, Trabajo de grado especialización, Especialización en Gerencia de Mantenimiento, Universidad de Antioquia, Medellín, Antioquia, Colombia, 2024.</p>



Especialización en Gerencia de Mantenimiento, Cohorte XVIII.



Centro de Documentación Ingeniería (CENDOI)

Repositorio Institucional: <http://bibliotecadigital.udea.edu.co>

Universidad de Antioquia - www.udea.edu.co

El contenido de esta obra corresponde al derecho de expresión de los autores y no compromete el pensamiento institucional de la Universidad de Antioquia ni desata su responsabilidad frente a terceros. Los autores asumen la responsabilidad por los derechos de autor y conexos.

RESUMEN

Se realiza el desarrollo de un modelo predictor para determinar la cantidad de fallos que tendrá un monitor de signos vitales (MSV) en el año subsiguiente con el fin de ajustar la periodicidad del mantenimiento preventivo de dichos equipos. Para ello, se creó una base de datos con la que se entrena el modelo predictor, utilizando como técnica de entrenamiento las máquinas de soporte vectorial. Así mismo, se realiza variación del kernel con el fin de encontrar el mejor predictor; para esto se hace uso de un kernel gaussiano, lineal y polinomial. Por último, se implementan dos métodos de reducción de dimensionalidad los cuales son: la selección de características por análisis de componentes de vecindad para regresión (FSRNCA) y el análisis de componentes principales (PCA).

Palabras clave — **Máquinas de soporte vectorial, Predicción de fallos, Monitores de signos vitales.**

ABSTRACT

A predictive model is developed to determine the failures number that a vital signs monitor (VSM) will have in the following year in order to adjust the preventive maintenance periodicity of such equipment. For this purpose, a database was created to train the predictor model, using support vector machines as a training technique. Likewise, kernel variation is performed in order to find the best predictor; for this purpose, a Gaussian, linear and polynomial kernel is used. Finally, two dimensionality reduction methods are implemented: feature selection using neighborhood component analysis for regression (FSRNCA) and principal component analysis (PCA).

Keywords — **Support vector machines, Failure prediction, Vital signs monitors.**

I. INTRODUCCIÓN

Según la Organización Mundial de la Salud (OMS), el 10% de los pacientes experimenta algún tipo de lesión durante la atención hospitalaria [1]. Entre los servicios hospitalarios se encuentran las unidades de cuidado intensivo (UCI), las cuales presentan una tasa de mortalidad que oscila entre el 40% [2] y 50% [3]. De este porcentaje, una fracción menor se atribuye a fallos asociados con la operación adecuada de los equipos responsables del monitoreo, terapia y tratamiento del paciente [4]. La probabilidad de fallo de un equipo biomédico puede mitigarse si se llevan a cabo los mantenimientos preventivos en los intervalos de tiempo sugeridos por el fabricante o determinados por el personal de ingeniería [5]. De igual forma, a medida que el tiempo de uso de los equipos se incrementa, también lo hace la probabilidad de que estos fallen, lo que resulta en una disminución del intervalo entre los mantenimientos preventivos [6].

En investigaciones anteriores se delineó una aproximación inicial para la predicción de fallas, empleando técnicas como la regresión logística multinomial, la regresión lineal y la regresión por pasos. Adicionalmente, se documentó la implementación de dos estrategias de reducción de dimensionalidad: PCA y FSRNCA. La aplicación de estos métodos resultó con una precisión en la predicción del 64,752% [7].

Tras la estandarización de los fallos, se procede a la generación de dos bases de datos utilizando el software de hoja de cálculo Excel. Cada base de datos encapsula la información de los mantenimientos realizados a los monitores durante un periodo de diez años. Ambas bases de datos constan de 20 columnas, las cuales corresponden a los fallos estandarizados de los equipos, es decir, las variables del sistema predictor, ver Fig. 2. La primera base de datos comprende 1567 filas, donde cada fila representa un fallo o parada que el equipo ha experimentado durante los diez años de operación. En contraste, la segunda base de datos consta de 460 filas, donde cada fila representa el resumen anual de fallos o paradas que cada monitor ha experimentado.

