Modelo predictivo para la vida útil restante de un componente crítico de un vehículo eléctrico

Catherine Johana Rincón Maya

Tesis de maestría presentada para optar al título de Magíster en Ingeniería

Olga Cecilia Úsuga Manco, Doctora (PhD) en Ciencias - Estadística

Carmen Elena Patiño Rodríguez, Doctora (PhD) en Ingeniería

Universidad de Antioquia

Facultad de Ingeniería

Maestría en Ingeniería

Medellín, Antioquia, Colombia

2024

Cita	(Rincón Maya, 2024)
Referencia **Estilo APA 7 (2020)**	Rincón Maya, C. (2024). *Modelo predictivo para la vida útil restante de un componente crítico de un vehículo eléctrico* [Tesis de maestría]. Universidad de Antioquia, Medellín, Colombia.

creative commons BY NC SA

Maestría en Ingeniería, Cohorte XXXV.

Grupo de Investigación Analítica e Investigación para la Toma de Decisiones (ALIADO).

Centro de Investigación Ambientales y de Ingeniería (CIA).

UNIVERSIDAD DE ANTIOQUIA
Vicerrectoría de Docencia

Sistema de Bibliotecas

Centro de Documentación de 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.

Agradecimientos

"Siempre hay que encontrar el tiempo para agradecer a las personas que hacen una diferencia en nuestras vidas" John F. Kennedy

Quiero agradecer a mis directoras de tesis, Olga Úsuga y Carmen Patiño, por recibirme dentro del grupo ALIADO de la Facultad de Ingeniería, por brindarme su conocimiento, guía y confianza para llevar a cabo mi trabajo de investigación. Quiero agradecer al profesor Freddy Hernández por brindarme su conocimiento, tiempo y paciencia para ayudarme en la resolución de las dudas que surgieron durante la realización de mi trabajo.

Quiero agradecer a mi familia, especialmente a mi madre por apoyarme y brindarme el espacio para realizar mis actividades académicas. A mi compañero de vida por alentarme, darme su apoyo y tiempo para culminar mi trabajo de investigación.

Quiero expresar mis agradecimientos a mi alma mater, mi segundo hogar, que me ha formado como persona y profesional y me ha permitido conocer grandes personas. Quiero agradecer a todas las personas que con su apoyo, motivación y paciencia permitieron la culminación de mis estudios.

También quiero extender mis agradecimientos a los profesores que me aconsejaron y dieron recomendaciones para llevar a cabo mi trabajo de investigación. Así mismo, quiero agradecer a los estudiantes que participaron en las convocatorias para hacer las pruebas y obtener los datos necesarios para desarrollar el modelo predictivo.

Resumen

El trabajo de investigación se enfocó en el desarrollo de una metodología para la implementación de un modelo predictivo de la vida útil restante de un componente crítico como la batería. Dentro de la metodología, la predicción de la vida útil se desarrolló a través de dos modelos predictivos: el primero implementa una red neuronal convolucional que permite el suavizamiento de los datos obtenidos por medio de sensores, y el segundo modelo predictivo, implementa una red de corto y largo plazo para una predicción más precisa. La implementación y evaluación de la metodología propuesta del modelo predictivo se aplicó a dos casos de estudio: un caso de estudio de baterías operando bajo condiciones controladas y otro caso de estudio desarrollado en un vehículo eléctrico de dos ruedas operando bajo condiciones de tráfico real. Así mismo, para la implementación de la metodología, se desarrolló un paquete en R llamado `cnnlstm`. Este paquete permite llevar a cabo la metodología propuesta a partir del procesamiento de los datos, la división necesaria para cada una de las redes neuronales, el ajuste de parámetros en cada una de las arquitecturas y la optimización de los hiperparámetros. El modelo predictivo propuesto genera un RMSE de menos de 0.04 para el caso de baterías en condiciones controladas y del 0.095 para las baterías en condiciones de tráfico real, lo que demuestra su buen desempeño.

Palabras clave: modelo predictivo, vida útil restante (RUL), red convolucional (CNN), red de corto y largo plazo (LSTM), vehículo eléctrico (EV), `cnnlstm`.

Abstract

The research focused on the development of a methodology for the implementation of a predictive model of the remaining useful life of a critical component such as the battery. Within the methodology, the prediction of the remaining useful life was developed through two predictive models: the first one implements a convolutional neural network that allows the smoothing of the data obtained through sensors, and the second predictive model implements a long-short term memory neural network to get a better prediction. The implementation and evaluation of the proposed methodology of the predictive model was applied on two case studies: a case study of batteries operating under controlled conditions and another case study developed in an electric two-wheeled vehicle operating under real traffic conditions. Likewise, for the implementation of the methodology, a package in R called `cnnlstm` was developed. This package allows carrying out the proposed methodology from the data processing, the necessary division for each of the neural networks, the adjustment of parameters in each of the architectures and the optimization of the hyperparameters. The proposed predictive model generates an RMSE of less than 0.04 for the case operating under controlled conditions and an RMSE of less than 0.095 for the case operating under real traffic conditions, which demonstrates its good performance.

Keywords: predictive model, remaining useful life (RUL), convolutional network (CNN), long-short term memory (LSTM), electric vehicle (EV), `cnnlstm`.

Tabla de contenido

1. INTRODUCCIÓN 16
2. OBJETIVOS DEL PROYECTO 20
2.1 Objetivo general 20
2.2 Objetivos específicos 20
3. MODELOS PREDICTIVOS EN EV 21
3.1 Descripción general 21
3.2 Modelos predictivos del SOH y RUL 22
3.3 Metodologías implementadas en modelos predictivos 28
4. REDES NEURONALES 38
4.1 Descripción general 38
4.2 Redes neuronales convolucionales – CNN 50
4.3 Redes neuronales de corto y largo plazo – LSTM 53
5. PROPUESTA DEL MODELO PREDICTIVO 57
5.1 Descripción general 57
5.2 Estructura del modelo predictivo 58
5.3 División de los datos 59
Paso 1. Obtener datos de degradación 60
Paso 2: Obtener datos para el modelo de suavizamiento 61
Paso 3: Obtener datos para el modelo predictivo 62
5.4 Modelo de suavizamiento con la CNN 64
Paso 1: Estructurar datos 65
Paso 2: Definir la arquitectura de la CNN 67
Paso 3: Entrenar el modelo de suavizamiento 68
Paso 4: Optimizar los hiperparámetros del modelo de suavizamiento 69
Paso 5: Realizar suavizamiento 71
5.5 Modelo predictivo con la red LSTM 72
Paso 1: Estructurar datos 73
Paso 2: Definir la arquitectura de la red LSTM 75
Paso 3: Entrenar el modelo predictivo 76
Paso 4: Optimizar los hiperparámetros del modelo predictivo 77
Paso 5: Realizar predicciones 79
5.6 Paquete en lenguaje de programación R – `cnnlstm` 80
6. IMPLEMENTACIÓN DEL MODELO PREDICTIVO 84
6.1 Escenario en condiciones controladas – Datos públicos de baterías 84

6.1.1 Análisis descriptivo de los datos 86
6.1.2 Implementación de la metodología 93
6.1.3 Evaluación de la precisión del RUL 99
6.2 Escenario en condiciones de tráfico real – Datos experimentales 104
6.2.1 Análisis descriptivo de los datos 106
6.2.2 Implementación de la metodología 119
6.2.3 Evaluación de la precisión del RUL 125
7. CONCLUSIONES 130
8. REFERENCIAS 133
9. ANEXOS 141

Lista de tablas

Tabla 1 Sistemas y variables a considerar en un estudio experimental para predecir una variable de desempeño....26
Tabla 2 Ventajas y desventajas de las metodologías....32
Tabla 3 Límites de evaluación de las medidas de desempeño de un modelo predictivo 36
Tabla 4 Funciones de activación implementadas en las redes neuronales 43
Tabla 5 Hiperparámetros y técnicas de regularización en una red neuronal 47
Tabla 6 Métodos implementados para la optimización de hiperparámetros....48
Tabla 7 Hiperparámetros de la CNN 52
Tabla 8 Hiperparámetros de la red LSTM 56
Tabla 9 Valores de los hiperparámetros fijos para la optimización en la CNN....70
Tabla 10 Hiperparámetros y valores para la optimización en la CNN 70
Tabla 11 Valores de los hiperparámetros fijos para la optimización en la red LSTM....77
Tabla 12 Hiperparámetros y valores para la optimización en la red LSTM....78
Tabla 13 Variables medidas en condiciones controladas 85
Tabla 14 Descripción datos de baterías publicados por la NASA....86
Tabla 15 Medidas descriptivas de los datos de las baterías en condiciones controladas 88
Tabla 16 Ciclos de degradación en condiciones controladas 95
Tabla 17 Hiperparámetros y valores a optimizar del modelo de suavizamiento en ambiente controlado 95
Tabla 18 Errores de entrenamiento y prueba obtenidos en el modelo de suavizamiento con los datos en ambiente controlado 96
Tabla 19 Errores en el suavizamiento de los datos en ambiente controlado 97
Tabla 20 Hiperparámetros y valores a optimizar del modelo predictivo en ambiente controlado 98
Tabla 21 Errores de entrenamiento, validación y prueba obtenidos en el modelo predictivo con los datos en ambiente controlado 99
Tabla 22 Planteamiento de escenarios para datos de baterías en ambiente controlado 99
Tabla 23 Errores obtenidos en cada uno de los escenarios y baterías en condiciones controladas .104
Tabla 24 Variables medidas bajo condiciones de tráfico real 106
Tabla 25 Descripción datos de baterías experimentales....107
Tabla 26 Medidas descriptivas de los datos de las baterías en condiciones de tráfico real 108
Tabla 27 Ciclos de degradación en condiciones controladas 120
Tabla 28 Hiperparámetros y valores a optimizar del modelo de suavizamiento en condiciones de tráfico real 121
Tabla 29 Errores de entrenamiento y prueba obtenidos en el modelo de suavizamiento con los datos en condiciones de tráfico real....122
Tabla 30 Errores en el suavizamiento de los datos en condiciones de tráfico real 123
Tabla 31 Hiperparámetros y valores a optimizar del modelo predictivo en condiciones de tráfico real 124
Tabla 32 Errores en el entrenamiento y prueba de la predicción del RUL en condiciones de tráfico real 125
Tabla 33 Planteamiento de escenarios para datos de baterías en condiciones de tráfico real 125
Tabla 34 Errores obtenidos en cada uno de los escenarios y baterías en condiciones de tráfico real 128
Tabla 35 Métricas de desempeño, definición y ecuación....141
Tabla 36 Librerías de Python para la implementación de redes neuronales....142
Tabla 37 Librerías desarrolladas en R para la implementación de redes neuronales....143
Tabla 38 Versión de paquetes implementados en R 145

Tabla 39 Importancia de las variables con Radom Forest para el escenario en condiciones controladas 146
Tabla 40 Importancia de las variables con Radom Forest para el escenario en condiciones de tráfico real 146
Tabla 41 Niveles de severidad.....149
Tabla 42 Niveles de la probabilidad del efecto de falla β 149
Tabla 43 Formato de la hoja de trabajo FMECA a partir de (Guedes Soares, 2002) 152

Lista de figuras

Figura 1. Relación entre las variables, estado de salud y predicción del RUL. Fuente: Elaboración propia a partir de (Ge et al., 2021) 27
Figura 2. Publicaciones de RUL, SOH y SOC en los últimos once años. Fuente: Elaboración propia28
Figura 3. Metodologías implementadas para estimar el SOH. Fuente: Elaboración propia a partir de (Ge et al., 2021) 29
Figura 4. Metodologías implementadas para predecir el RUL. Fuente: Elaboración propia a partir de (Ge et al., 2021) 31
Figura 5. Transformación geométrica de los datos que realiza una capa oculta. Fuente: Elaboración propia a partir de (Ghatak, 2019).....39
Figura 6. Nodos de una red neuronal con tres entradas, cuatro nodos de una capa oculta y una capa de salida. Fuente: Elaboración propia a partir de (Ghatak, 2019).....40
Figura 7. Red neuronal (a) estándar y (b) con tasa de abandono 46
Figura 8. Arquitectura común de una CNN. Fuente: Elaboración propia basada en (Ghatak, 2019) 51
Figura 9. Arquitectura de la LSTM. Fuente: Elaboración propia basada en (Listou Ellefsen et al., 2019) 54
Figura 10. Metodología desarrollada para el modelo predictivo.....58
Figura 11. Estructura de los pasos a seguir para realizar la división de datos.....60
Figura 12. Estructura de los pasos a seguir para realizar el suavizamiento con la CNN.....64
Figura 13. Arquitectura de la CNN 67
Figura 14. Arquitectura y cantidad de parámetros de la CNN 68
Figura 15. Estructura de los pasos a seguir para realizar la predicción con la red LSTM.....73
Figura 16. Arquitectura y cantidad de parámetros para la red LSTM.....76
Figura 17. Estado de salud de las baterías en ambiente controlado 87
Figura 18. Capacidad de las baterías en ambiente controlado 87
Figura 19. Matriz de correlaciones datos en condiciones controladas 90
Figura 20. Registros de la corriente por cada ciclo para baterías en ambiente controlado 91
Figura 21. Registros de la temperatura por cada ciclo para baterías en ambiente controlado 92
Figura 22. Registros del voltaje por cada ciclo para baterías en ambiente controlado 92
Figura 23. Gráficas de las medidas descriptivas de las variables de cada batería en ambiente controlado 93
Figura 24. Evaluación de los datos de degradación para la B05 en ambiente controlado 94
Figura 25. Resultados escenario 1 con datos en condiciones controladas.....100
Figura 26. Resultados escenario 2 con datos en condiciones controladas.....101
Figura 27. Resultados escenario 3 con datos en condiciones controladas.....102
Figura 28. Resultados escenario 4 con datos en condiciones controladas.....103
Figura 29. Criticidad de los componentes de la bicicleta eléctrica 105
Figura 30. RUL de las baterías en condiciones de tráfico real.....107
Figura 31. Matriz de correlaciones de los datos en condiciones de tráfico real 110
Figura 32. Registros de la aceleración en el eje x por ciclo para B01 y B02 en condiciones de tráfico real 111
Figura 33. Registros de la aceleración en el eje y por ciclo para B01 y B02 en condiciones de tráfico real 112
Figura 34. Registros de la aceleración en el eje z por ciclo para B01 y B02 en condiciones de tráfico real 112
Figura 35. Registros de la corriente por ciclo para B01 y B02 en condiciones de tráfico real.....113
Figura 36. Registros de la potencia por ciclo para B01 y B02 en condiciones de tráfico real.....114

Figura 37. Registros de la temperatura por ciclo para B01 y B02 en condiciones de tráfico real ...115
Figura 38. Registros del voltaje por ciclo para B01 y B02 en condiciones de tráfico real 116
Figura 39. Registros de la humedad ambiental para B01 y B02 en condiciones de tráfico real 117
Figura 40. Registros de la temperatura ambiental para B01 y B02 en condiciones de tráfico real..118
Figura 41. Evaluación de los datos de degradación para la B01 en condiciones de tráfico real 119
Figura 42. Resultados escenario 1 con datos en condiciones de tráfico real 126
Figura 43. Resultados escenario 2 con datos en condiciones de tráfico real 126
Figura 44. Resultados escenario 3 con datos en condiciones de tráfico real 127
Figura 45. Resultados escenario 4 con datos en condiciones de tráfico real 128
Figura 46. Sistemas y funciones de la bicicleta y patineta eléctrica.....147
Figura 47. Procedimiento para realizar el FMECA. Fuente: Elaboración propia basada en (Guedes Soares, 2002) 151

Lista de símbolos – Notación

Notación redes neuronales

p: Representa el número de *features*.

$\boldsymbol{X}_p^i$: Representa el vector de las observaciones de una p *feature* específica, donde i hace referencia a la i-ésima observación.

$\boldsymbol{X} = \boldsymbol{X}_1, \boldsymbol{X}_2, \dots, \boldsymbol{X}_p$: Representa la matriz de las p *features* de entrada.

$\boldsymbol{W} = \boldsymbol{W}_1, \boldsymbol{W}_2, \dots, \boldsymbol{W}_p$: Representa la matriz de pesos de cada una de las conexiones de la red neuronal.

Z: Representa la ecuación de transformación geométrica de los datos.

$\boldsymbol{b}$: Representa el vector del sesgo obtenido y que es sumado a la ecuación Z.

σ: Representa la función de activación que se debe aplicar a la ecuación Z.

$\hat{y}$ o y: Representa el *target*.

$h_k^{[l]}$: Representa un nodo de capa oculta, donde l hace referencia a la l-ésima capa oculta y k hace referencia al k-ésimo nodo de una capa oculta.

Notación redes neuronales convolucionales

$s(t)$: Representa el mapa de *features* en una red convolucional.

x: Corresponde a los valores de entrada.

w: Representa el *kernel* aplicado en la red convolucional.

$*$: Es el operador de convolución en la red convolucional.

t: Es la posición en el tiempo de la convolución.

$\boldsymbol{I}(m, n)$: Representa una entrada con dos dimensiones (m, n).

$\boldsymbol{K}(i, j)$: Representa el *kernel* aplicado en dos dimensiones (i, j).

$\boldsymbol{S}(i, j)$: Representa el mapa de *features* en dos dimensiones (i, j).

Notación redes neuronales de corto y largo plazo

f_t: Es la puerta de olvido de la red LSTM

i_t: Es la puerta de entrada de la red LSTM

$\tilde{C}_t$: Es un estado temporal de la puerta de entrada de la red LSTM

o_t: Es la puerta de salida de la red LSTM

Metodología

$\boldsymbol{Y}$: Vector del *target*

$\boldsymbol{X}$: Matriz de p *features* seleccionadas como variables de entrada para el modelo predictivo

$\boldsymbol{Y^m}$: Vector del *target* con m filas (número de observaciones). $\boldsymbol{Y^m} = Y^1, Y^2, \dots, Y^m$

$\boldsymbol{X_p^m}$: Matriz con dimensiones $p \times m$, donde p representa el número de *features* y m representa el número de filas que se tienen de cada una de las *features*. $\boldsymbol{X_p^m} = \boldsymbol{X_1^m}, \boldsymbol{X_2^m}, \dots, \boldsymbol{X_p^m}$, donde el número de p $features = 1, 2, \dots, p$, y el número de observaciones $= 1, 2, \dots, m$.

$\boldsymbol{X_s^m}$: Matriz con dimensiones $s \times m$, donde s representa el número de *features* obtenidas por medio de sensores y m representa el número de filas

$\boldsymbol{X_{p-s}^m}$: Matriz con dimensiones $p - s \times m$, donde $p - s$ representa el número de *features* obtenidas sin sensores y m representa el número de filas

k: Observaciones identificadas como filas de degradación

$\boldsymbol{X_p^{m-k}}$: Matriz con dimensiones $p \times m - k$, donde p representa el número de *features* y $m - k$ representa el número de filas después de eliminar el conjunto de observaciones k

Lista de abreviaturas

Abreviatura	Descripción en inglés	Descripción en español
AR	AutoRegressive	Autoregresivo
MA	Moving Average	Media Móvil
ARMA	AutoRegressive Moving Average	Autoregresivo de Media Móvil
ARIMA	AutoRegressive Integrated Moving Average	Autoregresivo Integrado de Media Móvil
KF	Kalman Filter	Filtro de Kalman
PF	Particle Filter	Filtro de partícula
ANN	Artificial Neural Network	Red Neuronal Artificial
CNN	Convolutional Neural Network	Red Neuronal Convolucional
LSTM	Long-Short Term Memory	Red Neuronal de Corto y Largo Plazo
DNN	Deep Neural Network	Red Neuronal Profunda
EM	Electric Motor	Motor Eléctrico
LIB	Lithium-Ion Battery	Batería de Ion Litio
EV	Electric Vehicle	Vehículo Eléctrico
SOC	State Of Charge	Estado De Carga
SOH	State of Health	Estado De Salud
RUL	Remaining Useful Life	Vida Útil Restante
RMSE	Root Mean Square Error	Raíz del Error Cuadrático Medio
MSE	Mean Square Error	Error Cuadrático Medio
MAE	Mean Absolute Error	Error Medio Absoluto
MAPE	Mean Absolute Percentage Error	Error Porcentual Medio Absoluto
FMECA	Failure Mode, Effects, and Criticality Analysis	Análisis de modo de falla, efecto y análisis de criticidad

Lista de anexos

Anexo 1. Definiciones y ecuaciones de las métricas de desempeño de modelos predictivos de regresión 141
Anexo 2. Modelos predictivos computacionales de Python y R.....142
Anexo 3. Versión de los paquetes implementados en el desarrollo de la metodología propuesta para la predicción con datos controlados y datos en condiciones de tráfico real. 145
Anexo 4. Análisis de importancia de las variables para cada escenario implementando Random Forest 146
Anexo 5. Descripción de la bicicleta eléctrica, FMECA, instrumentación y captura de datos experimentales bajo condiciones de tráfico real 147

1. INTRODUCCIÓN

En los últimos años, se ha incrementado la recolección de datos a partir de diferentes fuentes, lo cual ha generado la necesidad de análisis complejos que permitan obtener conocimiento del campo específico para la toma de decisiones. El análisis de los datos permite la detección de patrones, identificación de tendencias y clasificación de comportamientos, lo que posibilita la generación de modelos predictivos en un campo específico, con el objetivo de brindar herramientas y servicios que permitan generar un sistema para el proceso de toma de decisiones. Debido al gran potencial de los modelos predictivos para generar conocimiento, estos han sido implementados en diferentes campos tales como: comercio, economía, finanzas, medicina, educación, industria, transporte, entre otros (Dinov, 2018; Ge et al., 2021; Kuhn & Johnson, 2019; Zraibi et al., 2021).

El campo de la industria de la energía y el transporte, son los principales campos que contribuyen con la contaminación ambiental, las emisiones de carbono y la dependencia de combustibles fósiles (Hu et al., 2016; Hu et al., 2020; Nanaki, 2021; Zhang et al., 2022). Debido a la presión por la reducción de estas problemáticas a nivel mundial, los gobiernos deben implementar estrategias y políticas que permitan garantizar la seguridad energética y lograr el desarrollo sostenible de la sociedad (Ge et al., 2021; Hu et al., 2016; Zhang et al., 2022). La comisión europea se ha trazado una ruta energética al año 2050, la cual contempla la reducción de las emisiones en países como China, EEUU y Corea, entre otros (J. Hu et al., 2016). Es por esto, que existe una fuerza impulsadora significativa para el desarrollo de nuevas fuentes de energía renovable y sistemas de transporte limpios (Hu et al., 2016; Hu et al., 2020; Nanaki, 2021; Zhang et al., 2022). En el sector del transporte, el vehículo eléctrico (EV, por sus siglas en inglés: *Electric Vehicle*) ha representado un crecimiento exponencial en ventas a nivel mundial durante los últimos años, y se espera que para el año 2040, el 57% de las ventas globales sean de EV (Nanaki, 2021; UPME, 2019).

En Colombia, por medio del Plan Nacional de Desarrollo 2018-2022, se viene impulsando la transformación del sector transporte hacia tecnologías limpias, gracias a los diferentes beneficios definidos para la incorporación en el mercado de los EV, esperando para el 2030 llegar a la meta de 600.000 EV (UPME, 2019). A través de este plan de movilidad limpia, eficiente y sostenible, y acciones para mejorar la calidad del aire, se espera que para el año 2030 haya una reducción de emisiones de gases de efecto invernadero del 51% por parte de todos los sectores (UPME, 2019). El uso de estos EV permitirá reducir las emisiones, conservar la energía, proteger el medio ambiente y promover el desarrollo sostenible de la sociedad (Ge et al., 2021; J. Hu et al., 2016; R. Yang et al., 2021; X. Zhang et al., 2022). Así, la electrificación en la industria automotriz es un objetivo de consenso global para reducir las emisiones de gases de efecto invernadero,

brindando beneficios ambientales, técnicos, sociales y de salud, potenciando la mejora de la eficiencia y la seguridad energética (Marcos et al., 2021; Nanaki, 2021).

Dentro de los EV se encuentran los EV de dos ruedas, los cuales se destacan por su bajo peso, transporte personal, rápido y eficiente, baja necesidad energética y mayor eficiencia en el funcionamiento (Chien et al., 2023). Este tipo de EV es de gran interés para el uso en zonas urbanas y también para hacer recorridos de distancias cortas (Ajanovic et al., 2021). Las ventas de los EV de dos ruedas han venido en aumento en los últimos años, sin embargo, la incorporación en la flota mundial es relativamente baja, por lo que los fabricantes y gobiernos deben incrementar la disponibilidad de tecnología y definir políticas que incentiven el uso de EV (Nanaki, 2021; UPME, 2019).

Los EV de dos ruedas tienen diferentes componentes críticos cuya falla impide su correcto funcionamiento, y, en consecuencia, se genera una falla directa en el EV. Los componentes críticos más reportados son: la batería y el motor (EM, por sus siglas en inglés: *Electric Motor*) (H. C. Liu et al., 2013). La batería es el principal suministro de energía y componente de almacenamiento que puede costar hasta un 30% del costo total del EV de dos ruedas. El EM, es el encargado de usar la energía almacenada en las baterías para la propulsión del EV (Ge et al., 2021; X. Hu et al., 2020; Wu et al., 2016). Por esto, una batería y un EM de larga duración son fundamentales para lograr la viabilidad económica de los EV de dos ruedas, la implementación de energía renovable y el desarrollo de la infraestructura de redes inteligentes (Ge et al., 2021; X. Hu et al., 2020). Las baterías de iones de litio (LIB, por sus siglas en inglés: *Lithium-ion battery*), se consideran la fuente de energía más viable para este tipo de vehículos, debido a su buena eficiencia, ciclo de vida prolongado, alta potencia y densidades de energía (Z. Chen et al., 2021a; Fan et al., 2019; Ge et al., 2021; X. Hu, Deng, et al., 2021; Ji et al., 2021; W. Li et al., 2019; Marcos et al., 2021; Samanta et al., 2021; X. Zhang et al., 2022).

Los componentes críticos de un EV de dos ruedas deben equiparse de un sistema para administrar de manera efectiva su seguridad y confiabilidad (Ge et al., 2021). La seguridad se refiere a las características, sistemas y medidas implementadas para garantizar la protección de los ocupantes, peatones y el entorno durante la conducción, y la confiabilidad es entendida como la probabilidad de que el EV de dos ruedas cumpla la función para la cual fue diseñado durante un tiempo predeterminado. Para el caso de las baterías, se cuenta con un sistema de gestión de baterías (BMS, por sus siglas en inglés: *battery management system*) inteligente. Para que un BMS sea confiable, es importante estimar su estado de salud (SOH, por sus siglas en inglés: *state of health*), ya que mediante la estimación precisa del SOH y la predicción de la vida útil restante (RUL, por sus siglas en inglés: *remaining useful life*), se puede comprender el rendimiento de cada

componente crítico y obtener información clave para garantizar el funcionamiento seguro y confiable del EV de dos ruedas (Ge et al., 2021; X. Hu et al., 2020).

Uno de los problemas de estos componentes críticos es la degradación durante el funcionamiento, ya que es un proceso complejo, que incluye reacciones fisicoquímicas y variación de parámetros en el tiempo que presentan un comportamiento no lineal (Hu et al., 2020). Para evitar fallas potencialmente peligrosas y riesgos de seguridad tales como incendios, fugas y explosiones, es esencial monitorear la variación de los parámetros que intervienen en la degradación del componente crítico. En este sentido, predecir de manera precisa el RUL de los componentes críticos en condiciones de funcionamiento es una estrategia que permite garantizar la seguridad y confiabilidad del EV (Ge et al., 2021; Hu et al., 2020). Por esto, debido a la naturaleza del proceso de degradación de los componentes críticos, predecir el RUL de la batería y/o del EM se convierte en una tarea desafiante, ya que se debe considerar el comportamiento no lineal, las fallas que pueden ocurrir, y las relaciones e interacciones entre las variables (Hu et al., 2020).

Como metodologías para la predicción del RUL, se han implementado métodos basados en modelos, métodos basados en datos, así como métodos híbridos que corresponden a la combinación de los anteriores (X. Hu et al., 2020). Generalmente para la predicción del RUL se han usado datos públicos de experimentaciones controladas que limitan la afectación de una variable sobre el RUL, por lo que no se consideran variables asociadas al tráfico, variables ambientales o topográficas en la experimentación (J. C. Chen et al., 2021; Z. Chen et al., 2021; Ge et al., 2021; Gou et al., 2021; Ji et al., 2021; Jiang et al., 2019; Jorge et al., 2020; Kara, 2021; Khumprom & Yodo, 2019; Listou Ellefsen et al., 2019; K. Liu et al., 2021; Pan et al., 2020; Qu et al., 2019; Ren et al., 2018; Shenghao, 2021; Sun et al., 2019; Tong et al., 2021; J. Wang et al., 2019; Xue et al., 2020; X. Zhang et al., 2022; Zhu et al., 2019; Zraibi et al., 2021). Por otra parte, Abdelaty et al. (2021) realizó una experimentación en campo, considerando las variables relacionadas con factores externos como topografía de la ruta, comportamiento del conductor, condiciones climáticas y parámetros del tráfico, con el fin de identificar las variables que influyen en la predicción del RUL del EV (Abdelaty et al., 2021). Este tipo de experimentación en campo permite determinar la naturaleza de la relación entre las variables cuando están expuestas a cambios no controlables. Además, se identifican los parámetros significativos que afectan el desempeño de un EV de dos ruedas en una condición de operación real, y se obtienen resultados relevantes sobre el efecto en el desempeño del EV de factores externos que varían significativamente (Abdelaty et al., 2021).

Por esto, se torna fundamental desarrollar un modelo predictivo del RUL de un componente crítico de un EV implementado en dos escenarios: un escenario que permita evaluar el modelo predictivo en un ambiente

controlado, y otro escenario con un EV de dos ruedas con datos obtenidos en condiciones de funcionamiento reales. Estos escenarios darán resultados que permitan definir mantenimientos preventivos eficientes, de modo que se garantice la seguridad de los sistemas y se disminuyan los costos asociados (X. Hu, Deng, et al., 2021). A través de la recolección de los datos y la construcción del modelo, se pretende estudiar los comportamientos de degradación de los componentes críticos de un EV de dos ruedas. Se realizará la captura de los datos a través de un sistema de adquisición de datos para establecer las entradas del modelo. De esta forma, con el desarrollo del proyecto se pretende dar respuesta a la siguiente pregunta de investigación: ¿Cómo predecir el RUL de un EV a partir de un modelo predictivo evaluado en dos escenarios: uno en condiciones controladas y otro escenario en condiciones de tráfico real?

El documento se organiza de la siguiente forma: en el capítulo 2 se presentan los objetivos de la investigación, en el capítulo 3 se presenta una revisión de literatura sobre modelos predictivos, definición de conceptos y las tendencias en la implementación de modelos predictivos para el RUL. En el capítulo 4 se realiza una descripción general de las redes neuronales con su respectiva notación, así mismo, se hace la descripción de la red convolucional y la red de corto y largo plazo. En el capítulo 5 se presenta el modelo predictivo implementado para la predicción del RUL junto con la metodología desarrollada para el logro del objetivo. Además, se presenta el paquete desarrollado en el lenguaje de programación R para la implementación de la metodología propuesta que considera el suavizamiento de los datos obtenidos por medio de sensores a través de una CNN y que sirven como datos de entrada para el modelo predictivo con la red LSTM. En el capítulo 6 se presenta la implementación del modelo predictivo propuesto, descrito en los capítulos anteriores, por medio de dos escenarios: un escenario en el que se ponen a prueba cuatro baterías bajo condiciones controladas y otro escenario en el que se hace uso de una bicicleta eléctrica bajo condiciones de tráfico real. En el capítulo 7 se presentan las conclusiones del trabajo de investigación y finalmente, en el capítulo 8 y 9 se presentan las referencias y los anexos respectivamente.

2. OBJETIVOS DEL PROYECTO

2.1 Objetivo general

Desarrollar una metodología para la implementación de un modelo predictivo para el RUL de una batería de un EV de dos ruedas operando bajo condiciones de tráfico real.

2.2 Objetivos específicos

1. Identificar las metodologías implementadas en los modelos predictivos para predecir el RUL.

2. Proponer un modelo predictivo que integre metodologías para el suavizamiento de los datos y la predicción del RUL.

3. Implementar, en un lenguaje de programación de uso libre, un modelo predictivo para analizar el comportamiento de variables con características similares a las estudiadas.

4. Evaluar el modelo predictivo implementado, usando los datos de las variables recopilados en tiempo real.

3. MODELOS PREDICTIVOS EN EV

En este capítulo se presentan los conceptos relacionados con modelos predictivos, su definición, los campos de aplicación, su enfoque en EV y las variables de desempeño asociadas a éstos, lo cual es definido en la primera sección del capítulo. En la segunda sección, se presentan las metodologías implementadas en los modelos predictivos para la predicción de una variable de desempeño, los autores que han implementado este tipo de modelos y sus resultados. Así mismo, se explican las medidas de desempeño implementadas para evaluar el rendimiento del modelo predictivo. Finalmente, se presentan los vacíos de conocimiento que existen en cuanto a la implementación de una metodología para desarrollar un modelo predictivo junto con un paquete desarrollado en un lenguaje de programación que brinde las funciones necesarias para llevarlo a cabo.

3.1 Descripción general

Los modelos predictivos se pueden definir como el proceso en el que se crea y desarrolla un modelo matemático para tratar de representar el sistema a través de entradas y salidas, con el objetivo de predecir con buena precisión la probabilidad de un resultado (Kuhn & Johnson, 2013). El modelo predictivo usa formulaciones matemáticas, algoritmos de computación estadística, herramientas y servicios de software que permiten representar, interpretar y generar la predicción a partir de los datos (Dinov, 2018). El objetivo de un modelo predictivo es encontrar una aproximación útil que permita determinar una relación predictiva entre las variables de entrada y las variables de salida de un sistema analizado (Hastie et al., 2009). Además, este tipo de modelos tiene el objetivo de pronosticar tendencias, predecir patrones en los datos o pronosticar el comportamiento de un proceso (Dinov, 2018).

Un modelo se crea a partir del entendimiento de los datos que se realiza a través de un análisis exploratorio que permita identificar variables importantes para la modelación. Dentro del entendimiento también se podrían incluir otros análisis a partir de la evaluación de medidas de resumen simples o la identificación de variables que tengan correlaciones fuertes con el resultado, con un enfoque especial en encontrar relaciones entre las variables y el resultado (Kuhn & Johnson, 2019). Un modelo predictivo también puede obtener resultados a partir de un algoritmo de aprendizaje automático. Dentro de los algoritmos de aprendizaje automático, existen diferentes métodos para hacer las predicciones, tales como: aprendizaje supervisado, no supervisado y por refuerzo.

El aprendizaje supervisado implica la identificación de patrones entre unas variables de entrada que corresponden a los predictores y son definidas como *features* y unas variables de salida etiquetadas conocidas como *targets*. Estas variables de salida son las variables respuesta del modelo y sobre las cuales se realizan las predicciones (Dinov, 2018; Kuhn & Johnson, 2019). El aprendizaje supervisado implementa métodos para el análisis de la relación entre los *features* y el *target* por medio de la clasificación y la

regresión. El aprendizaje no supervisado se centra en la identificación de patrones entre las variables de entrada, es decir, las *features*. Este tipo de aprendizaje incluye métodos para el análisis como agrupación en clústeres, reducción de características, análisis de conglomerados y herramientas similares que permiten descubrir patrones en los datos (Kuhn & Johnson, 2019). El aprendizaje por refuerzo determina la forma en la que los agentes deben tomar decisiones en un entorno específico, con el fin de maximizar alguna noción de recompensa acumulativa, obteniendo una recopilación de experiencias que le permite mejorar las políticas para aumentar la recompensa (Levine et al., 2020).

Los modelos predictivos toman los patrones de comportamiento encontrados en los datos actuales e históricos para identificar riesgos y oportunidades, ayudando a tomar decisiones, brindando mayor confiabilidad en la predicción e incluso ayudando a disminuir costos (X. Hu, Deng, et al., 2021; Kuhn & Johnson, 2019). Por esto, los modelos predictivos se han implementado en diferentes campos de estudio tales como: comercio, economía, finanzas, medicina, educación, industria, transporte, entre otros, con el fin de desarrollar herramientas y servicios que permitan generar un sistema de toma de decisiones (Dinov, 2018; Ge et al., 2021; Kuhn & Johnson, 2019; Zraibi et al., 2021). A partir de los resultados del modelo predictivo, se puede extraer información importante y conocimiento basado en evidencia, identificando relaciones o asociaciones que permiten predecir resultados desconocidos, predecir probabilidades de estimación, generar clasificaciones o aportar predicciones agregadas o individuales en un campo de estudio determinado (Dinov, 2018).

3.2 Modelos predictivos del SOH y RUL

En el campo de estudio del transporte, especialmente con los EV, los modelos predictivos han contribuido en gran medida al desarrollo científico de esta importante área de investigación. Es así como en la última década se ha incrementado el número de estudios dedicados a la investigación de los EV y la predicción de sus componentes como la batería (87%), así como también en vehículos híbridos (13%). Los EV se pueden clasificar en dos categorías de acuerdo a su función, peso y capacidad, tales como vehículos pesados y medianos, y los vehículos ligeros y personales. Dentro de los vehículos pesados y medianos se encuentran los tranvías y trenes, camiones, autobuses y carros. La categoría de vehículos ligeros y personales se conforma por motos, patinetas y bicicletas. Dentro de los EV, sólo el 16% de estos artículos desarrolla la implementación en las dos categorías en las que se dividen los EV: los vehículos pesados y medianos (17%) y los vehículos ligeros y personales (83%) (Machedon-Pisu & Borza, 2019).

Según revisión de literatura, también se encuentra que la implementación de los modelos predictivos ha sido desarrollada a partir de datos públicos o datos recolectados por medio de una experimentación en laboratorio muy controlada, que puede ser muy costosa para su modelación, debido a los equipos, tiempo y personal que se requiere para llevarlos a cabo. Además, en este tipo de estudios no son consideradas las variables de

condiciones del tráfico, ambientales y topográficas en la experimentación (Z. Chen et al., 2021; Ge et al., 2021; Gou et al., 2021; Ji et al., 2021; Jiang et al., 2019; Jorge et al., 2020; Kara, 2021; Khumprom & Yodo, 2019; Listou Ellefsen et al., 2019; K. Liu et al., 2021; Pan et al., 2020; Qu et al., 2019; Ren et al., 2018; Shenghao, 2021; Sun et al., 2019; Tong et al., 2021; J. Wang et al., 2019; Xue et al., 2020; X. Zhang et al., 2022; Zhu et al., 2019; Zraibi et al., 2021). Por otra parte, en la revisión de literatura se encontró que Abdelaty et al. (2021) realizó un trabajo de investigación en campo, en el que tuvo en cuenta factores externos como topografía, comportamiento del conductor, condiciones climáticas y parámetros del tráfico, con el objetivo de identificar las variables que influyen en el modelo predictivo. Este tipo de experimentación en campo permite determinar la naturaleza de la relación entre las variables cuando están expuestas a cambios no controlables. Además, se identifican los parámetros significativos que afectan el desempeño de los EV en condición de operación real, y se obtienen resultados relevantes sobre el efecto en el desempeño del EV de factores externos que varían significativamente (Abdelaty et al., 2021).

Los EV tienen una combinación de diferentes sistemas, subsistemas y componentes, donde cada uno de estos sistemas interactúa entre sí para hacer que el EV funcione a través de diferentes tecnologías. Estas tecnologías pueden ser empleadas en los diferentes sistemas, donde el nivel de interacción varía según el subsistema al que pertenezca (Un-Noor et al., 2017). Un EV tiene sistemas y componentes críticos, tales como: el sistema de almacenamiento de energía (la batería), de tracción (el EM), de frenos, estructural, auxiliar, y el sistema eléctrico (Elwert et al., 2015). Este tipo de vehículos depende en gran medida de la energía almacenada en el paquete de baterías, ya que la batería proporciona la energía eléctrica al EM, para que éste genere la propulsión al vehículo (Xiong, 2020). Por lo tanto, la autonomía del vehículo depende directamente de la capacidad de la batería, ya que el desempeño de ésta afectará directamente la seguridad, la economía y el rendimiento dinámico del EV (Un-Noor et al., 2017; Xiong, 2020).

Así mismo, la batería determina la manejabilidad, seguridad y durabilidad del EV, lo cual, convierte a este componente en uno de los más críticos y principales dentro de sus sistemas (Hong et al., 2019). Existen diferentes tipos de baterías, tales como: plomo ácido, níquel-hierro, níquel-cadmio y iones de litio. Las LIB son un tipo especial de batería que se usa ampliamente en teléfonos celulares, dispositivos aeroespaciales, sistemas de almacenamiento de energía y EV debido a características como bajo costo, buena eficiencia, largo ciclo de vida, alta potencia y densidades de energía (Jo et al., 2021; Xiong et al., 2018).

Dentro de los sistemas que componen los EV, también se encuentra el BMS para que los componentes críticos puedan administrar de manera efectiva su confiabilidad, funcionamiento y seguridad, a través de la estimación precisa de su estado (Z. Chen et al., 2021a; Ge et al., 2021; X. Hu et al., 2020; Z. Wang et al., 2021). El BMS se encarga de monitorear el rendimiento de la batería y ajustar el sistema, por lo cual, también es considerado uno de los componentes críticos del EV (Xing et al., 2011). El BMS consiste de varios

sensores, actuadores y controladores de algoritmos y líneas de señal (Xiong, 2020). El BMS también puede controlar razonablemente el estado operativo y gestionar de manera inteligente cada unidad para prolongar la vida útil de los componentes críticos (J. C. Chen et al., 2021).

Cada uno de estos sistemas y componentes críticos de los EV tienen asociadas unas variables de desempeño que permiten conocer el estado del EV. Los modelos predictivos en EV se han desarrollado en torno a estas variables de desempeño asociadas a los sistemas y sus componentes críticos, con el fin de monitorear los indicadores necesarios para asegurar la confiabilidad del EV. En particular, los estudios se han enfocado en la estimación de dos estados: el estado de carga (SOC, por sus siglas en inglés: S*tate Of Charge*) y el SOH, los cuales están fuertemente correlacionados para mejorar la confiabilidad y garantizar la seguridad del componente crítico a través de la predicción del RUL (Z. Wang et al., 2021).

La estimación del SOC pretende verificar la capacidad restante de una batería durante un ciclo de carga-descarga, lo que evitaría la sobrecarga y la descarga excesiva (Z. Wang et al., 2021; Xu et al., 2020). La estimación del SOC se ve afectada por variables como: la tasa de carga, la temperatura, el ciclo de vida, la resistencia y la impedancia, entre otros parámetros (Xu et al., 2020). La estimación del SOC sirve para evaluar su comportamiento durante un ciclo de carga-descarga, por lo cual, es usado cuando se requiere de la evaluación de un ciclo específico, con el fin de validar el porcentaje de carga que tiene el vehículo para su correcto funcionamiento.

