

**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

Avance del proyecto

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 predecir si el motor está en condiciones normales o si está presentando comportamientos que indican una posible falla inminente. Para lograr esto, emplearemos un modelo de machine learning basado en Redes Neuronales Recurrentes (RNN) que permita predecir el estado de salud del motor eléctrico (saludable o

con fallas) utilizando las series temporales registradas. El modelo buscará identificar patrones temporales en variables como la corriente, voltaje, potencia y temperatura.

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

① 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

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 Referencias y resultados previos

En primer lugar, el conjunto de datos se dividió en dos partes: una destinada al entrenamiento y otra a la validación, siguiendo la proporción estándar de 70-30. Luego, se diseñó el modelo presente en la Ilustración 2 y se entrenó durante 10 epochs. La Ilustración 4 presenta la curva ROC obtenida con este modelo, mostrando resultados satisfactorios desde esta etapa, también en la Ilustración 3 se presentan las curvas de precisión y pérdida para evaluar cómo ha calibrado el modelo. A continuación, incluyo los enlaces que utilizaré como

Referencia para este proyecto:

- [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](#)

Model: "sequential_10"

Layer (type)	Output Shape	Param #
lstm_23 (LSTM)	(None, 1, 128)	72,192
dropout_22 (Dropout)	(None, 1, 128)	0
lstm_24 (LSTM)	(None, 64)	49,408
dropout_23 (Dropout)	(None, 64)	0
dense_9 (Dense)	(None, 1)	65

Total params: 121,665 (475.25 KB)

Trainable params: 121,665 (475.25 KB)

Non-trainable params: 0 (0.00 B)

Ilustración 2: Modelo inicial propuesto

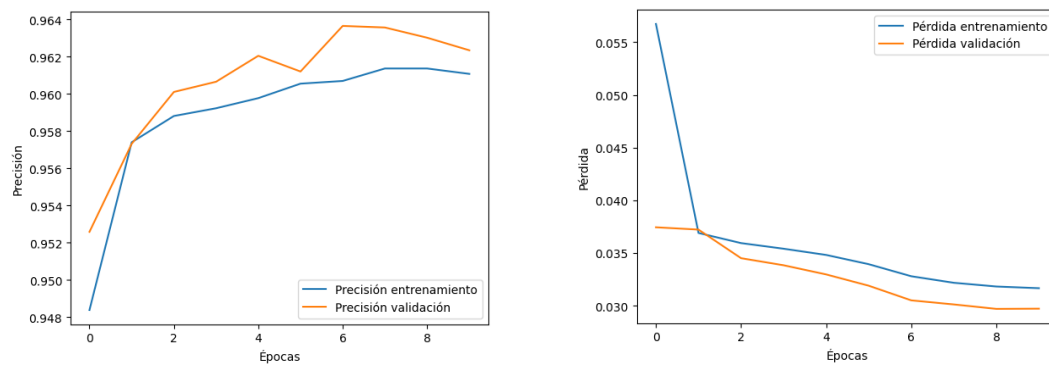


Ilustración 3: Visualizamos de las curvas de precisión y pérdida

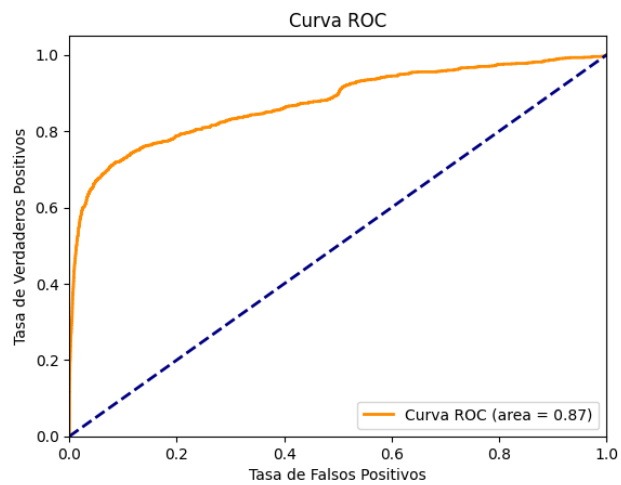


Ilustración 4: Curva ROC