

**Diagnóstico de Fallas en Motores Eléctricos para  
Vehículos de Dos Ruedas en Condiciones reales y no  
estacionarias**

**PROYECTO FINAL CURSO FUNDAMENTOS DEEP  
LEARNING**

**Carlos Mario García Peña, ingeniero industrial**

**C.C: 1066188667**

**Docente: Raúl Ramos Pollán**

**Facultad de ingeniería**

**Universidad de Antioquia**

**Medellín, Colombia**

**Año 2024**

## Proyecto Final

### 1.1 Contexto de aplicación:

El desarrollo y el uso de motores eléctricos en vehículos de dos ruedas, como las bicicletas eléctricas, están en crecimiento debido a su eficiencia energética y su contribución a la movilidad sostenible. En mi caso, estoy trabajando con un motor eléctrico de imanes permanentes sin escobillas, evaluado en condiciones de tráfico real y no estacionarias, lo que plantea ciertos desafíos en el monitoreo y el diagnóstico de su salud. Una de las problemáticas comunes en estos motores es identificar patrones de fallas o comportamientos anómalos a partir de las series de tiempo capturadas por sensores que monitorean variables críticas como corrientes, potencias, voltajes y temperaturas. El problema que abordaré en este trabajo es la clasificación de estas series temporales para predecir el estado de salud del motor eléctrico y, de esta manera, mejorar su mantenimiento predictivo. El rendimiento de los motores eléctricos en condiciones de operación no estacionarias puede ser afectado por múltiples factores como las corrientes, voltajes, temperaturas y potencias. Esto genera la necesidad de un modelo predictivo que permita detectar fallos o deterioro en el motor. El objetivo es desarrollar un modelo que clasifique el estado del motor (saludable o defectuoso) o en términos más estadísticos una ocurrencia o no ocurrencia del motor (falla o no) basándose en series temporales obtenidas de los sensores que monitorean su funcionamiento.

### 1.2 Objetivo de machine learning

El objetivo de este proyecto es construir un modelo de aprendizaje profundo que permita clasificar series temporales de las señales capturadas por los sensores del motor eléctrico para predecir su estado de salud. En particular, buscamos identificar si el motor se encuentra en condiciones normales o si exhibe comportamientos que sugieren una posible falla inminente. Para ello, utilizaremos un modelo de machine learning basado en Redes Neuronales Recurrentes (RNN), específicamente Long Short-Term Memory (LSTM). Las LSTM son

una arquitectura especializada en la captura de dependencias a largo plazo en series temporales, gracias a su capacidad para mantener información relevante durante múltiples pasos de tiempo y filtrar las irrelevantes mediante sus compuertas de entrada, salida y olvido. Esta propiedad es crucial en la predicción de fallas de motores eléctricos, donde el comportamiento anómalo puede no ser inmediato, sino manifestarse en patrones sutiles y prolongados en variables como la corriente, voltaje, potencia y temperatura. El uso de LSTM permite que el modelo reconozca estos patrones de forma más eficiente que otros enfoques tradicionales, lo que se traduce en una mejor capacidad de predicción y diagnóstico temprano de fallas. Por lo tanto, la implementación de este modelo no solo busca clasificar el estado de salud del motor, sino también anticiparse a los eventos críticos, brindando un soporte esencial en la gestión y mantenimiento preventivo de motores eléctricos.

### 1.3 Dataset

Los datos que se analizarán consisten en series de tiempo multivariadas obtenidas de sensores instalados en el motor eléctrico. Estas variables incluyen corrientes (CURRENT\_A\_CALC, CURRENT\_B\_CALC, CURRENT\_C\_CALC, CURRENT\_D\_CALC), voltajes (VOLTAGE\_A, VOLTAGE\_B, VOLTAGE\_C, VOLTAGE\_D), potencias (POWER\_A, POWER\_B, POWER\_C, POWER\_D) y temperaturas (TEMPERATURE\_MOTOR, TEMPERATURE\_B, TEMPERATURE\_C, TEMPERATURE\_D). Algo importante que se debe mencionar es que para este análisis se contará con las variables específicas del motor eléctrico, como lo son la de las tres fases CURRENT\_B\_CALC, CURRENT\_C\_CALC, CURRENT\_D\_CALC, VOLTAGE\_B, VOLTAGE\_C, VOLTAGE\_D, POWER\_B, POWER\_C, POWER\_D y TEMPERATURE\_MOTOR. Sin embargo, se registraron otras variables como las aceleraciones que en este modelo no se tendrá en cuenta por temas de condiciones no estacionarias, dado que estas se analizan con más frecuencias en sistemas estacionarios para analizar fallas por vibraciones en los motores en condiciones no dinámicas.