El SOC ha sido estudiado por Shenghao (2021), Wei et al. (2020) y Zhao et al. (2021), los cuales mencionan que el control preciso del SOC de la batería del EV es de gran importancia para la seguridad de la conducción y es clave para la gestión de baterías del EV. El SOC es estimado indirectamente por los parámetros físicos de la batería, y Shenghao (2021) propone un método basado en el filtro de Kalman con una transformación, que permita agregar ruido a los datos y generar un resultado adecuado para EV equipados con computadoras de vehículos de alto rendimiento. Xu et al. (2020) presentan una red neuronal para estimar el SOC de EV ligeros y personales, mejorando la precisión de la predicción y la optimización del modelo. Wei et al. (2020) proponen un método híbrido de redes neuronales que le permiten estimar el SOC de la batería con precisión, disminuyendo el error obtenido en comparación con un modelo base de una red neuronal.

Así mismo, una de las variables de desempeño asociadas a los EV es el SOH, el cual se usa para evaluar cuantitativamente el nivel de envejecimiento del componente crítico en términos de capacidad y resistencia interna (Ge et al., 2021; X. Hu et al., 2020; K. Liu et al., 2021; R. Yang et al., 2021; Zhou et al., 2016). La capacidad de la batería refleja la cantidad de energía que se puede almacenar en una batería completamente cargada y la resistencia interna representa el cambio de la curva de voltaje cuando se aplica una corriente de descarga (Ge et al., 2021). Debido a que la capacidad no se puede medir directamente, se usan métodos

de estimación para el SOH, los cuales sirven para predecir el RUL o los ciclos de carga-descarga restantes (X. Hu et al., 2019; W. Li et al., 2019; Xiong et al., 2018). La forma de expresar el SOH para un componente crítico de un EV en términos de capacidad es como se muestra en la ecuación (1) (Ge et al., 2021; Hosen et al., 2021; Qiao et al., 2020):

$$SOH = \frac{Capacidad\ o\ resistencia\ interna\ actual}{Capacidad\ o\ resistencia\ interna\ al\ inicio\ de\ la\ vida} * 100\% \quad (1)$$

Gou et al. (2021) proponen un novedoso método de ensamble para predecir el SOH de baterías, considerando el voltaje como un indicador clave de salud. Los autores aplican un análisis de correlación para validar las entradas al modelo predictivo y proponen una estructura de aprendizaje en conjunto para reducir los errores (Gou et al., 2021). Pan et al. (2020) consideran que el SOH es un factor clave en la predicción de la vida útil de una batería, ya que garantiza la operación segura del EV. Los autores realizan un comparativo entre diferentes métodos de filtro con el fin de analizar el rendimiento de los algoritmos, encontrando que un tipo de filtrado obtiene mayor precisión en la estimación, así mismo, recomiendan la implementación de otro tipo de filtro para obtener la estimación del SOH en menor tiempo (Pan et al., 2020). Burzyński & Kasprzyk (2021) proponen un proceso de regresión gaussiano para evaluar el SOH durante un ciclo de funcionamiento de la batería, encontrando una estimación de la degradación de la batería que permite hacer una predicción del RUL.

El RUL se refiere al número de ciclos o ciclos de carga-descarga antes de que el rendimiento o SOH se degrade hasta el punto de que el componente ya no pueda funcionar (Z. Chen et al., 2021a; W. Li et al., 2019; Si et al., 2011). Por esto, la predicción oportuna y precisa del RUL permite mejorar la durabilidad, confiabilidad y el mantenimiento del componente crítico a partir de indicadores de salud (X. Hu et al., 2020; Si et al., 2011). Así mismo, la predicción precisa del RUL de un componente crítico para un EV es importante para el diagnóstico de fallas de la batería y EM, y garantizar el suministro de energía y reducir el costo de operación (Ge et al., 2021; X. Hu et al., 2020). La predicción del RUL de un componente crítico puede determinar adecuadamente su capacidad de trabajo futuro y evitar inconvenientes causados por fallas repentinas (Ji et al., 2021). Así mismo, la predicción del RUL garantiza la confiabilidad operativa durante el ciclo de vida del componente crítico, permite un mantenimiento preventivo inteligente, reduce costos, aumenta la seguridad del sistema y extiende su vida útil (Gao et al., 2015; Qiao et al., 2020).

El RUL se puede obtener a partir de la ecuación (2), donde C_{actual} representa la capacidad real y C_{umbral} es el umbral de la capacidad cuando el componente llega al final de su vida útil (W. Li et al., 2019; Qiao et al., 2020). Generalmente, cuando el SOH se encuentra entre el 70% y 80%, se define que el componente

debe ser reemplazado (Ge et al., 2021; X. Hu et al., 2019, 2020; K. Liu et al., 2021; Xiong et al., 2018; R. Yang et al., 2021; X. Zhang et al., 2022).

$$RUL = C_{actual} - C_{umbral} \quad (2)$$

La predicción del RUL y la estimación de SOH del componente no pueden ser medidos directamente, por lo que es necesario tener información de variables propias de los sistemas del EV, variables asociadas al medio ambiente y variables de conducción (Abdelaty et al., 2021). Las variables que deben tomarse de los sistemas del EV son el ciclo de descarga, voltaje, corriente y temperatura para monitorear y reflejar el RUL, y cada uno de los estados en función de estas variables (X. Hu et al., 2019; Z. Wang et al., 2021). Algunos estudios han usado variables como temperatura ambiente, humedad relativa, cantidad de frenos, velocidad, entre otras, con el objetivo de capturar el comportamiento asociado al medio ambiente y al tipo de conducción (Abdelaty et al., 2021).

Actualmente, existen pocos estudios que combinan las variables operativas, topográficas, vehiculares y externas como entradas al modelo predictivo, esto hace que no se tenga en cuenta el impacto de estas variables en la predicción de una variable de desempeño asociada al EV. Teniendo en cuenta este vacío de conocimiento, es importante conocer las variables que se han estudiado de forma independiente y que deben ser consideradas en la definición de un estudio que las integre para validar el comportamiento y encontrar las relaciones entre ellas, las cuales, permitan obtener conocimiento sobre su relación. En la Tabla 1 se muestran los sistemas, las variables y los autores que han estudiado o han mencionado la importancia del estudio de estas variables para la implementación de modelos predictivos de las variables de desempeño asociadas a los EV.

Tabla 1

Sistemas y variables a considerar en un estudio experimental para predecir una variable de desempeño

Sistema	Variable	Autores de revisión
Operativas de los componentes (Batería, motor)	Temperatura, voltaje de carga/descarga, corriente de carga/descarga, tiempo de funcionamiento, ciclo	(Ardeshiri et al., 2020; Hosen et al., 2021; Z. Wang et al., 2021; R. Yang et al., 2021; Zhou et al., 2016)
Vehiculares	Vibraciones, ciclo de carga/descarga, velocidad, distancia recorrida	(Ardeshiri et al., 2020; Tong et al., 2021; Zhao et al., 2021)
Externas Ambiente	Presión, temperatura, humedad relativa	(Listou Ellefsen et al., 2019; Si et al., 2011; Zhao et al., 2021)
Conductor	Peso del conductor, comportamiento	(Abdelaty et al., 2021)

Estas variables se pueden monitorear a través de sensores ubicados en cada uno de los componentes y un sistema de adquisición de datos, que permita capturar los datos de las variables a través del tiempo y proporcionar una cantidad significativa de información. Esto permite realizar análisis, diagnóstico y pronóstico de la degradación de los componentes instrumentados (Gao et al., 2015; X. Hu et al., 2019; Z. Zhang et al., 2018). La relación entre las variables que deben ser monitoreadas respecto al SOH del componente y la predicción del RUL se muestran en la Figura 1. A partir de las variables de corriente, voltaje, temperatura, ciclo de carga/descarga, tiempo de funcionamiento, se puede calcular el SOH a partir de la capacidad. Cuando este SOH llega a un límite, definido según revisión de literatura entre el 70% y 80%, se indica que es el fin de la vida útil del componente, lo que permite hacer la predicción del RUL (Z. Chen et al., 2021a; Ge et al., 2021; X. Hu et al., 2020; Ji et al., 2021; Jorge et al., 2020; K. Liu et al., 2021; Xiong et al., 2018; Y. Yang, 2021; Z. Zhang et al., 2018; Zhou et al., 2016).

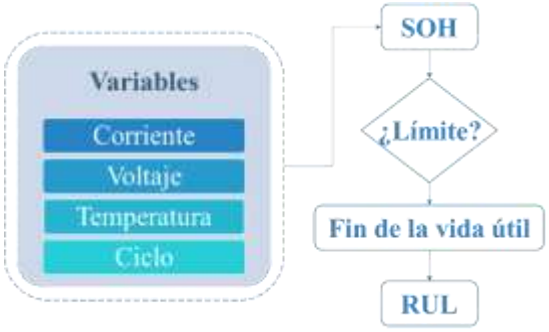


Figura 1. Relación entre las variables, estado de salud y predicción del RUL. Fuente: Elaboración propia a partir de (Ge et al., 2021)

En la Figura 2 se muestra el número de artículos publicados por cada una de las variables, en la que se destaca el número de artículos publicados para la variable del RUL y el SOH. Así mismo, se observa que, en los últimos años, las investigaciones sobre estas variables de desempeño de los EV han venido en aumento, lo que resalta la importancia de la investigación en este campo.

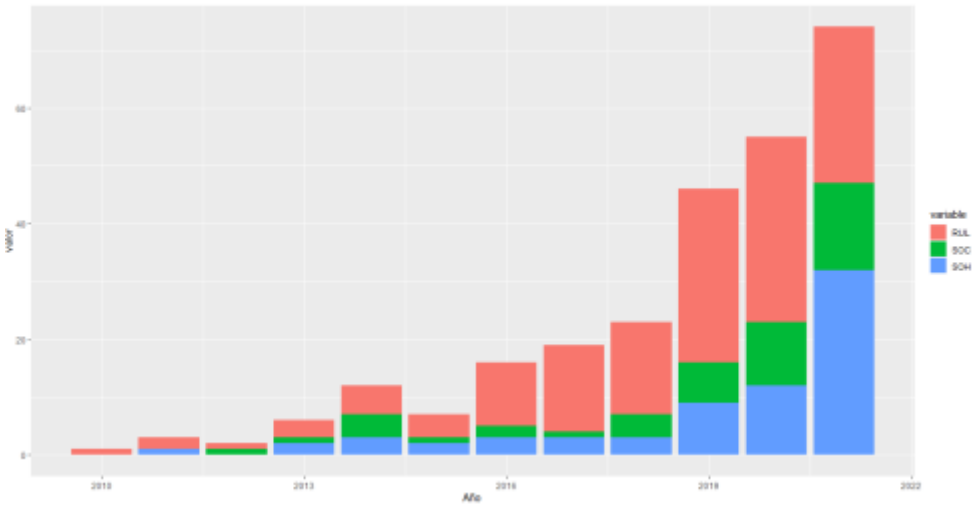


Figura 2. Publicaciones de RUL, SOH y SOC en los últimos once años. Fuente: Elaboración propia

3.3 Metodologías implementadas en modelos predictivos

Teniendo en cuenta que el SOC se implementa para la evaluación de un ciclo de carga-descarga específica, en esta sección se describen las metodologías implementadas para el desarrollo de modelos predictivos para las variables de desempeño SOH y RUL, las cuales permiten capturar el comportamiento en general de todos los ciclos de carga-descarga del EV. Para realizar estas estimaciones y predicciones de las variables, se han implementado diferentes tipos de modelos predictivos que permiten validar el comportamiento de la variable de desempeño y proporcionan información útil de su estado. La estimación del SOH puede realizarse a partir de diferentes metodologías, éstas se dividen en metodologías directas, las cuales usan alguna medida para su estimación, y las metodologías indirectas que se implementan a partir de diferentes modelos predictivos, las cuales se basan en modelos, en datos o metodologías híbridas que implementan una metodología basada en modelos con una basada en datos, o dos metodologías basadas en datos.

La estimación del SOH se puede realizar por medio de una medición directa a partir de la medición de la resistencia interna, la impedancia u otros métodos directos como el conteo de ciclos, o un análisis indirecto por medio de métodos basados en modelos, en datos o métodos híbridos (Ge et al., 2021; Qiao et al., 2020; Si et al., 2011). La medición directa requiere de una gran cantidad de experimentos llevados a cabo en el laboratorio para analizar el comportamiento del envejecimiento de los componentes críticos. Para realizar estos experimentos, se requiere de equipo experimental costoso y pruebas limitadas que no tienen en cuenta las condiciones reales de conducción, por lo cual, es un método difícil de llevar a cabo (Xiong et al., 2018). El SOH también se puede estimar a partir del conteo de eventos de carga y descarga o de ciclos para la LIB,

por lo cual, este método es considerado simple y directo, ya que sólo es necesario registrar el conteo de los ciclos. Sin embargo, es necesario que el fabricante proporcione el número total de ciclos de operación de la LIB, y que conozca la cantidad de ciclos que han experimentado, para estimar el SOH de estos componentes críticos (Xiong et al., 2018).

El análisis indirecto se puede llevar a cabo a partir de métodos basados en modelos, tales como: circuitos equivalentes, modelos electroquímicos y modelos combinados. Este tipo de modelamiento requiere de conocimientos en los principios de funcionamiento de los componentes eléctricos y un modelo de batería explícito (Z. Chen et al., 2021a; Xiong et al., 2018). Estos métodos se enfocan en analizar la disminución del rendimiento del componente en función de las propiedades de los materiales, de las reacciones electroquímicas, y del cambio de la impedancia que permita establecer un modelo de estimación del SOH (Z. Chen et al., 2021a). Sin embargo, debido a los complejos procesos físicos y químicos de la LIB y a la intervención de diferentes parámetros que pueden afectar el rendimiento de estos componentes, los métodos basados en modelos físicos no son ideales para situaciones prácticas (Z. Chen et al., 2021a; Fan et al., 2019).

En la Figura 3 se muestran las metodologías que se implementan para la estimación del SOH, las cuales se dividen en metodologías directas y metodologías indirectas. El 23% de los autores, según revisión de literatura, han realizado la estimación del SOH por medio de metodologías directas de medición. El 77% restante de revisión de literatura, se encuentra en la estimación del SOH a partir de medidas indirectas que se basan en modelos, datos o metodologías híbridas (Ge et al., 2021). Las metodologías implementadas en cada una de éstas se pueden observar en la Figura 3.

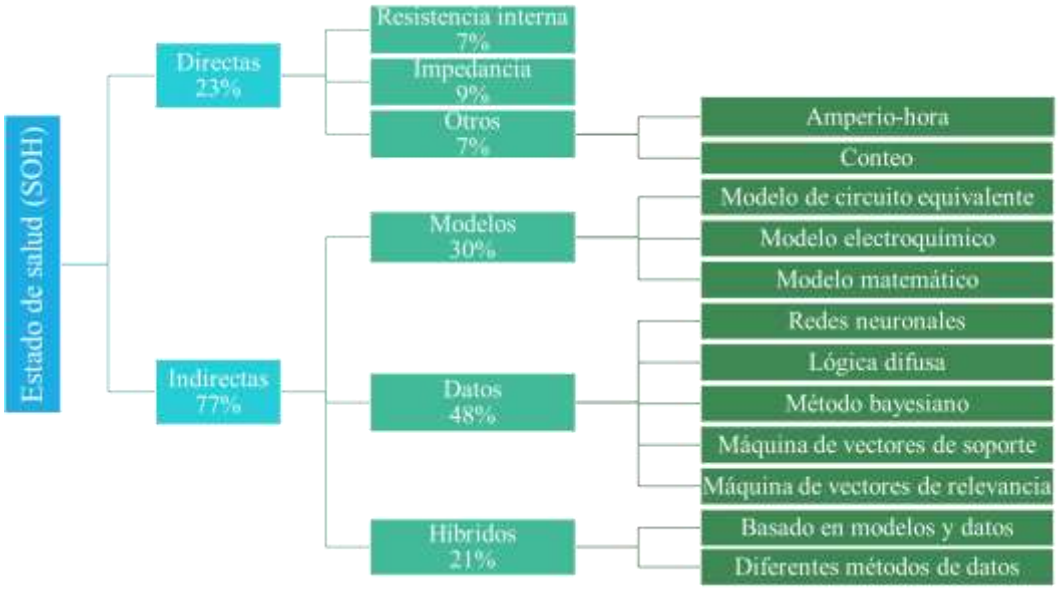


Figura 3. Metodologías implementadas para estimar el SOH. Fuente: Elaboración propia a partir de (Ge et al., 2021)

Para la predicción del RUL, también se hace uso de metodologías indirectas basadas en datos e híbridas que permiten generar el modelo predictivo para determinar la capacidad de la batería a través del tiempo. El análisis indirecto a partir de métodos basados en datos depende de los datos recolectados de la degradación de los componentes críticos, por lo que se requiere de experimentos de varios meses y que pueden ser costosos, debido a los recursos necesarios para llevarlos a cabo (Xiong et al., 2018). Este tipo de análisis se puede realizar a partir de métodos empíricos y de ajuste, algoritmos de optimización o *machine learning*, entre otros (Xiong et al., 2018). Los métodos basados en datos son los métodos que más se han estado implementando para la estimación del SOH y la predicción del RUL, debido a su potencial informático y capacidad de procesamiento (Z. Chen et al., 2021a; Ge et al., 2021; Xiong et al., 2018). Estos métodos se pueden dividir en series de tiempo, inteligencia artificial y métodos basados en filtros (Z. Chen et al., 2021a). Los métodos de serie de tiempo para la predicción del RUL que más se implementan son el modelo de auto regresión (AR, por sus siglas en inglés: A*utoregression*) y sus variantes tales como el modelo autorregresivo de media móvil (ARMA, por sus siglas en inglés: *Autoregressive Moving Average*) que consiste en un modelo de AR y un promedio móvil (MA, por sus siglas en inglés: *Moving Average*) (Ge et al., 2021; X. Hu et al., 2020). Así mismo, dentro de los métodos de series de tiempo se implementa el modelo de promedio móvil integrado autorregresivo (ARIMA, por sus siglas en inglés: *Autoregressive Integrated Moving Average*), que se desempeña bien con series de tiempo estacionarias no lineales (Ge et al., 2021; X. Hu et al., 2020). Sin embargo, estos métodos sólo consideran una variable para describir la degradación de la batería a través de la capacidad (Ge et al., 2021; X. Hu et al., 2020).

Los métodos basados en filtro se implementan para realizar seguimiento al SOH y predecir la degradación a partir de métodos comunes como el algoritmo de filtro de Kalman (KF, por sus siglas en inglés: *Kalman Filter*), el algoritmo de filtro de partículas (PF, por sus siglas en inglés: *Particle Filter*), y un algoritmo que combina ambos métodos (Ge et al., 2021; X. Hu et al., 2020). Estos métodos pertenecen a la familia de filtrado bayesiano y ofrecen un marco general para la predicción de la degradación de la batería (Ge et al., 2021; X. Hu et al., 2020).

Los métodos basados en inteligencia artificial enfocado en *machine learning* son: máquina de vectores de soporte (SVM, por sus siglas en inglés: *Support Vector Machine*), máquina de vectores de relevancia (RVM, por sus siglas en inglés: *Relevance Vector Machine*), regresión del proceso gaussiano (GPR, por sus siglas en inglés: *Gaussian Process Regression*), lógica difusa (FL, por sus siglas en inglés: *Fuzzy Logic*), red neuronal artificial (ANN, por sus siglas en inglés: *Artificial Neural Network*), y agrupación de medias C difusas (FCM, por sus siglas en inglés: *Fuzzy c-Means*) (Z. Chen et al., 2021a; Fan et al., 2019; Hosen et al., 2021; X. Hu et al., 2019; Qiao et al., 2020).

En la Figura 4 se presentan las metodologías implementadas para la predicción del RUL, en la que se destacan las herramientas basadas en datos como las series de tiempo, la inteligencia artificial y las metodologías basadas en filtro. Las metodologías basadas en datos tienen un porcentaje de participación del 80% según revisión de literatura. El 20% de participación restante, se implementa bajo metodologías híbridas que se observan en la Figura 4.

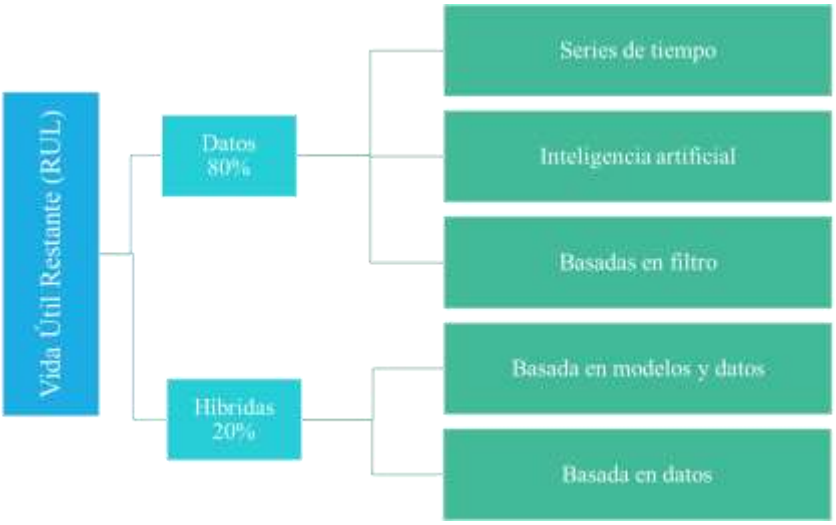


Figura 4. Metodologías implementadas para predecir el RUL. Fuente: Elaboración propia a partir de (Ge et al., 2021)

Los métodos basados en datos permiten que los algoritmos aprendan y mejoren sus predicciones a partir de los datos, los cuales se pueden obtener en tiempo real, y con esto, hacer predicciones en línea sobre el RUL (Z. Chen et al., 2021a; Qu et al., 2019). En los últimos años, los métodos híbridos se han venido generalizando, ya que al fusionar las ventajas de múltiples algoritmos se obtiene mayor precisión en las predicciones (Z. Chen et al., 2021a; Ge et al., 2021). Generalmente, los métodos híbridos más usados son aquellos que combinan dos tipos de métodos basados en datos, los cuales pueden mejorar las dificultades de modelamiento y también permiten mejorar la adaptabilidad deficiente y los problemas menos funcionales del uso de un solo método basado en datos (Z. Chen et al., 2021a; Ge et al., 2021).

En los últimos años, muchos investigadores han implementado métodos basados en datos y métodos híbridos para predecir el RUL y monitorear el SOH. En estos estudios, se destaca el uso de los métodos híbridos, ya que han tenido un crecimiento en los últimos años debido a su capacidad de precisión en la predicción del RUL y capacidad de procesamiento no lineal (Z. Chen et al., 2021a; Kara, 2021; Tong et al., 2021; Y. Yang, 2021). Los métodos híbridos más implementados para la predicción del RUL de un componente crítico están relacionados con inteligencia artificial, tales como: GPR, RVM, red neuronal convolucional (CNN, por sus siglas en inglés: *Convolutional Neural Network*), red neuronal profunda (DNN, por sus siglas en inglés: *Deep Neural Network*), red neuronal a corto y largo plazo (LSTM, por sus siglas en inglés: *Long-Short Term Memory*), variaciones de la LSTM, entre otros (Z. Chen et al., 2021a;

Hosen et al., 2021; X. Hu, Yang, et al., 2021; Lee et al., 2021; K. Liu et al., 2021; Tong et al., 2021; Y. Yang, 2021; Zraibi et al., 2021). En la Tabla 2 se muestran las ventajas y desventajas de cada una de las metodologías implementadas en los modelos predictivos.

Tabla 2

Ventajas y desventajas de las metodologías

Metodología	Clasificación de métodos	Ventajas (V) y desventajas (D)
Basado en modelos	Modelo electroquímico	**V:** Alta precisión, habilidad de generalización y considera el proceso electroquímico interno. **D:** Dificultad para establecer o identificar parámetros del modelo, requiere conocimiento experto, altos costos computacionales.
	Circuitos equivalentes	**V:** Describe eficazmente la tendencia de crecimiento de resistencia interna y el mecanismo de envejecimiento. **D:** Afectado fácilmente por condiciones externas, requiere un equipo de medición costoso, su prueba acelera el proceso de envejecimiento.
Basado en datos	ANN	**V:** Aprendizaje fuerte y capacidad de procesamiento no lineal. **D:** Estructura compleja y requiere de muchos datos.
	Máquina de vector de soporte (SVM)	**V:** Trata eficazmente los mínimos locales. **D:** No lineal y problemas de muestra, carece de escasez.
	Máquina de vector de relevancia (RVM)	**V:** Mejora la escasez, evita el sobreajuste. **D:** Carece de estabilidad.
	Regresión de procesos gaussianos (GPR)	**V:** Trata eficazmente con conjuntos de datos de muestra pequeña. **D:** Necesita mucha computación para grandes conjuntos de datos.
Métodos híbridos	Basado en modelo + Basado en datos	**V:** Formula exacta para describir el sistema con alta precisión. **D:** El cálculo es complejo y difícil encontrar el modelo que represente la situación real.
	Dos métodos basados en datos	**V:** Alta precisión, simula mejor la tendencia a la degradación. **D:** El modelo es complejo y requiere cálculo.

La red LSTM ha sido ampliamente usada para la predicción del RUL debido a que es buena para la predicción de series temporales (Park et al., 2020; Ren et al., 2018; J. Wang et al., 2019; Xia et al., 2020). La red LSTM tiene la capacidad de capturar a largo plazo las dependencias entre las capacidades de degradación de la batería, por lo que tiene buena aplicabilidad para la predicción del RUL de la batería (Ge et al., 2021; X. Hu, Yang, et al., 2021). Para la predicción del RUL, se han implementado diferentes modelos

predictivos basados en la red LSTM y la combinación de la red LSTM con otro tipo de metodologías para una mejor precisión. En los últimos años, el método híbrido CNN–LSTM ha sido usado con éxito en diferentes problemas de clasificación y predicción en campos como la agricultura y medicina (Zraibi et al., 2021).

Para la evaluación del rendimiento del modelo predictivo, se implementan diferentes medidas de desempeño, tanto gráficas como métricas cuantitativas estadísticas. Estas métricas son de gran importancia, ya que permiten evaluar la efectividad del modelo predictivo y generar conclusiones sobre su rendimiento (Kuhn & Johnson, 2019). Una métrica de desempeño incorrecta no permite evaluar adecuadamente el rendimiento de un modelo predictivo y como consecuencia, mejorar el modelo (Pawlus & Devine, 2020). Las definiciones y ecuaciones de cada una de las métricas mencionadas en esta sección, se explican en el Anexo 1.

Las herramientas gráficas permiten crear visualizaciones simples de un modelo predictivo para evaluar su rendimiento (Kuhn & Johnson, 2013). Una de las herramientas gráficas más usadas para evaluar el desempeño de un modelo predictivo de regresión, es la gráfica de valores observados o reales versus los valores predichos, con el fin de descubrir las áreas de los datos donde el modelo predictivo funciona bien o mal (Kuhn & Johnson, 2013). Este tipo de información cualitativa es fundamental para mejorar los modelos predictivos, ya que estas visualizaciones permiten observar la distancia entre la gráfica de valores reales versus valores predichos, donde lo ideal es que la distancia entre cada una de las gráficas sea reducida (Kuhn & Johnson, 2013).

Según Kuhn & Johnson (2013, 2019), cuando se predicen valores numéricos, la raíz del error cuadrático medio (RMSE, por sus siglas en inglés: *Root-Mean-Square Error*) es el método más común e implementado para caracterizar las capacidades predictivas de un modelo y evaluar su desempeño. El RMSE es una métrica en función de los residuos del modelo, que corresponde a los valores observados menos las predicciones del modelo predictivo (Kuhn & Johnson, 2013, 2019). El RMSE se puede interpretar como la distancia promedio de una muestra desde su valor observado hasta su valor predicho, es decir, cuanto menor sea el RMSE, mejor podrá predecir un modelo los resultados de las muestras (Kuhn & Johnson, 2019). Así mismo, una de las métricas para la evaluación del desempeño de un modelo predictivo es el error cuadrático medio (MSE, por sus siglas en inglés: *Mean Squared Error*), el cual se calcula elevando al cuadrado los residuos y sumándolos. El MSE es insumo principal para calcular el RMSE, ya que el RMSE es la raíz cuadrada del MSE. El resultado obtenido del MSE puede ser interpretado como la distancia promedio entre los valores observados y las predicciones obtenidas con el modelo (Kuhn & Johnson, 2013).

Kutkina & Feuerriegel, (2016) indican que una medida de desempeño común para la regresión también es el error absoluto medio (MAE, por sus siglas en inglés: *Mean Absolute Error*). Esta métrica es simple de interpretar y es útil cuando no hay necesidad de aplicar penalización adicional a errores grandes (Pawlus & Devine, 2020). Un ejemplo para determinar si se requiere de esta métrica para evaluar el desempeño del modelo, es validando si un error que es tres veces más grande que otro error y a su vez, es tres veces más malo que este error, entonces el MAE es una buena métrica para implementar (Pawlus & Devine, 2020). Sin embargo, hay diferentes casos en los que un error que es tres veces mayor que otro error, genera más complicaciones en el modelo predictivo, por lo cual debe agregarse una penalización adicional (Pawlus & Devine, 2020). Otra de las métricas para la evaluación del desempeño del modelo predictivo es el error de porcentaje medio absoluto (MAPE, por sus siglas en inglés: *Mean Absolute Percentage Error*), el cual, calcula el porcentaje de error respecto a los valores reales, obteniendo un valor ponderado y una medida más estandarizada que no depende de los rangos de valores de los datos.

Teniendo en cuenta estas métricas de desempeño para la evaluación de los modelos predictivos, se presentan los autores que han investigado e implementado diferentes modelos predictivos para la predicción del RUL. Zraibi et al. (2021) proponen un método híbrido que combina una CNN con una red LSTM y una DNN para hacer la predicción del RUL y evaluar el desempeño a través del RMSE y del coeficiente de correlación. Liu et al. (2021) proponen un enfoque basado en una descomposición de modo empírico (EMD, por sus siglas en inglés: *Empirical Mode Decomposition*) para descomponer el conjunto de datos de capacidad original, capturar la dependencia a través de una red LSTM y generar la incertidumbre de cada resultado de la predicción a través de una GPR, evaluando el desempeño del modelo a partir del RMSE. J. C. Chen et al. (2021) proponen un modelo basado en EMD con una red LSTM y agrega un análisis relacional gris para extraer las características de los datos, predecir el RUL y evaluar el desempeño a través del RMSE, el MAE y el MAPE.

Z. Chen et al. (2021) mencionan la importancia de la predicción del RUL para los EV e implementan un algoritmo híbrido que combina un sistema de aprendizaje amplio con una máquina de vectores de relevancia. Tong et al. (2021) implementan un algoritmo de predicción híbrido del RUL llamado ADLSTM-MC que combina una red LSTM optimizada que proporciona una dirección de actualización correcta y rápida, asegurando que el modelo se optimice continuamente. A este método se añade el método de abandono adaptativo basado en el algoritmo de optimización bayesiano, el cual proporciona un valor de abandono apropiado para entrenar el modelo con las características de la memoria adaptativa a corto y largo plazo y evalúan su desempeño a través del RMSE y el MAE.

X. Hu, Yang, et al. (2021) proponen un enfoque híbrido para la predicción basada en el filtro de partículas para actualizar iterativamente los parámetros del modelo y el uso de la red LSTM con abandono de Monte Carlo para la predicción del RUL y evaluación del desempeño a través del RMSE. Kara (2021) presenta un enfoque híbrido que combina una CNN y una red LSTM para extraer las relaciones, capturar las características no lineales y lograr una mejor precisión en la predicción del RUL, mejorando el desempeño, el cual es evaluado a través del RMSE y el MSE. Ren et al. (2021) proponen un método híbrido de predicción del RUL para una LIB llamado Auto-CNN-LSTM basado en una CNN mejorada que extrae la información en datos limitados y usa la capacidad de la red LSTM para procesar la información de serie de tiempo, evaluando su desempeño a partir del RMSE.

Sin embargo, en la mayoría de los trabajos de investigación, sólo se considera la variable de capacidad para hacer la predicción del RUL a partir de los modelos predictivos. De esta forma, no se consideran variables como voltaje, corriente y temperatura, las cuales son las encargadas de capturar el comportamiento de la batería a lo largo de los ciclos. Así mismo, los trabajos de investigación han validado su modelo predictivo propuesto a partir de conjuntos de datos de baterías bajo condiciones controladas, en las que se mantenía la temperatura ambiental de forma constante (Saha & Goebel, 2007). Considerando lo anterior, en las investigaciones en conjunto, no se ha estructurado un modelo predictivo que disminuya el ruido de las variables obtenidas por sensores, tales como: corriente, voltaje, temperatura a través de una ANN. El ruido de estas variables puede ser suavizado por medio de una CNN, que generará como salidas las variables suavizadas que serán el insumo principal para el modelo predictivo por medio de una red LSTM. Así mismo, en los trabajos de investigación, los modelos predictivos se han aplicado en datos de experimentación controlada, sin considerar los efectos de variables exteriores que permitan un análisis bajo condiciones de tráfico real.

Para la evaluación de los modelos predictivos bajo alguna de las metodologías mencionadas anteriormente, se construye una tabla con las principales medidas de desempeño que sirven para la evaluación de estos modelos predictivos. En la Tabla 3 se observan las medidas de desempeño implementadas para la evaluación de un modelo predictivo con sus respectivos porcentajes de participación de acuerdo a la revisión de literatura realizada. Así mismo, se presentan los límites para evaluar cada una de las medidas de desempeño de acuerdo a la apreciación de diferentes autores en los modelos predictivos implementados para la predicción del RUL.

Tabla 3

Límites de evaluación de las medidas de desempeño de un modelo predictivo

Medida de desempeño	% Participación en revisión	Rango	Límites aceptables	Autores
RMSE	54%	[0, ∞]	0 – 1.5	(Z. Chen et al., 2021a; Fan et al., 2019; Ge et al., 2021; X. Hu, Yang, et al., 2021; Kara, 2021; K. Liu et al., 2021; Park et al., 2020; Ren et al., 2018; Zraibi et al., 2021)
MAE	27%	[0, ∞]	< 29	(Z. Chen et al., 2021a; Ge et al., 2021; Jo et al., 2021; Park et al., 2020; Zraibi et al., 2021)
MSE	12%	[0, ∞]	< 0.9	(Ge et al., 2021; Jo et al., 2021; Kara, 2021)
MAPE	8%	[0%, >100%]	< 25%	(Z. Chen et al., 2021a; Ge et al., 2021; Jo et al., 2021; Park et al., 2020)

Estos modelos predictivos se pueden implementar en diferentes lenguajes de programación tales como Python y R, los cuales son lenguajes de uso libre desarrollados para el desarrollo de software, computación estadística y gráficos. Cada uno de estos lenguajes ha desarrollado diferentes paquetes y librerías que permite la implementación de modelos predictivos en diferentes contextos, ver Anexo 2 para conocer las librerías que existen actualmente en estos lenguajes de programación. Diferentes autores mencionan el uso del lenguaje de programación Python para llevar a cabo su modelo predictivo, donde algunos también mencionan el uso de `Tensorflow, PyTorch o Keras` para el diseño de la arquitectura de la ANN (X. Hu, Yang, et al., 2021; Jo et al., 2021; Mejia et al., 2021; Park et al., 2020; Zraibi et al., 2021).

Humphrey et al. (2017) desarrollaron el paquete `validann` en R que ofrece funciones para acceder de manera legible a los métodos de validación de las ANN, en la que se destaca la función *ann* para la creación de una ANN más flexible a la implementada en el paquete `nnet`. Ozkok & Celik (2022) desarrollaron un paquete en R que implementa un modelo CNN-LSTM, la CNN es usada para la extracción de características y la red LSTM es usada para la clasificación, por lo que el enfoque del modelo propuesto es la clasificación de la variable respuesta. En general, las librerías ofrecen funciones con las que se pueden crear redes neuronales para realizar los modelos predictivos, sin embargo, según revisión de literatura, no se encontró una librería, paquete o archivo que muestre los pasos necesarios para llevar a cabo los modelos predictivos

propuestos en los trabajos de investigación mencionados anteriormente. Esto muestra un vacío de conocimiento y una oportunidad para desarrollar una librería o paquete que integre una metodología completa e implemente los pasos necesarios para llevar a cabo una división de datos, un modelo de suavizamiento y que permita generar un modelo predictivo con predicciones más precisas.

Teniendo en cuenta esta oportunidad, se desarrolla un paquete en lenguaje de programación R, el cual, permite a los usuarios implementar la metodología propuesta descrita en este trabajo de investigación de una manera eficiente y fácil de usar. Se selecciona el lenguaje de programación R, ya que es gratuito, de código abierto y permite la distribución simple, el uso y mantenimiento del código (Humphrey et al., 2017). Además, el paquete permite a los usuarios realizar manipulaciones y adaptaciones de las funciones según sea necesario. De esta forma, se plantean en los siguientes capítulos la explicación de las redes neuronales, el desarrollo de la metodología para llevar a cabo el modelo predictivo propuesto y el paquete en lenguaje de programación R, el cual, integra las funciones necesarias para llevar a cabo la metodología.

4. REDES NEURONALES

En este capítulo se realizará la descripción general de las redes neuronales, desde su definición, sus funciones, ecuaciones y los hiperparámetros asociados a una ANN. De esta forma, se enfocará el capítulo en las dos redes que se implementan en el desarrollo del trabajo de investigación, la CNN y la red LSTM, realizando la definición de cada una, mostrando los parámetros, los hiperparámetros asociados y los valores de referencia que indican algunos de los autores, así como también, la implementación en el contexto de predicción del RUL.

4.1 Descripción general

La ANN surgió en 1943 con el desarrollo de teorías de aprendizaje biológico con Warren McCulloch y Walter Harry Pitts (Goodfellow et al., 2016). En 1949, Hebb da lugar a la teoría Hebbiana en la que el aprendizaje se localiza en las conexiones entre las neuronas. En 1958, Rosenblatt implementa el primer modelo conocido como el perceptrón que permite el entrenamiento de una sola neurona. Con estas teorías, se definió la primera ola de las redes neuronales y fue llamada la ola de la cibernética que se desarrolló entre los años 1940 y 1960. La segunda ola se llamó conexionismo y se desarrolló entre los años 1980 y 1995, en esta ola, Rumelhart et al. en 1986 presentaron la retropropagación, con el objetivo de entrenar una red neuronal con una o dos capas ocultas. Finalmente, la tercera ola es la actual, y se conoce como la ola del aprendizaje profundo que inició alrededor del 2006 con autores como Hinton et al., (2006), Bengio et al., (2007) y Ranzato et al., (2007) (Goodfellow et al., 2016).

Es así como las redes neuronales tratan de simular el mecanismo de aprendizaje de organismos biológicos a través de la imitación de una neurona biológica (Aggarwal, 2018). Una ANN está compuesta por nodos también conocidos como neuronas, las cuales son las unidades básicas de procesamiento de las redes neuronales. Estos nodos se organizan en capas, donde la primera capa se llama capa de entrada, la última se llama capa de salida y las capas del medio se conocen como capas ocultas (Nasiriany et al., 2019). La ANN simple es conocida como perceptrón y se caracteriza por tener una sola capa de entrada y un nodo de salida (Aggarwal, 2018). Las ANN pueden ser multicapas, donde las neuronas están dispuestas en capas, y su arquitectura es conocida como una red feed-forward (Aggarwal, 2018). En la Figura 5 se presenta la estructura del proceso que se realiza dentro de una neurona a través de la transformación geométrica de los datos. En esta se consideran cada una de las entradas, los pesos asociados a las conexiones de las entradas y las capas ocultas, así como la función de activación que define el valor de la salida. El sesgo (bias en inglés) del modelo refleja la capacidad del modelo predictivo para ajustarse a la estructura teórica subyacente de los datos (Kuhn & Johnson, 2019).

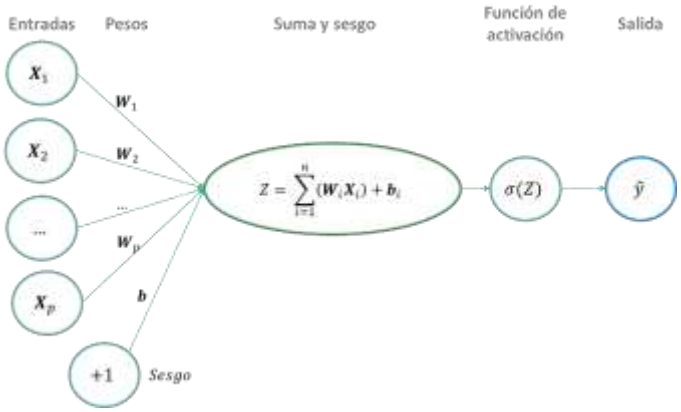


Figura 5. Transformación geométrica de los datos que realiza una capa oculta. Fuente: Elaboración propia a partir de (Ghatak, 2019)

La notación en la Figura 5 significa (Aggarwal, 2018; Ghatak, 2019; Skansi, 2018):

- La capa de entrada contiene p número de nodos representados por la matriz de entradas: $\boldsymbol{X} = \boldsymbol{X}_1, \boldsymbol{X}_2, \dots, \boldsymbol{X}_p$, los cuales, corresponden al número de *features* de entrada a la ANN.
- Los pesos de cada una de las conexiones de entrada a un nodo de la capa oculta representados por la matriz de pesos: $\boldsymbol{W} = \boldsymbol{W}_1, \boldsymbol{W}_2, \dots, \boldsymbol{W}_p$.
- Z que representa la ecuación de transformación geométrica de los datos.
- El sesgo con peso $\boldsymbol{b}$ representa el vector del sesgo obtenido y es sumado a la ecuación Z.
- σ representa la función de activación que se debe aplicar a la ecuación Z.
- $\hat{y}$ representa el valor a predecir.

Dentro de cada uno de los nodos de las capas ocultas de la ANN, se realizan funciones de transformación geométrica de los datos a través de formulación (Ghatak, 2019). En la red, cada neurona tiene conexiones de entrada asociadas a las *features*, y cada conexión tiene asociada un peso que permite realizar cálculos internos a través de diferentes funciones que generan valores de salida en las conexiones a las siguientes neuronas (Skansi, 2018). De esta forma, la información se va procesando desde la primera capa de entrada, pasando por las capas ocultas que se tengan definidas y realizando una función que determina según los pesos de las conexiones, las entradas y un sesgo, cuál es el valor que adquiere ese nodo o neurona dentro de la capa oculta. Donde el sesgo estadísticamente se describe como el grado en el que un valor se desvía de lo esperado (Kuhn & Johnson, 2019).

Después de este proceso, el valor obtenido en el nodo, pasa por una función de activación que permite brindar el resultado final del modelo predictivo. Si una de las salidas se ha activado con mayor frecuencia, entonces a esta salida se le asigna un mayor peso y a las demás salidas menor peso, con el fin de que la suma

de las salidas dé un valor a 1. Así, la red neuronal artificial calcula la función de entrada propagando los valores calculados desde las neuronas de entrada a las neuronas de salida, usando los pesos como parámetros intermedios en las capas ocultas (Aggarwal, 2018). De esta forma, la predicción de $\hat{y}$ se calcula de acuerdo a la ecuación (3) (Aggarwal, 2018):

$$\hat{y} = \sigma\left(\sum_{i=1}^{n}(\boldsymbol{W}_i\boldsymbol{X}_i) + \boldsymbol{b}_i\right) \quad (3)$$

Cada una de estas funciones de transformación geométrica de los datos se realiza en cada uno de los nodos que se tienen en las capas ocultas de la arquitectura de la ANN. La capa de entrada tiene p números de nodos que corresponden al número p *features* de entrada a la red neuronal. El número de capas ocultas y el número de nodos o neuronas de cada capa oculta es definido por la persona que está diseñando la red neuronal (Ghatak, 2019; Skansi, 2018). El número de nodos de la capa de salida corresponde al número de salidas o *target* que se requiera en la situación analizada, en este caso, si se requiere de la predicción de un número, se tendrá un nodo en la capa de salida que corresponderá al *target*. En la Figura 6 se observa el esquema de los nodos de una ANN con p *features* como vectores en la capa de entrada, la primera capa oculta es representada por $h_k^{[1]}$ y un *target* en la capa de salida representado por $\hat{y}$.