Variable 1	Variable 2	Variable 3	Variable 4
Daño sensor SpO2	Daño extensión SpO2	Daño latiguillos ECG	Daño extensión ECG
Variable 5	Variable 6	Variable 7	Variable 8
Daño sensor de temperatura	Daño línea de capnografía	Daño brazalete NIBP	Daño manguera de NIBP
Variable 9	Variable 10	Variable 11	Variable 12
Daño batería	Daño cable AC	Daño módulo NIBP	Daño tarjeta SpO2
Variable 13	Variable 14	Variable 15	Variable 16
Daño tarjeta adquisición	Daño tarjeta UDOM	Daño CPU	Daño de pantalla
Variable 17	Variable 18	Variable 19	Variable 20
Daño de la bomba	Daño acoples NIBP	Daño racores	Otros (ajuste - mal uso)

Fig. 2. Variables de entrenamiento.

Las bases de datos previamente descritas contienen variables de tipo categoricial. Sin embargo, las funciones de entrenamiento de las máquinas de soporte vectorial requieren exclusivamente datos numéricos. Por lo tanto, es necesario transformar estas variables de tipo categoricial a un formato numérico. Para este propósito, se emplea el método de codificación one-hot encoding, ya que ha demostrado ofrecer un rendimiento superior en el entrenamiento del modelo predictor [9].

B. Entrenamiento

1) Segmentación training, validation y test

Para el entrenamiento del sistema predictor se segmenta cada una de las bases de datos en tres subconjuntos. El primer subconjunto corresponde a los fallos experimentados por los 46 monitores durante el año 2023, los cuales se utilizarán para la evaluación final del modelo predictor, simulando así su aplicación en un entorno real. Este subconjunto de datos consta

originalmente de 162 filas y 20 columnas. Sin embargo, su objetivo principal es ser comparado con la salida del modelo predictor para determinar la precisión real del sistema [10]. Por esta razón, los datos se agrupan por equipo y se suman las 20 variables, obteniendo un resumen de los fallos de los 46 monitores en el año 2023. Para fines de este documento, a este subconjunto de datos se le denominará como datos ‘test’.

Una vez extraídos los datos test, la base de datos se divide en dos matrices. La primera matriz, denominada ‘training’, comprende el 70% de los datos, es decir, los datos de fallos de 32 MSV. La segunda matriz, denominada ‘validation’, contiene los datos de fallos de los 14 MSV restantes, lo que representa el 30% de los datos [11]. Esta segmentación se realiza de manera aleatoria utilizando la función ‘randperm’ del software de cálculo numérico Matlab [12]. Cada vez que se ejecuta el código se realiza esta segmentación de la base de datos en ‘training’ y ‘validation’ con el objetivo de generalizar el modelo predictor. Esta técnica es conocida como validación cruzada o ‘cross-validation’ [13].

2) *Generación modelo predictor*

El modelo predictor se obtiene tras el entrenamiento del sistema utilizando máquinas de soporte vectorial realizando variaciones en la función del kernel. Para esto, se implementan kernels Gaussianos, lineales y polinomiales. El procedimiento mencionado se lleva a cabo utilizando la función ‘fitrsvm’ del software de cálculo numérico Matlab [14]. Esta función tiene como parámetros de entrada la matriz de datos training. Además, para garantizar la generalización del modelo, se realizan 500.000 ejecuciones del código de entrenamiento.

a) *Sin reducción de dimensionalidad*

El entrenamiento del modelo predictor sin la implementación de técnicas de reducción de dimensionalidad puede presentar desventajas, principalmente debido al alto costo computacional que implica. Además, la presencia de variables potencialmente irrelevantes para el sistema o datos susceptibles de provocar sobreajuste pueden disminuir la precisión del modelo predictor [15, 16].

b) *FSRNCA*

La reducción de dimensionalidad de los datos se lleva a cabo mediante la selección de características para regresión utilizando el análisis de componentes de vecindad. Este proceso se implementa con la función ‘fsrnca’ del software de cálculo numérico Matlab

[17]. De esta función, se obtiene una variable de tipo objeto, de la cual se extrae la matriz denominada 'FeatureWeights'. Esta matriz se representa gráficamente para facilitar la visualización de las variables que son relevantes para el entrenamiento del sistema, ver Fig. 3. Las variables cuyo peso se aproxima significativamente a cero no se consideran relevantes para el entrenamiento del modelo. Por lo tanto, se establece un criterio de selección basado en un peso de 0.09. Cualquier variable que no alcance este umbral es excluida del conjunto de datos.