#### 1.3.1 Tamaño del dataset:

El conjunto de datos abarca múltiples ciclos de operación del motor, con una duración aproximada de 1800 segundos por ciclo. En total, contamos con alrededor de 78886 registros, distribuidos en ciclos normales y anómalos. Distribución de clases: La clase objetivo será

binaria, indicando si el motor está en estado "normal" o "falla". La distribución de las clases es aproximadamente del 95% de eventos saludables y 5% de eventos fallidos.

### 1.3.2 Capacidad computacional

Especificaciones del dispositivo	
Nombre del dispositivo	Carlosg
Procesador	12th Gen Intel(R) Core(TM) i5-12450H 2.50 GHz
RAM instalada	24,0 GB (23,7 GB usable)
Identificador de dispositivo	033D00E6-458D-4685-9F38-36629A3A9A5B
Id. del producto	00331-10000-00001-AA884
Tipo de sistema	Sistema operativo de 64 bits, procesador basado en x64
Lápiz y entrada táctil	La entrada táctil o manuscrita no está disponible para esta pantalla

*Ilustración 1: Información de la capacidad computacionales*

## 1.4 Métricas de desempeño evaluadas

Para esta investigación, y considerando estudios previos en el campo de la clasificación de series temporales y modelos secuenciales, se han identificado las métricas más relevantes para la evaluación del modelo. En primer lugar, la precisión (accuracy), que mide la proporción de predicciones correctas, es una de las métricas más utilizadas. Además, la pérdida (loss), calculada mediante la función de pérdida de entropía cruzada binaria (binary crossentropy), permite evaluar el desempeño del modelo durante el entrenamiento. Por último, las curvas ROC y el AUC se emplean para medir la capacidad del modelo para distinguir entre las dos clases, proporcionando una validación adicional de su efectividad, además otra de las métricas más comunes es la matriz de confusión dada en clase, la cual nos dará un plus para observar que tanto está calibrando el modelo, es decir, observar su desempeño.

## 1.5 Resultados del modelo final

El conjunto de datos utilizado en este trabajo se dividió en dos partes: 70% para entrenamiento y 30% para validación, tras probar diferentes proporciones y confirmar que esta configuración proporcionaba los mejores resultados para la calibración del modelo. El modelo diseñado, ilustrado en la Ilustración 2, consiste en dos capas LSTM con 128 y 64 unidades respectivamente, intercaladas con capas de Dropout para mitigar el riesgo de sobreajuste, y culmina con una capa densa de salida. Tras un entrenamiento de 100 épocas,

las curvas de precisión mostradas en la Ilustración 3 revelan un desempeño creciente y consistente, con la precisión de validación superando el 97% al final del proceso, lo que refleja una buena generalización del modelo. Asimismo, las curvas de pérdida presentadas en la Ilustración 3 evidencian una disminución constante en la pérdida tanto del conjunto de entrenamiento como de validación, lo que denota un aprendizaje estable sin sobreajuste. La Ilustración 4, que muestra la curva ROC obtenida, subraya el rendimiento robusto del modelo, confirmando su capacidad para distinguir con alta precisión entre las clases desde las primeras fases de entrenamiento. Estos resultados demuestran que el enfoque adoptado fue adecuado y efectivo para el objetivo de este estudio, logrando un equilibrio entre precisión y estabilidad del modelo.

Model: "sequential"

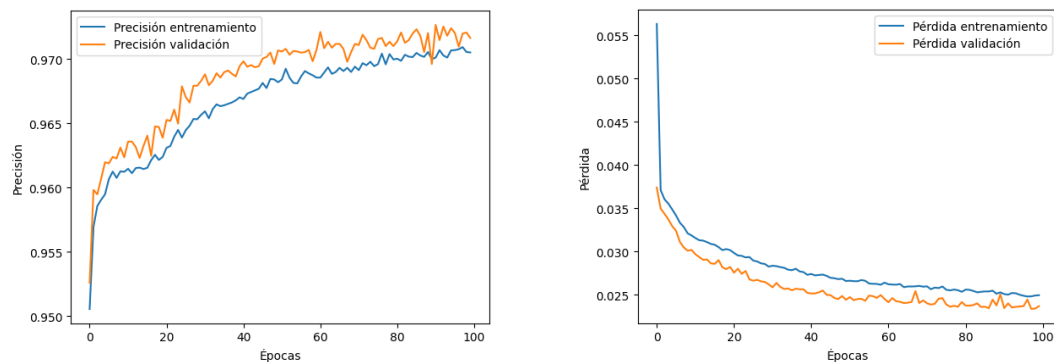
Layer (type)	Output Shape	Param #
lstm (LSTM)	(None, 1, 128)	71,680
dropout (Dropout)	(None, 1, 128)	0
lstm_1 (LSTM)	(None, 64)	49,408
dropout_1 (Dropout)	(None, 64)	0
dense (Dense)	(None, 1)	65