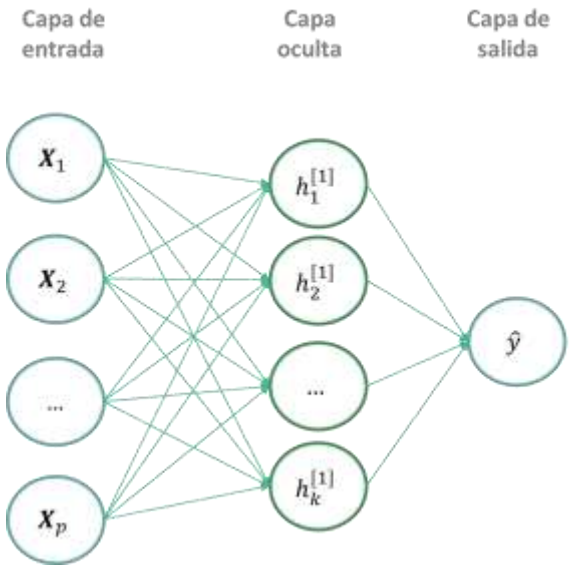


Figura 6. Nodos de una red neuronal con tres entradas, cuatro nodos de una capa oculta y una capa de salida. Fuente: Elaboración propia a partir de (Ghatak, 2019)

La notación usada en la Figura 6 indica (Ghatak, 2019; Skansi, 2018):

- Capa de entrada representada por el vector de p *features* $\boldsymbol{X}_p$, donde el vector de entradas se puede representar así: $\boldsymbol{X}_p^i \in \mathbb{R}^{p_x}$ (p *features*):

- $\boldsymbol{X}_p^i$: Representa el vector de las observaciones de una p *feature* específica.
- i: Hace referencia a la i-ésima observación. Por ejemplo, $\boldsymbol{X}^1$, corresponde al valor de entrada de la primera observación. Donde $i = 1, 2, \dots, n$.
- p: Hace referencia al p-ésimo número de *features.* Por ejemplo, $\boldsymbol{X}_2^1$, corresponde a la primera observación de la segunda *feature.*

- Capas ocultas representadas a partir de $h_k^{[l]}$:
  - l: Hace referencia a la l-ésima capa oculta. Por ejemplo, $h^{[1]}$, corresponde a la primera capa oculta de la red neuronal.
  - k: Hace referencia al k-ésimo nodo de una capa oculta. Por ejemplo, $h_1^{[1]}$, corresponde al primer nodo de la primera capa oculta de la red neuronal.
- Capa de salida representada por $\hat{y}$ como *target.*

De esta forma, se considera un conjunto de ecuaciones lineales con p *features* de acuerdo a la ecuación (4) (Abhijit Ghatak):

$$y^{(i)} = w_0 + w_1 x_1^{(i)} + w_2 x_2^{(i)} + \cdots + w_p x_p^{(i)} + \in_1^{(i)} \ para\ i = 1, \dots, n \qquad (4)$$

Donde cada una de estas ecuaciones lineales pueden ser representadas por medio de la siguiente notación matricial:

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} 1 & x_1^{(1)} & \cdots & x_p^{(1)} \\ 1 & x_1^{(2)} & \cdots & x_p^{(2)} \\ \vdots & \vdots & \cdots & \vdots \\ 1 & x_1^{(n)} & \cdots & x_p^{(n)} \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_p \end{bmatrix} + \begin{bmatrix} \epsilon_0 \\ \epsilon_1 \\ \vdots \\ \epsilon_p \end{bmatrix}$$

En este caso, así como también en el aprendizaje automático, en las redes neuronales existen tres tipos de errores que están incluidos dentro del modelo predictivo (Ghatak, 2019):

1. **Error irreducible:** Es el error que no se puede reducir y su ocurrencia se debe a la aleatoriedad inherente presente en los datos.
2. **Error de sesgo (*b*):** Este error aparece cuando se introduce un modelo más simple para un conjunto de datos más complejo. Para evitar este error, se considera aumentar la capacidad o complejidad del modelo.
3. **Varianza:** Este error aparece cuando se introduce un modelo más complejo para un conjunto de datos menos complejo. Para evitar este tipo de error, se aumenta el conjunto de datos.

Dentro de la red neuronal, los pesos y el error son parámetros que se ajustan de acuerdo a la arquitectura definida y el valor de los hiperparámetros dentro del entrenamiento del modelo. Los hiperparámetros son

los parámetros que no se ajustan dentro de la calibración del modelo y que deben ser configurados antes de iniciar el entrenamiento del modelo predictivo (Elgeldawi et al., 2021; L. Yang & Shami, 2020). Los hiperparámetros indican cómo se estructura el modelo para iniciar su proceso de calibración. En este punto, se puede realizar el ajuste de hiperparámetros conocido como la optimización de los mismos, con el fin de encontrar la mejor combinación de hiperparámetros que produzca el mejor rendimiento. El número de nodos, el número de las capas ocultas y la función de activación son hiperparámetros a elegir por el diseñador de la red (Nasiriany et al., 2019).

Dentro de los hiperparámetros que tiene una red neuronal se encuentra el número de capas que indica cuántas capas tiene la red neuronal en su arquitectura. Así mismo, se tiene el número de neuronas que se define como el número de nodos que tiene una capa en la red (Aggarwal, 2018; Goodfellow et al., 2016). Debido a que las redes neuronales tienen una muy baja interpretabilidad, no es posible conocer con certeza hasta qué punto este hiperparámetro influye en una predicción, y debe consumirse un tiempo de computación elevado para probar el resultado con varios valores del mismo (Aggarwal, 2018).

La función de activación también es uno de los hiperparámetros de una red neuronal y se define como una función matemática que transforma los datos de entrada, activando una señal de salida cuando se ha alcanzado un umbral de entrada específico (Dinov, 2018; Ghatak, 2019). Existen diferentes funciones de activación que se pueden implementar en las redes neuronales, estas funciones de activación pueden ser lineales y no lineales (Ghatak, 2019; Pawlus & Devine, 2020). La función de activación lineal es una línea recta y su salida oscila entre $[-\infty, +\infty]$, sin embargo, este tipo de funciones no ayudan con la complejidad de los datos y los parámetros de las redes neuronales (Aggarwal, 2018; Ghatak, 2019). Las funciones de activación no lineales ayudan a la simulación de la mayoría de problemas del mundo real, las cuales toman cualquier entrada de $[-\infty, +\infty]$ y asignarla a una salida que oscila entre $[0,1]$ y $[-1,1]$ (Aggarwal, 2018; Ghatak, 2019).

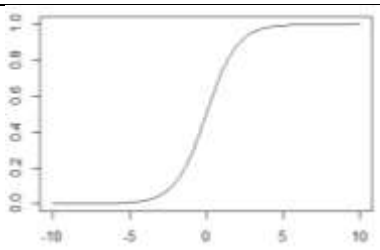
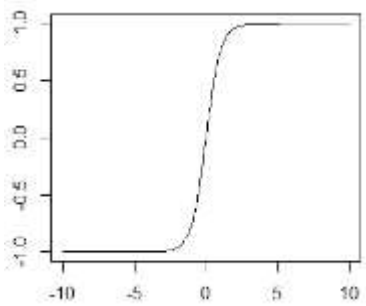
La función de activación sigmoidea se usa para la clasificación binaria en regresión logística e implementaciones de redes neuronales para la clasificación binaria. Esta función asigna la entrada lineal a una salida no lineal y es una función monótonamente creciente (Aggarwal, 2018; Ghatak, 2019). Sin embargo, la función es propensa a "atascarse", es decir, los valores de salida estarían muy cerca de cero cuando los valores de entrada son fuertemente negativos y viceversa (Ghatak, 2019). La función tangente hiperbólica, al igual que la función sigmoidea, tiene forma de "S", pero genera valores entre $[-1,1]$ (Aggarwal, 2018; Ghatak, 2019). En esta función, grandes entradas negativas generan salidas negativas, además, solo las entradas de valor cero se asignan a salidas cercanas a cero, estas propiedades hacen que la red sea menos propensa a atascarse, a diferencia de la función sigmoidal (Ghatak, 2019).

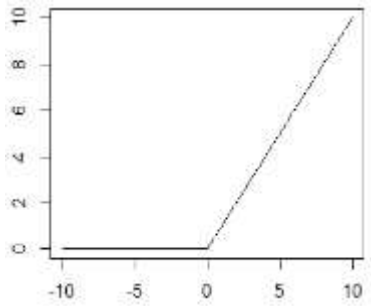
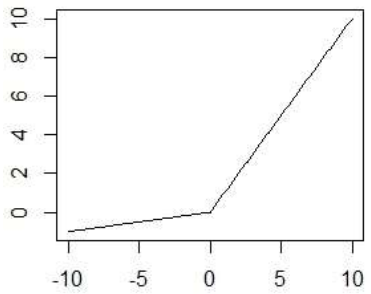
La función de activación unidad lineal rectificada (*relu*) es la función más utilizada para las capas ocultas, esta función se rectifica a la mitad (Aggarwal, 2018; Ghatak, 2019; Goodfellow et al., 2016). De esta forma, la función tiene una salida de cero si la entrada es menor que cero, y si la entrada es mayor que cero, la salida es la misma que la entrada. La función *relu* generalmente brinda un entrenamiento más rápido para redes grandes y se debe tener cuidado con los valores negativos, ya que los convierte a cero. Esto afecta el resultado al no mapear los valores negativos de manera adecuada, además, también disminuye la capacidad del modelo para ajustarse o entrenarse con los datos correctamente (Ghatak, 2019). Con el fin de mitigar estos problemas, se generó la función *Leaky relu*, la cual, usa un parámetro adicional que aumenta el rango de la función *relu* (Aggarwal, 2018; Ghatak, 2019).

En la Tabla 4 se observan las funciones de activación que se pueden implementar en las redes neuronales, así como su ecuación, los rangos de los valores de salida y los gráficos de cada una de las funciones de activación.

Tabla 4

Funciones de activación implementadas en las redes neuronales

Función de activación	Ecuación	Valores de salida	Gráfica
Sigmoidea (σ)	$\sigma(Z) = \frac{1}{1 + exp^{-Z}}$	[0,1]	
Tangente hiperbólica (*tanh*)	$g(Z) = \frac{exp^{Z} - exp^{-Z}}{exp^{Z} + exp^{-Z}}$	$[-1,1]$	

Unidad lineal rectificada (*relu*)	$g(Z) = \max(Z, 0)$	$[0, +\infty]$	
Leaky relu	$g(Z) = \begin{cases} Z & if\ Z > 0 \\ aZ & otherwise \end{cases}$ Donde a es un número pequeño y positivo	$[-\infty, +\infty]$	

Otro de los hiperparámetros que debe definirse en las redes neuronales es el tamaño del lote conocido también como *batch size*, el cual, corresponde al número de ejemplos que se introducen en la red para que entrenen cada vez (Aggarwal, 2018; Goodfellow et al., 2016). Si el número es pequeño, significa que la red tiene en memoria poca cantidad de datos, y entrena más rápido. Sin embargo, es posible que no aprenda las características y detalles que pueden ser significativos en la predicción, además, puede ser ineficiente desde el punto de vista computacional (Aggarwal, 2018). Si el tamaño del lote es grande, ocurre lo contrario, es más probable que se tenga una estimación más precisa del gradiente, pero con rendimientos más inferiores y un tiempo de entrenamiento más lento (Goodfellow et al., 2016). Generalmente se usan potencias de dos para seleccionar el tamaño del lote, ya que esta opción proporciona la mejor eficiencia en la mayoría de las arquitecturas (Aggarwal, 2018; Goodfellow et al., 2016).

Así mismo, uno de los hiperparámetros que debe determinarse es el número de épocas (*epochs*), que se define como el número de iteraciones que se deben realizar con los datos de entrenamiento para que la red se calibre de acuerdo a los pesos en cada nodo (Aggarwal, 2018; Dinov, 2018; Goodfellow et al., 2016). Con un número alto de épocas, se puede producir sobreajuste u *overfitting*, es decir, el modelo predictivo se aprende los datos y requiere más tiempo de computación. Con un número bajo de épocas, se puede generar un ajuste insuficiente o *underfitting*, es decir, el modelo no se calibra lo suficiente para generar las predicciones y no captura la tendencia en los datos (Ghatak, 2019).

El optimizador es otro de los hiperparámetros que debe definirse en la compilación de la red neuronal. El optimizador es aquel que se encarga de optimizar los valores de los parámetros para reducir el error cometido

por la red (Goodfellow et al., 2016; Pawlus & Devine, 2020). El optimizador se encarga de actualizar los valores de los parámetros, con el objetivo de seleccionar los mejores parámetros para el desempeño de la red. Se pueden implementar diferentes métodos de optimización, tales como: el descenso del gradiente estocástico, RMSprop, Adam, Momentum, Adagrad/Adadelta, entre otros (Aggarwal, 2018; Goodfellow et al., 2016; Pawlus & Devine, 2020). En la actualidad, el optimizador Adam es el más usado, ya que incorpora las ventajas de otros algoritmos y suele desempeñarse de manera competitiva con respecto a los otros optimizadores (Aggarwal, 2018; J. C. Chen et al., 2021; X. Hu, Yang, et al., 2021; Park et al., 2020).

En la compilación de la red neuronal, también es importante definir el hiperparámetro conocido como la tasa de aprendizaje, la cual, es un valor constante aplicado al gradiente de la función de pérdida que se usa para ajustar los pesos (Aggarwal, 2018; Ghatak, 2019; Pawlus & Devine, 2020). Los cambios en la tasa de aprendizaje deben ser suaves y graduales para evitar errores en el ajuste del modelo, por lo que se recomienda reducir la tasa a la mitad cada 5 épocas, o en 0,1 cada 20 épocas (Ghatak, 2019; Pawlus & Devine, 2020). Cuando la tasa de aprendizaje es alta, el descenso del gradiente puede aumentar inadvertidamente en lugar de disminuir el error de entrenamiento, lo que puede generar resultados NaN (Goodfellow et al., 2016). Como solución a este problema, generalmente se va reduciendo a la mitad la tasa de aprendizaje hasta que los NaN desaparecen (Aggarwal, 2018). Una tasa de aprendizaje baja genera tiempos de cómputo más altos, por lo que es importante seleccionar una tasa que permita obtener buenos resultados (Ghatak, 2019; Goodfellow et al., 2016; Pawlus & Devine, 2020).

Para generar la compilación del modelo, también debe definirse la función de pérdida asociada, la cual, actúa como medida de desempeño dentro de la compilación. Esta función es la que se implementa dentro de la red para realizar la propagación hacia atrás conocida como *backpropagation*, con el objetivo de reducir la pérdida (Aggarwal, 2018; Berk, 2016; Ghatak, 2019). La función permite evaluar el rendimiento del modelo y ajustar los parámetros. Para los problemas de regresión, la función de pérdida suele ser el error cuadrático medio (MSE) (Ghatak, 2019; Skansi, 2018).

Las redes neuronales pueden presentar sobreajuste dentro de la calibración del modelo predictivo, por lo cual, existen unos hiperparámetros dentro de la red neuronal conocidos como técnicas de regularización, las cuales, son procesos usados para reducir la complejidad del modelo y permiten ajustar los valores en la red neuronal para evitar el sobreajuste en el modelo predictivo (Ghatak, 2019; X. Hu, Yang, et al., 2021; Y. Zhang et al., 2018). Las redes neuronales usan los métodos de regularización conocidos como L1 y L2 para evitar el sobreajuste en el modelo predictivo, sin embargo, Hinton et al. (2012) propusieron una técnica de abandono para evitar el sobreajuste de manera más eficiente (Y. Zhang et al., 2018). Esta técnica permite reducir significativamente la varianza de un modelo con redes neuronales, sin aumentos sustanciales en su sesgo (Ghatak, 2019; Goodfellow et al., 2016). La tasa de abandono o también conocida como tasa de

dropout, elimina aleatoriamente algunas neuronas junto con sus conexiones de la red neuronal durante su entrenamiento (X. Hu, Yang, et al., 2021; Y. Zhang et al., 2018). De esta forma, se define un porcentaje en el que se deben desactivar aleatoriamente o "abandonar" al azar algunas neuronas, generando que las conexiones se desactiven y reduciendo la tendencia de la red a un sobreajuste por el uso de todas las neuronas (Ghatak, 2019; Y. Zhang et al., 2018). Esto hace que los datos cambien en cada iteración para que el modelo se generalice mejor, tenga una representación más equilibrada de los datos y no ajuste la solución específica a los datos de entrenamiento (sobreajuste) (Ghatak, 2019; Hinton et al., 2012; X. Hu, Yang, et al., 2021). En la Figura 7 se representa una red neuronal estándar versus una red neuronal con una tasa de abandono.

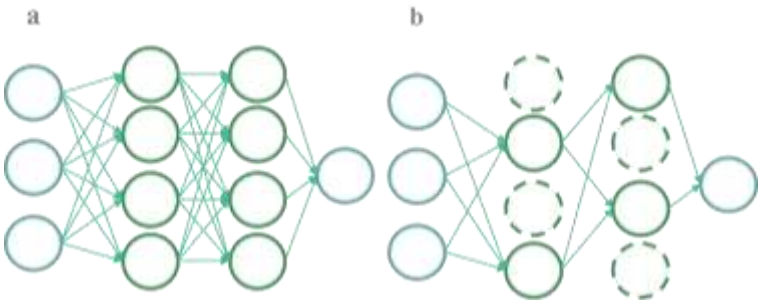


Figura 7. Red neuronal (a) estándar y (b) con tasa de abandono

Otra de las técnicas de regularización que también es considerado como un hiperparámetro, es la parada temprana o conocida como *early stopping*, la cual, es aplicada para monitorear el rendimiento del modelo predictivo y evitar que éste continue entrenándose con los datos cuando el modelo predictivo no mejora eficientemente su función de pérdida durante un periodo de tiempo determinado (Goodfellow et al., 2016). En este caso, el descenso del gradiente finaliza después de unas iteraciones, ya que se reduce el tamaño del espacio de los parámetros a un espacio más pequeño, actuando como regularizador dentro de la red (Aggarwal, 2018). Por lo general, la aplicación de este método de regularización se determina según la función de pérdida definida, de esta forma, se determina un valor mínimo de mejora en la función de pérdida para que la red neuronal se entrene (Goodfellow et al., 2016).

En la

Tabla ***5*** se muestran los hiperparámetros y las técnicas de regularización descritas anteriormente, como la tasa de abandono y la parada temprana. Se muestra el rango de valores que puede adquirir cada uno de los hiperparámetros y los valores típicos que se recomiendan explorar para su optimización según revisión de literatura (Elgeldawi et al., 2021; Ghatak, 2019; Pawlus & Devine, 2020).

Tabla 5

Hiperparámetros y técnicas de regularización en una red neuronal

Hiperparámetro	Rango	Valores sugeridos	Autores
Número de capas	[1, 50]	1, 2, 3, 4	(J. C. Chen et al., 2021; X. Hu, Yang, et al., 2021; Y. Liu et al., 2019)
Número de neuronas	[1, 500]	1, 8, 16, 32, 50, 100, 128, 256, 500	(J. C. Chen et al., 2021; X. Hu, Yang, et al., 2021; Qiao et al., 2020)
Función de activación	*sigmoid, tanh, relu, leaky relu*	*relu, tanh*	(J. C. Chen et al., 2021; Ghatak, 2019; X. Hu, Yang, et al., 2021; Park et al., 2020)
Tamaño del lote	[2, 1024]	32, 64, 128, 256	(Aggarwal, 2018; J. C. Chen et al., 2021; Goodfellow et al., 2016; Qiao et al., 2020)
Épocas	[10, 1000]	20, 50, 100, 500	(J. C. Chen et al., 2021; X. Hu, Yang, et al., 2021; Park et al., 2020)
Optimizador	RMSprop, Adam, Momentum, Adagrad/Adadelta	Adam	(Aggarwal, 2018; J. C. Chen et al., 2021; X. Hu, Yang, et al., 2021; Jo et al., 2021; Park et al., 2020; Qiao et al., 2020; Ren et al., 2021)
Tasa de aprendizaje	[0,0001 – 0,01]	0,001	(X. Hu, Yang, et al., 2021; Jo et al., 2021; Park et al., 2020)
Función de pérdida	MSE, MAE	MSE	(J. C. Chen et al., 2021; Ghatak, 2019; X. Hu, Yang, et al., 2021; Jo et al., 2021; Y. Liu et al., 2019; Qiao et al., 2020; Ren et al., 2018)
Tasa de abandono (*dropout*)	[0%, 100%]	Desde el 10% hasta el 50%	(J. C. Chen et al., 2021; Ghatak, 2019; Y. Liu et al., 2019)
Parada temprana (*early stopping*)	De acuerdo a la función de pérdida		(Goodfellow et al., 2016; Y. Liu et al., 2019; Park et al., 2020)

Cada uno de los valores de estos hiperparámetros debe ser probado en las arquitecturas de la red neuronal definida, con el fin de validar los valores que permiten obtener un buen desempeño del modelo predictivo con una predicción más precisa. Para esto, existen diferentes herramientas para realizar la optimización de hiperparámetros, tales como: búsqueda en cuadrícula (*grid search*), búsqueda aleatoria (*random search*),

búsqueda codiciosa (*greedy search*), optimización bayesiana (*Bayesian Optimization*), entre otros (Aggarwal, 2018; Elgeldawi et al., 2021; Ghatak, 2019). Estas herramientas buscan minimizar el error de validación de un modelo parametrizado de acuerdo a los hiperparámetros seleccionados antes del entrenamiento, es decir, encontrar la configuración de hiperparámetros que le permita al modelo generalizar bien ante nuevos datos (Ghatak, 2019). En la Tabla 6 se muestran los autores que han implementado cada uno de los métodos de optimización de hiperparámetros para definir los valores de cada uno de los hiperparámetros que se deben implementar en la optimización de estos.

Tabla 6

Métodos implementados para la optimización de hiperparámetros

Método de optimización de hiperparámetros	**Definición**	**Autores**
Búsqueda en cuadrícula (*grid search*)	Todas las combinaciones son probadas	(Elgeldawi et al., 2021; Huotari et al., 2021; Mejia et al., 2021)
Búsqueda aleatoria (*random search*)	El número de combinaciones definidas son testeadas	(Elgeldawi et al., 2021)
Optimización bayesiana (*Bayesian Optimization*)	Se alcanza el número máximo de evaluación	(J. C. Chen et al., 2021; Elgeldawi et al., 2021)

Después de la optimización de los hiperparámetros, se hace el entrenamiento de los modelos que permiten realizar las predicciones a partir de diferentes conjuntos de datos. Para el entrenamiento de cualquier tipo de red neuronal, se requiere de la división de los datos de entrada en mínimo dos conjuntos de datos, con el objetivo de realizar el entrenamiento de la red. Es así como la división del conjunto de datos de entrada se puede dividir en dos conjuntos tales como: datos de entrenamiento y datos de prueba (Fan et al., 2019; Y. Liu et al., 2019; Ren et al., 2018). También se puede realizar la división en tres conjuntos de datos: datos de entrenamiento, datos de validación y datos de prueba (Z. Chen et al., 2020; Jo et al., 2021; X. Li et al., 2021; Park et al., 2020). Los datos de entrenamiento permiten al modelo calibrarse de acuerdo a las entradas y salidas esperadas; los datos de validación se usan para ajustar los hiperparámetros del modelo predictivo y los datos de prueba se usan para evaluar la precisión y robustez del modelo (X. Li et al., 2021; Park et al., 2020).

Para cada conjunto de datos, se debe realizar una división porcentual del número de datos que tendrá cada conjunto. Existen diferentes porcentajes para la división en dos conjuntos de datos: los datos de entrenamiento y los datos de prueba, tales como: 70% para datos de entrenamiento y 30% para datos de prueba, así como también, 80% y 20%, 90% y 10% o incluso, el porcentaje de división puede ser definido

de acuerdo al número de conjuntos de datos que se tengan, según revisión de literatura (Fan et al., 2019; Y. Liu et al., 2019; Ren et al., 2018). Así mismo, existen diferentes porcentajes para la división en tres conjuntos de datos: los datos de entrenamiento, los datos de validación y los datos de prueba, tales como: 60% para datos de entrenamiento, 10% para datos de prueba y 30% para datos de validación, así como también, 70%, 10% y 20% (Z. Chen et al., 2020; Jo et al., 2021; X. Li et al., 2021; Park et al., 2020).

Después del entrenamiento del modelo con sus respectivos datos y la calibración del modelo predictivo, se podrán realizar predicciones con nuevos datos. Estas predicciones por medio de modelos que implementan redes neuronales han sido usadas para reconocimiento de caracteres, imágenes, voz, predicción bursátil, generación de texto, traducción de idiomas, conducción autónoma, series temporales y predicción en general de diferentes campos (Dinov, 2018; Ge et al., 2021; Kuhn & Johnson, 2019; Zraibi et al., 2021). Así mismo, algunas de las aplicaciones en estos campos generan datos que muestran una interdependencia no lineal entre las muestras, y las mediciones suelen estar contaminadas por el ruido generado por los sensores o por el entorno en el que se realizaron las mediciones (Mejia et al., 2021). Por esto, es importante el uso de las redes neuronales, ya que éstas son capaces de manejar grandes volúmenes de información de sistemas complejos no lineales y son independientes de los principios electroquímicos de la batería (Ge et al., 2021).

El uso de las redes neuronales ha incrementado en los últimos años debido a su precisión en las predicciones. Existen diferentes tipos de redes neuronales tales como: CNN, redes neuronales recurrentes (RNN, por sus siglas en inglés: *Recurrent Neural Network*), red LSTM, entre otras. Dentro de las redes implementadas para la predicción de series de tiempo se encuentra la red LSTM, debido a su capacidad de capturar las dependencias de corto y largo plazo en una secuencia (Mejia et al., 2021). Además, este tipo de redes se ha aplicado con éxito en la predicción del RUL de las baterías a través de diferentes series temporales como la capacidad, el voltaje, la corriente, la temperatura y demás variables importantes en la predicción.

En esta investigación, el modelo predictivo se trabajará con el método supervisado de regresión a partir de la implementación de redes neuronales. De las cuales, se hará uso de la CNN como un filtro clásico para disminuir el ruido que tienen los datos obtenidos a través de sensores. Así mismo, se destaca el uso de las redes LSTM, ya que son una de las metodologías más efectivas para la predicción de series temporales adaptativas (Ge et al., 2021). Además, las redes LSTM son capaces de capturar el comportamiento no lineal de los datos, la dependencia a corto y largo plazo y permiten obtener predicciones más precisas. Dentro del análisis del trabajo de investigación, se destaca el comportamiento no lineal de las variables del EV, tales como: la capacidad de la batería, el voltaje, la temperatura, entre otras. Por esto, es importante implementar métodos como la red LSTM que permita capturar el comportamiento y generar una predicción precisa.

4.2 Redes neuronales convolucionales – CNN

Las redes neuronales convolucionales (CNN) se introdujeron en los trabajos realizados por LeCun et al. en 1990 y 1998, para el reconocimiento de manuscritos de caracteres (Mejia et al., 2021). LeCun et al. en 1990 y 1998 se basaron en las ideas de David H. Hubel and Torsten Weisel presentadas en el artículo publicado en 1968 sobre campos receptivos para la generación de las CNN (Skansi, 2018). Desde el 2012, las CNN se han convertido en el algoritmo de referencia para las tareas de visión artificial (Chollet & Allaire, 2016; Ghatak, 2019). Estas CNN están inspiradas en la corteza visual del cerebro humano, allí hay áreas específicas que son sensibles a ciertas características de entrada, como por ejemplo algunas neuronas (Mejia et al., 2021). Es así como las ANN son capaces de extraer *features* útiles de los datos al ser capaces de detectar patrones de comportamiento en éstos (Chollet & Allaire, 2016).

Para realizar este proceso, las CNN requieren de una operación matemática llamada convolución como componente fundamental de la red (Ghatak, 2019; Goodfellow et al., 2016; Jo et al., 2021). Esta operación de convolución es un tipo de operación lineal y se refiere a la aplicación de un filtro o también llamado *kernel* en los datos de entrada, lo que permite especificar los bordes y obtener datos que permitan generar una buena predicción (Ghatak, 2019; Goodfellow et al., 2016). Es por esto, que las CNN han sido ampliamente usadas en problemas de clasificación de imágenes y también en predicciones de acciones y aprendizaje de la dinámica de los robots en sistemas autónomos (Goodfellow et al., 2016; Nasiriany et al., 2019). La operación de convolución matemática se representa mediante la ecuación (5) (Goodfellow et al., 2016):

$$s(t) = (x * w)(t), \tag{5}$$

donde x corresponde a los valores de entrada, w representa el kernel aplicado, $*$ es el operador de convolución, t es la posición en el tiempo y $s(t)$ representa el mapa de *features*. En aplicaciones de aprendizaje automático, x suele ser una matriz multidimensional de datos y el *kernel* suele ser una matriz de parámetros que se adaptan según el entrenamiento de la red (Goodfellow et al., 2016; Zraibi et al., 2021). De esta forma, si se tiene una entrada con dos dimensiones representada por $\boldsymbol{I}(m,n)$ y un kernel correspondiente representado por $\boldsymbol{K}(i,j)$, la ecuación para representar el mapa de *features* $\boldsymbol{S}(i,j)$ es la ecuación (6):

$$\boldsymbol{S}(i,j) = (\boldsymbol{K} * \boldsymbol{I})(i,j) = \sum_m \sum_n \boldsymbol{I}(i-m, j-n)\boldsymbol{K}(m,n), \tag{6}$$

donde $\boldsymbol{K}(i,j)$ representa el *kernel* de tamaño i,j, $\boldsymbol{I}(m,n)$ representa la matriz de entrada con dos dimensiones m,n. De esta forma, para la generación de las CNN en regresión, se requieren de tres componentes: aplanar las imágenes o matrices a vectores, tomar los vectores de entrada y entregar a una

neurona para su procesamiento, con el fin de generar el campo receptivo (Skansi, 2018). Generalmente las CNN constan de capas convolucionales, seguidas de una capa de agrupación, que se usa para comprimir los datos sin perder mucha información (Goodfellow et al., 2016; Mao et al., 2018; Mejia et al., 2021). Las CNN también permiten suavizar los datos y disminuir el ruido de las mediciones obtenido por los sensores que toman el valor de cada una de las variables implicadas (Mejia et al., 2021). En la Figura 8 se muestra una arquitectura común de una CNN.

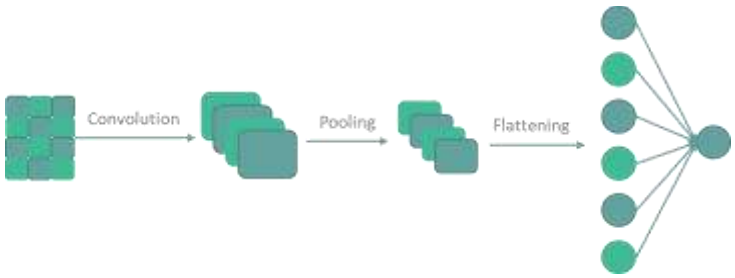


Figura 8. Arquitectura común de una CNN. Fuente: Elaboración propia basada en (Ghatak, 2019)

Inicialmente se cuenta con una matriz de entrada a la cual se le realizan diferentes operaciones de convolución determinadas por las capas convolucionales. Seguido, se realiza el proceso de agrupación, donde se reducen las dimensiones de la entrada y el cálculo, permitiendo que los detectores de *features* sean más invariantes, esto indica que se puede reconocer un objeto o variable incluso cuando sus datos varían (Chollet & Allaire, 2016; Ghatak, 2019). Existen dos tipos de capas de agrupación: la capa de agrupación máxima (*max-pooling*) y la capa de agrupación promedio (*average-pooling*). La capa de agrupación máxima es la más implementada, ya que genera mejores resultados al deslizar el filtro sobre la entrada y almacenar el máximo de los valores de entrada. La capa de agrupación promedio almacena el valor promedio de las entradas, lo que en ocasiones genera en la red neuronal que las salidas colapsen (Ghatak, 2019).

Una CNN requiere que los datos de entrada estén establecidos como una matriz, con el fin de aplicar los filtros y extraer las características principales de los datos de entrada (Goodfellow et al., 2016). Dentro de la CNN se deben definir: las capas convolucionales con el número de filtros, para los cuales se recomienda la implementación de la potencia de dos, como por ejemplo 32 o 64 filtros con un tamaño de filtro impar de 3 o 5 y una función de activación *relu*, que permita capturar los patrones de entrada (Ghatak, 2019; Zraibi et al., 2021). Así mismo, se debe definir la capa de *pooling* con su tamaño, con el objetivo de reducir las dimensiones de entrada, en este paso, se recomienda el uso de *max-pooling* con un tamaño de *pooling* igual a 2 (Ghatak, 2019). Luego de realizar este proceso, si se requiere como salida una probabilidad o en general un número, se agrega una capa de aplanamiento que permita convertir la matriz resultante en un vector, y este resultado, se pasa por dos capas densas que permiten arrojar como salida el número (Zraibi et al., 2021).

La red convolucional tiene diferentes hiperparámetros a seleccionar dentro de la arquitectura de cada una de sus capas, los valores de referencia o recomendados según los autores revisados se muestran en la Tabla 7. Además de estos hiperparámetros, también deben ser consideradas las definiciones anteriormente brindadas en la

Tabla ***5*** para el número de capas y función de activación. Así mismo, se deben tener en cuenta los hiperparámetros necesarios para hacer la compilación y entrenamiento de la CNN, tales como tamaño del lote, épocas, optimizador, tasa de aprendizaje y función de pérdida, los cuales fueron presentados en la

Tabla ***5***.

Tabla 7

Hiperparámetros de la CNN

Hiperparámetro	**Rango**	**Valores sugeridos**	**Autores**
Número de capas convolucionales	[1, 10]	1, 2	(Goodfellow et al., 2016; Mao et al., 2018; Mejia et al., 2021)
Número de filtros	[16, 512]	32, 64	(Ghatak, 2019; Mao et al., 2018; Zraibi et al., 2021)
Tamaño del filtro	[1, 15]	2, 3, 5	(Ghatak, 2019; Mao et al., 2018; Zraibi et al., 2021)
Función de activación	*sigmoid, tanh, relu, leaky relu*	*relu*	(Ghatak, 2019; Zraibi et al., 2021)
Número de capas de agrupación	[1, 10]	1, 2	(Goodfellow et al., 2016; Mao et al., 2018; Mejia et al., 2021)
Función de agrupación	*max-pooling* o *average-pooling*	*max-pooling*	(Ghatak, 2019; Mao et al., 2018)
Tamaño del *pool*	[2, 32]	2, 3	(Ghatak, 2019; Mao et al., 2018)
Número de capas de aplanamiento	[1, 10]	1, 2	(Mao et al., 2018; Ren et al., 2021; Zraibi et al., 2021)
Número de capas densas	[1, 10]	1, 2	(Zraibi et al., 2021)

Número de neuronas en capas densas	[1, 500]	1, 16, 32, 50, 100	(J. C. Chen et al., 2021; X. Hu, Yang, et al., 2021; Qiao et al., 2020)

4.3 Redes neuronales de corto y largo plazo – LSTM

Las redes neuronales LSTM son un tipo especial de red neuronal recurrente (RNN) con capacidades de aprendizaje profundo que fue planteada por Sepp Hochreiter y Jürgen Schmidhuber en 1997 (Hochreiter & Schmidhuber, 1997), diseñada para capturar información de dependencia a largo plazo (Xiong, 2019). La arquitectura de la LSTM consta de un conjunto de subredes conectadas de forma recurrente conocidas como celdas de memoria, las cuales se encargan de mantener su estado a lo largo del tiempo, regulando el flujo de la información a través de unidades de activación no lineales (van Houdt et al., 2020).

La arquitectura de la red LSTM fue modificada por Gers et al. (1999), al agregar a la arquitectura una puerta de olvido para permitir que la red reestableciera su estado; a esta modificación se le conoce como "*Vanilla* LSTM" y es la arquitectura más conocida actualmente (van Houdt et al., 2020). Por ser la arquitectura más conocida se seguirá nombrando como red LSTM dentro de la investigación. La red LSTM contiene una puerta lógica para controlar y proteger la información, resolviendo de manera efectiva la entrada a largo plazo (Zhang et al., 2021). La red LSTM permite que las salidas en cada paso de tiempo se utilicen como entradas para procesar datos secuenciales, y se diferencia de una red RNN, ya que el peso en el bucle está condicionado por el contexto en lugar de permanecer fijo, lo que genera un problema en el uso de la red RNN (Jo et al., 2021).

Las redes LSTM se han aplicado en campos tales como: calificación de emociones, retrasos en vuelos, análisis predictivo de salud, modelado de idiomas, reconocimiento de imágenes, predicción de series temporales, entre otros problemas que requieran memoria temporal, debido a su gran desempeño (van Houdt et al., 2020; Xiong, 2019). Se considera que la red LSTM es una de las redes más avanzadas para procesar secuencias temporales (van Houdt et al., 2020). La red LSTM se ha aplicado a las predicciones de series temporales debido a características complejas como la no linealidad, la no estacionariedad y la correlación de secuencias, por lo cual, este tipo de datos plantea un gran desafío de pronóstico (van Houdt et al., 2020).

La arquitectura de la red LSTM tiene tres tipos de estructuras de compuertas, donde el núcleo de la LSTM es el estado de la celda, el cual almacena la información de principio a fin (X. Hu, Yang, et al., 2021). Así mismo, el estado de la celda permite que el gradiente se propague bien, aunque el estado ya haya transcurrido, por lo tanto, la red LSTM alivia el problema de las redes RNN. Los tres tipos de estructuras

son: puerta de olvido, puerta de entrada y puerta de salida y estas puertas suelen estar representadas por la función de activación sigmoide (X. Li et al., 2019). En la Figura 9 se muestra la arquitectura de la red LSTM.

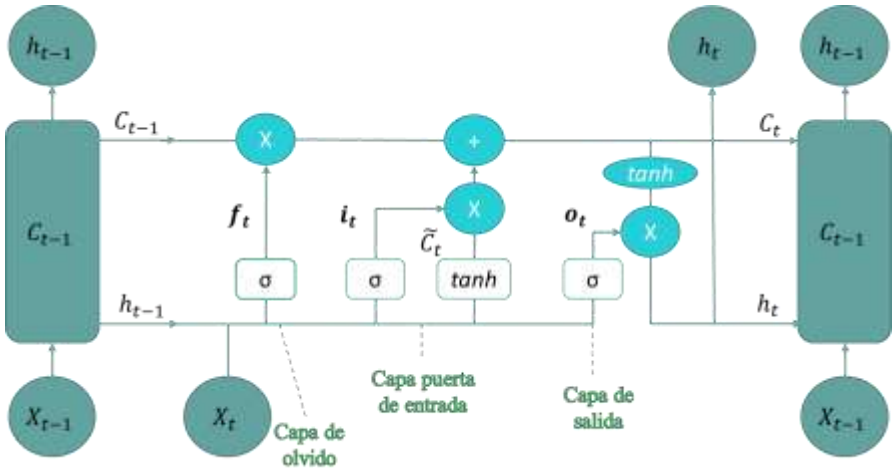


Figura 9. Arquitectura de la LSTM. Fuente: Elaboración propia basada en (Listou Ellefsen et al., 2019)

En la Figura 9, f_t es la puerta de olvido, i_t es la puerta de entrada, $\tilde{C}_t$ es un estado temporal de la puerta de entrada y o_t es la puerta de salida. A continuación, se describen cada una de ellas.

Puerta de olvido: Decide si retener u olvidar la información anterior, esta toma de decisiones se hace a partir de una función de activación, compuesta por el vector de entrada y el estado oculto anterior, generando un valor entre 0 y 1. La ecuación (7) representa la puerta de olvido.

$$f_t = \sigma(\boldsymbol{W}_f \boldsymbol{X}_t + \boldsymbol{U}_f h_{t-1} + \boldsymbol{b}_f), \quad (7)$$

donde f_t es el vector de activación de la puerta de olvido, σ corresponde a la función de activación, $\boldsymbol{W}_f/\boldsymbol{U}_f$ son las matrices de pesos asociadas a la puerta de olvido, $\boldsymbol{X}_t$ es el vector de entrada, h_{t-1} es el vector de estado oculto anterior y $\boldsymbol{b}_f$ es el vector del sesgo asociado a la puerta de olvido.

Puerta de entrada: Define si actualizar el estado usando la entrada de información actual o no. Esta puerta tiene dos pasos, en el primero se elige una función de activación tangente hiperbólica para formar la nueva memoria y la función sigmoidea determina qué información en la nueva memoria se puede actualizar al estado de la celda. La ecuación (8) representa la puerta de entrada de la red LSTM.

$$i_t = \sigma(\boldsymbol{W}_i \boldsymbol{X}_t + \boldsymbol{U}_i h_{t-1} + \boldsymbol{b}_i), \quad (8)$$

donde i_t es el vector de activación de la puerta de entrada, σ corresponde a la función de activación, $\boldsymbol{W}_i/\boldsymbol{U}_i$ son las matrices de pesos asociadas a la puerta de entrada, $\boldsymbol{X}_t$ es el vector de entrada, h_{t-1} es el vector de estado oculto anterior y $\boldsymbol{b}_i$ es el vector del sesgo asociado a la puerta de entrada.

Antes de calcular la puerta de salida, se calcula $\tilde{C}_t$, que es un valor temporalmente guardado antes de ser actualizado el nuevo valor del estado de la celda. Este estado previo a la puerta de salida se representa por medio de la ecuación (9).

$$\tilde{C}_t = Activación(\boldsymbol{W}_g \boldsymbol{X}_t + \boldsymbol{U}_g h_{t-1} + \boldsymbol{b}_g), \quad (9)$$

donde $\tilde{C}_t$ es el vector temporal, $Activación$ corresponde a la función de activación, $\boldsymbol{W}_g/\boldsymbol{U}_g$ son las matrices de pesos asociadas a la compuerta temporal, $\boldsymbol{X}_t$ es el vector de entrada, h_{t-1} es el vector de estado oculto anterior y $\boldsymbol{b}_g$ es el vector del sesgo asociado.