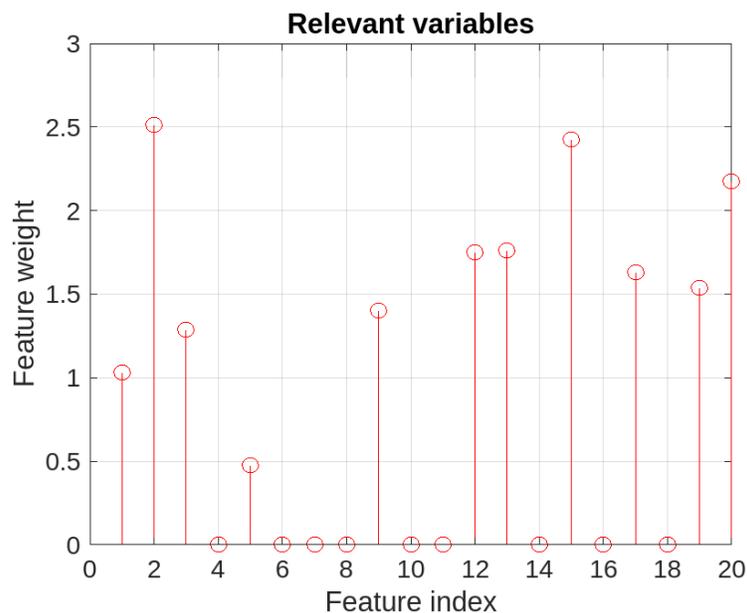


Fig. 3. Variables relevantes FSRNCA.

c) PCA

La técnica de reducción de dimensionalidad mediante el análisis de componentes principales se implementa utilizando la función 'pca' del software de cálculo numérico Matlab [18]. Esta función devuelve dos variables: la varianza acumulada y los coeficientes de los componentes principales. La varianza acumulada se visualiza a través de un gráfico de Pareto, ver Fig. 4. Como criterio de selección se considera el 90% de la varianza acumulada. Por lo tanto, se identifican las variables que comprenden este porcentaje de los datos y se descartan las variables restantes, es decir, aquellas que representan aproximadamente el 10% de la información residual.

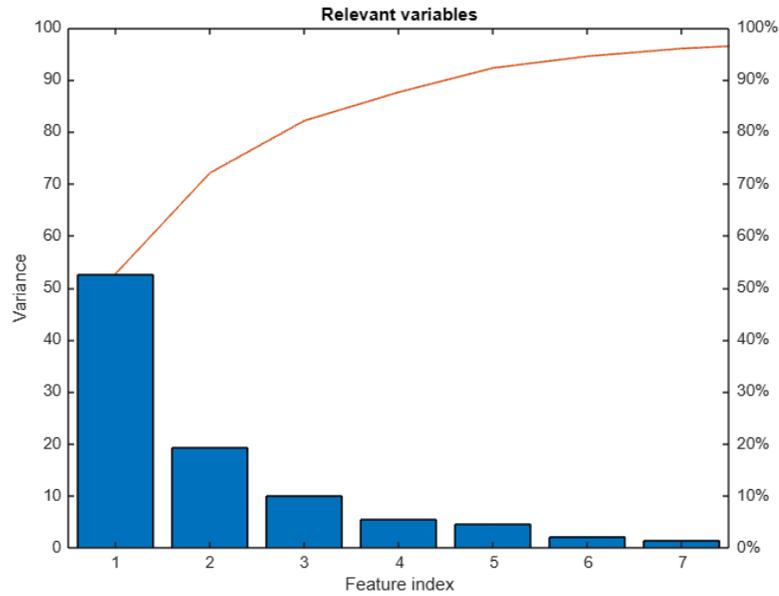


Fig. 4. Variables relevantes PCA.

3) Validación modelo predictor

La validación del modelo predictor se lleva a cabo mediante la implementación de la función 'predict' incorporada en el software de cálculo numérico, Matlab [19]. Esta función tiene como argumentos de entrada la matriz validation y el modelo predictor, que es una variable de tipo objeto. Como resultado de la ejecución de la función predict, se genera una estimación del número de intervenciones de mantenimiento que cada monitor requerirá en el año subsiguiente. El modelo que demuestre un mayor grado de precisión en sus predicciones será seleccionado para ser implementado en el sistema predictor.