Total params: 121,153 (473.25 KB)

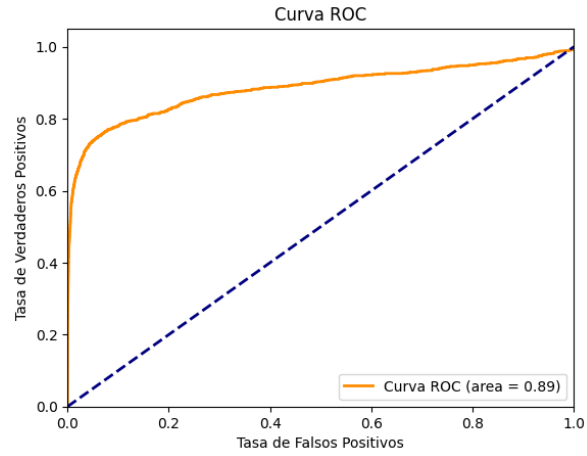
Trainable params: 121,153 (473.25 KB)

Non-trainable params: 0 (0.00 B)

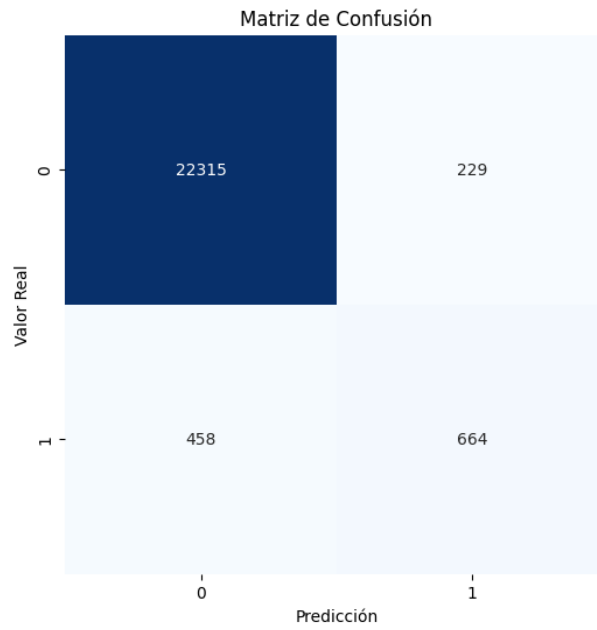
*Ilustración 2: Modelo propuesto*



*Ilustración 3: Visualizamos de las curvas de precisión y pérdida*



*Ilustración 4: Curva ROC*



*Ilustración 5. Matriz de Confusión*

## 1.6 Referencias

Bidirectional LSTM-RNN-based hybrid deep learning frameworks for univariate time series classification

Guía de Tensorflow

Predicción de fallas futuras en motores de turboventiladores con Machine Learning

*An In-Depth Study and Improvement of Isolation Forest | IEEE Journals & Magazine | IEEE*

*Xplore.* (s. f.). Recuperado 31 de octubre de 2024, de <https://ieeexplore.ieee.org/document/9684896>

Khan, M., Wang, H., Riaz, A., Elfatyany, A., & Karim, S. (2021). Bidirectional LSTM-RNN-based hybrid deep learning frameworks for univariate time series classification. *The Journal of Supercomputing*, 77(7), 7021-7045. <https://doi.org/10.1007/s11227-020-03560-z>

Kumar, S., & Ganga, D. (2023). Classification of Rolling Bearing Fault Based on Long Short Term Memory Neural Network. *2023 2nd International Conference for Innovation in Technology (INOCON)*, 1-5. <https://doi.org/10.1109/INOCON57975.2023.10101096>

Mohanty, P. K., Jena, P., & Padhy, N. P. (2022). Electric Vehicle State-of-charge Prediction using Deep LSTM Network Model. *2022 IEEE International Conference on Power Electronics, Drives and Energy Systems (PEDES)*, 1-6. <https://doi.org/10.1109/PEDES56012.2022.10080466>

Sridhar, B., Allirani, S., Reshmin Shafi, R. S., Rithvik Abinav, K. R., & Siddharth, A. (2024). Electric Vehicle Battery Temperature Prediction Using LSTM Algorithm. *2024 IEEE 4th International Conference on Sustainable Energy and Future Electric Transportation (SEFET)*, 1-6. <https://doi.org/10.1109/SEFET61574.2024.10718084>