Puerta de salida: Esta puerta está diseñada para decidir qué información se convierte del estado de celda a la capa oculta actual. Después de que pase el estado de celda por una función de activación tangente hiperbólica, su valor oscila entre -1 y 1 y luego se multiplica por una función sigmoidea, y así, la información del estado de la celda se transfiere a la capa oculta. La ecuación (10) presenta la forma de obtener la puerta de salida.

$$o_t = \sigma(\boldsymbol{W}_o \boldsymbol{X}_t + \boldsymbol{U}_o h_{t-1} + \boldsymbol{b}_o), \quad (10)$$

donde o_t es el vector de activación de la puerta de salida, σ corresponde a la función de activación, $\boldsymbol{W}_o/\boldsymbol{U}_o$ son las matrices de pesos asociadas a la puerta de salida, $\boldsymbol{X}_t$ es el vector de entrada, h_{t-1} es el vector de estado oculto anterior y $\boldsymbol{b}_o$ es el vector del sesgo asociado a la puerta de salida.

Existen diferentes estructuras para la LSTM: Uno a uno, esta es una red densa con una entrada y una salida. Uno a muchos, se tiene una entrada y varias capas ocultas que conservan la información para obtener como salida una secuencia de observaciones como, por ejemplo, la generación de la descripción de una imagen. Muchos a uno, en este tipo de estructura, se tienen varias entradas y como salida un único valor, como por ejemplo las series de tiempo, problemas de clasificación, entre otros. Muchos a muchos, se tiene una secuencia de entrada y una secuencia de salida, en esta estructura se pueden presentar dos tipos de configuraciones: la secuencia de salida se genera después de la secuencia de entrada como, por ejemplo, una secuencia de traducción; o la secuencia de salida se genera al mismo tiempo como, por ejemplo, la asignación de una etiqueta a una palabra (Ge et al., 2021; Ghatak, 2019; Park et al., 2020).

Estas estructuras son empleadas en diferentes campos de investigación, tanto individualmente como en combinación con otras arquitecturas de aprendizaje profundo (Aggarwal, 2018; Zraibi et al., 2021). En esta investigación, se desarrollará un modelo predictivo implementando la CNN para el suavizamiento de los datos, cuya salida serán los datos suavizados que, junto con datos sin sensores, serán la entrada a la red LSTM con estructura de muchos a uno, lo que indica muchas variables de entrada y como salida la predicción del RUL. La red LSTM tiene como hiperparámetros el número de neuronas dentro de cada capa LSTM, el número de pasos seleccionados para realizar la predicción, la definición de si se requiere la

secuencia de retorno o no y la función de activación, estos hiperparámetros con sus respectivos valores sugeridos, según revisión de literatura, son mostrados en la Tabla 8.

Además de estos hiperparámetros, debido al sobreajuste que puede presentar la red LSTM, se pueden implementar capas de regularización como el *dropout*, descrito anteriormente, el cual, permite controlar el sobreajuste que se puede presentar en la red. Así mismo, se deben tener en cuenta los hiperparámetros necesarios para hacer la compilación y entrenamiento de la red LSTM, tales como tamaño del lote, épocas, optimizador, tasa de aprendizaje y función de pérdida, los cuales fueron presentados en la

Tabla ***5***.

Tabla 8

Hiperparámetros de la red LSTM

Hiperparámetro	Rango	Valores	Autores
Número de capas LSTM	[1, 50]	1, 2, 3, 4	(J. C. Chen et al., 2021; X. Hu, Yang, et al., 2021; Y. Liu et al., 2019)
Número de neuronas en capa LSTM	[2, 256]	128	(Ghatak, 2019)
Número de pasos	[1, 30]	5, 10, 20, 39	(X. Hu, Yang, et al., 2021; Jo et al., 2021; Y. Liu et al., 2019; Park et al., 2020)
Secuencia de retorno	Sí o No	Sí	(Zraibi et al., 2021)
Función de activación	*sigmoid, tanh, relu, leaky relu*	*relu, tanh*	(J. C. Chen et al., 2021; Ghatak, 2019; X. Hu, Yang, et al., 2021; Park et al., 2020)
Número de capas *dropout*	[1, 50]	1, 2, 3	(J. C. Chen et al., 2021; X. Hu, Yang, et al., 2021; Y. Liu et al., 2019)
Tasa de *dropout*	[0%, 100%]	Desde el 10% hasta el 50%	(J. C. Chen et al., 2021; Ghatak, 2019; Y. Liu et al., 2019)
Número de capas densas	[1, 50]	1, 2	(J. C. Chen et al., 2021; X. Hu, Yang, et al., 2021; Y. Liu et al., 2019; Tong et al., 2021)
Número de neuronas en capa densa	[1, 500]	1, 64, 128	(J. C. Chen et al., 2021; X. Hu, Yang, et al., 2021; Qiao et al., 2020)

5. PROPUESTA DEL MODELO PREDICTIVO

En este capítulo se presenta la metodología propuesta para el desarrollo del modelo predictivo, considerando el tipo de datos a analizar, la división en diferentes conjuntos y la generación de la estructura de los datos como entrada para el modelo de suavizamiento y el modelo predictivo. Además, se utiliza una CNN para suavizar los datos obtenidos a partir de sensores, con el objetivo de disminuir el ruido. Los datos suavizados por medio de la CNN son los datos de entrada para la red LSTM, los cuales permiten realizar la predicción del *target*.

5.1 Descripción general

Teniendo en cuenta la revisión de literatura, en la que se destaca el uso de modelos predictivos a través de las redes neuronales, se propone una metodología para realizar un modelo predictivo que integre una CNN y una red LSTM con arquitectura de muchos a uno. La precisión en la predicción del RUL garantiza la durabilidad, confiabilidad y seguridad de la batería, la cual, corresponde a un componente crítico del EV. Por esto, se destaca la importancia de los modelos predictivos a través de la implementación de redes neuronales que permitan capturar su comportamiento de degradación a través del tiempo. Este modelo predictivo consta de dos redes neuronales que en conjunto permiten una predicción más precisa, debido a que se combinan las ventajas de la CNN para hacer un suavizamiento de los datos obtenidos de las *features* medidas por medio de sensores, y que sirven como datos de entrada para el modelo con la red LSTM. Así mismo, se destacan las ventajas del uso de la red LSTM para realizar predicciones de series con comportamientos no lineales que conserva la memoria de corto y largo plazo, lo que permite realizar una predicción más ajustada. La metodología propuesta se muestra en la Figura 10, la cual es implementada en el paquete `cnnlstm` del lenguaje de programación R que integra diferentes funciones para hacer la normalización de los datos, la división de los mismos, la creación de los modelos y la evaluación de desempeño (Rincón, 2023).

La metodología propuesta para el modelo predictivo se divide en tres etapas que consideran: la división de los datos, el suavizamiento de los datos obtenidos a partir de sensores y la predicción a partir de un modelo predictivo usando una red LSTM. Este modelo predictivo puede ser evaluado con diferentes métricas de desempeño, las cuales, pueden estar acompañadas de un análisis gráfico de los resultados que permita tomar

decisiones para hacer los ajustes necesarios y generar una predicción más precisa. Cada una de las etapas de la metodología propuesta es explicada, teniendo en cuenta los pasos que se deben realizar para llevarla a cabo.

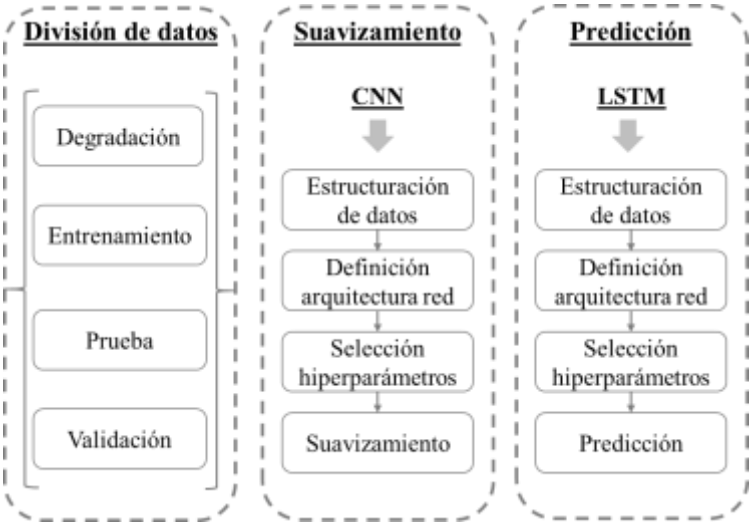


Figura 10. Metodología desarrollada para el modelo predictivo

La metodología propuesta puede ser utilizada cuando se tienen variables obtenidas a partir de sensores (*features*) y una variable que presenta un comportamiento no lineal a través del tiempo (*target*). A partir de esta estructura, se propone una división de los datos en cuatro conjuntos que permitan la identificación de datos especiales llamados datos de degradación, el entrenamiento del modelo, el ajuste de parámetros dentro de la red con los datos de prueba y la optimización de los hiperparámetros con los datos de validación (J. C. Chen et al., 2021; X. Li et al., 2021). Así mismo, con los datos obtenidos a partir de sensores, se realiza una estructuración de los datos con el objetivo de realizar la suavización de los mismos por medio de la CNN, lo cual, genera datos de entrada sin ruido para hacer una predicción precisa (Mejia et al., 2021). Además, se considera una red LSTM para hacer la predicción de muchos (*p features*) a uno (*y target*), lo que permite a partir de diferentes variables hacer la predicción (Park et al., 2020). En el modelo predictivo, también se tiene en cuenta la selección de hiperparámetros para mejorar el desempeño en la predicción.

5.2 Estructura del modelo predictivo

Sea $\boldsymbol{Y}$ el vector del *target* y $\boldsymbol{X}$ la matriz de *p features* seleccionadas como variables de entrada para el modelo predictivo. De esta forma, se obtiene un vector $\boldsymbol{Y^m}$, que indica *target* con m filas (número de observaciones), y una matriz $\boldsymbol{X_p^m}$ con dimensiones $p \times m$, donde p representa el número de *features* y m representa el

número de filas que se tienen de cada una de las *features*. Es así, como se representa la estructura matricial del vector $\boldsymbol{Y^m}$ y la matriz $\boldsymbol{X_p^m}$.

$$\boldsymbol{X_p^m} = \begin{bmatrix} x_1^1 & x_2^1 & \cdots & x_p^1 \\ x_1^2 & x_2^2 & \cdots & x_p^2 \\ \vdots & \vdots & \ddots & \vdots \\ x_1^m & x_2^m & \cdots & x_p^m \end{bmatrix} \; donde \; \boldsymbol{Y^m} = \begin{bmatrix} y^1 \\ y^2 \\ \vdots \\ y^m \end{bmatrix} = f \begin{bmatrix} x_1^1 & x_2^1 & \cdots & x_p^1 \\ x_1^2 & x_2^2 & \cdots & x_p^2 \\ \vdots & \vdots & \ddots & \vdots \\ x_1^m & x_2^m & \cdots & x_p^m \end{bmatrix}$$

Donde la notación implementada significa:

- La matriz de p *features*: $\boldsymbol{X_p^m} = \boldsymbol{X_1^m}, \boldsymbol{X_2^m}, \dots, \boldsymbol{X_p^m}$, donde el número de $p\ features = 1, 2, \dots, p$, y el número de observaciones $= 1, 2, \dots, m$.
- El vector del *target* corresponde a: $\boldsymbol{Y^m} = Y^1, Y^2, \dots, Y^m$.

De la matriz $\boldsymbol{X_p^m}$, se deben evaluar los datos para determinar el número de observaciones que corresponden a los k datos de degradación y que deben ser eliminados de la matriz $\boldsymbol{X_p^m}$. Al identificar estos datos, se obtiene una matriz que se representa así: $\boldsymbol{X_p^{m-k}}$. Las p features que se tienen en la matriz $\boldsymbol{X_p^{m-k}}$, pueden ser representadas a partir de dos matrices $\boldsymbol{X_s^{m-k}}$ y $\boldsymbol{X_{p-s}^{m-k}}$, las cuales representan la matriz de las s *features* obtenidas por sensores y las $p - s$ *features* obtenidas sin sensores respectivamente. De esta forma, $\boldsymbol{X_p^{m-k}} = \boldsymbol{X_s^{m-k}} + \boldsymbol{X_{p-s}^{m-k}}$. A la matriz $\boldsymbol{X_s^{m-k}}$ se le realiza un proceso de suavizamiento por medio de un modelo con una CNN. Después de realizar el suavizamiento con la CNN, se une la matriz con los datos suavizados y la matriz de los datos obtenidos sin sensores. Al unir ambas matrices, se obtiene una matriz de entrada para el modelo predictivo con la red LSTM.

5.3 División de los datos

A partir de la matriz $\boldsymbol{X_p^m}$, se propone una etapa de división de los datos que debe ser aplicada en diferentes momentos de la metodología propuesta, en la Figura 11 se muestra la estructura de los pasos a seguir. La división de los datos considera cuatro conjuntos para el modelo predictivo tales como: datos de degradación, entrenamiento, prueba y validación, con el objetivo de:

1. Identificar los datos especiales de degradación, los cuales se separan de los datos para obtener un mejor desempeño en el modelo predictivo.
2. Realizar el suavizamiento de los datos obtenidos por medio de sensores a través de la división en datos de entrenamiento y prueba.

3. Realizar la predicción por medio de tres conjuntos de datos: entrenamiento, validación para realizar el ajuste de los hiperparámetros y prueba para evaluar la precisión del modelo.

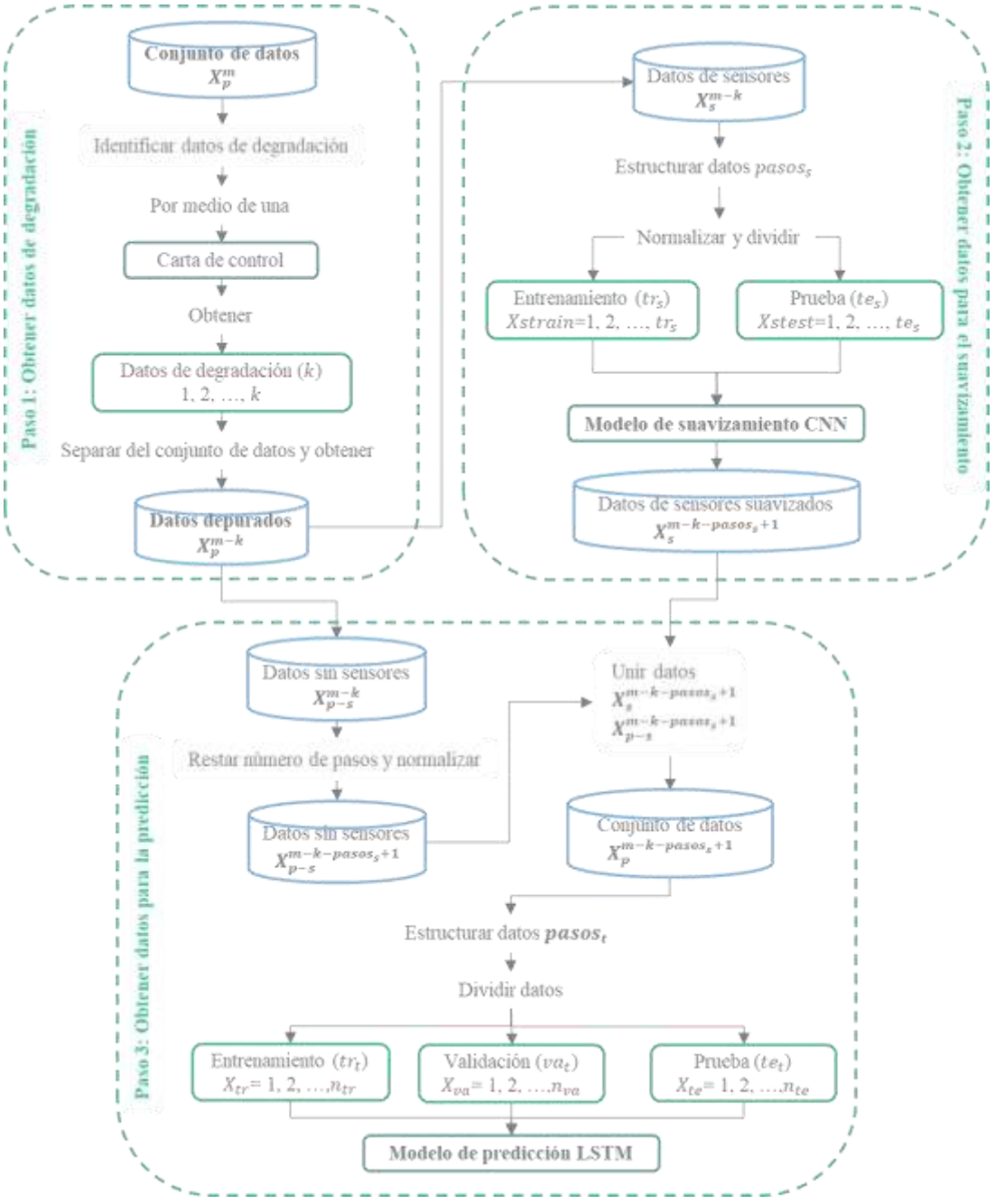


Figura 11. Estructura de los pasos a seguir para realizar la división de datos

Paso 1. Obtener datos de degradación

Inicialmente los datos de degradación deben ser identificados en todo el conjunto de datos, con el fin de obtener datos limpios que permitan entrenar adecuadamente los modelos de suavizamiento y predicción.

Los datos de degradación hacen parte de los datos especiales que tuvieron un comportamiento diferente comparado con el conjunto de datos normales y que es necesario sacar del conjunto de datos para su posterior evaluación en los modelos de suavizamiento y predicción (X. Li et al., 2021). La separación de estos datos tiene como objetivo la identificación de posibles cambios, daños o degradaciones que pueden afectar la predicción del *target* dentro del modelo predictivo (X. Li et al., 2021). Por esto, dentro de la metodología, se propone hacer la división de los datos en cuatro conjuntos, con el objetivo de calibrar adecuadamente el modelo predictivo sin tener en cuenta estos datos de degradación que pueden generar ruido en los modelos predictivos (X. Li et al., 2021).

Estos datos de degradación se pueden identificar a partir de diferentes métricas, en la metodología se propone la siguiente: la evaluación de los datos a partir de una carta de control evaluada como $\bar{X} \pm L\sigma$, donde L indica 1, 2 o 3 desviaciones estándar. A partir de la evaluación de los datos con la carta de control, se obtienen los datos que no siguen el mismo comportamiento de los datos normales y es necesario su separación para la precisión del modelo (X. Li et al., 2021). Para este caso, al implementar el criterio con la carta de control para una *feature* específica, se observa un comportamiento por fuera de los límites de la media ± L desviaciones estándar seleccionadas, lo que permite catalogar los datos como datos de degradación. Así mismo, como criterio para la identificación de este tipo de datos de degradación, se puede establecer un periodo de tiempo determinado en el que se haya detectado un periodo de fallas, que permita realizar la separación de estos datos para una predicción más precisa (X. Li et al., 2021).

Después de implementar el criterio que permita la identificación de las filas del conjunto $\boldsymbol{X_p^m}$, que muestran un comportamiento diferente y que son llamados datos de degradación k, se procede con la eliminación de estas filas del conjunto de datos $\boldsymbol{X_p^m}$, con el objetivo de tener un conjunto de datos depurados $\boldsymbol{X_p^{m-k}}$. El resultado de este primer paso es un conjunto de datos $\boldsymbol{X_p^{m-k}}$ que se implementa como datos de entrada para el paso 2 y el paso 3 de la división de datos mostrada en la Figura 11 dentro de la metodología propuesta. El procedimiento para llevar a cabo este primer paso dentro de la metodología de división de datos propuesta es el siguiente:

I. Obtener el conjunto de datos $\boldsymbol{X_p^m}$.
II. Definir la métrica de identificación para los datos de degradación (carta de control).
III. Aplicar la métrica definida en el paso II a los datos $\boldsymbol{X_p^m}$.
IV. Identificar las filas de datos de degradación k de acuerdo al paso III.
V. Eliminar los datos de degradación k identificados en el paso IV del conjunto de datos $\boldsymbol{X_p^m}$.
VI. Guardar los datos depurados en el conjunto de datos $\boldsymbol{X_p^{m-k}}$.

Paso 2: Obtener datos para el modelo de suavizamiento

En este paso, se realiza la división de los datos en dos conjuntos: datos de entrenamiento y prueba para el modelo de suavizamiento con la CNN. Para realizar la división de los datos, se considera como datos de entrada, los datos provenientes de sensores $\boldsymbol{X_s^{m-k}}$, los cuales, hacen parte del conjunto de datos $\boldsymbol{X_p^{m-k}}$, como se observa en la Figura 11. Cada una de las *features* de los datos de entrada $\boldsymbol{X_s^{m-k}}$, se estructuran como series de tiempo individuales definiendo el número de pasos ($pasos_s$), este paso, es explicado en la etapa de estructuración de datos del modelo de suavizamiento con la CNN. Luego de la estructuración de los datos de entrada $\boldsymbol{X_s^{m-k}}$, se realiza un proceso de normalización de los datos y se realiza el proceso de división de los mismos en dos conjuntos: datos de entrenamiento $\boldsymbol{Xs_{tr}}$ y prueba $\boldsymbol{Xs_{te}}$ en un porcentaje determinado.

Para llevar a cabo la metodología, se propone una división del conjunto de datos del 70% en datos de entrenamiento $\boldsymbol{Xs_{tr}}$ y 30% en datos de prueba $\boldsymbol{Xs_{te}}$, con el objetivo de evaluar el desempeño del modelo. En este caso, se proponen sólo dos conjuntos, ya que se realiza el suavizamiento de cada una de las *features* de sensores que pertenecen al conjunto $\boldsymbol{X_s^{m-k}}$ y por medio de los dos conjuntos se garantiza el entrenamiento y prueba del modelo de suavizamiento. El procedimiento para llevar a cabo este segundo paso dentro de la metodología de división de datos propuesta es el siguiente:

I. Obtener el conjunto de datos de sensores $\boldsymbol{X_s^{m-k}}$.
II. Determinar el número de *features* obtenidas por sensores s.
III. Estructurar los datos de acuerdo al número de pasos definido $pasos_s$.
IV. Normalizar el conjunto de datos $\boldsymbol{X_s^{m-k}}$ a través de la normalización min-max.
V. Definir el porcentaje de división de los datos para el modelo de suavizamiento.
VI. Dividir el conjunto de datos $\boldsymbol{X_s^{m-k}}$ según el porcentaje definido en el paso V.
VII. Guardar los dos conjuntos de datos generados en el paso VI como datos de entrenamiento $\boldsymbol{Xs_{tr}}$ y datos de prueba $\boldsymbol{Xs_{te}}$.
VIII. Entrenar el modelo de suavizamiento con la CNN para cada *feature* s.
IX. Obtener el conjunto de datos de sensores suavizados para cada *feature* s.
X. Integrar cada conjunto de datos suavizados $\boldsymbol{X_s^{m-k-pasos_s+1}}$ de cada *feature* s.
XI. Guardar el conjunto de datos suavizados $\boldsymbol{X_s^{m-k-pasos_s+1}}$ integrados en el paso X.

Paso 3: Obtener datos para el modelo predictivo

Se realiza la división de los datos para el modelo predictivo con la red LSTM en tres conjuntos: entrenamiento, prueba y validación. En este paso, los datos de entrada corresponden a la unión de los datos

depurados sin sensores $\boldsymbol{X_{p-s}^{m-k}}$, a los cuales, se les debe restar el número de pasos ($pasos_s$) definido en el paso anterior, con el objetivo de normalizar y conformar un conjunto de datos de las mismas dimensiones $\boldsymbol{X_{p-s}^{m-k-pasos_s+1}}$. Este conjunto de datos $\boldsymbol{X_{p-s}^{m-k-pasos_s+1}}$ junto con el conjunto de datos suavizados de sensores $\boldsymbol{X_{s}^{m-k-pasos_s+1}}$, conforman el conjunto de datos de entrada $\boldsymbol{X_{p}^{m-k-pasos_s+1}}$ para el modelo predictivo, como se observa en la Figura 11. El conjunto de datos $\boldsymbol{X_{p}^{m-k-pasos_s+1}}$ se estructura en series con p *features* definiendo el número de pasos ($pasos_t$), la estructuración de los datos es explicada en la etapa de estructuración del modelo predictivo con la red LSTM.

Después de la estructuración de los datos en series con p *features*, se realiza el proceso de división de los datos para el entrenamiento del modelo predictivo, validación y prueba del modelo. Para llevar a cabo la metodología, se propone una división del conjunto de datos en 60% para los datos de entrenamiento $\boldsymbol{X_{tr}}$, 20% para los datos de validación $\boldsymbol{X_{te}}$, y 20% para los datos de prueba $\boldsymbol{X_{va}}$ (J. C. Chen et al., 2021). El procedimiento para llevar a cabo este tercer paso dentro de la metodología de división de datos propuesta es el siguiente:

I. Obtener el conjunto de datos de sensores suavizados $\boldsymbol{X_{s}^{m-k-pasos_s+1}}$.
II. Obtener el conjunto de datos sin sensores $\boldsymbol{X_{p-s}^{m-k-pasos_s+1}}$ depurados obtenidos en el paso 1 de la división de datos de degradación.
III. Normalizar el conjunto de datos $\boldsymbol{X_{p-s}^{m-k-pasos_s+1}}$ a través de la normalización min-max.
IV. Unir ambos conjuntos de datos $\boldsymbol{X_{s}^{m-k-pasos_s+1}}$ y $\boldsymbol{X_{p-s}^{m-k-pasos_s+1}}$ para formar el conjunto de datos $\boldsymbol{X_{p}^{m-k-pasos_s+1}}$.
V. Estructurar los datos $\boldsymbol{X_{p}^{m-k-pasos_s+1}}$ en el número de pasos definido $pasos_t$.
VI. Definir el porcentaje de división de los datos $\boldsymbol{X_{p}^{m-k-pasos_s+1}}$ para el modelo predictivo.
VII. Dividir el conjunto de datos $\boldsymbol{X_{p}^{m-k-pasos_s+1}}$ según el porcentaje definido en el paso VI.
VIII. Guardar los tres conjuntos de datos generados: datos de entrenamiento $\boldsymbol{X_{tr}}$, datos de validación $\boldsymbol{X_{va}}$ y datos de prueba $\boldsymbol{X_{te}}$.
IX. Entrenar el modelo predictivo con la red LSTM.

Luego de la división de datos en cada una de las etapas que se plantean en la metodología, se explican los pasos necesarios para llevar a cabo el modelo de suavizamiento con la CNN, considerando la estructuración de los datos, la definición de la arquitectura planteada para el modelo de suavizamiento, la optimización y selección de los hiperparámetros para realizar el suavizamiento y las predicciones.

5.4 Modelo de suavizamiento con la CNN

Para el modelo de suavizamiento, se implementa la CNN, con el objetivo de tomar los datos obtenidos a partir de sensores o aquellos datos que se identifica tienen ruido en sus mediciones, para realizar el suavizamiento y disminuir el ruido obtenido por diferentes fuentes: el sensor, el entorno, la captura de los datos. Después de este proceso, se obtienen los datos de entrada para la red LSTM suavizados, lo que permite realizar una predicción más precisa (Mejia et al., 2021). En esta sección, se describen los pasos que se llevaron a cabo para la estructuración de los datos de entrada, la definición de la arquitectura de red y la optimización de los hiperparámetros para generar el suavizamiento. En la Figura 12 se muestra la estructura de los pasos necesarios para realizar el suavizamiento de los datos obtenidos por medio de sensores.

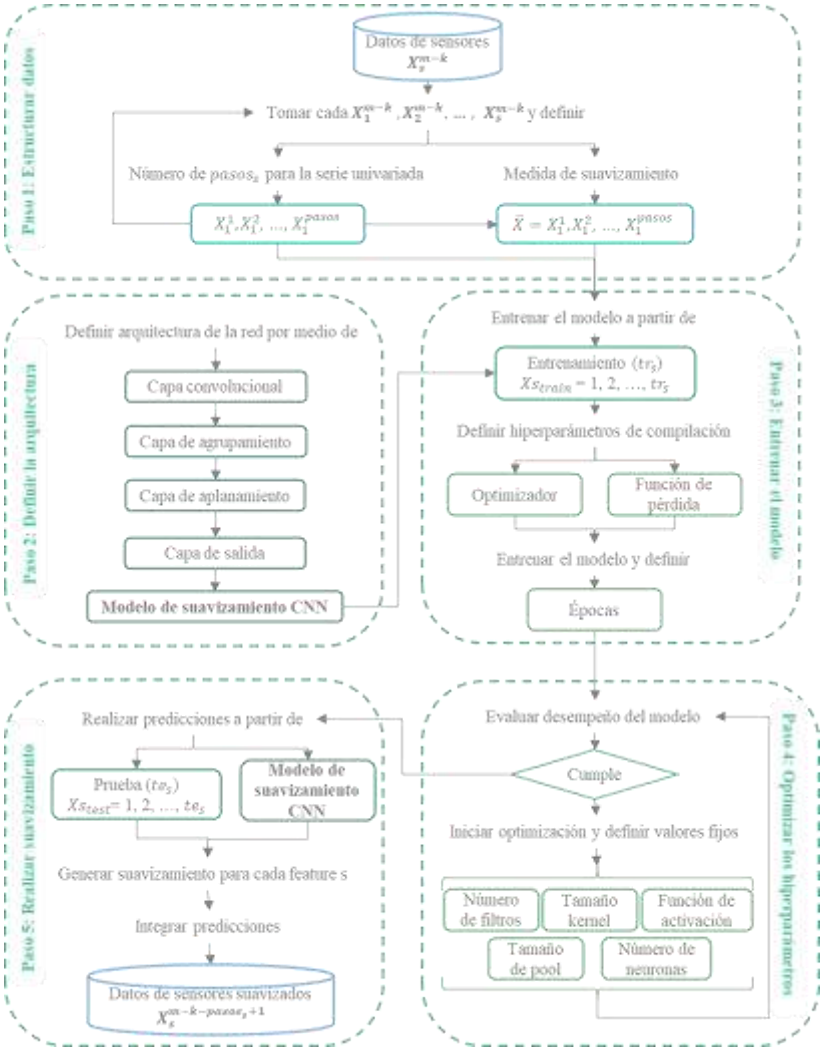


Figura 12. Estructura de los pasos a seguir para realizar el suavizamiento con la CNN

Paso 1: Estructurar datos

Se consideran como datos de entrada los datos provenientes de sensores $\boldsymbol{X_s^{m-k}}$ y que hacen parte del conjunto de datos depurados $\boldsymbol{X_p^{m-k}}$, como se observa en la Figura *11*. A partir de los datos de entrada, se define la cantidad de *features* s que pertenece al conjunto de datos de sensores $\boldsymbol{X_s^{m-k}}$, a las cuales, se les realiza el proceso de estructuración y suavizamiento. Para esto, se toma cada *feature* s y se define el número de $pasos_s$, los cuales representan la cantidad de observaciones a tomar para realizar las series univariadas de cada *feature* s. Estas series univariadas se estructuran en forma de matriz como datos de entrada para el modelo de suavizamiento con la CNN. En este caso, cada una de las *features* s obtenidas por medio de sensores, se toman como series de tiempo individuales que son desintegradas en un conjunto de nuevas series de tiempo que son más fáciles de modelar y suavizar, a partir del número de observaciones definidas de acuerdo al número de $pasos_s$ (Mejia et al., 2021).

Para definir el número de observaciones que se deben tomar para cada *feature* s, es importante tener en cuenta los siguientes criterios:

a. El número mínimo de observaciones que requiere la CNN como entrada es de tres.
b. A menor número de $pasos_s$ (observaciones a tomar), menor pérdida de datos en el suavizamiento.
c. A mayor número de $pasos_s$, es más probable incrementar el error en el suavizamiento debido a la variabilidad de las observaciones (Jo et al., 2021; Mejia et al., 2021).

Por este motivo, en la metodología se plantea tomar tres observaciones como el número de $pasos_s$ requeridas para generar las series univariadas de cada *feature* s, con el objetivo de realizar el proceso de suavizamiento, minimizando la pérdida de los datos iniciales que se requieren y tomando el número mínimo de muestras que se pueden considerar dentro de la CNN. Después de definir el número de $pasos_s$ para hacer el suavizamiento de cada *feature* s, se debe realizar la definición de una función que permita realizar el suavizamiento para obtener el *target* de cada *feature* s. El *target* puede ser definido como el promedio, la mediana, la moda, el último valor de las observaciones tomadas como el número de $pasos_s$ o la observación siguiente (Mejia et al., 2021).

Cada una de las *features* s obtenidas por medio de sensores generan nuevas series de tiempo de acuerdo al número de $pasos_s$ y al *target* definido, los cuales, se estructuran de una forma matricial como entrada para el modelo de suavizamiento. Estos nuevos datos generados para cada *feature* s obtenida por medio de sensores, debe ser dividida en dos conjuntos: datos de entrenamiento $\boldsymbol{Xs_{tr}}$ y prueba $\boldsymbol{Xs_{te}}$, con el objetivo de entrenar el modelo de suavizamiento con la CNN. De esta forma y como un caso de ejemplo, para realizar

el suavizamiento de los datos tomando *s features* de entrada del conjunto de datos $\boldsymbol{X_s^{m-k}}$, la forma matricial de visualizar este conjunto de datos es así:

$$\boldsymbol{X_s^{m-k}} = \begin{bmatrix} x_1^1 & x_2^1 & \cdots & x_s^1 \\ x_1^2 & x_2^2 & \cdots & x_s^2 \\ \vdots & \vdots & \ddots & \vdots \\ x_1^{m-k} & x_2^{m-k} & \cdots & x_s^{m-k} \end{bmatrix}$$

Donde cada una de estas *s features* tiene un número $m-k$ de observaciones, es decir, la primera *s feature* del conjunto de datos $\boldsymbol{X_s^{m-k}}$ está representada por: $\boldsymbol{X_1^{m-k}} = \{x_1^1, x_1^2, \ldots, x_1^{m-k}\}$. Tomando el vector $\boldsymbol{X_1^{m-k}}$ como una *feature s* obtenida por medio de un sensor, se define un número de $pasos_s$ igual a tres observaciones como las nuevas observaciones de entrada para el modelo de suavizamiento, y el *target* como el promedio de las tres observaciones. De esta forma, teniendo en cuenta la ecuación $\boldsymbol{Y} = f(\boldsymbol{X})$, se presenta la forma matricial del ejemplo considerando la estructuración de la primera *s feature* $\boldsymbol{X_1^{m-k}}$ del conjunto de datos $\boldsymbol{X_s^{m-k}}$, los datos para el suavizamiento quedarían así:

$$\begin{bmatrix} y^1 \\ y^2 \\ \vdots \\ y^{m-k-2} \end{bmatrix} = \begin{bmatrix} x_1^1 & x_1^2 & x_1^3 \\ x_1^2 & x_1^3 & x_1^4 \\ \vdots & \vdots & \vdots \\ x_1^{m-k-2} & x_1^{m-k-1} & x_1^{m-k} \end{bmatrix}$$

Donde $y_1 = \frac{x_1^1 + x_1^2 + x_1^3}{3}$ y corresponde al *target* con el que será entrenado el modelo de suavizamiento con la CNN. Es así como se genera el conjunto de datos $\boldsymbol{X_s^{m-k-3+1}} = \boldsymbol{X_s^{m-k-2}}$, donde tres corresponde al número de $pasos_s$, y dos, corresponde al número de observaciones de la *s feature* $\boldsymbol{X_1^{m-k}}$ que se pierden al realizar la estructuración matricial de los datos de entrada para la CNN. Para cada una de las *s features* obtenidas por medio de sensores, se debe realizar este proceso, formando una estructura matricial de dimensión: [$pasos_s$, ($m-k-pasos_s$)], así mismo, la dimensión para el *target* es [1, ($m-k-pasos_s$)]. Luego de la estructuración de los datos de entrada en forma de matriz para la CNN, se debe realizar el proceso de división de los datos en dos conjuntos: datos de entrenamiento $\boldsymbol{Xs_{tr}}$ y prueba $\boldsymbol{Xs_{te}}$ en el porcentaje determinado en el paso 2 de la división de datos. El procedimiento para llevar a cabo la estructuración de los datos dentro de la metodología del modelo de suavizamiento es el siguiente:

I. Obtener el conjunto de datos de sensores $\boldsymbol{X_s^{m-k}}$.
II. Determinar el número *s features* obtenidas por sensores.
III. Definir el número de $pasos_s$ a tomar como número de observaciones para realizar el suavizamiento de cada *s feature*.
IV. Definir la función de suavizamiento para el *target* (promedio, mediana, observación siguiente).

V. Dividir cada *feature* s en una secuencia univariada de un número de $pasos_s$ y el *target* definido en el paso IV.

VI. Convertir en una estructura matricial la división realizada en el paso V.

VII. Guardar el conjunto de datos $\boldsymbol{X_s^{m-k-2}}$ generado de cada variable s.

Paso 2: Definir la arquitectura de la CNN

Para la arquitectura de la CNN se propone tener una capa convolucional, seguida de una capa de agrupamiento (*pooling*), que permita capturar el mapa de las *features* según las matrices de entrada. Seguida de la capa de agrupamiento, se propone una capa de aplanamiento (*flatten*) para transformar las matrices resultantes en vectores, y de esta forma, se extraen las *features* de los datos (Goodfellow et al., 2016; Jo et al., 2021; Mao et al., 2018; Mejia et al., 2021). Luego de esta transformación, se pasan por una serie de capas densas que permiten obtener el suavizamiento. Esta arquitectura propuesta se puede observar en la Figura 13.

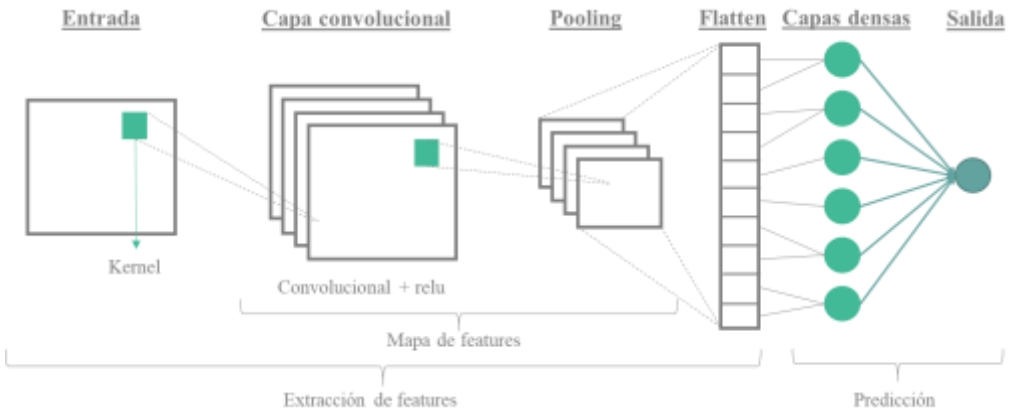


Figura 13. Arquitectura de la CNN

La arquitectura contiene una capa convolucional que permite hacer las convoluciones de los datos de entrada con una función de activación *relu* y un filtro alrededor de la matriz de entrada. Donde el número y el tamaño del filtro son hiperparámetros a optimizar y permiten capturar las relaciones bidimensionales (Pawlus & Devine, 2020; Ren et al., 2021; Zraibi et al., 2021). Seguido de la capa convolucional, se encuentra una capa de *maxpooling*, que actúa como regulador de la red y hace que el modelo sea más robusto a los cambios (Ghatak, 2019; Mao et al., 2018). En esta capa de *maxpoling* se define un tamaño de agrupación o *pool size* para reducir el tamaño de entrada y extraer las características relevantes.

Así mismo, se considera una capa de *flatten* para aplanar los datos de salida, los cuales pasan por dos capas densas, la primera capa densa que permite obtener unos datos suavizados de acuerdo al entrenamiento del

modelo de suavizamiento con la CNN y la otra capa densa que permite brindar como salida el suavizamiento del modelo con la CNN (Zraibi et al., 2021). En la Figura 14 se observa la arquitectura de la CNN definida en la metodología con los valores de los parámetros tomando tres observaciones como $pasos_s$ para el modelo de suavizamiento. Además, como ejemplificación, se consideran 32 filtros de tamaño 2 para la capa convolucional, y 50 neuronas para la primera capa densa.

Layer (type)	Output Shape	Param #
conv1d (Conv1D)	(None, 2, 32)	96
max_pooling1d (MaxPooling1D)	(None, 1, 32)	0
flatten (Flatten)	(None, 32)	0
dense_1 (Dense)	(None, 50)	1650
dense (Dense)	(None, 1)	51

Total params: 1,797

Figura 14. Arquitectura y cantidad de parámetros de la CNN

El procedimiento para definir la arquitectura del modelo de suavizamiento con la CNN es el siguiente:

I. Definir el número de capas convolucionales:
   a. Determinar el valor inicial del número de filtros.
   b. Fijar el tamaño del filtro.
   c. Fijar la función de activación
II. Definir el número de capas de agrupamiento:
   a. Fijar la función de agrupamiento.
   b. Fijar el tamaño del agrupamiento.
III. Definir el número de capas de aplanamiento.
IV. Definir el número de capas densas:
   a. Determinar el valor inicial de neuronas.
   b. Fijar a 1 las neuronas de la capa densa de salida para el suavizamiento.

Paso 3: Entrenar el modelo de suavizamiento

Después de definir la arquitectura de la CNN para el modelo de suavizamiento, se toman los datos de entrenamiento $\boldsymbol{Xs_{tr}}$ generados en el paso 2 de la división de los datos, con el objetivo de realizar el entrenamiento del modelo de suavizamiento. Para esto, se definen los valores iniciales de los hiperparámetros del compilador para el modelo de suavizamiento con la CNN, tales como: el optimizador que se va a implementar y la función de pérdida. En este caso, se selecciona dentro de la metodología para el método de compilación, el optimizador Adam, ya que es uno de los optimizadores que mejor desempeño ha mostrado en la actualidad, según revisión de literatura (Park et al., 2020; Pawlus & Devine, 2020). Para

la tasa de aprendizaje, se selecciona el valor inicial de 0.001, el cual corresponde al valor definido por defecto, y se probaran más valores según la optimización de hiperparámetros descrita en el paso 4.

Dentro de la compilación del modelo de suavizamiento, se define como función de pérdida el MSE, ya que es la función de pérdida que implementan varios autores, ya que permite una penalización mayor a los errores más grandes al elevar al cuadrado dichos errores y genera mejores predicciones en un modelo de regresión (J. C. Chen et al., 2021; Ghatak, 2019; X. Hu, Yang, et al., 2021; Jo et al., 2021; Y. Liu et al., 2019; Pawlus & Devine, 2020; Qiao et al., 2020; Ren et al., 2018). De esta forma, se realiza la compilación del modelo de suavizamiento y se define como número de épocas iniciales el valor de 500, ya que es uno de los valores que se han usado en otros estudios (J. C. Chen et al., 2021; X. Hu, Yang, et al., 2021; Park et al., 2020). Se entrena el modelo de suavizamiento y se validan los resultados para iniciar con la optimización de los hiperparámetros. El procedimiento para llevar a cabo el entrenamiento del modelo de suavizamiento es el siguiente:

I. Obtener datos de entrenamiento $\boldsymbol{Xs_{tr}}$.
II. Definir el optimizador Adam para la compilación del modelo.
III. Definir como función de pérdida para la compilación del modelo el MSE.
IV. Definir como valor inicial para la tasa de aprendizaje el valor de 0.001.
V. Compilar el modelo de suavizamiento.
VI. Definir el valor inicial de las épocas como 500.
VII. Realizar el entrenamiento del modelo de suavizamiento con las definiciones anteriores y los datos $\boldsymbol{Xs_{tr}}$.