C. Prueba sistema predictor

La evaluación del sistema predictor se lleva a cabo a través de la función predict. El primer argumento de entrada es un conjunto de datos que encapsula los registros de fallos de los equipos durante un intervalo de nueve años, específicamente desde el año 2014 hasta el 2022. El segundo argumento es el modelo de predicción que ha sido seleccionado en la etapa previa. La función predict devuelve un vector que contiene las estimaciones del número de fallos que los monitores podrían experimentar en el año subsiguiente, es decir, en el año 2023. Este vector de predicciones se contrasta posteriormente con el conjunto de datos test para determinar la exactitud del sistema predictor. Esta comparación permite calcular el porcentaje de acierto real del sistema, proporcionando una medida cuantitativa de su rendimiento y fiabilidad.

III. RESULTADOS

Se han implementado nueve metodologías distintas para el entrenamiento del sistema predictor. Estas estrategias se basan en tres tipos de kernel: lineal, polinomial y Gaussiano. Además, se han aplicado dos técnicas de reducción de dimensionalidad a dos conjuntos de datos distintos, los cuales contienen la misma información, pero están organizados de forma diferente.

Al examinar las Tablas I, II y III, que presentan los porcentajes de acierto en la predicción para el primer conjunto de datos, se puede observar que el modelo de predicción más eficaz es aquel que ha sido entrenado utilizando un kernel de función gaussiana y los datos de entrenamiento originales, sin aplicar ninguna técnica de reducción de dimensionalidad.

TABLA I
BASE DE DATOS 1 SIN REDUCCIÓN DE DIMENSIONALIDAD

Sin reducción de dimensionalidad		
<i>Kernel</i>	<i>Medida</i>	<i>Valores</i>
<i>Lineal</i>	Media	8,6893 %
	Desviación estándar	± 6.366
<i>Polinomial</i>	Media	9,8571 %
	Desviación estándar	± 7,0907
<i>Gaussiana</i>	Media	58,6093 %
	Desviación estándar	± 11,1977

TABLA II
BASE DE DATOS 1 FSRNCA

FSRNCA		
<i>Kernel</i>	<i>Medida</i>	<i>Valores</i>
<i>Lineal</i>	Media	4,1857 %
	Desviación estándar	± 4,6237
<i>Polinomial</i>	Media	15,0786 %
	Desviación estándar	± 8,1378
<i>Gaussiana</i>	Media	36,9714 %
	Desviación estándar	± 10,7078

TABLA III
BASE DE DATOS 1 PCA

PCA		
<i>Kernel</i>	<i>Medida</i>	<i>Valores</i>
<i>Lineal</i>	Media	6,5957 %
	Desviación estándar	± 5,5942
<i>Polinomial</i>	Media	15,095 %
	Desviación estándar	± 8,1202
<i>Gaussiana</i>	Media	39,2721
	Desviación estándar	± 11,0542

Por otro lado, al examinar las Tablas IV, V y VI, que presentan los porcentajes de acierto en la predicción para el segundo conjunto de datos, se puede observar que el modelo de predicción más eficaz es aquel que ha sido entrenado utilizando un kernel de función polinomial y los datos de entrenamiento a los que se ha aplicado el análisis de componentes principales como técnica de reducción de dimensionalidad.

TABLA IV
BASE DE DATOS 2 SIN REDUCCIÓN DE DIMENSIONALIDAD

Sin reducción de dimensionalidad		
<i>Kernel</i>	<i>Medida</i>	<i>Valores</i>
<i>Lineal</i>	Media	71,7611 %
	Desviación estándar	± 10,1429
<i>Polinomial</i>	Media	87,0786 %
	Desviación estándar	± 7,6052
<i>Gaussiana</i>	Media	84,7985 %
	Desviación estándar	± 8,0906

TABLA V
BASE DE DATOS 2 FSRNCA

FSRNCA		
<i>Kernel</i>	<i>Medida</i>	<i>Valores</i>
<i>Lineal</i>	Media	67,4244 %
	Desviación estándar	± 10,5531
<i>Polinomial</i>	Media	86,8921 %
	Desviación estándar	± 7,6679
<i>Gaussiana</i>	Media	84,7752 %
	Desviación estándar	± 8,0797

TABLA VI
BASE DE DATOS 2 PCA

PCA		
<i>Kernel</i>	<i>Medida</i>	<i>Valores</i>
<i>Lineal</i>	Media	71,7999 %
	Desv. Est.	± 10,1727
<i>Polinomial</i>	Media	89,1249 %
	Desv. Est.	± 7,0348
<i>Gaussiana</i>	Media	82,6071 %
	Desv. Est.	± 8,5437