Paso 4: Optimizar los hiperparámetros del modelo de suavizamiento

Para este paso, se definen unos hiperparámetros como fijos y otros variables de acuerdo a los valores encontrados en la revisión de literatura. Para los hiperparámetros variables, se considera una búsqueda en cuadrícula (*grid search*), donde se especifica manualmente los valores de los hiperparámetros que el modelo puede tomar. A partir de la búsqueda en cuadrícula, se espera determinar la mejor combinación de valores de los hiperparámetros en la que el desempeño del modelo de suavizamiento es mejor para las variables obtenidas por medio de sensores.

Es así como se tienen en cuenta los hiperparámetros que intervienen en la creación de la arquitectura de la CNN para el modelo de suavizamiento, así como también, los hiperparámetros relacionados con la compilación y el entrenamiento del modelo. En la Tabla 9 se presentan los valores fijos de los hiperparámetros para cada una de las etapas que se deben llevar a cabo para el modelo de suavizamiento a

partir de la CNN. Estos valores se definen como fijos para cada uno de los hiperparámetros, ya que, según revisión de literatura, estos valores han presentado un buen desempeño en el modelo.

Tabla 9

Valores de los hiperparámetros fijos para la optimización en la CNN

Etapa	Hiperparámetro	Valor	Autores
Datos	Número de datos	3	(Mejia et al., 2021)
Arquitectura	Número de capas convolucionales	1	(Goodfellow et al., 2016; Mao et al., 2018; Mejia et al., 2021)
	Función de activación	*relu*	(Ghatak, 2019; Zraibi et al., 2021)
	Número de capas de agrupación	1	(Goodfellow et al., 2016; Mao et al., 2018; Mejia et al., 2021)
	Tamaño del filtro en la capa convolucional	2	(Ghatak, 2019; Mao et al., 2018; Zraibi et al., 2021)
	Función de agrupación	*max-pooling*	(Ghatak, 2019; Zraibi et al., 2021)
	Tamaño de *pool* en la capa de agrupación	2	(Ghatak, 2019; Mao et al., 2018)
	Número de capas de aplanamiento	1	(Mao et al., 2018; Ren et al., 2021; Zraibi et al., 2021)
	Número de capas densas	2	(Zraibi et al., 2021)
	Número de neuronas en la segunda capa densa	1	(Zraibi et al., 2021)
Compilación	Optimizador	Adam	(Aggarwal, 2018; J. C. Chen et al., 2021; X. Hu, Yang, et al., 2021; Jo et al., 2021; Park et al., 2020; Qiao et al., 2020; Ren et al., 2021; Zraibi et al., 2021)
	Función de pérdida	MSE	(J. C. Chen et al., 2021; Ghatak, 2019; X. Hu, Yang, et al., 2021; Jo et al., 2021; Y. Liu et al., 2019; Qiao et al., 2020; Ren et al., 2018)

Así mismo, se definen los diferentes valores de los hiperparámetros en cada una de las etapas que serán evaluados a partir de la búsqueda en cuadrícula, con el objetivo de obtener la mejor combinación de valores de los hiperparámetros para mejorar el desempeño del modelo de suavizamiento. Estos valores son presentados en la Tabla 10. En este caso, se generan 24 combinaciones que corresponden a 24 modelos que serán evaluados para determinar la mejor combinación de hiperparámetros que permite obtener el mejor desempeño del modelo de suavizamiento.

Tabla 10

Hiperparámetros y valores para la optimización en la CNN

Etapa	Hiperparámetro	Valores	Autores

Arquitectura	Número de filtros en la capa convolucional	16, 32, 64	(Ghatak, 2019; Mao et al., 2018; Zraibi et al., 2021)
	Número de neuronas en la primera capa densa	50, 100	(J. C. Chen et al., 2021; X. Hu, Yang, et al., 2021; Qiao et al., 2020)
Compilación	Tasa de aprendizaje	0.001, 0.01	(X. Hu, Yang, et al., 2021; Jo et al., 2021; Park et al., 2020)
Entrenamiento	Épocas	100, 500	(J. C. Chen et al., 2021; X. Hu, Yang, et al., 2021; Park et al., 2020)

Después de realizar la búsqueda en cuadrícula para determinar los valores de cada hiperparámetro que permite un mejor desempeño en el suavizamiento, se entrena nuevamente el modelo de suavizamiento para realizar el suavizamiento de las *features* que son entrada para el modelo predictivo con la CNN. El procedimiento para llevar a cabo la optimización de los hiperparámetros del modelo de suavizamiento es el siguiente:

I. Definir el valor de los hiperparámetros fijos.
II. Definir el valor de los hiperparámetros variables.
III. Para cada hiperparámetro variable:
    a. Generar la arquitectura del modelo de suavizamiento.
    b. Compilar el modelo de suavizamiento.
    c. Entrenar el modelo con los datos $\boldsymbol{Xs_{tr}}$.
    d. Calcular el desempeño del modelo de acuerdo a los datos $\boldsymbol{Xs_{tr}}$.
    e. Calcular el desempeño del modelo de acuerdo a los datos $\boldsymbol{Xs_{te}}$.
IV. Determinar la mejor combinación de hiperparámetros que mejora el desempeño del modelo.

Paso 5: Realizar suavizamiento

Luego de la optimización de los hiperparámetros presentados en el paso anterior, se realiza el suavizamiento de cada *s feature* para obtener los datos de sensores suavizados de acuerdo a la mejor combinación de hiperparámetros. Con esta combinación, se fijan los valores de los hiperparámetros para generar la arquitectura, la compilación y el entrenamiento del modelo con la CNN, el cual, permite generar el suavizamiento. El proceso de suavizamiento se realiza para cada una de las *features s* que se tienen del conjunto de datos $\boldsymbol{X_s^{m-k}}$. Así mismo, se calcula el desempeño del modelo de acuerdo a métricas como el RMSE, MAPE, MAE y MSE para cada una de las *s features* suavizadas. Después de generar el suavizamiento, se integran los resultados de cada una de las *s features* en un nuevo conjunto de datos $\boldsymbol{X_s^{m-k-2}}$. Este conjunto de datos pertenece a uno de los conjuntos de datos de entrada del modelo predictivo con la red LSTM. El procedimiento para llevar a cabo el suavizamiento de las *s features* es el siguiente:

I. Tomar cada *s feature* del conjunto de datos $\boldsymbol{X_s^{m-k}}$.
II. Definir y fijar el valor de los hiperparámetros de acuerdo a la mejor combinación obtenida en el paso 4:
    a. Generar la arquitectura del modelo.
    b. Realizar la compilación del modelo.
    c. Entrenar el modelo de acuerdo al valor de los hiperparámetros y los datos de la *s feature*.
III. Calcular el desempeño del modelo de suavizamiento a través de métricas como el RMSE, MAPE, MAE y MSE.
IV. Guardar los resultados de la *s feature* suavizada.
V. Integrar los datos de cada *s feature* suavizada en el conjunto de datos $\boldsymbol{X_s^{m-k-2}}$.

5.5 Modelo predictivo con la red LSTM

Para el modelo predictivo se implementa una red LSTM con estructura de muchos a uno, la cual, permita capturar el comportamiento de las *features*, con las cuales se realiza la predicción del *target*, caracterizado por presentar un comportamiento no lineal (Ge et al., 2021; X. Hu et al., 2020). En esta sección, se describen los pasos necesarios para llevar a cabo el modelo predictivo a partir de la red LSTM, considerando la estructuración de los datos, la definición de la arquitectura de la red, el entrenamiento del modelo, la optimización de los hiperparámetros y la predicción del *target*. En la Figura 15 se muestran los pasos a seguir para realizar la predicción a partir del modelo predictivo con la red LSTM.

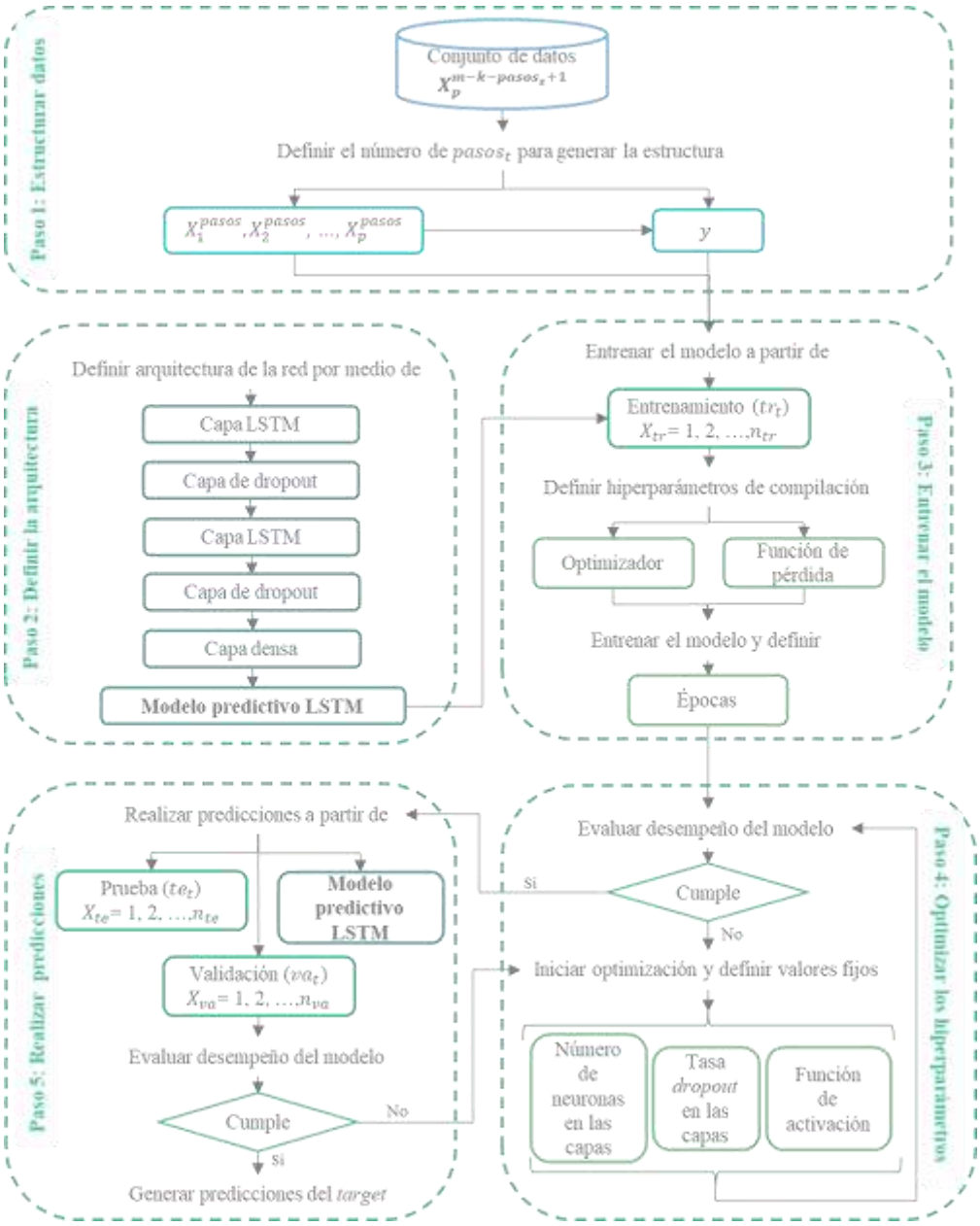


Figura 15. Estructura de los pasos a seguir para realizar la predicción con la red LSTM

Paso 1: Estructurar datos

Se consideran dos tipos de datos de entrada para el modelo predictivo con la red LSTM: los datos suavizados $\boldsymbol{X_s^{m-k-2}}$ obtenidos del modelo de suavizamiento explicados en el paso anterior y los datos sin sensores depurados $\boldsymbol{X_{p-s}^{m-k-2}}$ obtenidos en el paso 1, como se observa en la Figura *11*. Los dos conjuntos de datos de entrada se integran en un nuevo conjunto de datos $\boldsymbol{X_p^{m-k-2}}$ que debe ser dividido en nuevas series de datos que considera el número de *features* necesarias para la predicción y considera el número de $pasos_t$ definido.

En este caso, se define el número de $pasos_t$ para hacer la predicción igual a diez, ya que este número de pasos ha sido usado en la literatura para realizar predicciones con la red LSTM (Park et al., 2020).

Luego de hacer la definición del número de $pasos_t$ y generar las nuevas series de datos con las *p features* y el *y target*, se genera la estructuración en forma matricial de los datos $\boldsymbol{X_p^{m-k-2}}$, teniendo en cuenta que la red LSTM recibe los datos de entrada como matrices de la siguiente forma: [número de series = (número de filas – número de $pasos_t$ + 1), número de $pasos_t$, número de *p features*]. De esta forma, se considera un conjunto de datos $\boldsymbol{X_p^{m-k-2}}$, que contiene los datos obtenidos sin sensores $\boldsymbol{X_{p-s}^{m-k-2}}$ y los datos suavizados $\boldsymbol{X_s^{m-k-2}}$ obtenidos a partir del modelo de suavizamiento con la CNN. Este conjunto de datos $\boldsymbol{X_p^{m-k-2}} = \{\boldsymbol{X_1^{m-k-2}}, \boldsymbol{X_2^{m-k-2}}, \dots, \boldsymbol{X_s^{m-k-2}}, \boldsymbol{X_{p-s}^{m-k-2}}\}$ cuenta con *p features* y el *y target* definido. Teniendo en cuenta la ecuación $\boldsymbol{Y} = f(\boldsymbol{X})$, la forma matricial de visualizar los datos es así:

$$\begin{bmatrix} y^1 \\ y^2 \\ \vdots \\ y^{m-k-2} \end{bmatrix} = f\left[\boldsymbol{X_p^{m-k-2}}\right] \; donde \; \boldsymbol{X_p^{m-k-2}} = \begin{bmatrix} x_1^1 & x_2^1 & \cdots & x_s^1 & x_{p-s}^1 \\ x_1^2 & x_2^2 & \cdots & x_s^2 & x_{p-s}^2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ x_1^{m-k-2} & x_2^{m-k-2} & \cdots & x_s^{m-k-2} & x_{p-s}^{m-k-2} \end{bmatrix}$$

Donde cada una de las *p features* tiene un número $m-k-2$ de observaciones, es decir, la primera *feature* del conjunto de datos $\boldsymbol{X_p^{m-k-2}}$ está representada por: $\boldsymbol{x_1^{m-k-2}} = \{x_1^1, x_1^2, \dots, x_1^{m-k-2}\}$. Teniendo en cuenta las p *features* y un número de $pasos_t$ de diez, según revisión de literatura, se tendría una matriz con dimensiones [$(m-k-2-\ pasos_t+1)$, $pasos_t$, p *features*], donde la primera, segunda y última matriz están conformadas de la siguiente manera:

$$[y^1] = f\begin{bmatrix} x_1^1 & x_2^1 & \cdots & x_{p-s}^1 \\ x_1^2 & x_2^2 & \cdots & x_{p-s}^2 \\ \vdots & \vdots & \ddots & \vdots \\ x_1^{10} & x_2^{10} & \cdots & x_{p-s}^{10} \end{bmatrix}, [y^2] = f\begin{bmatrix} x_1^2 & x_2^2 & \cdots & x_{pt}^2 \\ x_1^3 & x_2^3 & \cdots & x_{pt}^3 \\ \vdots & \vdots & \ddots & \vdots \\ x_1^{11} & x_2^{11} & \cdots & x_{pt}^{11} \end{bmatrix}, \dots,$$

$$[y^{m-k-2-10+1}] = f\begin{bmatrix} x_1^{m-k-2-9} & x_2^{m-k-2-9} & \cdots & x_{p-s}^{m-k-2-9} \\ x_1^{m-k-2-8} & x_2^{m-k-2-8} & \cdots & x_{p-s}^{m-k-2-8} \\ \vdots & \vdots & \ddots & \vdots \\ x_1^{m-k-2} & x_2^{m-k-2} & \cdots & x_{p-s}^{m-k-2} \end{bmatrix}$$

Donde $y^{m-k-2-10+1}$ corresponde al *target* que se va a predecir por medio del entrenamiento del modelo predictivo con la red LSTM. Es así como se genera el conjunto de datos $\boldsymbol{X_p^{m-k-2-10+1}} = \boldsymbol{X_p^{m-k-11}}$ de las series que contiene los datos de entrada en la forma específica para la red LSTM. Después de realizar las series, se debe realizar un proceso de división de los datos para el entrenamiento del modelo predictivo. El procedimiento para llevar a cabo la estructuración de los datos para el modelo predictivo es el siguiente:

I. Obtener el conjunto de datos de $\boldsymbol{X_p^{m-k-pasos_s+1}}$.

II. Definir el número de $pasos_t$ a tomar para la división de la serie.

III. Dividir el conjunto de datos $\boldsymbol{X_p^{m-k-pasos_s+1}}$ en una secuencia de acuerdo al número de pasos seleccionado en el paso II.

IV. Convertir en una estructura matricial la división realizada en el paso III.

Paso 2: Definir la arquitectura de la red LSTM

La arquitectura considera dos capas LSTM, que permiten capturar las dependencias de corto y largo plazo en la predicción (Park et al., 2020; Zraibi et al., 2021). La primera capa LSTM, contiene un número de neuronas que cambiará de acuerdo a la optimización de hiperparámetros definidos en el paso 3 y una función de activación igual a *tanh*. Así mismo, en esta primera capa se genera una secuencia de retorno que genera todos los estados ocultos de cada paso del tiempo y brinda como salida una matriz en las mismas dimensiones de entrada para la segunda capa LSTM. Entre cada una de las capas LSTM, se considera dentro de la arquitectura dos capas de *dropout* como método de regularización para controlar el sobreajuste que pueda tener el modelo predictivo. Para estas capas *dropout*, se considera como hiperparámetro la tasa de abandono, para las cuales, se definirá un valor inicial de la tasa y se determinan diferentes valores como tasa en la búsqueda en cuadrícula desarrollada para la optimización de hiperparámetros, con el objetivo de determinar la tasa que mejor desempeño genera en el modelo predictivo (J. C. Chen et al., 2021; Y. Liu et al., 2019; Park et al., 2020). Además, se tiene en cuenta que el número de neuronas en la primera capa LSTM podría ser mayor a la segunda capa LSTM, ya que esto permitirá capturar mayor información en la primera capa de entrada.

Luego de generar la segunda capa LSTM y la segunda capa de *dropout* que controlará el sobreajuste de la red, se agrega a la arquitectura del modelo una capa densa que permite realizar la predicción de la capacidad, esta capa se caracteriza por tener una neurona, la cual se encarga de generar un dato de salida. De esta forma, con la arquitectura planteada para el modelo predictivo, se obtiene una estructura de muchos a uno (Park et al., 2020). En la Figura 16 se presenta la arquitectura y el número de parámetros necesarios para el entrenamiento del modelo predictivo con la red LSTM, considerando un número de pasos de diez para obtener las series con p *features*, como se explicó en el paso de división de datos. Además, como ejemplificación, se toma un valor inicial de 128 neuronas para la primera capa de la LSTM y 64 neuronas para la segunda capa LSTM.

```
 Layer (type)                Output Shape              Param #
=================================================================
 lstm (LSTM)                 (None, 10, 128)           70144

 dropout (Dropout)           (None, 10, 128)           0

 lstm_1 (LSTM)               (None, 64)                49408

 dropout_1 (Dropout)         (None, 64)                0

 dense_2 (Dense)             (None, 1)                 65

=================================================================
Total params: 119,617
```

Figura 16. Arquitectura y cantidad de parámetros para la red LSTM

El procedimiento para definir la arquitectura del modelo predictivo con la red LSTM es el siguiente:

I. Definir el número de capas LSTM:
   a. Determinar el valor inicial del número de neuronas.
   b. Fijar la función de activación
II. Definir el número de capas *dropout*:
   a. Determinar el valor inicial de la tasa de *dropout*.
III. Definir el número de capas densas:
   a. Fijar a 1 las neuronas de la capa densa de salida para la predicción.

Paso 3: Entrenar el modelo predictivo

Después de definir la arquitectura de la red LSTM para el modelo predictivo, se realiza el entrenamiento del modelo a partir de los datos de entrenamiento $\boldsymbol{X_{tr}}$ generados en el paso 3 de la división de datos. Para el entrenamiento, también se definen los valores iniciales de los hiperparámetros del compilador del modelo, tales como: el optimizador y la función de pérdida que serán implementadas para hacer el entrenamiento inicial del modelo. Para la red LSTM, se plantea el optimizador Adam, ya que, según revisión de literatura, es uno de los optimizadores que mejor desempeño ha mostrado en la evaluación de los modelos predictivos (Park et al., 2020; Pawlus & Devine, 2020). Dentro del compilador, se debe definir el valor inicial de la tasa de aprendizaje, la cual se inicializa con el valor de 0.001, y se plantea realizar una búsqueda en cuadrícula con un conjunto de valores de los hiperparámetros para la optimización de cada uno, el cual, es explicado en el siguiente paso.

Así mismo, para realizar la compilación del modelo predictivo con la red LSTM, se define como función de pérdida el MSE, ya que es la función de pérdida que mejor desempeño ha demostrado en los modelos predictivos de regresión, como se ha mencionado anteriormente (J. C. Chen et al., 2021; Ghatak, 2019; X.

Hu, Yang, et al., 2021; Jo et al., 2021; Y. Liu et al., 2019; Pawlus & Devine, 2020; Qiao et al., 2020; Ren et al., 2018). Teniendo en cuenta la definición de los hiperparámetros para la compilación inicial del modelo predictivo, se define el número de épocas para realizar el entrenamiento del modelo igual a 500, ya que es uno de los valores que se han implementado en el entrenamiento, según revisión de literatura (J. C. Chen et al., 2021; X. Hu, Yang, et al., 2021; Park et al., 2020). El valor de las épocas, es un valor inicial y se plantean diferentes valores para evaluar el desempeño del modelo en el paso de optimización de los hiperparámetros para realizar la selección de los mejores valores para éstos. Es así, como se realiza el entrenamiento inicial del modelo y se continua con la etapa de optimización como lo muestra la Figura 15. El procedimiento para llevar a cabo el entrenamiento del modelo predictivo es el siguiente:

I. Obtener datos de entrenamiento $\boldsymbol{X_{tr}}$.
II. Definir el optimizador Adam para la compilación del modelo predictivo.
III. Definir como función de pérdida para la compilación del modelo el MSE.
IV. Definir como valor inicial para la tasa de aprendizaje el valor de 0.001.
V. Compilar el modelo predictivo con la red LSTM.
VI. Definir el valor inicial de las épocas como 500.
VII. Realizar el entrenamiento del modelo predictivo con las definiciones anteriores y los datos de entrenamiento $\boldsymbol{X_{tr}}$.

Paso 4: Optimizar los hiperparámetros del modelo predictivo

Con la arquitectura de la red definida y los valores iniciales de los hiperparámetros para la compilación y el entrenamiento inicial del modelo predictivo con la red LSTM, se procede a la definición de los valores de los hiperparámetros a optimizar. Se definen los hiperparámetros que son necesarios validar con una búsqueda en cuadrícula, con el objetivo de tomar aquellos valores que permitan mejorar el desempeño en el modelo predictivo. De esta forma, se definen unos hiperparámetros como fijos y otros como variables, en los que se determinan unos valores de acuerdo a la revisión de literatura realizada. En la Tabla 11 se observan los valores de los hiperparámetros fijos del modelo predictivo, en esta, se plantean los valores para cada uno de los hiperparámetros que intervienen en la generación de los datos, la arquitectura y la compilación del modelo predictivo. Estos valores se definen como fijos para cada uno de los hiperparámetros presentados, ya que, según los resultados de diferentes trabajos de investigación, han presentado un buen desempeño en el modelo predictivo.

Tabla 11

Valores de los hiperparámetros fijos para la optimización en la red LSTM

Etapa	Hiperparámetro	Valor	Autores
Datos	Número de pasos	10	(Park et al., 2020)
Arquitectura	Número de capas LSTM	2	(J. C. Chen et al., 2021)
	Secuencia de retorno primera capa LSTM	Sí	(Zraibi et al., 2021)
	Función de activación	tanh	(J. C. Chen et al., 2021)
	Número de capas *dropout*	2	(J. C. Chen et al., 2021)
	Número de capas densas	1	(J. C. Chen et al., 2021; X. Hu, Yang, et al., 2021; Y. Liu et al., 2019)
	Número de neuronas en la segunda capa densa	1	(J. C. Chen et al., 2021; X. Hu, Yang, et al., 2021; Qiao et al., 2020)
Compilación	Optimizador	Adam	(Aggarwal, 2018; J. C. Chen et al., 2021; X. Hu, Yang, et al., 2021; Jo et al., 2021; Park et al., 2020; Qiao et al., 2020; Ren et al., 2021; Zraibi et al., 2021)
	Función de pérdida	MSE	(J. C. Chen et al., 2021; Ghatak, 2019; X. Hu, Yang, et al., 2021; Jo et al., 2021; Y. Liu et al., 2019; Qiao et al., 2020; Ren et al., 2018)

Luego de la definición de los valores fijos para el modelo predictivo con la red LSTM, se definen los valores que se toman como el conjunto especificado para la búsqueda en cuadrícula y que permiten generar la optimización de los hiperparámetros. El valor de cada uno de los hiperparámetros que cambia en la arquitectura, compilación y entrenamiento del modelo predictivo, se observa en la Tabla 12. En este caso, también se generan 144 combinaciones que corresponden a 144 modelos que serán evaluados con cada uno de los valores propuestos para determinar la mejor combinación de hiperparámetros que permite obtener el mejor desempeño del modelo predictivo con la red LSTM.

Tabla 12

Hiperparámetros y valores para la optimización en la red LSTM

Etapa	Hiperparámetro	Valores	Autores
Arquitectura	Número de neuronas primera capa LSTM	64, 128	(J. C. Chen et al., 2021)
	Tasa de *dropout* primera capa	0.1, 0.2, 0.3	(J. C. Chen et al., 2021; Ghatak, 2019; Y. Liu et al., 2019)
	Número de neuronas segunda capa LSTM	64, 128	(J. C. Chen et al., 2021)
	Tasa de *dropout* segunda capa	0.1, 0.2, 0.3	(J. C. Chen et al., 2021; Ghatak, 2019; Y. Liu et al., 2019)
Compilación	Tasa de aprendizaje	0.001, 0.01	(X. Hu, Yang, et al., 2021; Jo et al., 2021; Park et al., 2020)
Entrenamiento	Épocas	100, 500	(J. C. Chen et al., 2021; X. Hu, Yang, et al., 2021; Park et al., 2020)

El procedimiento para llevar a cabo la optimización de los hiperparámetros del modelo predictivo con la red LSTM es el siguiente:

IV. Definir el valor de los hiperparámetros fijos.
V. Definir el valor de los hiperparámetros variables.
VI. Para cada hiperparámetro variable:
    a. Generar la arquitectura del modelo predictivo.
    b. Compilar el modelo predictivo.
    c. Entrenar el modelo con los datos $\boldsymbol{X_{tr}}$.
    d. Calcular el desempeño del modelo predictivo según los datos $\boldsymbol{X_{tr}}$.
    e. Calcular el desempeño del modelo predictivo según los datos $\boldsymbol{X_{va}}$.
VII. Determinar la mejor combinación de hiperparámetros que mejora el desempeño del modelo predictivo.

Paso 5: Realizar predicciones

Luego de la definición de la arquitectura, la compilación y el entrenamiento inicial del modelo predictivo, se procede con la optimización de los hiperparámetros, explicados en los pasos anteriores. Después de la optimización de los hiperparámetros, se evalúa el modelo predictivo con los datos de prueba, con el objetivo de determinar si se requiere de la optimización de alguno de los hiperparámetros nuevamente o si el desempeño del modelo predictivo es el adecuado en sus predicciones. Para llevar a cabo la evaluación del desempeño del modelo predictivo, se pueden usar diferentes métricas que deben estar acompañadas de un análisis gráfico que sirva de apoyo para visualizar los resultados obtenidos. Las métricas implementadas para la evaluación del desempeño del modelo predictivo son el RMSE, MAPE, MAE y MSE.

De esta forma, se define la mejor combinación de valores de los hiperparámetros que permite obtener la predicción del *target* necesario a través de la integración de un modelo de suavizamiento y un modelo predictivo que captura el comportamiento del *target*. El procedimiento para llevar a cabo la predicción del *target* a partir de una red LSTM y la combinación de los conjuntos de datos $\boldsymbol{X_s^{m-k-2}}$ y $\boldsymbol{X_{p-s}^{m-k-2}}$ el siguiente:

I. Integrar los conjuntos de datos $\boldsymbol{X_s^{m-k-2}}$ y $\boldsymbol{X_{p-s}^{m-k-2}}$ en el conjunto de datos $\boldsymbol{X_p^{m-k-2}}$.
II. Definir y fijar el valor de los hiperparámetros de acuerdo a la mejor combinación obtenida en el paso 4:
    a. Generar la arquitectura del modelo predictivo.
    b. Realizar la compilación del modelo predictivo.
    c. Entrenar el modelo predictivo de acuerdo al valor de los hiperparámetros y los datos $\boldsymbol{X_p^{m-k-2}}$.

III. Generar la predicción del *target*.

IV. Calcular el desempeño del modelo predictivo a través de métricas como el RMSE, MAPE, MAE y MSE.

5.6 Paquete en lenguaje de programación R – `cnnlstm`

En esta sección se detalla el paquete desarrollado en el lenguaje de programación de R para la implementación de la metodología propuesta para el modelo predictivo presentada anteriormente. El paquete se llama `cnnlstm` y se puede consultar en la página web https://cjrincon.github.io/cnnlstm/ o en el siguiente repositorio: https://github.com/cjrincon/cnnlstm. El paquete se basa en `TensorFlow`, una biblioteca de código abierto para el cálculo numérico mediante gráficos dirigidos. El paquete `cnnlstm` usa el `TensorFlow` creado por Al-laire & Tang (2021) para integrar `TensorFlow` en el entorno R. Además, el paquete `cnnlstm` utiliza `Keras`, una API de redes neuronales de código abierto fácil de usar y ampliamente usada, la cual, proporciona una interfaz de alto nivel para crear y entrenar modelos de aprendizaje profundo (Chollet et al., 2015). La documentación completa de cada una de las funciones que ofrece el paquete `cnnlstm` se puede observar en la página web.

Para usar el paquete `cnnlstm`, se recomienda seguir los siguientes pasos:

1. Instalar el paquete `Reticulate`
    a. install.packages("reticulate")
    b. library(reticulate)
2. Instalar el paquete `Tensorflow`
    a. install.packages("remotes")
    b. remotes::install_github("rstudio/tensorflow"))
    c. library(tensorflow)
3. A continuación, valide si Python está instalado en el sistema, si no está instalado, se puede descargar: www.python.org/download o llamar
    a. reticulate::install_python()
4. Instalar el módulo `TensorFlow` y crear un ambiente virtual aislado llamado “r-tensorflow”
    a. install_tensorflow(envname = "r-tensorflow")
5. Instalar el paquete `keras`
    a. install.packages("keras")
    b. library(keras)
    c. install_keras()

6. Confirmar la instalación exitosa de `Tensorflow`
    a. tf$constant("Hello Tensorflow!")
7. Instalar el paquete `devtools`
    a. install.packages("devtools")
8. Instalar el paquete `cnnlstm` desde GitHub
    a. devtools::install_github("cjrincon/cnnlstm")
    b. library(cnnlstm)

El objetivo del paquete es ofrecer las funciones necesarias para desarrollar la metodología propuesta del modelo predictivo, incluyendo funciones para el preprocesamiento de los datos, la normalización, la división de los datos, la preparación de los datos en matrices como entrada para cada uno de los modelos de predicción, usando la CNN y la red LSTM. Así mismo, el paquete ofrece funciones para la creación de modelos con diferentes valores de hiperparámetros según la selección del usuario y funciones para realizar la evaluación del desempeño de los modelos.

Las funciones que comprende este paquete pueden implementarse en conjunto, con el fin de llevar a cabo la metodología propuesta, en la que se tienen en cuenta datos obtenidos por sensores que deben ser suavizados. Para realizar este proceso, se proponen funciones para realizar el proceso de normalización de los datos a partir de una normalización min-max que permite realizar la transformación de los datos, así como también, la inversa de la transformación para obtener los datos en la escala real de cada una de las variables. Así mismo, el paquete ofrece una función para organizar los datos en las matrices necesarias, que sirven de entrada para el modelo de suavizamiento con la CNN, para el cual, también existe una función que permite la selección de los valores de los hiperparámetros en cada una de las capas que contiene la arquitectura del modelo de suavizamiento con la CNN definida en el capítulo anterior. En este caso, si no se seleccionan los valores para los hiperparámetros del modelo con la CNN, la función tomará por defecto los valores de los hiperparámetros presentados en la metodología descrita en el capítulo anterior. Además, el paquete cuenta con funciones para realizar la evaluación del desempeño del modelo de suavizamiento por medio de la CNN.

A partir de los resultados obtenidos con el modelo de suavizamiento de la CNN de los datos que requerían de este proceso, el paquete ofrece una función para realizar la adecuación de los datos necesarios para el modelo predictivo a partir de una red LSTM, en la cual, los datos se organizan en formas de matrices según el número de pasos definido, separando las *features* y el *target* respectivo. Así mismo, se presenta una función que permite digitar el valor de los hiperparámetros para la arquitectura de la red LSTM, según elección del usuario. En este caso, también se cuenta con la opción de no digitar ningún valor, y es así, como

la función tomará los valores de los hiperparámetros que se explicaron en la metodología del capítulo anterior.

El paquete ofrece flexibilidad en cuanto al uso de sus funciones, ya que si se tiene una situación en la que sólo se requiere de un modelo de suavizamiento y se quiere implementar y evaluar un modelo a partir de una CNN, se puede hacer uso de las funciones específicas que permitan obtener como resultado datos suavizados según lo requerido. Esto también aplica para el modelo predictivo implementado con la red LSTM, para el cual, se puede hacer uso de las funciones propuestas para la estructuración de los datos, la definición del valor de los hiperparámetros dentro de la arquitectura de la red LSTM, así como la evaluación del desempeño del modelo predictivo.

Este paquete se diferencia de los demás, ya que ofrece diferentes funciones que permiten realizar el suavizamiento por medio de un modelo con una CNN y una predicción precisa por medio de una red LSTM. Dentro de las funciones se encuentra la función ***conv_array***, la cual permite como entrada una matriz o dataframe para transformarlo en un array con dimensiones de acuerdo al número de filas y columnas que se tengan en la variable de entrada. El paquete también cuenta con dos funciones que se encargan de realizar la normalización min-max, determinando el rango de las variables de entrada por medio de la función ***min_max_range***, que sirve de insumo para la función ***norm_min_max***. Esta función requiere como parámetros la matriz o dataframe que se requiere normalizar, la información del rango de la matriz presentada en el primer argumento y la definición del argumento *scaler*, el cual, tiene dos opciones: *transform* para realizar la normalización de los datos e *inverse* que ofrece la transformación inversa de los datos para tenerlos en la escala real.

El paquete también ofrece una función llamada ***split_unv***, que sirve para realizar una división univariada de los datos. Esta función permite preparar los datos de entrada en series univariadas para el modelo de suavizamiento a partir de la CNN. La función recibe como argumentos la variable a configurar y el número de pasos necesarios para hacer la división de la variable entre *features* y *target.* Los *features* corresponderán al número de muestras definido en el argumento de número de pasos de la variable, y el *target*, será una función determinada por el usuario. En la metodología explicada en el capítulo anterior se describen algunas de estas funciones que pueden ser implementadas en la función, tales como: media, mediana, moda, último dato, entre otras elegidas por el usuario para realizar el suavizamiento de los datos por medio de la CNN.

Así mismo, el paquete ofrece una función para realizar la división con *p features* de un conjunto de datos. En este caso, la función se llama ***split_seq*** y permite preparar los datos de entrada para el modelo predictivo mediante la red LSTM. Los parámetros que recibe la función son: *x*, que corresponde al conjunto de datos de las *features*, *y*, que corresponde al *target*, y como último parámetro, se recibe el número de pasos

requeridos para realizar la división de la secuencia de datos. Como resultado, la función brinda dos resultados en formato array, las *features* que tendrán una dimensión igual al [número de pasos, número de columnas, número de filas], además del array del *target*. Estos resultados sirven de insumo para el modelo predictivo desarrollado mediante una red LSTM.

Teniendo listos los datos de entrada para cada uno de los modelos, el de suavizamiento a partir de la CNN y el predictivo a partir de la red LSTM, el paquete ofrece dos funciones que permiten la definición del valor de los hiperparámetros para cada una de las arquitecturas. Para el modelo de suavizamiento con la CNN, se encuentra la función ***param_CNN,*** la cual tiene como valores de entrada el número de pasos que determina la forma de entrada de los datos en la primera capa de la red convolucional, el número de filtros que se deben aplicar en la capa convolucional, junto con el tamaño del filtro que determina el kernel a aplicar a los datos de entrada en la capa convolucional. Así mismo, se cuenta con el parámetro de tamaño de pool y el número de neuronas para hacer el suavizamiento y la predicción final en el modelo de suavizamiento por medio de la CNN.

Para el modelo predictivo que implementa la red LSTM, la función para la definición del valor de los hiperparámetros se llama ***param_LSTM,*** la cual recibe como parámetros el número de pasos y el número de *features* que determinan la forma de entrada de los datos para la primera capa LSTM. Dado que en la arquitectura del modelo predictivo se contemplan dos capas LSTM y dos capas de dropout como método de regularización, se reciben como parámetros el número de neuronas de las dos capas LSTM y las dos tasas de dropout que pueden ser aplicadas en la estructura del modelo para realizar la predicción.

Además, el paquete ofrece funciones de optimización de hiperparámetros para el modelo de suavizamiento con la CNN y el modelo predictivo con la red LSTM. Para la CNN, se implementa la función ***hp_tuning_CNN,*** la cual recibe como parámetros de entrada los datos, el número de pasos seleccionado, y recibe como datos en forma de lista para la optimización de los valores: el número de filtros a optimizar, el número de neuronas, la tasa de aprendizaje y las épocas. Para la optimización de los hiperparámetros de la red LSTM, se implementa la función de optimización ***hp_tuning_LSTM,*** la cual toma como parámetros de entrada los datos, el número de pasos implementado en la red LSTM, el número de neuronas a probar en la optimización en la primera y segunda capa de la arquitectura definida, la tasa de *dropout* en cada una de las capas de *dropout*, la tasa de aprendizaje y las épocas.

6. IMPLEMENTACIÓN DEL MODELO PREDICTIVO

En este capítulo se presenta la metodología propuesta de la división de datos, el modelo de suavizamiento en conjunto con el modelo predictivo. La metodología es implementada en dos conjuntos de datos, por lo cual, este capítulo se divide en dos secciones: en la primera se muestran los resultados obtenidos al implementar el modelo predictivo en datos públicos de baterías que fueron evaluadas en condiciones controladas y han sido usados para la evaluación de modelos predictivos del RUL. En la segunda sección, se describen los datos experimentales obtenidos en las pruebas realizadas a un EV de dos ruedas operando bajo condiciones de tráfico real, una bicicleta eléctrica.

Ambas secciones del capítulo muestran el análisis descriptivo de los datos, la implementación de la metodología propuesta para la división de los datos, el suavizamiento con la CNN de los datos obtenidos por medio de sensores y el modelo predictivo a partir de la red LSTM. Así mismo, se proponen escenarios para el análisis de los datos con el modelo propuesto anteriormente y la búsqueda en cuadrícula implementada para la optimización de los hiperparámetros. Los escenarios permiten evaluar el desempeño del modelo y validar la precisión de la metodología propuesta, resaltando el potencial que tiene la implementación de la metodología en estos casos de estudio. Finalmente, se muestran los resultados obtenidos del modelo predictivo en cada uno de los escenarios planteados con su respectiva métrica de desempeño y conclusiones.

La implementación del modelo predictivo se realizó con el lenguaje de programación R a través del entorno de desarrollo integrado RStudio, con un sistema operativo Windows, procesador Intel Core i5- 4210U CPU 2.40 GHz y una GPU NVIDIA GeForce 820M. La versión de los paquetes implementados en el desarrollo del trabajo se presenta en el Anexo 3. Se adjunta como anexo un archivo de R con la ejemplificación del código implementado para el desarrollo de la metodología. Así mismo, se realiza un análisis de importancia de las variables para cada uno de los escenarios y los resultados son presentados en el Anexo 4.

6.1 Escenario en condiciones controladas – Datos públicos de baterías

Los datos experimentales de baterías bajo condiciones controladas son datos públicos que se obtuvieron del Centro de Excelencia de Pronósticos (PCoE) de la Administración Nacional de Aeronáutica y el Espacio (NASA) y serán usados para la implementación de la metodología propuesta para la predicción del RUL (Saha & Goebel, 2007). Estos datos han sido ampliamente usados para la predicción del RUL de baterías (Fan et al., 2019; Jo et al., 2021; Park et al., 2020; Qiao et al., 2020; Qu et al., 2019; Ren et al., 2018; Zhu et al., 2019). En el trabajo de investigación se usan los datos de las cuatro LIB llamadas B05, B06, B07 y B18, las cuales fueron instrumentadas para capturar el valor de variables como corriente, temperatura y

voltaje a partir de sensores. Las baterías fueron sometidas a procesos de carga y descarga en un ambiente controlado con una temperatura ambiente constante de 24 °C, por lo cual, esta variable es considerada una variable tomada sin sensor. En la Tabla 13 se presentan las variables, su nombre y unidad de medida, donde el 91% de las variables corresponde a variables numéricas y la variable fecha, que representa la fecha en la que se desarrolló la prueba, es de tipo fecha. Para la corriente y el voltaje, se capturó el valor obtenido en el proceso de descarga y en el proceso de carga, por esto, para las variables medidas durante el proceso de descarga, se seguirán llamando por su nombre y para las variables medidas durante el proceso de carga, se añade la palabra de carga después del nombre de la variable. La capacidad es una variable calculada a partir de los amperios hora que brinda la batería en cada una de las pruebas. La variable SOH es calculada teniendo en cuenta la ecuación 1 en la que se describe cómo realizar el cálculo del SOH a partir de la capacidad inicial de la batería. Es así, como el conjunto de datos cuenta con 11 variables que representan el comportamiento de cada batería. De estas variables, la variable a predecir se llama capacidad y representa la cantidad de energía que puede almacenar la batería, la cual cambia ciclo a ciclo.

Tabla 13

Variables medidas en condiciones controladas

Variable	Nombre	Modo de captura	Unidad de medida
Corriente	Corriente	Sensor	Amperios (A)
	Corriente de carga	Sensor	
Voltaje	Voltaje	Sensor	Voltios (V)
	Voltaje de carga	Sensor	
Temperatura	Temperatura medida	Sensor	°C
Tomadas sin sensor	Capacidad	Cálculo	Amperios hora (Ah)
	Temperatura ambiente	Registro	°C
	Fecha y hora	Registro	Fecha
	Ciclo	Registro	Conteo
	Tiempo	Registro	Segundos
	Estado de salud	Cálculo	%

Los pasos para obtener los datos de las baterías fueron los siguientes:

a. *Proceso de carga*: Se realizó la carga de las LIB a corriente constante de 1.5 amperios hasta que el voltaje de la LIB alcanzó los 4.2 voltios, y luego, se mantuvo el voltaje constante a 4.2 voltios hasta que la corriente de carga bajó a 20 miliamperios.
b. *Proceso de descarga*: La LIB se descargó a una corriente constante de 2 amperios hasta que el voltaje de las LIB B05, B06, B07 y B18, cayera a 2.7 voltios, 2.5 voltios, 2.2 voltios y 2.5 voltios respectivamente (Fan et al., 2019).

6.1.1 Análisis descriptivo de los datos

El conjunto de datos de las baterías B05, B06, B07 y B18 se caracteriza por la cantidad de ciclos de carga y descarga, la cantidad de registros y el número de ciclos para cada batería. En total, se tienen 185.721 datos de los 4 conjuntos de baterías y 636 ciclos de carga y descarga. Así mismo, teniendo en cuenta la ecuación 1, se realizó el cálculo del SOH como una variable nueva para el conjunto de datos. La Tabla 14 presenta la capacidad inicial de cada batería, la cantidad de registros, la cantidad de ciclos y el SOH final para cada una de las baterías. En la Tabla 14 se observa que las baterías B05 y B18 inician con la misma capacidad, además, la batería B18 presenta 132 ciclos de carga y descarga, lo cual representa menos ciclos de carga y descarga en comparación con las demás baterías que tuvieron mediciones por 168 ciclos de carga y descarga. Por esta razón, también se observa una menor cantidad de registros para la B18 en comparación con las demás baterías.

Tabla 14

Descripción datos de baterías publicados por la NASA

N° Batería	Capacidad inicial (Ah)	Cantidad registros	Cantidad ciclos	SOH final
B05	1.86	50.285	168	69.3%
B06	2.04	50.285	168	56.7%
B07	1.89	50.285	168	74.1%
B18	1.86	34.866	132	72.3%

En la Figura 17, se presenta el comportamiento del SOH para cada una de las baterías, teniendo en cuenta que el SOH de una batería nueva inicia en un 100% y va disminuyendo a medida que aumentan los ciclos de carga y descarga. Este comportamiento demuestra que la batería a medida que aumentan los ciclos, el SOH se va deteriorando. Así mismo, se grafican los límites del SOH del 70% y 80%, ya que, según revisión de literatura, estos son los límites establecidos del SOH e indican el fin de la vida útil de la batería. En esta, se destaca el comportamiento de la batería B07 y B18, las cuales, no alcanzan a tocar el 70% del límite establecido. Además, la B06 muestra que alcanzó incluso el 60% del SOH en su funcionamiento.

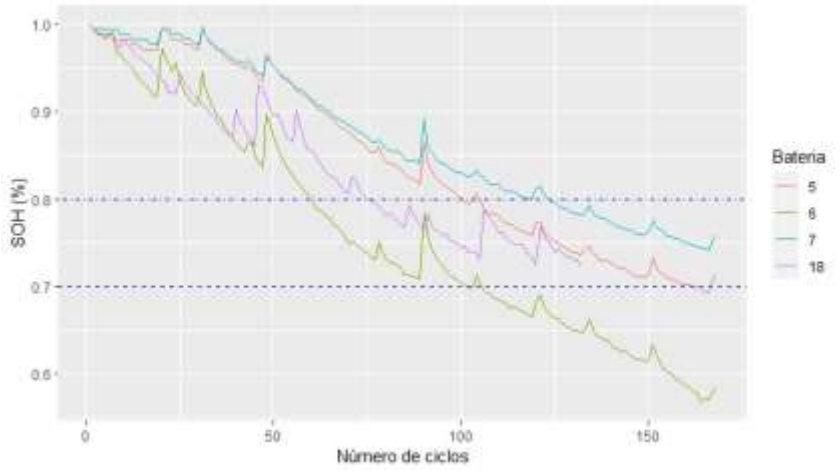


Figura 17. Estado de salud de las baterías en ambiente controlado

El comportamiento de la capacidad a través de los ciclos de carga y descarga, representa el comportamiento de degradación de la batería. A medida que disminuye la capacidad de la batería, disminuye su estado de salud, aumenta su proceso de degradación y su capacidad de funcionamiento es menor. En la Figura 18, se observa el comportamiento decreciente de la capacidad, la cual va disminuyendo a través de los ciclos para cada una de las baterías. Esto indica que la batería va perdiendo capacidad en cada ciclo de carga y descarga. Así mismo, se observan algunos picos de regeneración de la capacidad de la batería, lo que produce un comportamiento no lineal de la capacidad a través del tiempo.

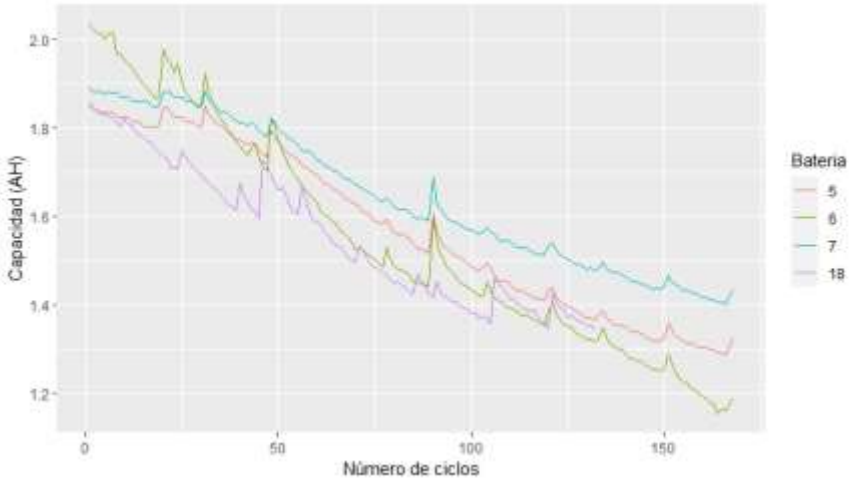


Figura 18. Capacidad de las baterías en ambiente controlado

Se realiza un análisis descriptivo a cada una de las variables de las baterías para describir su comportamiento. En la Tabla 15 se muestran las medidas descriptivas para cada variable de las baterías B05, B06, B07 y B18, las cuales estarán representadas por 5, 6, 7 y 18 respectivamente. Teniendo en cuenta que la variable de temperatura ambiente es una variable constante para todas las pruebas y baterías, no se muestra

en este análisis, así mismo, no se muestra el análisis de la variable ciclo, ya que cada batería inicia en el ciclo 1 y termina en la cantidad de ciclos mostrada anteriormente en la Tabla 14. Según los datos presentados, la B06 es la que inicia con una capacidad mayor y finaliza con la menor capacidad presentada, generando un porcentaje estimado del SOH menor y una mayor desviación estándar en comparación con las demás baterías. Además, se destaca que la mediana de la capacidad es inferior a la media para las baterías B05, B06 y B07, sin embargo, para la B18, se observa que la mediana es mayor a la media. Para la B06 también se observa una mayor desviación estándar en la capacidad en comparación con las demás baterías y la B07 es la que menor desviación presenta. De esta forma, se presenta mayor dispersión de los datos en la B06 y menor dispersión en la B07 y B18.

En promedio, las baterías B05 y B07 llegaron al mismo valor medio del voltaje, seguido por la B18 y finalizando con la B06 que obtuvo el menor valor medio del voltaje y un mayor rango intercuartílico (IQR) comparada con las demás. Para la corriente, se observan valores similares de la mediana para la B05, B06 y B18, y una mediana menor para la B07. El IQR para la corriente es 0, lo que indica que el 50% de los datos de las baterías están distribuidos por valores similares, donde la B06 y B05 son las de mayor desviación estándar. En cuanto a la temperatura de las baterías, se observa que la B06 presenta un mayor valor de mediana, media, desviación estándar e IQR en comparación con las demás, lo que indica, mayor dispersión de los datos para esta batería. La B18 es la que presenta un menor valor de temperatura para la mediana y la media, además de una menor desviación estándar y un IQR menor en comparación con las demás baterías.

Para las variables medidas en el proceso de carga, se destaca la mediana de la corriente de carga, la cual es de 2 voltios para todas las baterías, donde la B06 es la que mayor desviación estándar presenta. En este caso, también se presenta un IQR igual a 0, lo que indica que el 50% datos oscilan entre valores similares. Para el voltaje de carga se observa que la B18 presenta una mayor mediana con una media menor, lo que indica una distribución asimétrica hacia la izquierda. La B06 presenta una mayor desviación estándar con un IQR mayor en comparación con las demás baterías, lo que representa una mayor dispersión en los datos. También se observa que la mediana del tiempo de prueba para las baterías es el mismo, sin embargo, la media para la B18 es mayor, con una desviación estándar más alta y un IQR mayor en comparación con las demás baterías. En este caso, la B18 tuvo 36 ciclos menos de duración en comparación con las demás baterías. El menor SOH se presentó en la B06 con un valor promedio de 74.9% y el mayor porcentaje promedio del SOH fue obtenido por la B07 con un SOH del 86.4%. Donde la B06 y B05 obtuvieron la mayor desviación estándar y un IQR mayor.

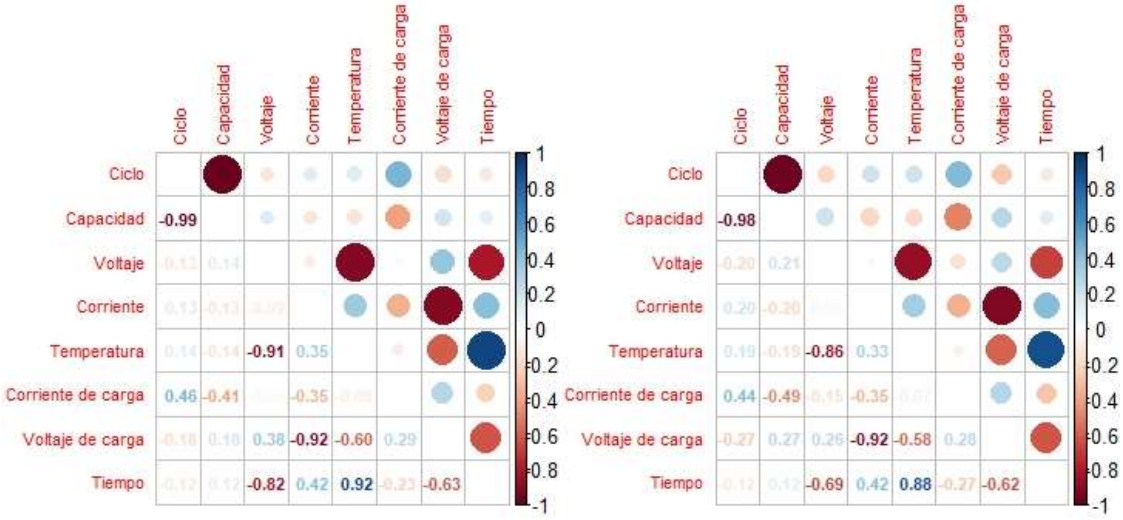
Tabla 15

Medidas descriptivas de los datos de las baterías en condiciones controladas

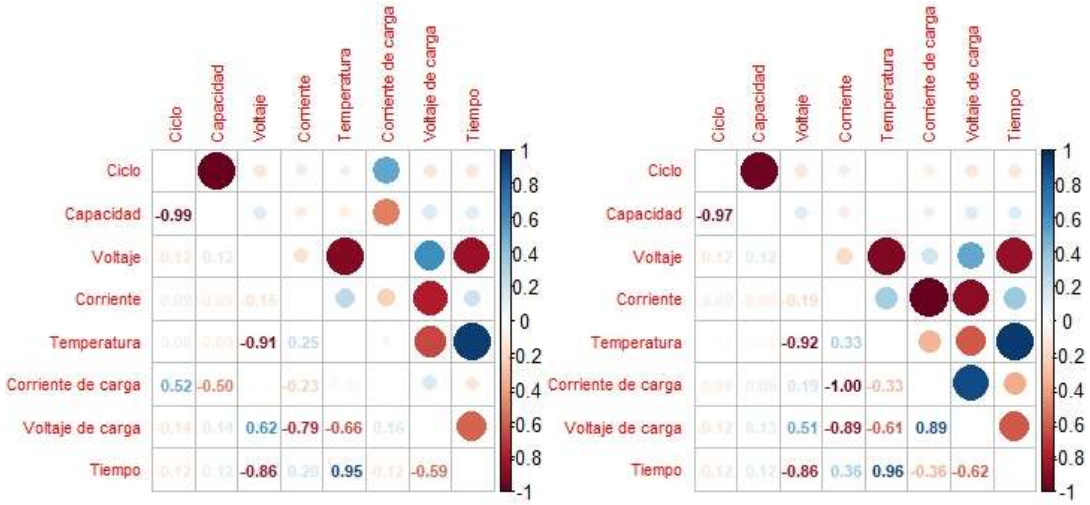
Variable	Unidad	Batería	Mediana	Media	Desviación	IQR
Capacidad	Amperios hora (Ah)	5	1.54	1.56	0.18	0.36
		6	1.47	1.53	0.23	0.38
		7	1.61	1.63	0.15	0.29
		18	1.61	1.58	0.16	0.28
Voltaje	Voltios (V)	5	3.51	3.52	0.23	0.26
		6	3.47	3.46	0.25	0.30
		7	3.52	3.52	0.27	0.27
		18	3.50	3.50	0.25	0.28
Corriente	Amperios (A)	5	-2.01	-1.81	0.61	0.00
		6	-2.01	-1.77	0.65	0.00
		7	-1.99	-1.91	0.39	0.00
		18	-2.01	-1.84	0.55	0.00
Temperatura	°C	5	32.83	32.82	3.99	5.90
		6	32.94	32.84	4.14	6.44
		7	32.29	32.37	4.00	5.55
		18	31.12	31.08	3.65	5.52
Corriente carga	Amperios (A)	5	2.00	1.36	1.31	0.00
		6	2.00	1.31	1.35	0.00
		7	2.00	1.47	1.29	0.00
		18	2.00	1.83	0.55	0.00
Voltaje carga	Voltios (V)	5	2.53	2.31	0.80	0.30
		6	2.53	2.26	0.85	0.36
		7	2.58	2.50	0.55	0.27
		18	2.59	2.41	0.74	0.29
Tiempo	Segundos (s)	5	1537	1546	906.64	1537.42
		6	1537	1546	906.64	1537.42
		7	1537	1546	906.64	1537.42
		18	1537	1547	908.37	1549.63
SOH	Porcentaje (%)	5	82.9%	84.1%	10%	19%
		6	72.4%	74.9%	11%	19%
		7	84.9%	86.4%	8%	15%
		18	86.6%	85.4%	8%	15%

Dentro del análisis también se realiza una matriz de correlaciones de acuerdo al coeficiente de *Pearson*, la cual puede ser observada en la Figura 19. Para la B05, en la Figura 19 (a), se observan relaciones directas fuertes entre la temperatura y el tiempo, lo que indica que a medida que aumenta el tiempo, la temperatura de la batería también aumentó con una correlación del 92%. Además, se observan relaciones inversamente proporcionales entre el ciclo y la capacidad, lo que indica que a medida que aumentan los ciclos de carga y descarga, la capacidad de la batería disminuye como se observó anteriormente, entre el ciclo y la capacidad de la B05 se tiene una correlación del 99%. Esta relación también se presenta para el voltaje y la temperatura con una correlación del 91%, lo que indica que a medida que aumenta la temperatura, el voltaje de la batería disminuye. En general, a medida que aumentan los ciclos y el tiempo de funcionamiento de la batería, la temperatura empieza a aumentar, y el voltaje y la capacidad empiezan a disminuir. Este comportamiento

también se observa en las demás baterías, en las cuales, la correlación entre las variables mencionadas es similar.



(a) Batería 5 (b) Batería 6



(a) Batería 7 (b) Batería 18

Figura 19. Matriz de correlaciones datos en condiciones controladas

Luego del análisis descriptivo, se realiza la selección de las variables, teniendo en cuenta la eliminación de las variables constantes o aquellas que no aportan información al comportamiento de la capacidad a través de los ciclos. De esta forma, se descarta del análisis, la variable temperatura ambiente, por ser una variable constante y controlada en los experimentos. Así mismo, se observa que se tienen diferentes mediciones a través del tiempo en cada uno de los ciclos de las baterías que representan el comportamiento de esa variable

en un ciclo específico. Para el análisis y modelación, se realiza la predicción de la variable capacidad que se caracteriza por cambiar ciclo a ciclo, por lo cual, es necesario seleccionar las medidas descriptivas adecuadas para representar cada una de las variables tomadas a partir de los sensores para cada uno de los ciclos de carga y descarga, con el objetivo de hacer una predicción ciclo a ciclo de la capacidad.

Por esto, se realiza una ilustración considerando el primer ciclo y los siguientes 25 ciclos (1, 25, 50, 75, 100, 125 y 150) para determinar el comportamiento de cada una de las variables tomadas a partir de sensores tales como: corriente, voltaje y temperatura. Teniendo en cuenta que el comportamiento en general de cada una de las variables es similar, para las gráficas se toma como referencia la B05. En la Figura 20 (a) se observa el comportamiento de la corriente en cada uno de los ciclos y se identifica que cuando la corriente en un ciclo determinado se encuentra en el mínimo, la batería está en funcionamiento, y cuando la corriente llega a cero, el ciclo finaliza. Por esta razón, se toma como medida descriptiva para la corriente, el valor mínimo que toma la variable en cada uno de los ciclos de carga y descarga. En la Figura 20 (b) se presenta el comportamiento final del valor mínimo de la corriente obtenida en cada uno de los ciclos de carga y descarga de cada una de las baterías. En esta, se observa que el valor mínimo para la B07 es mayor en comparación con el mínimo obtenido por las demás baterías, lo cual, también es observado en la Tabla 15. Así mismo, se observa que la corriente de una batería oscila en un valor específico y presenta algunos picos en los que la corriente disminuye.

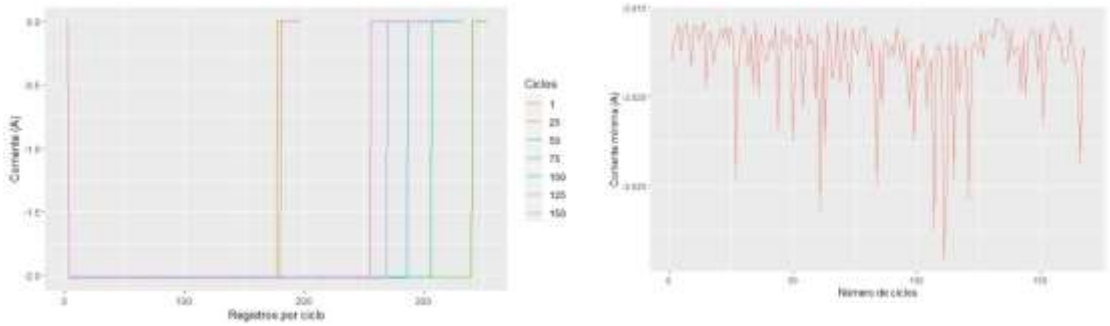


Figura 20. Registros de la corriente por cada ciclo para baterías en ambiente controlado

Para la temperatura obtenida en cada uno de los ciclos, se realiza un análisis similar al anterior, en el que se grafica el comportamiento de la temperatura en cada uno de los ciclos. En la Figura 21 se observa el comportamiento de la temperatura, la cual inicia en un valor de temperatura similar a la temperatura ambiente y a medida que aumenta el tiempo, la temperatura de la batería empieza a subir. Es así, como se identifica que cuando la batería llega al máximo de la temperatura en un ciclo determinado, el ciclo de la batería está finalizando. Por tal motivo, se toma como medida descriptiva para la temperatura, el máximo valor que toma esta variable en cada uno de los ciclos de carga y descarga del funcionamiento de la batería. Se observa que, al inicio de los ciclos, la temperatura de las baterías tiende a disminuir, sin embargo, después

de unos ciclos, se observa un comportamiento creciente para la temperatura, lo que indica que a medida que aumentan los ciclos de carga y descarga de la batería, la temperatura empieza a aumentar. Así mismo, se observa que la B18 es la batería que menor temperatura presentó a lo largo de los ciclos.

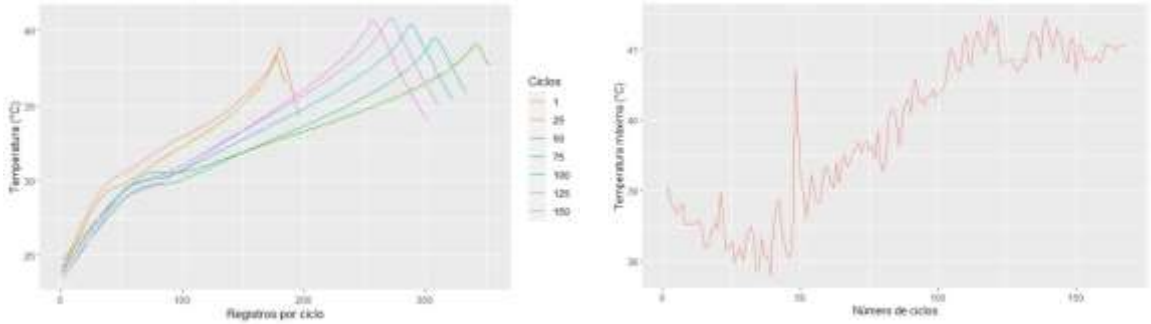


Figura 21. Registros de la temperatura por cada ciclo para baterías en ambiente controlado

Para el voltaje, en la Figura 22 se presenta el comportamiento que adquiere esta variable en cada uno de los ciclos. En esta gráfica, se observa que el voltaje inicia en un valor de 4 voltios y va descendiendo en cada una de las mediciones que se realiza en el ciclo. Además, se identifica que cuando el voltaje de la batería llega a un punto mínimo, la batería finaliza su periodo de funcionamiento y debe ponerse a cargar de nuevo la batería para un próximo ciclo de descarga. De esta forma, el valor mínimo del voltaje obtenido en el ciclo es tomado como medida descriptiva para representar el comportamiento del voltaje ciclo a ciclo. En la Figura 22 se observa que un comportamiento diferente al inicio de los ciclos, un comportamiento más variable en comparación con los resultados obtenidos en los demás ciclos de funcionamiento de la batería. Este comportamiento especial presentado en las baterías, está asociado a un comportamiento de degradación inicial donde la batería se está estabilizando. En general, se observa que la B05 tuvo un mayor valor en el voltaje, la B07 presentó el menor valor obtenido en el voltaje ciclo a ciclo, y para las B06 y B18, se observa un comportamiento similar en el valor obtenido del voltaje en cada uno de los ciclos.

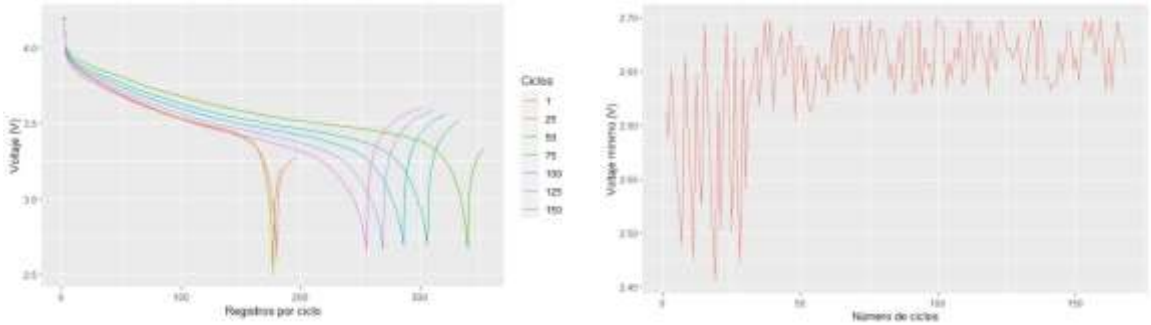


Figura 22. Registros del voltaje por cada ciclo para baterías en ambiente controlado

Teniendo en cuenta la información anterior, se realiza la selección de las medidas descriptivas para cada una de las variables, consolidando un nuevo conjunto de datos con la información. De esta forma, en la

Figura 23 se observan los resultados de las medidas descriptivas seleccionadas para la corriente, temperatura y voltaje de cada una de las baterías. Estos resultados se toman como las variables de entrada para la implementación de la metodología propuesta, la corriente mínima obtenida en cada uno de los ciclos de carga y descarga, la temperatura máxima obtenida en cada ciclo y el voltaje mínimo que tomó la batería en cada uno de los ciclos. Al conjunto de datos de las baterías generado, se le realiza el proceso de división de los datos en los conjuntos de datos necesarios según los pasos explicados en la metodología para llevar a cabo el modelo predictivo. Este conjunto de datos con las variables analizadas, es el insumo de entrada para la primera etapa de la metodología donde se realiza la división de los datos considerando: datos de degradación, datos para la CNN y datos para la red LSTM.

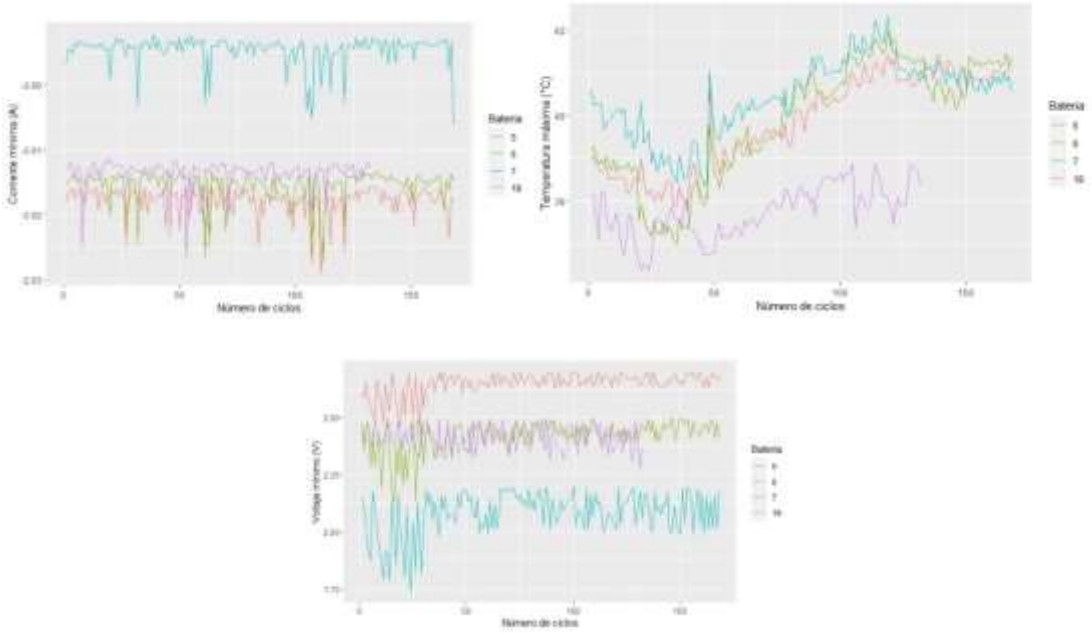


Figura 23. Gráficas de las medidas descriptivas de las variables de cada batería en ambiente controlado

6.1.2 Implementación de la metodología

Para la implementación de la metodología propuesta usando los datos de baterías en condiciones controladas, se hizo uso del paquete desarrollado en lenguaje de programación R `cnnlstm`. El paquete brinda las funciones necesarias para realizar cada uno de los pasos necesarios planteados en la metodología. En esta sección se detallan cada uno de los pasos: división de los datos, suavizamiento de los datos obtenidos por medio de sensores con la CNN y el modelo predictivo propuesto con la red LSTM.

6.1.2.1 División de los datos

Obtener datos de degradación

Para realizar la división de los datos, es necesario evaluar los datos para determinar aquellos datos que hacen parte de los datos de degradación y que deben ser eliminados del análisis para generar una predicción más precisa. En este caso, se tomará como variable de desempeño para determinar los datos de degradación el voltaje, ya que, según el análisis mostrado anteriormente, es la variable que refleja mejor el comportamiento de los datos clasificados como datos de degradación. Así mismo, se analizarán los valores que toma esta variable en cada uno de los ciclos para determinar por medio de tres desviaciones estándar, los valores que deben ser eliminados del conjunto de datos de cada una de las baterías. En la Figura 24 se muestra el comportamiento del voltaje de la B05 evaluado a partir de una carta de control $\bar{X} \pm 3\sigma$, en esta se observa el límite superior (UCL) y el límite inferior (LCL). En esta, se observa que algunos los primeros ciclos, se salen de los límites de control por lo cual, se eliminan estos ciclos y se hace la evaluación con la carta de control, después de eliminar los ciclos, es decir, los datos degradados, se obtienen los ciclos que se encuentran dentro de los límites de control. Por esto, los ciclos que se eliminan según el criterio de la carta de control, son considerados como datos de degradación del conjunto de datos de la B05 y por esta razón, se determina que este periodo de datos es necesario sacarlo del conjunto de datos para realizar el análisis.

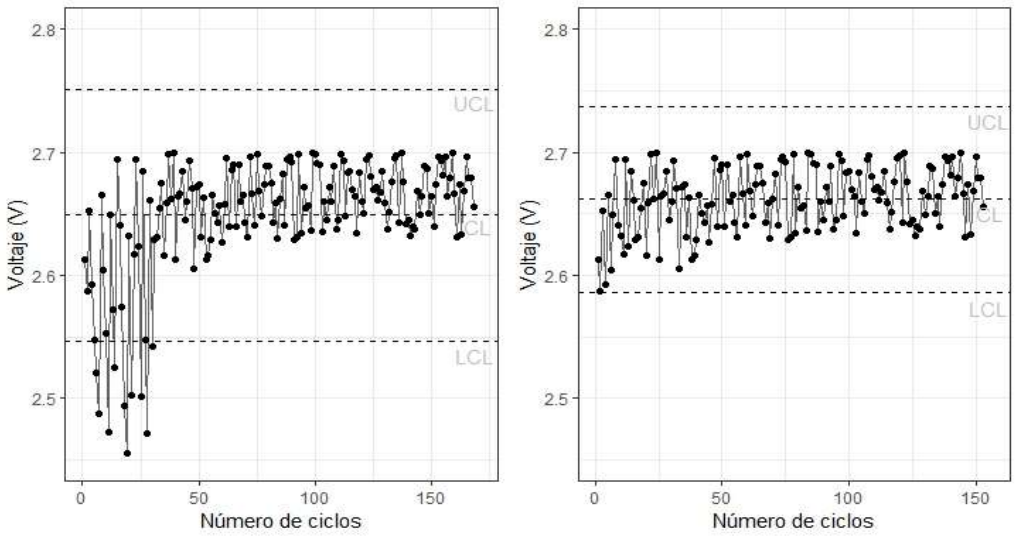


Figura 24. Evaluación de los datos de degradación para la B05 en ambiente controlado

Teniendo en cuenta este criterio, en la Tabla 16 se muestra la cantidad de ciclos determinados como datos de degradación k para cada una de las baterías, junto con la identificación de los ciclos eliminados. De esta forma, se eliminan 15 ciclos de la B05 identificados como ciclos de degradación, 10 ciclos para la B06, 10 ciclos para la B07 y no se identifican datos de degradación en el conjunto de datos de la B18. Estos ciclos identificados como datos de degradación, se eliminan del conjunto de datos de cada una de las baterías para evaluar los escenarios planteados usando estos datos y sin usar estos datos, con el objetivo de validar el desempeño del modelo predictivo.

Tabla 16

Ciclos de degradación en condiciones controladas

Batería	Número de ciclos Datos de degradación ***k***	Identificación de ciclos
B05	15 ciclos	6 ,7, 11, 14, 18 ,19, 21, 25, 28, 30, 5, 10, 13, 27, 17
B06	10 ciclos	10, 15, 16, 18, 20, 22, 26, 5, 7, 30
B07	10 ciclos	11, 14, 21, 24, 26, 29, 5, 10, 17, 23
B18	Sin ciclos de degradación	N/A

6.1.2.2 Modelo de suavizamiento con la CNN

En la Tabla 13 se presentaron las variables obtenidas por medio de sensores, tales como: corriente, temperatura y voltaje, las cuales, según la metodología propuesta, corresponden a las variables a suavizar por medio del modelo de suavizamiento con la CNN. De esta forma, se tienen 3 variables a suavizar para cada una de las baterías con 153 registros para la B05, 158 registros para la B06, 158 registros para la B07 y 132 registros para la B18. Para el modelo, se inicia con la estructuración de los datos de cada una de las variables, teniendo en cuenta un número de pasos de tres, lo que indica tomar tres observaciones para realizar el suavizamiento. Para éste, se considera el promedio de las tres observaciones tomadas como medida de suavizamiento.

Luego de la estructuración de los datos, se considera la arquitectura de la CNN propuesta en el capítulo de metodología para el suavizamiento de cada una de las variables obtenidas por medio de sensores. En esta, se tiene un modelo de suavizamiento que considera unos hiperparámetros fijos tales como: una capa de convolución con un tamaño de filtro de 2 y función de activación *relu.* Así mismo, en la arquitectura, se plantea una capa de *maxpooling* de tamaño 2, una capa de *flatten* y finalmente dos capas densas, donde la última capa densa con 1 neurona corresponde a la capa que brinda el suavizamiento de cada una de las variables obtenidas por medio de sensores. El modelo con la CNN es compilado con el optimizador Adam y la función de pérdida MSE. Este planteamiento de los hiperparámetros fijos permite realizar la optimización de los hiperparámetros variables modificando los valores de éstos a partir de una búsqueda en cuadrícula con dos o tres valores para cada hiperparámetro que cambia. En la Tabla 17 se presentan los hiperparámetros y los valores que toman cada uno.

Tabla 17

Hiperparámetros y valores a optimizar del modelo de suavizamiento en ambiente controlado

Hiperparámetro	Valores
Número de filtros	16, 32, 64
Número de neuronas	50, 100
Tasa de aprendizaje	0.001, 0.01
Épocas	100, 500

Teniendo en cuenta estos valores, se generaron y evaluaron 24 modelos con la combinación de cada uno de los valores que pueden tomar los hiperparámetros y que corresponden a la combinación de cada uno de los hiperparámetros para cada una de las variables de cada batería. Se ejecuta la función con los datos de entrenamiento (70%) y prueba (30%) de cada una de las variables y se seleccionan los hiperparámetros que menor error producen en conjunto para los datos de entrenamiento y prueba de las variables. Como resultado, la función muestra la combinación de valores de los hiperparámetros que minimiza el error. La función genera como resultado usar 64 filtros, 100 neuronas, 0.001 como tasa de aprendizaje y 500 épocas para el entrenamiento.

En la Tabla 18 se presentan los errores obtenidos tanto en entrenamiento como en prueba para cada una de las variables de cada una de las baterías. En esta se observa que la corriente es la variable mejor suavizada en cada una de las baterías con menores errores obtenidos con el modelo, seguido por el voltaje y la temperatura. La corriente obtuvo un menor error en el suavizamiento tanto en los datos de entrenamiento como de prueba en la B18, seguido por la B05, B07 y B06. Para el voltaje, se observan menores errores de suavizamiento para la B06, B05, B07 y B18. Para la B06 y B18 se obtuvo un menor error de entrenamiento en la mayoría de las variables en comparación con el error obtenido en los datos de prueba. Para la B05 y B07, se obtuvo un mayor error de entrenamiento en la mayoría de las variables en comparación con el error obtenido en los datos de prueba. Esto resalta la generalización del modelo para suavizar las variables obtenidas por medio de sensores a partir de la CNN.

Tabla 18

Errores de entrenamiento y prueba obtenidos en el modelo de suavizamiento con los datos en ambiente controlado

Batería	Variable	Error de entrenamiento				Error de prueba			
		RMSE	MAPE	MAE	MSE	RMSE	MAPE	MAE	MSE
5	Corriente	0.0155	1.27	0.0095	0.0002	**0.0080**	**0.78**	**0.0064**	**0.0001**
	Temperatura	0.0272	4.35	0.0195	**0.0007**	0.0358	4.84	0.0210	0.0013
	Voltaje	0.0275	1.75	0.0127	0.0008	**0.0143**	1.36	0.0090	0.0002
6	Corriente	0.0120	1.05	0.0079	**0.0001**	0.0133	1.23	0.0090	0.0002
	Temperatura	0.0312	4.06	0.0183	0.0010	0.0304	4.23	0.0216	0.0009
	Voltaje	**0.0131**	**1.24**	**0.0088**	**0.0002**	0.0117	**1.10**	**0.0078**	**0.0001**
7	Corriente	0.0147	1.19	0.0085	0.0002	0.0124	0.99	0.0071	0.0006
	Temperatura	**0.0259**	3.85	**0.0164**	**0.0007**	0.0237	2.99	0.0155	0.0006
	Voltaje	0.0175	1.67	0.0113	0.0003	0.0214	1.86	0.0133	0.0005

	Corriente	**0.0072**	**0.72**	**0.0060**	**0.0001**	0.0123	0.95	0.0075	0.0002
18	Temperatura	0.0273	**3.63**	0.0186	**0.0007**	**0.0183**	**2.90**	**0.0141**	**0.0003**
	Voltaje	0.0304	3.56	0.0201	0.0009	0.0332	3.60	0.0221	0.0011

Luego de realizar el proceso de entrenamiento y optimización de hiperparámetros, se selecciona la combinación que minimizan los errores y genera el mejor desempeño en el modelo de suavizamiento. De esta forma, se realiza el suavizamiento para cada una de las variables obtenidas por medio de sensores para cada una de las baterías, obteniendo los siguientes resultados presentados en la Tabla 19. En general, se observa un buen desempeño en el suavizamiento de las *features* obtenidas por medio de sensores, las cuales, serán las *features* de entrada para el modelo predictivo con la red LSTM. Como se había observado en la Tabla 18, se obtuvo un mejor suavizamiento para la corriente en comparación con las demás variables. Se observa un menor RMSE, MAPE y MAE obtenido en el suavizamiento de la corriente en la B18, un menor RMSE en la temperatura de la B07 y un menor RMSE, MAPE, MAE y MSE en el suavizamiento del voltaje de la B05.

Tabla 19

Errores en el suavizamiento de los datos en ambiente controlado

Batería	Variable	RMSE	MAPE	MAE	MSE
	Corriente	0.0115	0.86	0.0068	**0.0001**
5	Temperatura	0.0181	4.90	0.0117	0.0003
	Voltaje	**0.0145**	**1.32**	**0.0086**	**0.0002**
	Corriente	0.0112	0.95	0.0071	**0.0001**
6	Temperatura	**0.0152**	4.63	**0.0096**	**0.0002**
	Voltaje	0.0223	1.54	0.0105	0.0005
	Corriente	0.0131	0.94	0.0065	0.0002
7	Temperatura	0.0169	4.24	0.0109	0.0003
	Voltaje	0.0249	1.98	0.0133	0.0006
	Corriente	**0.0071**	**0.63**	**0.0053**	**0.0001**
18	Temperatura	0.0162	**4.01**	0.0107	0.0003
	Voltaje	0.0261	2.56	0.0149	0.0007

Una vez se realiza el proceso de suavizamiento a cada una de las variables de cada batería, se realiza un proceso de integración del suavizamiento generado, con el objetivo de crear un nuevo conjunto de datos con las variables obtenidas por medio de sensores suavizadas. Estas variables también se integran junto con las variables obtenidas sin sensores, consolidando la información en el nuevo conjunto de datos de entrada para el modelo predictivo LSTM.

6.1.2.3 Modelo predictivo con la red LSTM

El modelo predictivo toma como datos de entrada, los datos suavizados predichos anteriormente y los une con los datos obtenidos sin sensores, a los cuales, se les restan las dos primeras observaciones para cada variable obtenida sin sensores, ya que estas observaciones corresponden a los datos que se perdieron en el suavizamiento realizado en el paso anterior. Después de restar las dos observaciones, se normaliza el conjunto de datos obtenidos sin sensores a partir de la normalización min-max. Como resultado, se genera el conjunto de datos de las variables obtenidas sin sensores normalizadas.

Luego, este conjunto de datos obtenidos sin sensores normalizados, se unen con el conjunto de datos suavizados en el paso anterior para conformar el conjunto de datos que permite realizar el modelo predictivo. Este conjunto de datos está conformado por las variables de ciclo, corriente, temperatura, voltaje, SOH y tiempo para cada una de las baterías y 151 registros para la B05, 156 registros para la B06, 156 registros para la B07 y 130 registros para la B18. Estos datos se pasan por la función *split_seq*, la cual, toma los datos de entrada y el número de pasos igual a diez para generar las series con *p features* con la información de las *features* y el *target*. La información generada permite tener los datos con la estructura necesaria para la red LSTM propuesta.

Para la arquitectura de la red LSTM, se fijan unos hiperparámetros según la metodología propuesta, en la que la arquitectura de la red está compuesta por dos capas LSTM, donde cada capa tiene en medio dos capas *dropout*. Además, se considera una capa densa como salida con una neurona que permite generar la predicción del modelo. El modelo predictivo con la red LSTM es compilado con el optimizador Adam y la función de pérdida implementada es el MSE. Con este planteamiento, se realiza el proceso de optimización de los hiperparámetros modificando los valores de éstos a partir de la búsqueda en cuadrícula con dos o tres valores en el conjunto especificado. En la Tabla 20 se presentan los hiperparámetros y los valores que tomará cada uno de éstos en la optimización de los hiperparámetros.

Tabla 20

Hiperparámetros y valores a optimizar del modelo predictivo en ambiente controlado

Hiperparámetro	Valores
Número de neuronas primera capa	64, 128
Tasa *dropout* primera capa	0.1, 0.2, 0.3
Número de neuronas segunda capa	32, 64
Tasa *dropout* segunda capa	0.1, 0.2, 0.3
Tasa de aprendizaje	0.001, 0.01
Épocas	100, 500

De acuerdo a los valores presentados, se generaron, entrenaron y evaluaron 144 modelos predictivos con la combinación de cada uno de los valores que pueden tomar los hiperparámetros. Para esto, se implementa la función de optimización de los hiperparámetros con los datos de entrenamiento, y con los datos de validación, se selecciona la combinación de hiperparámetros que minimizan el error en la predicción. La función genera como resultado usar 128 neuronas en la primera capa LSTM, una tasa de *dropout* del 0.3 en la primera capa, 32 neuronas en la segunda capa LSTM y una capa de *dropout* con una tasa de 0.2. Además, la función de optimización define como tasa de aprendizaje un valor de 0.01 con 500 épocas para el entrenamiento del modelo predictivo. En la Tabla 21 se presentan cada uno de los errores obtenidos en el proceso de entrenamiento, validación y los errores de los datos de prueba para el modelo predictivo de la capacidad.