Este último modelo de predicción ha demostrado ser el más exacto durante la fase de entrenamiento del sistema, alcanzando un porcentaje de acierto del 89,1249% y una desviación estándar de $\pm 7,0348$. Estas métricas de rendimiento son la razón por la cual este modelo ha sido seleccionado para la implementación y evaluación en un entorno real. En la evaluación de este modelo se utiliza el conjunto de datos test, los cuales no han sido expuestos al modelo durante la fase de entrenamiento. Este procedimiento es esencial para evaluar la capacidad de generalización del modelo, es decir, su habilidad para realizar predicciones precisas sobre datos no vistos previamente. Con este conjunto de datos de prueba, el modelo ha demostrado un porcentaje de acierto en la predicción del 72,9713%.

IV. CONCLUSIONES

Al analizar los resultados presentados en las Tablas I, II y III, correspondientes a la base de datos 1, se evidencia que la estructura de este conjunto de datos no es óptima para el entrenamiento del modelo predictor. Esto se atribuye a que cada fallo de los equipos se registra en una fila distinta, generando una dispersión significativa en los datos, lo que dificulta la identificación de un patrón para la predicción.

Al aplicar métodos de reducción de dimensionalidad, no se logran porcentajes de acierto en la predicción sustancialmente superiores a los obtenidos con las bases de datos completas, esto debido a la naturaleza de los datos. Al ser transformados mediante la técnica one-hot encoding se tiene como resultado valores binarios de 0 y 1. Las técnicas de reducción de dimensionalidad que realizan una transformación lineal de los datos o un agrupamiento de los mismos en función de su tendencia y dispersión, no son tan eficaces en estas situaciones. No obstante, el modelo predictor con mayor porcentaje de acierto se obtuvo tras la aplicación de PCA, razón por la que se recomienda implementar al menos dos técnicas de selección de características [15].

Asimismo, dada la importancia de utilizar técnicas de selección de características, es crucial implementar variaciones en la técnica de entrenamiento del modelo predictor. En este caso, las variaciones son proporcionadas por el tipo de función que se utiliza en el kernel. Al implementar estas variaciones, se tiene la posibilidad de seleccionar los parámetros más óptimos para el entrenamiento [20]. Finalmente, se obtuvo un porcentaje de acierto en la predicción del 72,9713%, lo que proporciona una aproximación considerable al comportamiento de los equipos en los años subsiguientes. Sin embargo, este porcentaje podría intentar mejorarse implementando técnicas de regresión más robustas, como lo son las redes neuronales y el aprendizaje profundo (Deep Learning).

REFERENCIAS

- [1] Seguridad del paciente. World Health Organization (WHO). Accedido el 11 de diciembre de 2023. [En línea]. Disponible: <https://www.who.int/es/news-room/fact-sheets/detail/patient-safety>.
- [2] D. L. Leyes, F. Verga, M. Godino, and M. Barbato, “Tiempo de deterioro clínico extra UCI y sobrevida: una casuística.”
- [3] Azcuy Castro Amanda Lucía and Miranda Pedroso Rafael, “Mortality in intensive care unit in Pinar del Rio province hospital,” Universidad Médica Pinareña, 2019, [Online]. Available: <http://galeno.pri.sld.cu>
- [4] A. N. Thomas and I. Galvin, “Patient safety incidents associated with equipment in critical care: A review of reports to the UK National Patient Safety Agency,” *Anaesthesia*, vol. 63, no. 11. pp. 1193–1197, Nov. 2008. doi: 10.1111/j.1365-2044.2008.05607.x.
- [5] E. Iadanza, V. Gonnelli, F. Satta, and M. Gherardelli, “Evidence-based medical equipment management: a convenient implementation,” *Med Biol Eng Comput*, vol. 57, no. 10, pp. 2215–2230, Oct. 2019, doi: 10.1007/s11517-019-02021-x.
- [6] A. Jamshidi, S. A. Rahimi, D. Ait-Kadi, and A. Ruiz Bartolome, “Medical devices Inspection and Maintenance; A Literature Review.”
- [7] J. P. V. Silva, C. J. L. Salas, M. F. M. Cordero, L. J. A. Forero, and C. J. A. Pereira, “Development of a predictor software to determine the periodicity of patient monitors preventive maintenance in the intensive care unit of a health care institution,” in 2021 2nd Sustainable Cities Latin America Conference, SCLA 2021, Institute of Electrical and Electronics Engineers Inc., Aug. 2021. doi: 10.1109/SCLA53004.2021.9540072.
- [8] Iadanza, E., Luschi, A. (2024). Standardization of Failure Codes and Nomenclature of Medical Devices for Evidence-Based Maintenance. In: Badnjević, A., Gurbeta Pokvić, L. (eds) MEDICON’23 and CMBEBIH’23. MEDICON CMBEBIH 2023 2023. IFMBE Proceedings, vol 94. Springer, Cham. https://doi.org/10.1007/978-3-031-49068-2_19
- [9] C. Seger, “An investigation of categorical variable encoding techniques in machine learning: binary versus one-hot and feature hashing”, 2018.
- [10] O. Aziz et al., “Validation of accuracy of SVM-based fall detection system using real-world fall and non-fall datasets”, *PLoS One*, vol. 12, núm. 7, jul. 2017, doi: 10.1371/journal.pone.0180318.
- [11] P. P. Anudeep, S. Kumari, A. S. Rajasimman, S. Nayak, y P. Priyadarsini, “Machine learning predictive models of LDL-C in the population of eastern India and its comparison with directly measured and calculated LDL-C”, *Ann Clin Biochem*, vol. 59, núm. 1, pp. 76–86, ene. 2022, doi: 10.1177/00045632211046805.
- [12] “Permutación aleatoria de enteros - MATLAB randperm- MathWorks América Latina”. MathWorks - Creadores de MATLAB y Simulink - MATLAB y Simulink - MATLAB & Simulink. Accedido el 6 de mayo de 2024. [En línea]. Disponible:

<https://la.mathworks.com/help/matlab/ref/randperm.html>

- [13] D. Krstajic, L. J. Buturovic, D. E. Leahy, y S. Thomas, “Cross-validation pitfalls when selecting and assessing regression and classification models”, *J Cheminform*, vol. 6, núm. 1, mar. 2014, doi: 10.1186/1758-2946-6-10.
- [14] “Fit a support vector machine regression model - MATLAB fitrsvm- MathWorks América Latina”. MathWorks - Creadores de MATLAB y Simulink - MATLAB y Simulink - MATLAB & Simulink. Accedido el 6 de mayo de 2024. [En línea]. Disponible: <https://la.mathworks.com/help/stats/fitrsvm.html>
- [15] B. Venkatesh y J. Anuradha, “A review of Feature Selection and its methods”, *Cybernetics and Information Technologies*, vol. 19, núm. 1, pp. 3–26, 2019, doi: 10.2478/CAIT-2019-0001.
- [16] B. Remeseiro y V. Bolon-Canedo, “A review of feature selection methods in medical applications”, *Computers in Biology and Medicine*, vol. 112. Elsevier Ltd, el 1 de septiembre de 2019. doi: 10.1016/j.combiomed.2019.103375.
- [17] “Feature selection using neighborhood component analysis for regression - MATLAB fsrnca- MathWorks América Latina”. MathWorks - Creadores de MATLAB y Simulink - MATLAB y Simulink - MATLAB & Simulink. Accedido el 7 de mayo de 2024. [En línea]. Disponible: <https://la.mathworks.com/help/stats/fsrnca.html>
- [18] “Análisis de los componentes principales de datos sin procesar - MATLAB pca- MathWorks América Latina”. MathWorks - Creadores de MATLAB y Simulink - MATLAB y Simulink - MATLAB & Simulink. Accedido el 7 de mayo de 2024. [En línea]. Disponible: <https://la.mathworks.com/help/stats/pca.html>
- [19] “Predecir respuestas de un modelo de regresión lineal - MATLAB predict- MathWorks América Latina”. MathWorks - Creadores de MATLAB y Simulink - MATLAB y Simulink - MATLAB & Simulink. Accedido el 7 de mayo de 2024. [En línea]. Disponible: <https://la.mathworks.com/help/stats/linearmodel.predict.html>
- [20] W. J. Lee, H. Wu, H. Yun, H. Kim, M. B. G. Jun, y J. W. Sutherland, “Predictive maintenance of machine tool systems using artificial intelligence techniques applied to machine condition data”, en *Procedia CIRP*, Elsevier B.V., 2019, pp. 506–511. doi: 10.1016/j.procir.2018.12.019.