Tabla 21

Errores de entrenamiento, validación y prueba obtenidos en el modelo predictivo con los datos en ambiente controlado

Tipo de error	RMSE	MAPE	MAE	MSE
Entrenamiento	0.0123	4.80	0.0101	0.0002
Validación	0.0165	6.28	0.0130	0.0003
Prueba	0.0174	8.66	0.0142	0.0003

6.1.3 Evaluación de la precisión del RUL

Luego de hacer el entrenamiento del modelo predictivo y la optimización de lo hiperparámetros, se realiza la evaluación de la precisión de la predicción del RUL. Para esto, se plantean diferentes escenarios para la evaluación de la precisión en la predicción del RUL. En este caso, se consideran cuatro escenarios teniendo en cuenta lo siguiente: un escenario en el que no se consideran los datos de degradación, no se realiza el suavizamiento y se implementa el modelo predictivo con la red LSTM. El segundo escenario tampoco considera los datos de degradación, pero en este caso, si considera el modelo de suavizamiento y el modelo predictivo CNN-LSTM. En el tercer escenario, se consideran los datos de degradación, sin embargo, no se considera el modelo de suavizamiento y se implementa la red LSTM como modelo predictivo. Por último, en el cuarto escenario, se consideran los datos de degradación, el suavizamiento a partir de la CNN y el modelo predictivo con la red LSTM. En la Tabla 22 se muestra el resumen de cada uno de los escenarios planteados para la evaluación de los modelos.

Tabla 22

Planteamiento de escenarios para datos de baterías en ambiente controlado

Escenario	División de datos	Suavizamiento	Modelo predictivo
1	Train, validation, test	No	LSTM
2	Train, validation, test	Sí	CNN-LSTM
3	Train, validation, test, degradation	No	LSTM
4	Train, validation, test, degradation	Sí	CNN-LSTM

Para cada uno de los escenarios, se muestran los resultados obtenidos de manera gráfica y los errores obtenidos de acuerdo al RMSE, MAPE, MAE y MSE. Los resultados para el primer escenario se muestran en la Figura 25, donde se toman 3 conjuntos de datos: entrenamiento, validación y prueba y se implementa el modelo predictivo con la red LSTM. En la figura, se presentan los resultados de la B05, B06, B07 y B18. Se observa que la B06 tiene un buen ajuste de los valores predichos respecto a los valores reales. Para la B05 y la B07, se observa que se subestima varias predicciones en comparación con los datos reales, ya que la curva de puntos está por debajo de la línea. Además, para la B18, no se observa un buen ajuste en las predicciones, ya que el modelo predictivo implementado sobre ajusta las predicciones respecto a los valores reales de la capacidad de la batería.

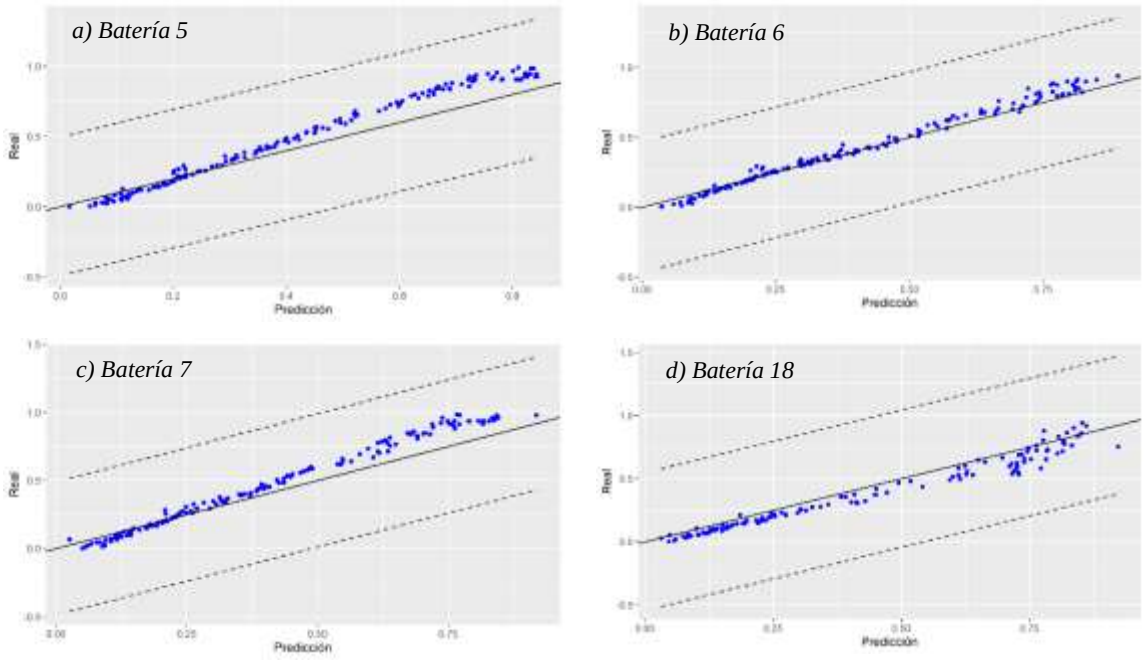


Figura 25. Resultados escenario 1 con datos en condiciones controladas

En la Figura 26, se observan los resultados obtenidos del escenario 2 en el que se toman 3 conjuntos de datos: entrenamiento, validación y prueba, además, se implementa el modelo de suavizamiento por medio de una CNN y el modelo predictivo con la red LSTM. En general, en este escenario se observa un buen ajuste de la predicción en comparación con los valores reales de la capacidad en cada una de las baterías.

Este escenario, considera la suavización de los datos obtenidos por medio de sensores, lo que resalta la importancia del modelo de suavizamiento en la generación del modelo predictivo con la red LSTM. Para la B06 y la B18, se observa un pequeño sobre ajuste en las predicciones con respecto a los valores reales, ya que varios puntos se encuentran por encima de la línea.

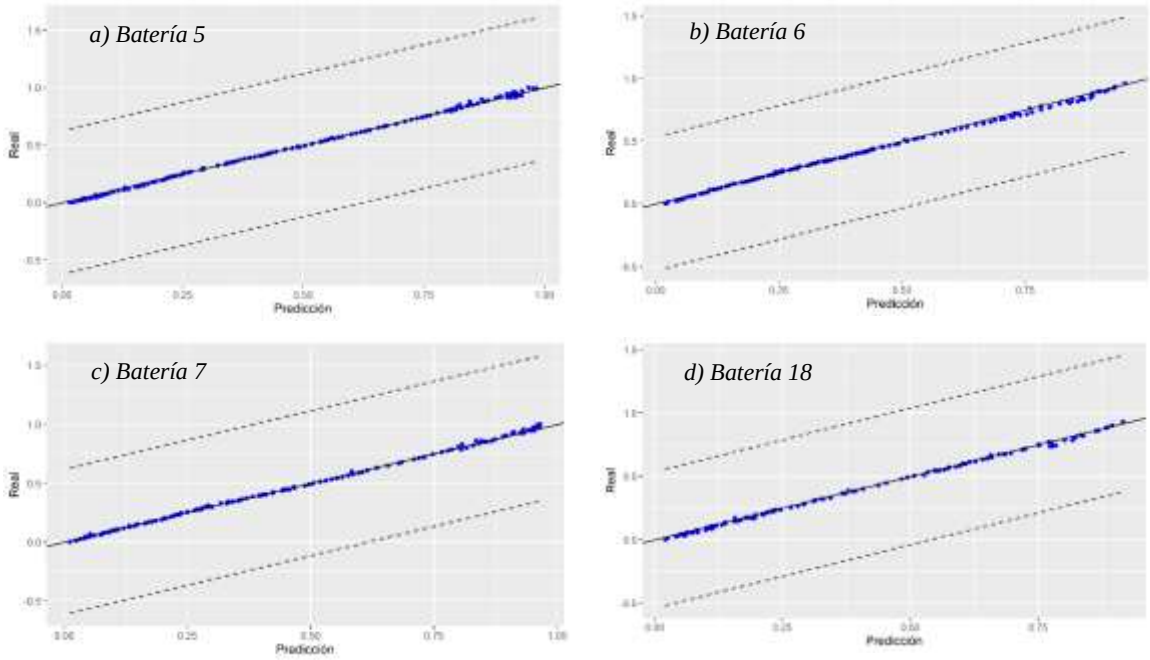


Figura 26. Resultados escenario 2 con datos en condiciones controladas

El escenario 3 presenta los resultados obtenidos en la implementación de cuatro conjuntos de datos: entrenamiento, validación, prueba y degradación. En este caso, solo se hace uso del modelo predictivo con la red LSTM para realizar las predicciones. En general, no se observa un buen ajuste de las predicciones para cada una de las baterías. La B05 presenta un sobre ajuste al inicio de las predicciones y una subestimación de las mismas al finalizar, lo que no permite un buen ajuste. Para la B06, se presenta en general un sobre ajuste en las predicciones, ya que la mayoría de los puntos permanece por encima de la línea. La B07, presenta un comportamiento similar en la predicción de la B05, en la cual, se realiza un sobre ajuste al inicio y una subestimación al final, lo que no permite al modelo una buena precisión en los resultados. Al evaluar la B18, se observa menos ajuste en el modelo predictivo, ya que se observa más variabilidad en los puntos respecto a la línea en la que deberían permanecer, indicando que el modelo no es capaz de capturar con buena precisión el comportamiento de la capacidad en esta batería.

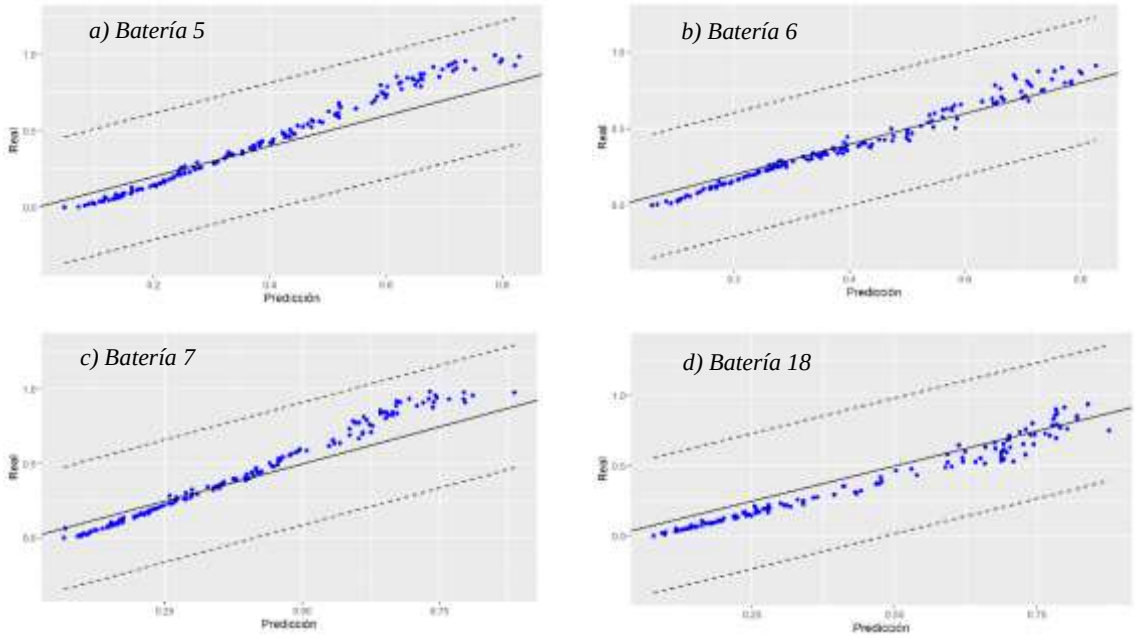
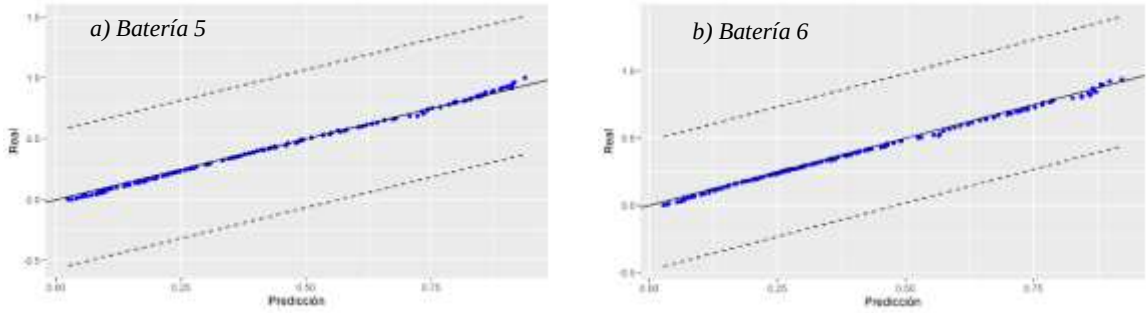


Figura 27. Resultados escenario 3 con datos en condiciones controladas

La Figura 28 muestra los resultados obtenidos al evaluar el cuarto escenario, en el que se considera la división de los datos en cuatro conjuntos: datos de entrenamiento, validación, prueba y degradación. En este caso, también se implementa el modelo de suavizamiento por medio de la CNN y el modelo predictivo por medio de la red LSTM. La combinación de la división de los datos, el modelo de suavizamiento y el modelo predictivo permite generar un buen ajuste en las predicciones de la capacidad respecto a los valores reales obtenidos. Este escenario, considera aspectos importantes en la modelación y predicción de la capacidad a lo largo de los ciclos de carga y descarga, resaltando la importancia de la suavización de los datos obtenidos por medio de sensores para realizar una predicción más precisa de la capacidad. Así mismo, la identificación de los datos de degradación permite un mejor ajuste del modelo, ya que para las cuatro baterías se observa que los valores predichos versus los valores reales se ajustan adecuadamente a la línea recta.



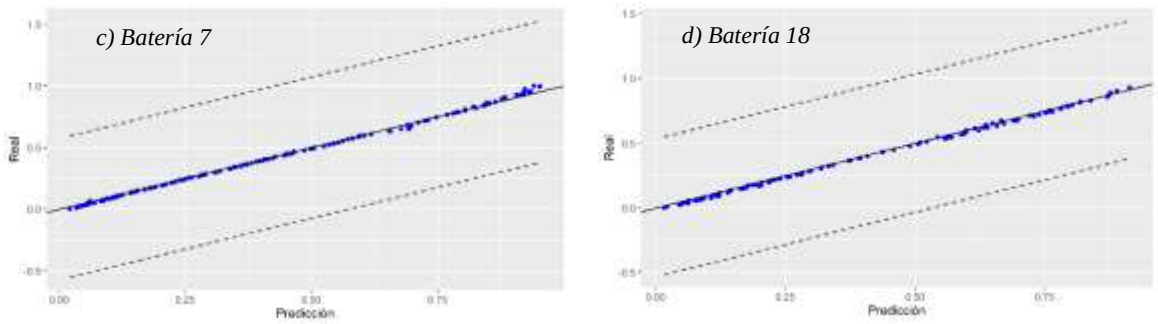


Figura 28. Resultados escenario 4 con datos en condiciones controladas

En la Tabla 23 se presentan los resultados de la evaluación del rendimiento de los modelos para cada una de las baterías a partir de métricas de desempeño como el RMSE, MAPE, MAE y MSE para cada una de las baterías. En la tabla se observa que el escenario 4, en el que se implementa la división en cuatro conjuntos: entrenamiento, validación, prueba y degradación y el modelo de suavizamiento y el modelo predictivo CNN-LSTM, es el escenario que mejores resultados genera en la predicción de la capacidad de la batería a lo largo de los ciclos. Así mismo, se observa que el escenario que tiene un mayor error en la predicción es el escenario 1 en el que se consideran los datos originales de las variables y se realiza la predicción mediante un modelo predictivo con la red LSTM. Seguido de este escenario, se observa que el escenario 3 que implementa la división de los datos en cuatro conjuntos y el modelo predictivo con la red LSTM, genera mejores predicciones en comparación con el modelo anterior, lo que resalta la importancia de la identificación de este tipo de datos de degradación que afectan la precisión en la predicción de la capacidad.

El escenario 2, donde se implementan tres conjuntos de datos, el suavizamiento por medio de la CNN y el modelo predictivo con la red LSTM, presenta mejores resultados en cuanto a la evaluación de los errores a partir de las métricas RMSE, MAPE, MAE y MSE, en comparación con el escenario 1 y 3 descritos anteriormente, indicando que es importante dentro de la metodología propuesta, el suavizamiento de los datos obtenidos por medio de sensores. Finalmente, el escenario 4 resalta el potencial que tiene la metodología propuesta al incorporar la división de los datos en cuatro conjuntos, el suavizamiento de los datos obtenidos por sensores a través de una CNN y el modelo predictivo a partir de la red LSTM que permite capturar el comportamiento a corto y largo plazo.

Tabla 23

Errores obtenidos en cada uno de los escenarios y baterías en condiciones controladas

Escenario	Error	Batería 5	Batería 6	Batería 7	Batería 18
1	RMSE	0.0983	0.0430	0.0878	0.1100
	MAPE	25	15.1	21.6	31.5
	MAE	0.0841	0.0377	0.0729	0.1030
	MSE	0.0097	0.0019	0.0077	0.0121
2	RMSE	0.0174	0.0257	0.0157	0.0190
	MAPE	8.66	6.40	5.28	7.67
	MAE	0.0142	0.0189	0.0127	0.0140
	MSE	0.0003	0.0007	0.0003	0.0004
3	RMSE	0.0878	0.0379	0.0835	0.0930
	MAPE	21.8	11.7	18.3	24.6
	MAE	0.0688	0.0292	0.0621	0.0797
	MSE	0.0077	0.0014	0.0070	0.0087
4	RMSE	**0.0166**	**0.0118**	**0.0125**	**0.0143**
	MAPE	**8.54**	**5.71**	**5.19**	**7.11**
	MAE	**0.0135**	**0.0089**	**0.0090**	**0.0117**
	MSE	**0.0003**	**0.0001**	**0.0002**	**0.0002**

6.2 Escenario en condiciones de tráfico real – Datos experimentales

Para el desarrollo de la investigación en EV de dos ruedas bajo condiciones de tráfico real, se hizo uso de una bicicleta eléctrica, la cual, se caracteriza por ser una bicicleta Stärker Urban ST que tiene acoplado un EM eléctrico con potencia nominal de 350 watts (W). Tiene una velocidad máxima de 25 km/h y una LIB de 36 voltios con capacidad de 10 amperios hora (Ah) que suministra energía y brinda un recorrido autónomo de 40 km. Con el objetivo de determinar los componentes críticos, se llevó a cabo un análisis de modos, efectos y criticidad (FMECA, por sus siglas en inglés: *Failure Mode, Effects, and Criticality Analysis*). La descripción de los componentes de la bicicleta, el FMECA, el proceso de instrumentación y realización de pruebas se muestra en el Anexo 3.

El FMECA tenía como objetivo determinar los componentes críticos a través de la caracterización de los modos de falla y la determinación de las condiciones operacionales y ambientales extremas para la bicicleta. De esta forma, se caracterizó la bicicleta a partir de 5 sistemas: eléctrico, de tracción, de frenado, estructural y auxiliar. Para cada uno de los sistemas se realizó un análisis cualitativo en el que se determinaron 60 modos de falla y a partir de estos resultados, se realizó un análisis cuantitativo para determinar la criticidad de los componentes. En la Figura 29 se muestran los resultados obtenidos al evaluar la criticidad de los componentes, en esta figura, se destacan los componentes del motor, frenos, levas, cadena, baterías (banco de baterías) y controlador como los componentes más críticos. Al evaluar el nivel de criticidad obtenido con los costos de cada uno de los componentes, se obtiene que el motor y la batería son los componentes más

críticos de la bicicleta eléctrica, ya que son los componentes más costosos del EV, donde la batería puede costar hasta un 30% del costo total del EV. Por este motivo, se instrumentaron la batería y el motor, y se analizó la vida útil de la batería, ya que este componente es intercambiable y permite prolongar la vida útil de la bicicleta a través del intercambio de la batería.

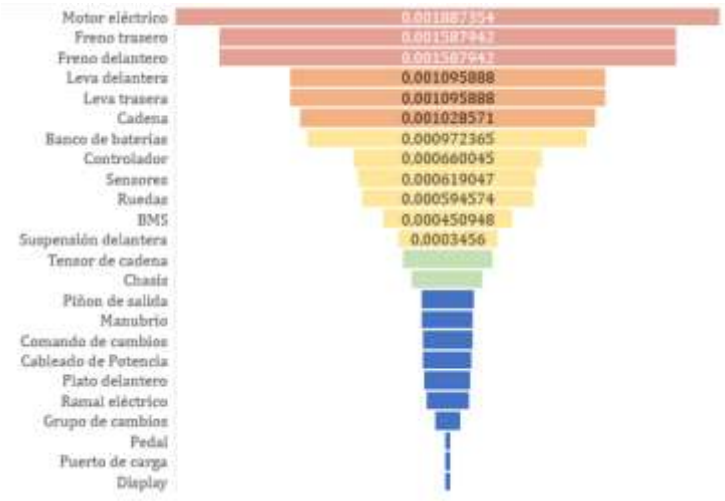


Figura 29. Criticidad de los componentes de la bicicleta eléctrica

Es así, como se evalúan los componentes de la bicicleta eléctrica y se realiza su instrumentación para la captura de las variables de interés. Las variables que no fueron tomadas por sensores, fueron registradas en cada una de las pruebas para complementar la base de datos. En la Tabla 24 se presentan las variables, su nombre, modo de captura (sensor o no sensor) y su unidad de medida, donde el 86% de las variables son numéricas, el 7% corresponde a una variable de tipo fecha y el 7% restante corresponde a una variable tipo factor para el número de experimento realizado. El tipo de variable aceleración representa la aceleración obtenida en los ejes x, y, z mientras se llevaba a cabo el experimento. Para el tipo de variable corriente y voltaje, se hizo uso de sensores para capturar su valor en la batería. También se hizo uso de un sensor para capturar el valor de la temperatura obtenido en el centro de la batería y en un extremo de la misma. Para las variables ambientales, se hizo uso de un mismo sensor para capturar la humedad y la temperatura ambiental mientras se realizaban las pruebas.

La potencia fue una variable calculada a partir de la corriente y el voltaje obtenido en cada momento del registro. La variable ciclo fue registrada en cada una de las pruebas y representa el ciclo de funcionamiento de la batería. La variable fecha representa el día en el que se realizó la prueba, y la variable experimento, representa el número del experimento que se realizó en ese día, la cual puede ser 1 o 2. La variable RUL se obtuvo a partir del cálculo de la duración en minutos que tuvo la batería en su funcionamiento. El RUL representa el total de minutos que duró en funcionamiento la batería, y para la medición del RUL en cada

uno de los ciclos, se toma la duración total de funcionamiento de la batería. Esta duración representa el tiempo de vida útil de la batería en el primer ciclo de funcionamiento, para el segundo ciclo de funcionamiento, se toma el tiempo total y se resta la duración del primer ciclo y así sucesivamente hasta llegar al último ciclo de funcionamiento. Debido a que la duración de cada prueba no es la misma, se tiene un comportamiento decreciente no lineal en cada batería. Las pruebas en condiciones de tráfico real se realizaron a dos LIB de la bicicleta eléctrica, las cuales, se nombran como B01 y B02. Las dos baterías fueron instrumentadas para capturar los valores de las mismas variables a partir de sensores en cada una de las pruebas realizadas.

Tabla 24

Variables medidas bajo condiciones de tráfico real

Variable	Nombre	Modo de captura	Unidad de medida
Aceleración	Eje x Eje y Eje z	Sensor ADXL345	m/s^2
Corriente	Batería	Sensor INA229	Amperios (A)
Temperatura	Batería en el centro Batería en un extremo	Sensor NTCALUG03A103GC	°C
Voltaje	Batería	Sensor INA229	Voltios (V)
Ambientales	Humedad ambiental Temperatura ambiental	Sensor SHT30 Sensor SHT30	g/m^3 °C
Potencia	Batería	Cálculo	Watt (W)
Tomadas sin sensor	Ciclo Fecha Experimento RUL	Registro Registro Registro Cálculo	Conteo Fecha Conteo Minutos

Así mismo, se determinaron los pasos para obtener los datos de las baterías a partir de las siguientes definiciones:

a. *Proceso de carga*: Se realizó la carga de las LIB en un toma corriente de a 110V/60Hz, según las recomendaciones del manual de garantía y mantenimiento.
b. *Proceso de descarga*: Se capturaban los datos cuando iniciaba la ruta de prueba y finalizaba la captura de los datos cuando la LIB se descargaba.

Los experimentos con las LIB en la bicicleta eléctrica se realizaron hasta que la batería ya no funcionaba.

6.2.1 Análisis descriptivo de los datos

El conjunto de datos experimentales se obtuvo a partir de la realización de pruebas con una persona que pesa en promedio 112 kg. Las pruebas en la bicicleta eléctrica se realizaron alrededor de la Universidad

Nacional sede del Río en Medellín pasando por el Metroplús. En general, el tiempo promedio de experimentación desde que salía la persona hasta que volvía a la sede fue de una hora y media, con un tiempo promedio de carga de aproximadamente cinco horas. En la Tabla 25 se observa que ambas baterías, según la información del fabricante, inician con la misma capacidad inicial, una cantidad de registros similar y tienen una duración de ciclos de carga y descarga diferente, con una diferencia en el RUL de 10 minutos. En los experimentos se calcula la variable RUL a partir de la duración total en minutos del funcionamiento de la batería, la cual, va disminuyendo ciclo a ciclo. El RUL, es la variable objetivo a modelar dentro de la experimentación realizada.

Tabla 25

Descripción datos de baterías experimentales

N° Batería	Capacidad inicial (Ah)	Cantidad registros	Cantidad ciclos	RUL (minutos)
B01	10	35.985	21	1.459
B02	10	35.854	27	1.449

El comportamiento de esta variable a través de cada ciclo de carga y descarga, representa el comportamiento de degradación de la batería. A medida que aumentan los ciclos de carga y descarga, el RUL de la batería disminuye, aumentando su proceso de degradación. En la Figura 30, se observa el comportamiento decreciente del RUL por ciclo, el cual va disminuyendo a través de los ciclos para cada una de las baterías. Para la B02 se observa un comportamiento más lineal en comparación con la B01, ya que la B01 muestra un comportamiento de degradación mayor que la B02. Así mismo, durante la realización de las pruebas, se observó un comportamiento más estable en el tiempo de funcionamiento de la B02 en comparación con la B01.

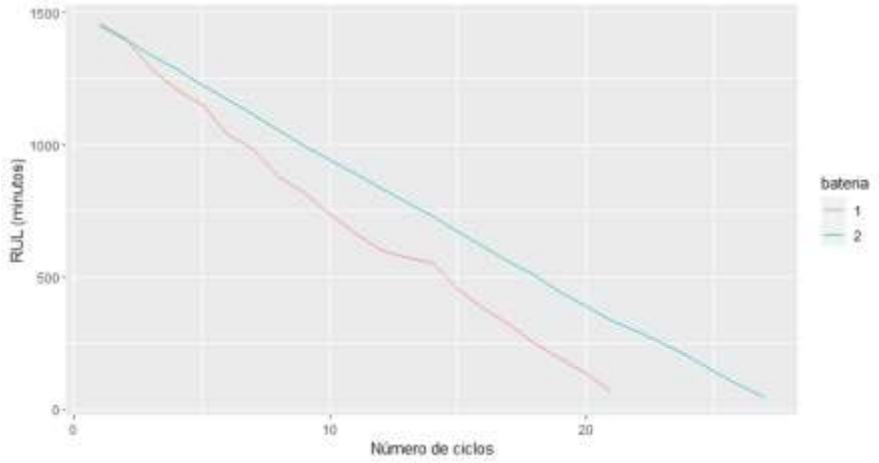


Figura 30. RUL de las baterías en condiciones de tráfico real

En la Tabla 26 se muestran las medidas descriptivas para cada una de las variables de las baterías B01 y B02. Según el análisis descriptivo, se observa una mayor desviación en la aceleración del eje x en la B01, lo que indica que la B01 en la bicicleta tuvo mayor movimiento en el eje x en comparación con la B02. Así mismo, para las aceleraciones, se observa que la B01 tiene una asimetría hacia la izquierda, y la B02 es simétrica, ya que la media y la mediana son similares. En general, el promedio y la mediana de la corriente para la B02 fue mayor en comparación con la B01, y, en consecuencia, la potencia de la B02 también fue mayor respecto a los valores obtenidos en la B01, por esto, se observa mayor suministro de corriente para la B02 en comparación con la B01. Se presenta una asimetría hacia la derecha para la corriente y la potencia de la B01 y una asimetría hacia la izquierda para las mismas variables de la B02. Así mismo, se ve un aumento en el promedio del voltaje de la B02 respecto a la B01, sin embargo, se observa una temperatura promedio mayor en la B01 respecto a la B02. Esto indica que la temperatura de la B01 fue mayor en comparación con la B02, alcanzando mayores niveles de temperatura.

Además, se observa un menor valor promedio de la humedad ambiental en las pruebas realizadas con la B01 en comparación con la B02. Sin embargo, la desviación estándar de la B01 es mayor en comparación con el valor de la desviación obtenida en la B02, lo que indica mayor variabilidad en la humedad ambiental de las pruebas realizadas con la B01. En promedio, se observa que la temperatura ambiental a la que estuvo sometida la B01 fue menor en comparación con la B02, así mismo, se observa un valor de mediana en la B02 mayor que la mediana obtenida en la B01. Para el RUL, el cual es medido a partir de los minutos de duración de la batería respecto a cada uno de los ciclos, se observa un mayor valor promedio para la B02 en comparación con la B01. Así mismo, el RUL para la B01 tiene una asimetría hacia la derecha, ya que la media es mayor a la mediana, y para la B02, se tiene una asimetría hacia la izquierda ya que la media es menor a la mediana, lo que refleja una mayor duración en minutos para la B02. La B01 presenta una mayor variabilidad en la humedad y la temperatura ambiental, ya que se tiene una asimetría hacia la izquierda para la humedad ambiental y una asimetría hacia la derecha para la temperatura ambiental. En general, se observa una mayor vida útil para la B02 con una mediana mayor a la media en comparación con la B01, lo que puede estar asociado a una mayor temperatura de la B01, un menor suministro de corriente, potencia y voltaje.

Tabla 26

Medidas descriptivas de los datos de las baterías en condiciones de tráfico real

Variable	Unidad	Batería	Mediana	Media	Desviación	IQR
Aceleración eje x	m/s^2	1	9.36	7.06	6.68	1.87
		2	9.52	9.52	1.69	1.17
Aceleración eje y		1	1.56	1.48	1.30	1.24
		2	1.95	1.98	1.26	1.10
Aceleración eje z		1	2.26	2.15	1.61	1.63
		2	2.34	2.34	1.26	1.24
Corriente		1	1.09	1.33	1.11	1.82

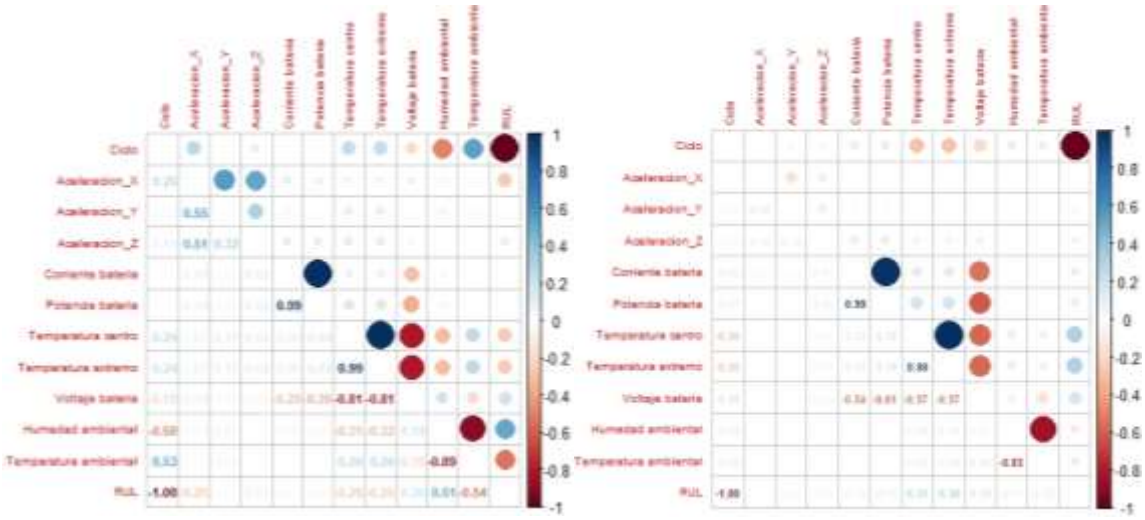
	Amperios (A)	2	1.97	1.82	1.10	1.48
Potencia	Watt (W)	1	0.03	0.04	0.03	0.05
		2	0.06	0.05	0.03	0.04
Temperatura batería en el centro	°C	1	52.46	54.77	19.99	22.57
		2	38.35	39.87	12.11	8.79
Temperatura batería en un extremo		1	52.49	54.79	20.04	22.57
		2	38.26	39.89	12.15	8.79
Voltaje	Voltios (V)	1	35.27	35.11	3.16	5.06
		2	35.56	35.48	2.49	3.45
Humedad ambiental	g/m^3	1	66.81	64.73	9.52	13.14
		2	66.26	66.00	7.10	5.60
Temperatura ambiental	°C	1	23.50	23.91	1.79	2.08
		2	24.40	24.32	1.63	2.14
RUL	Minutos	1	666.0	719.4	407.67	662
		2	784.0	761.1	419.62	726

Dentro del análisis también se realiza una matriz de correlaciones de acuerdo al coeficiente de *Pearson*, la cual puede ser observada en la Figura 31. Para la B01, en la Figura 31 (a), se observan relaciones directas fuertes entre la corriente y la potencia con una correlación del 99% y una relación directa entre la temperatura de la batería en el centro y en un extremo con una correlación del 99%. Así mismo, se observa una relación directa entre el RUL y la humedad ambiental registrada en los ciclos, donde se tiene una correlación del 51%. Para las relaciones inversas, se observa que existe una relación inversamente proporcional entre el ciclo y el RUL, ya que a medida que aumentan los ciclos de funcionamiento de la batería, el RUL disminuye, con una correlación del 100%. De igual forma, se observa una relación inversa entre el RUL y la temperatura ambiental, con una correlación del 54% y que indica que a medida que disminuye el RUL, aumenta la temperatura ambiental.

Así mismo, se observa una relación inversa entre las temperaturas obtenidas en el centro y el extremo de la batería y el voltaje de la batería, con una correlación del 81%, lo que indica que a medida que aumenta la temperatura, el voltaje de la batería disminuye. También se observa una relación inversa entre la humedad y la temperatura ambiental, donde a medida que aumenta la temperatura, la humedad tiende a disminuir con una correlación del 89%. En general, este análisis indica que a medida que aumenta la corriente, aumenta la potencia, así mismo, cuando la temperatura aumenta, el voltaje de la batería empieza a disminuir, lo que se relaciona con el comportamiento decreciente del RUL de las baterías.

Para la B02, en la Figura 31 (b), se observa la relación inversa entre el ciclo y el RUL con una correlación del 100%, lo que indica que a medida que aumentan los ciclos de funcionamiento, el RUL de la batería empieza a disminuir. También se observan relaciones inversas entre el voltaje y la corriente con una correlación del 54%, el voltaje y la potencia con una correlación del 61%, y entre el voltaje y las temperaturas de la batería con una correlación del 57%. Lo que indica que a medida que aumentan la

corriente, la potencia y las temperaturas, el voltaje de la batería tiende a disminuir. El análisis descriptivo muestra una mayor duración de la B02 en comparación con la B01, con unas relaciones directas e inversas entre las covariables, donde se destaca la relación inversa entre el ciclo y el RUL. Así mismo, se resalta la relación inversa entre el voltaje y las variables de corriente, potencia y temperatura, lo que indica que, a menor voltaje, mayor corriente, potencia y temperatura en la batería.



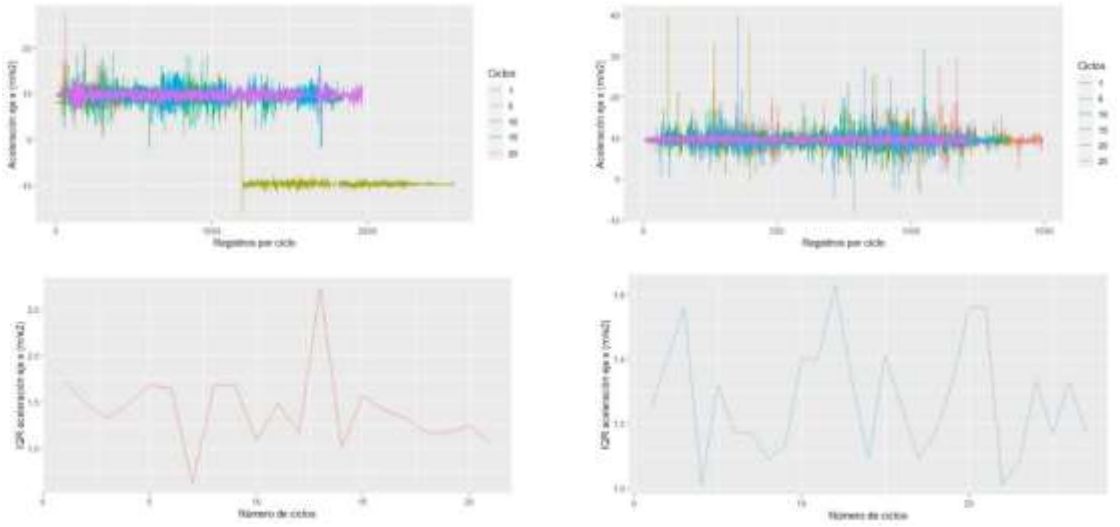
(a) Batería 1 (b) Batería 2

Figura 31. Matriz de correlaciones de los datos en condiciones de tráfico real

Luego del análisis descriptivo, se realiza la selección de las variables, teniendo en cuenta que la variable a predecir es el RUL, el cual, cambia ciclo a ciclo, por lo que se requiere que las variables para el análisis estén medidas ciclo a ciclo. Por esto, se seleccionan las medidas descriptivas para representar cada una de las variables tomadas a partir de sensores para cada uno de los ciclos de funcionamiento, con el fin de llevar a cabo una predicción ciclo a ciclo del RUL obtenido en cada una de las baterías. Para esto, se realiza un análisis que considera el comportamiento de cada una de las variables como aceleraciones, corriente, potencia, temperatura, voltaje, humedad y temperatura ambiental en los ciclos de funcionamiento. Como ilustración, se tomarán los ciclos 1, 5, 10, 15, 20 para la B01 y los ciclos 1, 5, 10, 15, 20 y 25 para la B02, con el fin de mostrar el comportamiento de las variables. Este análisis permite observar el comportamiento general de cada una de las variables y obtener la medida descriptiva que permita representar el comportamiento de la variable en el ciclo determinado.

La Figura 32 presenta el comportamiento de la aceleración obtenida en el eje x de la B01 y la B02 en cada uno de los ciclos de funcionamiento. Esta aceleración representa los cambios obtenidos en el eje x al realizar las pruebas. En estas, se observa que la aceleración oscila entre 10 m/s^2 y -10 m/s^2, donde la B01 muestra un cambio en el comportamiento de la aceleración en el eje x en el ciclo 5 que corresponde a un cambio en

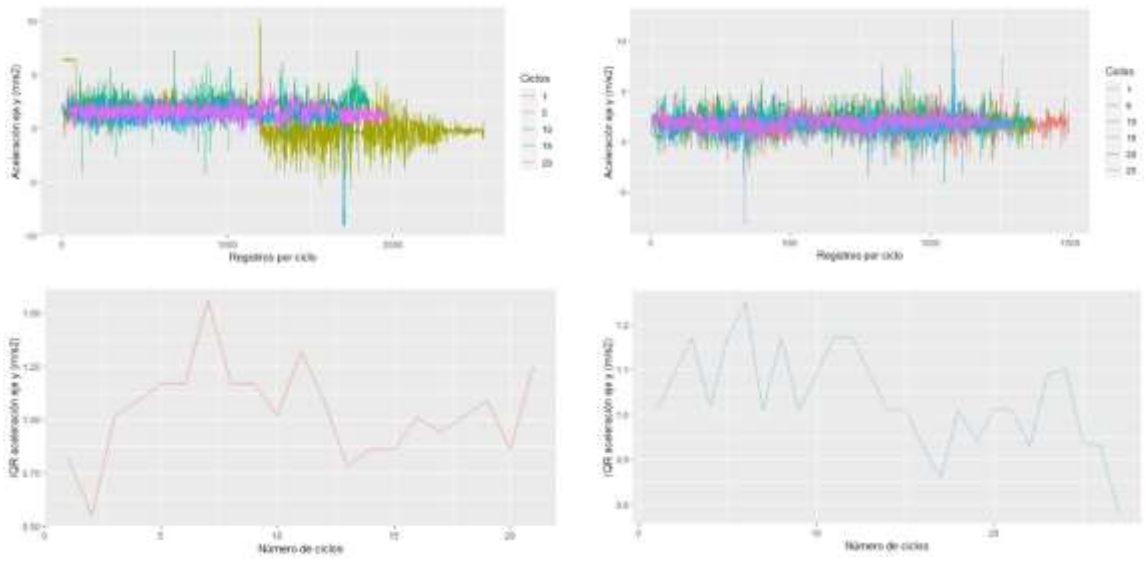
la dirección del EV al realizar la prueba. Debido al comportamiento de la aceleración, se toma como medida descriptiva el rango intercuartílico IQR como medida para representar la variabilidad de la variable obtenida en cada uno de los ciclos de funcionamiento. En general, se observa una mayor amplitud del IQR en la B01 en comparación con la B02, debido al mayor movimiento en el eje x de la B01 en comparación con la B02.



(a) Batería 1 *(b) Batería 2*

Figura 32. Registros de la aceleración en el eje x por ciclo para B01 y B02 en condiciones de tráfico real

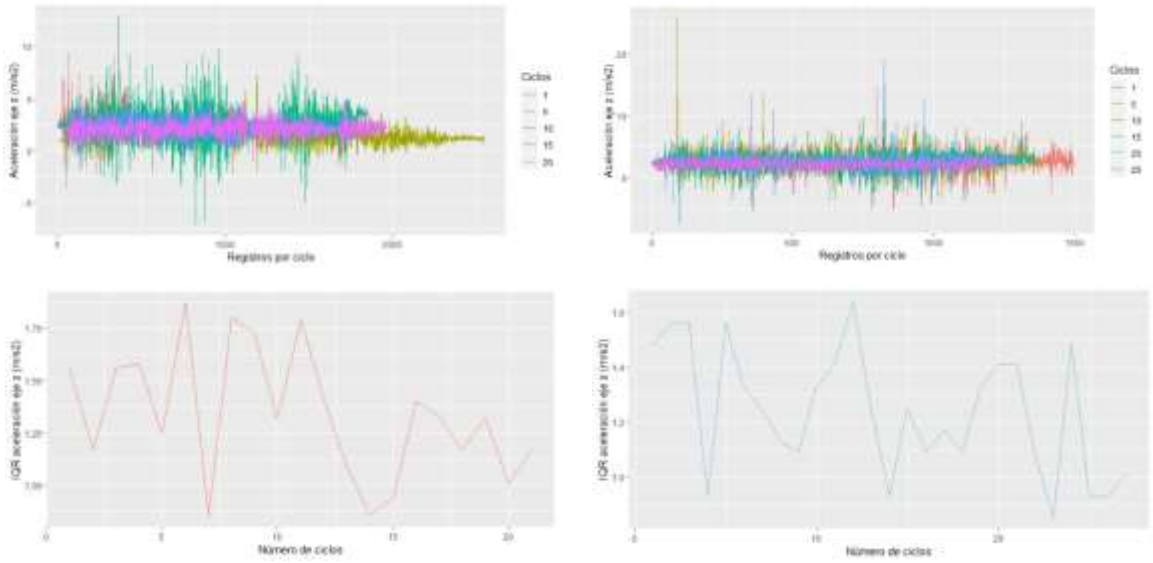
La aceleración en el eje y representa los cambios obtenidos en cada momento que se estaba realizando la prueba, en la Figura 33 se muestran los resultados para la B01 en (a) y para la B02 en (b). En este caso, también se observa que la aceleración en el eje y oscila entre -10 m/s^2 y 10 m/s^2 para la B01 y entre -5 m/s^2 y 10 m/s^2 para la B02. Para representar la variabilidad obtenida en cada uno de los ciclos de la aceleración en el eje y, se toma como medida descriptiva, el IQR obtenido en cada uno de los ciclos de funcionamiento.



(a) Batería 1 *(b) Batería 2*

Figura 33. Registros de la aceleración en el eje y por ciclo para B01 y B02 en condiciones de tráfico real

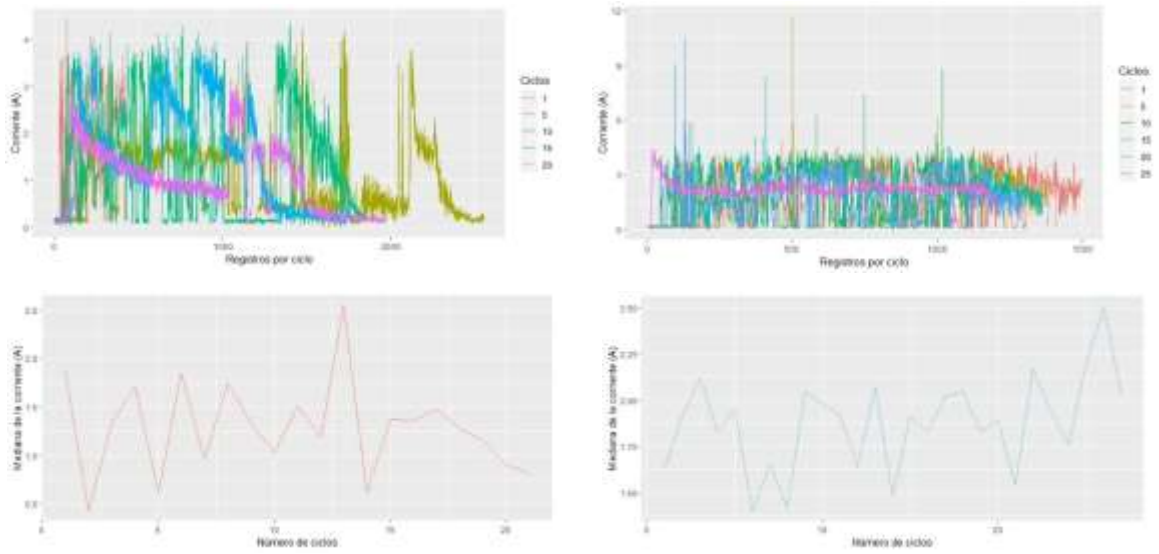
En la Figura 34 se presentan los resultados obtenidos de la aceleración obtenida en el eje z para la B01 en la (a) y para la B02 en la (b). En este caso, la aceleración obtenida en el eje z para la B01 tiene un rango entre -5 m/s^2 y 10 m/s^2, y para la B02, se presenta un comportamiento similar con algunos picos en la aceleración que suben hasta 20 m/s^2. Al igual que con las aceleraciones en el eje x y en el eje y, para la aceleración en el eje z, se toma como medida descriptiva para representar el comportamiento el IQR, el cual varía entre 0.75 m/s^2 y 1.75 m/s^2 para la B01 y entre 0.8 m/s^2 y 1.6 m/s^2 para la B02.



(a) Batería 1 *(b) Batería 2*

Figura 34. Registros de la aceleración en el eje z por ciclo para B01 y B02 en condiciones de tráfico real

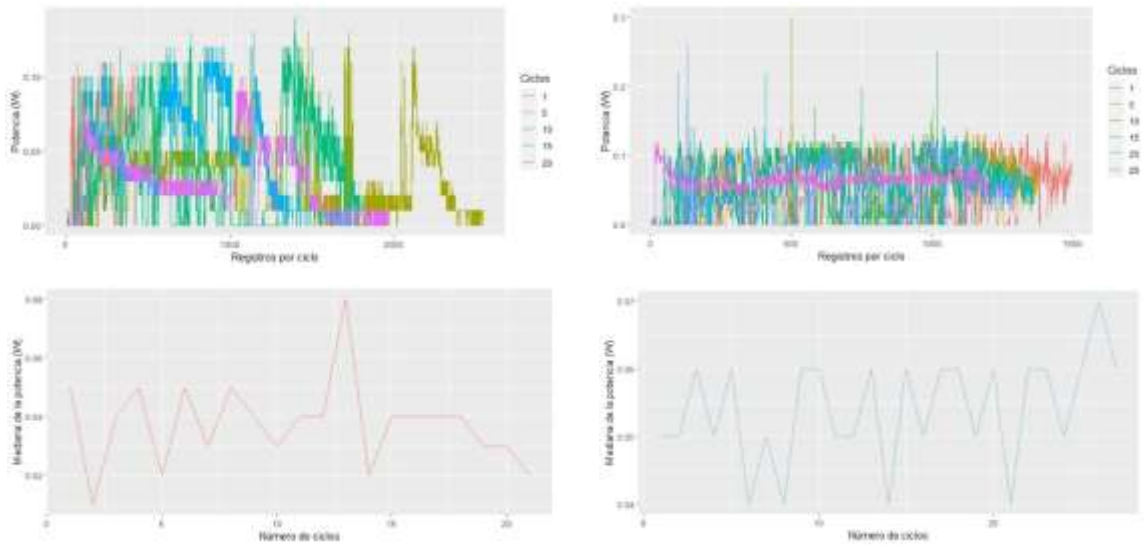
En la Figura 35 se observa el comportamiento de la corriente de la B01 y B02 en cada uno de los ciclos analizados, la cual varía entre 0 amperios y 4 amperios con algunos picos que se generan en el comportamiento de la batería en condiciones de tráfico real. De acuerdo a los resultados descriptivos, se tiene que la corriente tiene una distribución sesgada, por esto, se selecciona como medida descriptiva para representar el comportamiento ciclo a ciclo, la mediana obtenida en cada uno de los ciclos de funcionamiento. Así mismo, se muestra el comportamiento de la corriente al obtener el valor de la mediana para cada uno de los ciclos. En general, se observa un comportamiento de la corriente entre 0.5 amperios y 2.5 amperios. Además, se observa un comportamiento más estable del valor de la corriente para la B02 en comparación con la B01 que muestra más variabilidad en la mediana de la corriente durante los ciclos de funcionamiento de la batería.



(a) Batería 1 (b) Batería 2

Figura 35. Registros de la corriente por ciclo para B01 y B02 en condiciones de tráfico real

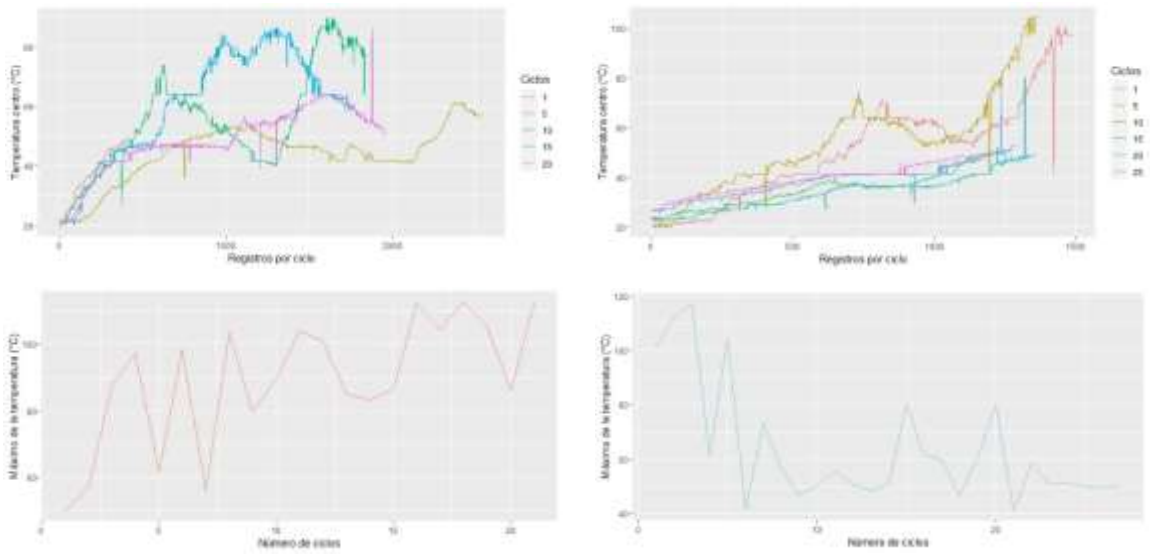
La potencia de la batería a través de cada uno de los ciclos de funcionamiento se presenta en la Figura 36. En esta figura se observa un comportamiento similar a la corriente, donde la potencia oscila entre 0 watts y 0.15 watts para la B01 y de 0 watts a 0.30 watts para la B02 con algunos picos en la potencia, debido al funcionamiento en condiciones de tráfico real. La potencia también presenta un comportamiento sesgado, por lo cual, se toma como medida descriptiva para representar el comportamiento de la potencia la mediana obtenida en cada uno de los ciclos. En la Figura 36 se observa el comportamiento de la mediana de la potencia en cada una de las baterías, en este caso, la potencia oscila entre 0.01 watts y 0.08 watts para la B01 y 0.04 watts y 0.07 watts para la B02. En este caso, la B01 muestra mayor variabilidad en comparación con la B02.



(a) Batería 1 (b) Batería 2

Figura 36. Registros de la potencia por ciclo para B01 y B02 en condiciones de tráfico real

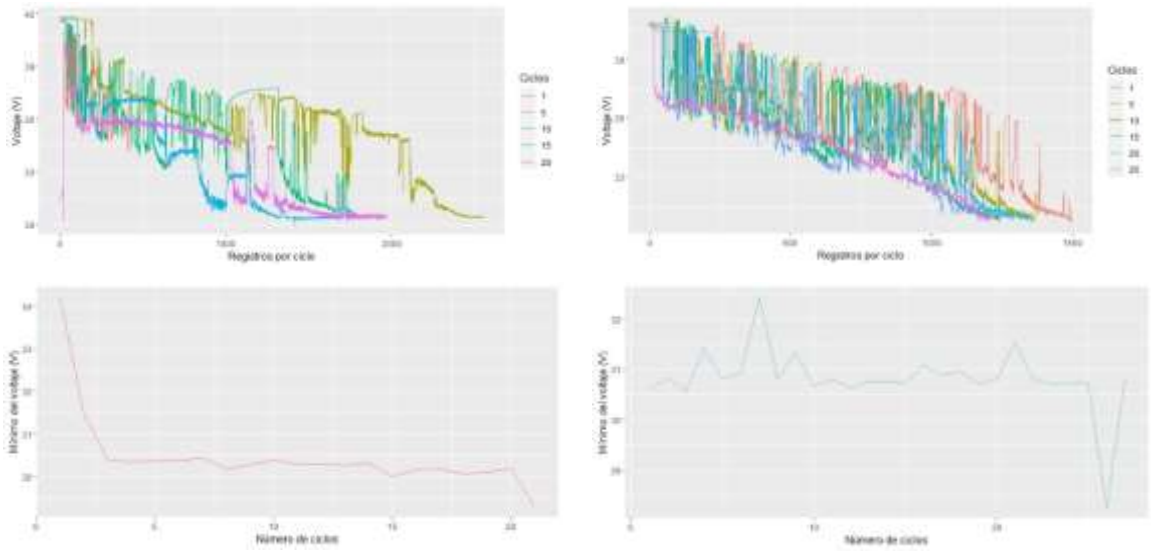
Para el análisis gráfico de la temperatura, se toma la temperatura de la batería en el centro, ya que el comportamiento de la temperatura en un extremo de la batería es similar, y su correlación es del 99%. Para el modelo predictivo, se incluyen ambas temperaturas, la temperatura del centro y del extremo medidas en la batería. En general, la temperatura presenta un comportamiento ascendente en cada uno de los ciclos de funcionamiento, tanto en el centro de la batería como en un extremo. En la Figura 37 se muestran los resultados obtenidos para la temperatura en el centro de la batería. La temperatura de la batería inicia en un valor similar a la temperatura ambiente cuando inician las pruebas y va aumentando a medida que aumentan los ciclos de funcionamiento. Es así, como se identifica que cuando la batería llega al máximo nivel, el ciclo de funcionamiento de la batería tiende a finalizar. Por esta razón, se toma como medida descriptiva, el valor máximo de la temperatura para representar el comportamiento de esta variable en cada uno de los ciclos. En este caso, se observa que la temperatura en general de la B01 va en aumento con algunos picos de variabilidad, sin embargo, en la B02, se observa que al inicio de los ciclos, la batería alcanzó unos niveles de temperatura mayores en comparación con el valor máximo en los demás ciclos de funcionamiento de la B02.



(a) Batería 1 (b) Batería 2

Figura 37. Registros de la temperatura por ciclo para B01 y B02 en condiciones de tráfico real

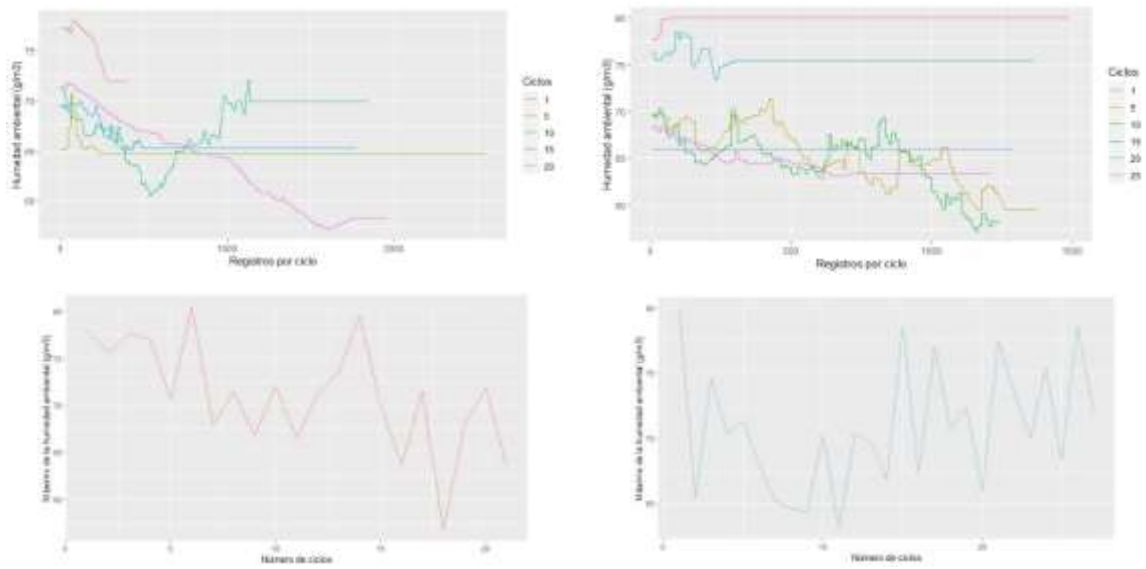
En la Figura 38 se muestra el comportamiento del voltaje de la batería en cada uno de los ciclos de funcionamiento. En esta figura se observa que a medida que aumenta el número de registros por ciclo, el voltaje de la batería empieza a disminuir y el ciclo de funcionamiento finaliza cuando el voltaje toma un valor mínimo, este comportamiento se asemeja con el comportamiento del RUL, ya que a medida que transcurren los ciclos de funcionamiento, el RUL empieza a disminuir. Por este motivo, se toma como medida descriptiva para el voltaje, el valor mínimo que adquiere en cada uno de los ciclos. En cada uno éstos, se observa que el voltaje inicia alrededor de los 40 voltios y va disminuyendo su valor a medida que aumenta el tiempo. Así mismo, se observa un comportamiento más variable al inicio de los ciclos de ambas baterías, luego, se estabiliza el comportamiento del voltaje durante el resto de los ciclos. En general, se observa que ambas baterías inician con un nivel similar de voltaje. Para la B02 se observa una caída en el voltaje al finalizar los ciclos de funcionamiento.



(a) Batería 1 (b) Batería 2

Figura 38. Registros del voltaje por ciclo para B01 y B02 en condiciones de tráfico real

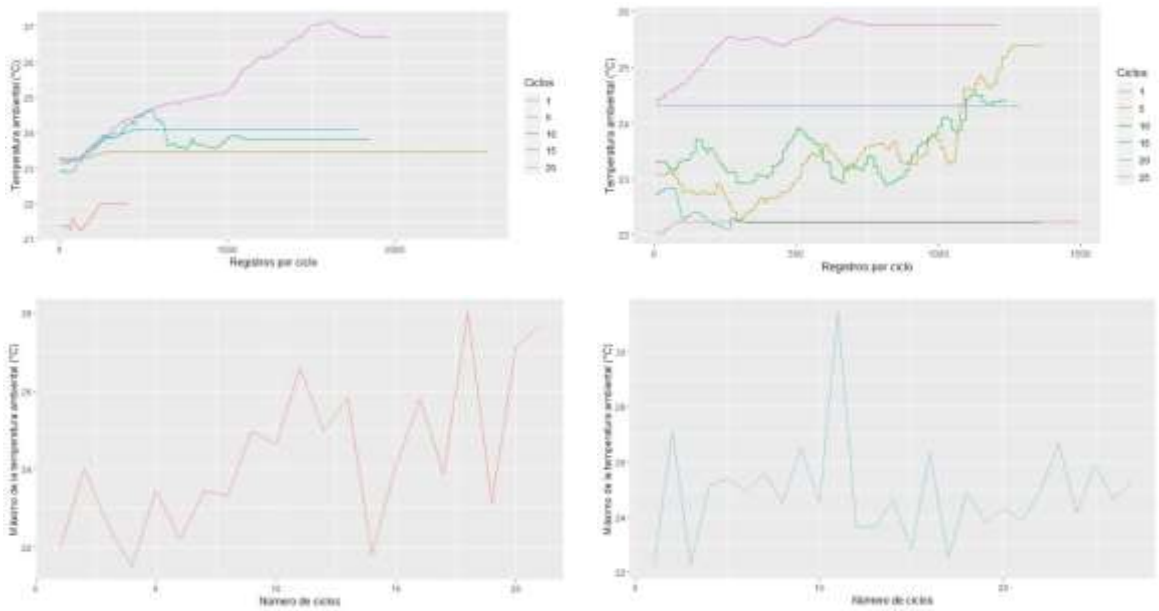
En el análisis descriptivo también se incluyen las variables ambientales capturadas en las pruebas tales como la humedad y la temperatura ambiental, las cuales fueron capturadas por medio de un sensor durante las pruebas desarrolladas. En la Figura 39 se presenta la humedad ambiental obtenida en los ciclos de funcionamiento seleccionados como ilustración para la B01 (a) y para la B02 (b). En este caso, la humedad oscilo entre 55 g/m^3 y 80 g/m^3 para ambas baterías. Debido a que la batería se puede ver más afectada con una humedad ambiental mayor, se selecciona como medida descriptiva para representar el comportamiento de la humedad ciclo a ciclo, la humedad ambiental máxima obtenida en cada uno de los ciclos de funcionamiento. En general, se observa una mayor amplitud de los valores de la humedad ambiental obtenidos en la B01 en comparación con la B02. Además, se observa un comportamiento decreciente para el máximo obtenido de la humedad en la B01 y un comportamiento creciente del máximo de la humedad en la B02.



(a) Batería 1 (b) Batería 2

Figura 39. Registros de la humedad ambiental para B01 y B02 en condiciones de tráfico real

Los resultados obtenidos de la temperatura ambiental para la B01 y la B02 se observan en la Figura 40, como (a) y (b) respectivamente. En este caso, se observa que la temperatura ambiental oscila en valores entre 21°C y 27°C. Para el análisis, se toma la temperatura ambiental máxima obtenida en cada uno de los ciclos de funcionamiento, ya que como se observó con la temperatura medida en el centro de la batería, cuando se alcanza el límite de la temperatura, la batería tiende a terminar su ciclo. De esta forma, se selecciona como medida descriptiva para el comportamiento de la temperatura ambiental, el valor máximo de temperatura obtenido en cada uno de los ciclos. En los resultados de la temperatura ambiental máxima, se observa una mayor amplitud de la temperatura ambiental obtenida en la B02 en comparación con la B01, debido a un pico de temperatura máxima obtenida en el ciclo 11 de las pruebas para la B02. Además, se observa un comportamiento creciente del máximo de la temperatura ambiental para la B01 y un comportamiento más estable para la B02 con algunos picos en los valores obtenidos de la temperatura ambiental máxima.



(a) Batería 1 (b) Batería 2

Figura 40. Registros de la temperatura ambiental para B01 y B02 en condiciones de tráfico real

El análisis anterior permite determinar las medidas descriptivas que muestran el comportamiento de cada una de las variables ciclo a ciclo. Las variables que deben ser suavizadas y las medidas descriptivas seleccionadas para cada una de ellas son: la aceleración en el eje x, y, z, representadas por medio del IQR. La corriente y la potencia, representadas por medio de la mediana obtenida en cada uno de los ciclos. Así mismo, se selecciona la máxima temperatura de la batería en el centro y en un extremo obtenida en cada uno de los ciclos, y se selecciona el mínimo valor obtenido para el voltaje. Para las variables ambientales, se selecciona el máximo valor obtenido en cada uno de los ciclos para la humedad y la temperatura ambiental. Luego del análisis para la selección de variables, se realiza el preprocesamiento de los datos, teniendo en cuenta las medidas descriptivas para cada una de las variables analizadas anteriormente, con las cuales, se genera un nuevo conjunto de datos con esta información. De esta forma, se obtienen los datos de cada una de las baterías, lo que permite realizar el proceso de división de los datos en los conjuntos necesarios según los pasos explicados en la metodología para llevar a cabo el modelo predictivo. El conjunto de datos con las variables analizadas, es el insumo para la primera etapa de la metodología donde se realiza la división de los datos considerando: datos de degradación, datos para la CNN y los datos para el modelo predictivo con la red LSTM.

6.2.2 Implementación de la metodología

Para la implementación de la metodología propuesta usando los datos de baterías en condiciones de tráfico real, se hizo uso del paquete en el lenguaje de programación R `cnnlstm`, el cual, brinda las funciones necesarias para realizar cada uno de los pasos necesarios y generar las predicciones. En esta sección se detallan cada uno de los pasos propuestos de la metodología como la división de los datos, el modelo de suavizamiento con la CNN y el modelo predictivo con la red LSTM.

6.2.2.1 División de los datos

Obtener datos de degradación

Para obtener los datos de degradación, se tiene en cuenta el análisis anterior en el que se destaca el comportamiento del voltaje, el cual, es similar al comportamiento del RUL, ya que a medida que aumenta el tiempo en cada uno de los ciclos, el voltaje empieza a disminuir. Por esto, se toma como variable de desempeño el voltaje, ya que, según el análisis presentado anteriormente, el voltaje es la variable que muestra el comportamiento de degradación que se genera en las baterías (Hong et al., 2019). Así mismo, debido a la variabilidad de los datos en condiciones de tráfico real se selecciona como método para la identificación de los datos de degradación una carta de control $\bar{X} \pm L\sigma$, considerando tres desviaciones estándar de acuerdo al valor del voltaje (X. Li et al., 2021). En la Figura 41 se muestra el comportamiento del voltaje evaluado con la carta de control $\bar{X} \pm 3\sigma$ para la B01, en esta se presentan los límites UCL y LCL. En esta, se observa que deben ser eliminados inicialmente 3 ciclos para seguir con el proceso de la carta de control y determinar el número de ciclos a eliminar para obtener los resultados de los ciclos dentro de los límites definidos en la carta.

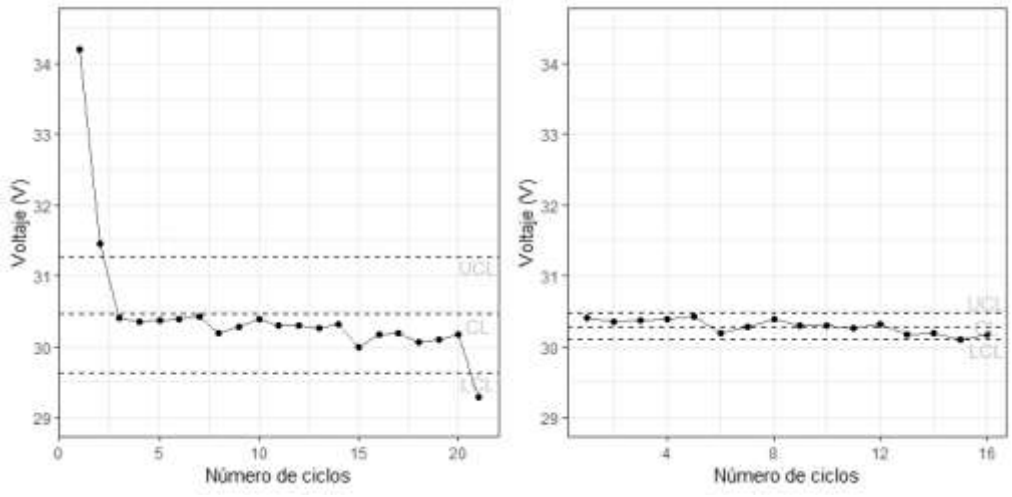


Figura 41. Evaluación de los datos de degradación para la B01 en condiciones de tráfico real

Teniendo en cuenta este criterio, en la Tabla 27 se presenta la cantidad de ciclos determinados como datos de degradación *k* que deben ser eliminados del conjunto de datos para el análisis y la identificación de cada uno de ellos. De esta forma, se eliminan 5 ciclos de la B01 y 2 ciclos de la B02 identificados como datos de degradación. Estos ciclos eliminados se evalúan en cada uno de los escenarios planteados, con el fin de validar el desempeño del modelo predictivo sin tener en cuenta estos datos de degradación.

Tabla 27

Ciclos de degradación en condiciones controladas

Batería	**Número de ciclos Datos de degradación *k***	**Identificación de ciclos**
B01	5 ciclos	1, 2, 21, 15, 17
B02	2 ciclos	7, 26

6.2.2.2 Modelo de suavizamiento con la CNN

En la Tabla 24 se presentaron las variables obtenidas por medio de sensores, de las cuales, se toman las variables de aceleración en el eje x, eje y, eje z, corriente, potencia, voltaje, temperatura en el centro y extremo de la batería, humedad y temperatura ambiental, con el objetivo de suavizarlas a partir del modelo de suavizamiento con la CNN. De esta forma, se tienen 10 variables a suavizar para cada una de las baterías con 16 registros para la B01 y 25 registros para la B02, las cuales deben estructurarse una a una para cada batería con el fin de generar los datos de entrada para el modelo de suavizamiento. Para el modelo, se inicia con la estructuración de los datos, en los que se tienen en cuenta cada una de las 10 variables mencionadas anteriormente y un número de pasos de tres observaciones para cada una de las variables. De esta forma, se generan series univariadas para cada una de las variables y cada una de las baterías a suavizar y se toma el promedio de las tres observaciones como el valor predicho para cada variable, con el objetivo de generar el suavizamiento.

Luego de la estructuración de los datos, se considera la arquitectura de la CNN propuesta en el capítulo de metodología para el suavizamiento de cada una de las variables obtenidas por medio de sensores. En esta, según los valores de revisión de literatura presentados en la Tabla 7, se genera un modelo de suavizamiento que considera unos hiperparámetros fijos para la arquitectura tales como una capa de convolución con un tamaño de filtro de 2 y una función de activación *relu*, luego se considera una capa de *maxpooling* de tamaño 2, una capa de *flatten* y finalmente dos capas densas, donde la última capa tiene 1 neurona de salida que se encarga de brindar el resultado del suavizamiento.

El modelo con la CNN es compilado con el optimizador Adam y la función de pérdida MSE. Luego de la definición de los hiperparámetros fijos presentados en la metodología, se realiza la optimización de los hiperparámetros modificando los valores de éstos a partir de una búsqueda en cuadrícula con dos o tres valores definidos para cada hiperparámetro que cambia. En la Tabla 28 se presentan los hiperparámetros y los valores que toman cada uno de éstos dentro de la optimización.

Tabla 28

Hiperparámetros y valores a optimizar del modelo de suavizamiento en condiciones de tráfico real

Hiperparámetro	Valores
Número de filtros	16, 32, 64
Número de neuronas	50, 100
Tasa de aprendizaje	0.001, 0.01
Épocas	100, 500

Al definir estos valores, se generaron y entrenaron 24 modelos que combinan cada uno de los valores de los hiperparámetros, con el fin de encontrar la mejor combinación de hiperparámetros que genera un mejor desempeño en el suavizamiento de las variables de cada batería. Al ejecutar la función con los datos de entrenamiento y prueba de cada una de las variables, se selecciona la combinación de hiperparámetros que menor error de entrenamiento y prueba generan para ambos conjuntos de datos, y se obtiene como resultado los siguientes valores: 32 filtros para la capa convolucional con 50 neuronas para la capa densa, una tasa de aprendizaje con valor de 0.001 y 500 épocas para el proceso de entrenamiento de los modelos. En la Tabla 29 se presentan los errores obtenidos en cada uno de los conjuntos de datos: entrenamiento y prueba para cada una de las variables de las baterías y se resaltan los menores errores obtenidos en cada variable durante el entrenamiento o la prueba.

En esta se observa que la variable mejor suavizada es el voltaje de la B02, ya que es la variable que menos variabilidad tiene en comparación con las demás. Así mismo, se observa que el suavizamiento del modelo con los datos de entrenamiento genera un buen suavizamiento para los datos de prueba para la B01 y en la B02 se observan menores errores en ambos conjuntos de datos: entrenamiento y prueba. En general, se observa un mejor suavizamiento para los datos de prueba de la B01 en comparación con la B02. La temperatura ambiental de la B02 es la variable que obtuvo un mayor valor de error comparada con las demás variables para ambas baterías, esto puede deberse al comportamiento de la variable a través de cada uno de los ciclos de funcionamiento. Para la corriente y la potencia, se observa una mejor predicción para la B01 en comparación con la B02, sin embargo, los errores en el suavizamiento son similares, lo que permite la generalización del modelo de suavizamiento para realizar el proceso de suavización a las variables obtenidas por medio de sensores.

Tabla 29

Errores de entrenamiento y prueba obtenidos en el modelo de suavizamiento con los datos en condiciones de tráfico real

Batería	Variable	Error de entrenamiento				Error de prueba			
		RMSE	**MAPE**	**MAE**	**MSE**	**RMSE**	**MAPE**	**MAE**	**MSE**
1	Aceleración eje x	0.0408	7.94	0.0256	0.0017	**0.0187**	**3.43**	**0.0166**	**0.0004**
	Aceleración eje y	0.0460	5.76	0.0271	0.0021	**0.0149**	**3.16**	**0.0136**	**0.0002**
	Aceleración eje z	0.0914	13.0	0.0678	0.0083	**0.0236**	**5.8**	**0.0200**	**0.0006**
	Corriente	0.0341	8.16	0.0287	0.0012	**0.0310**	**5.02**	**0.0277**	**0.0010**
	Potencia	0.0317	7.29	0.0265	0.0010	**0.0231**	**3.94**	**0.0207**	**0.0005**
	Temperatura centro	0.0587	8.72	0.0408	0.0035	**0.0367**	**4.24**	**0.0339**	**0.0014**
	Temperatura en un extremo	0.0981	12.3	0.0772	0.0096	**0.0380**	**4.51**	**0.0348**	**0.0014**
	Voltaje	0.0497	9.47	0.0267	0.0025	**0.0185**	**8.26**	**0.0170**	**0.0003**
	Humedad ambiental	0.0384	5.25	0.0297	0.0015	**0.0122**	**2.06**	**0.0094**	**0.0002**
	Temperatura ambiental	**0.0832**	**10.8**	**0.0579**	**0.0069**	0.1330	17.7	0.108	0.0178
2	Aceleración eje x	**0.0652**	**11.2**	**0.0538**	**0.0043**	0.0888	19.8	0.0702	0.0078
	Aceleración eje y	0.0518	8.52	0.0447	0.0027	**0.0280**	**3.54**	**0.0199**	**0.0008**
	Aceleración eje z	0.0733	**12.1**	0.0561	0.0054	**0.0661**	13.5	**0.0499**	**0.0044**
	Corriente	0.0397	8.21	0.0333	0.0016	**0.0349**	**5.94**	**0.0268**	**0.0012**
	Potencia	**0.0353**	**8.23**	**0.0312**	**0.0013**	0.0657	11.6	0.0568	0.0043
	Temperatura centro	0.0355	11.9	0.0287	0.0013	**0.0276**	**11.4**	**0.0236**	**0.0008**
	Temperatura en un extremo	**0.0324**	**8.25**	**0.0251**	**0.0011**	0.0912	20.0	0.0695	0.0083
	Voltaje	**0.0023**	**0.26**	**0.0016**	**0.0000**	0.0061	0.75	0.0047	0.0000
	Humedad ambiental	**0.0482**	**6.84**	**0.0356**	**0.0023**	0.0710	13.9	0.0548	0.0050
	Temperatura ambiental	**0.0862**	**18.8**	**0.0667**	**0.0074**	0.0935	22.0	0.0814	0.0087

Luego de realizar el entrenamiento y optimización del modelo, se selecciona la combinación que minimiza los errores y genera un mejor desempeño en el modelo de suavizamiento. Es así como se realiza el suavizamiento para cada una de las variables de cada una de las baterías. Los resultados del suavizamiento para cada variable y cada batería se muestran en la Tabla 30. En general, se observa un buen desempeño en el suavizamiento de las *features* obtenidas por medio de sensores, las cuales, serán las *features* de entrada para el modelo predictivo con la red LSTM. En esta se observa un menor error en la predicción del voltaje de la B02, el cual, puede estar asociado al comportamiento de la misma. Así mismo, se observa un mayor valor de error para las temperaturas de la B02, ya que presenta un comportamiento más variable. Se presentaron menores errores en el suavizamiento de las variables obtenidas en la B01 en comparación con

la B02. Sin embargo, los errores obtenidos en cada una de las variables suavizadas son similares y se resaltan en la tabla los errores de las variables que obtuvieron un menor error de suavizamiento comparando los resultados entre las dos baterías.

Tabla 30

Errores en el suavizamiento de los datos en condiciones de tráfico real

Batería	Variable	RMSE	MAPE	MAE	MSE
1	Aceleración eje x	**0.0589**	**9.26**	**0.0428**	**0.0035**
	Aceleración eje y	**0.0434**	**5.96**	**0.0300**	**0.0019**
	Aceleración eje z	0.0957	16.5	0.0805	0.0092
	Corriente	**0.0751**	**13.4**	**0.0626**	**0.0057**
	Potencia	**0.0729**	**12.6**	**0.0571**	**0.0053**
	Temperatura centro	**0.0716**	**10.2**	**0.0593**	**0.0051**
	Temperatura en un extremo	**0.0683**	**9.16**	**0.0562**	**0.0047**
	Voltaje	0.0476	10.2	0.0282	0.0023
	Humedad ambiental	**0.0684**	**10.3**	**0.0577**	**0.0047**
	Temperatura ambiental	**0.0626**	**13.5**	**0.0514**	**0.0039**
2	Aceleración eje x	0.0851	13.2	0.0617	0.0072
	Aceleración eje y	0.0577	9.33	0.0483	0.0033
	Aceleración eje z	0.0846	12.4	0.0633	0.0072
	Corriente	0.0792	15.0	0.0618	0.0063
	Potencia	0.0904	15.8	0.0738	0.0082
	Temperatura centro	0.0768	21.2	0.0623	0.0059
	Temperatura en un extremo	0.0999	19.4	0.0792	0.0100
	Voltaje	**0.0087**	**0.91**	**0.0059**	**0.0001**
	Humedad ambiental	0.0738	15.1	0.0632	0.0055
	Temperatura ambiental	0.0840	14.8	0.0561	0.0071

Después de obtener estos resultados en el suavizamiento a partir del modelo con la CNN, se hace una integración de cada una de las variables de las baterías, que permita obtener un nuevo conjunto de datos con las variables obtenidas por medio de sensores suavizadas y con las variables normalizadas a través de la normalización del mínimo y máximo de las variables obtenidas sin sensores. Esto permite obtener el conjunto de datos de entrada para el modelo predictivo con la red LSTM.

6.2.2.3 Modelo predictivo con la red LSTM

El modelo predictivo toma como datos de entrada, los datos suavizados anteriormente y los une con los datos obtenidos sin sensores, a los cuales, se les restan las dos primeras observaciones para cada variable obtenida sin sensores, ya que estas observaciones corresponden a los datos que se perdieron en el suavizamiento realizado en el paso anterior. Después de restar las dos observaciones, se normaliza el conjunto de datos obtenidos sin sensores a partir de la normalización min-max. Como resultado, se genera el conjunto de datos de las variables obtenidas sin sensores normalizadas.

Luego, este conjunto de datos obtenidos sin sensores normalizados, se une con el conjunto de datos suavizados en el paso anterior para conformar el conjunto de datos que permite realizar el modelo predictivo. Este conjunto de datos está conformado por las variables de ciclo, aceleración en el eje x, aceleración en el eje y, aceleración en el eje z, corriente, potencia, temperatura (centro y extremo de la batería), voltaje de la batería, humedad y temperatura ambiental, y la variable tiempo de cada ciclo para hacer la predicción con 14 registros para la B01 y 23 registros para la B02. De esta forma, para cada una de las baterías se toman las variables y un número de pasos igual a tres, con el fin de reducir la pérdida de datos. Esta información permite generar los datos de entrada en la estructura necesaria para la red LSTM.

Para la arquitectura de la red LSTM, según los datos presentados en la revisión de literatura en la Tabla 8 se inicia con dos capas LSTM, de las cuales, la primera capa LSTM genera secuencia de retorno para la siguiente capa. También se consideran como parámetros fijos la función de activación tangente hiperbólica, dos capas de *dropout* en el medio de cada capa LSTM y una capa densa al final de la red con una neurona de salida para brindar la predicción. El modelo predictivo con la red LSTM es compilado con el optimizador Adam y la función de pérdida implementada es el MSE. Así mismo, se realiza el entrenamiento del modelo con los datos de entrenamiento obtenidos en la etapa de división de los datos. Con este planteamiento, se realiza el proceso de optimización de los hiperparámetros modificando los valores de éstos a partir de la búsqueda en cuadrícula con dos valores en el conjunto especificado. En la Tabla 31 se presentan los hiperparámetros y los valores que tomará cada uno de éstos en la optimización de los hiperparámetros.

Tabla 31

Hiperparámetros y valores a optimizar del modelo predictivo en condiciones de tráfico real

Hiperparámetro	Valores
Número de neuronas primera capa	64, 128
Tasa *dropout* primera capa	0.1, 0.2, 0.3
Número de neuronas segunda capa	32, 64
Tasa *dropout* segunda capa	0.1, 0.2, 0.3
Tasa de aprendizaje	0.001, 0.01
Épocas	100, 500

De acuerdo a los valores presentados, se deben generar, entrenar y evaluar 144 modelos predictivos con la combinación de cada uno de los valores que pueden tomar los hiperparámetros. Para esto, se implementa la función de optimización de los hiperparámetros con los datos para la selección de la mejor combinación de hiperparámetros que minimizan el error en la predicción. La función genera como resultado usar 128 neuronas en la primera capa LSTM, una tasa de *dropout* del 0.1 en la primera capa, 32 neuronas en la segunda capa LSTM y una capa de *dropout* con una tasa de 0.2. Además, la función de optimización define

como tasa de aprendizaje un valor de 0.01 con 500 épocas para el entrenamiento del modelo predictivo. En la Tabla 32 se presentan cada uno de los errores obtenidos con los datos de entrenamiento, validación y prueba de la predicción del RUL con la optimización realizada anteriormente. En general, se obtiene un buen error en la predicción comparado con los resultados obtenidos en la revisión de literatura.

Tabla 32

Errores en el entrenamiento y prueba de la predicción del RUL en condiciones de tráfico real

Tipo de error	RMSE	MAPE	MAE	MSE
Entrenamiento	0.0175	9.28	0.0151	0.0003
Validación	0.0253	11.1	0.0226	0.0006
Prueba	0.0361	14.2	0.0261	0.0013

6.2.3 Evaluación de la precisión del RUL

Después del entrenamiento del modelo y optimización de los hiperparámetros, se realiza la evaluación de la precisión en la predicción del RUL considerando diferentes escenarios para la evaluación del modelo de suavizamiento y el modelo predictivo. En este caso, se consideran cuatro escenarios teniendo en cuenta lo siguiente: un escenario en el que no se consideran los datos de degradación, no se realiza el suavizamiento y se implementa el modelo predictivo con la red LSTM. El segundo escenario tampoco considera los datos de degradación, pero en este caso, si considera el modelo de suavizamiento y el modelo predictivo CNN-LSTM. En el tercer escenario, se consideran los datos de degradación, sin embargo, no se considera el modelo de suavizamiento y se implementa la red LSTM como modelo predictivo. Por último, en el cuarto escenario, se consideran los datos de degradación, el suavizamiento a partir de la CNN y el modelo predictivo con la red LSTM. En la Tabla 33 se muestra el resumen de cada uno de los escenarios planteados para la evaluación de los modelos.

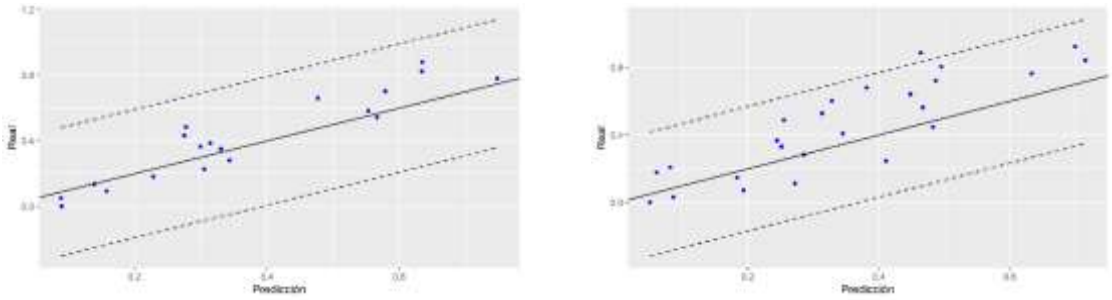
Tabla 33

Planteamiento de escenarios para datos de baterías en condiciones de tráfico real

Escenario	División de datos	Suavizamiento	Modelo predictivo
1	Train, validation, test	No	LSTM
2	Train, validation, test	Sí	CNN-LSTM
3	Train, validation, test, degradation	No	LSTM
4	Train, validation, test, degradation	Sí	CNN-LSTM

Para cada uno de los escenarios, se muestran los resultados obtenidos de manera gráfica y los errores obtenidos de acuerdo al RMSE, MAPE, MAE y MSE obtenidos al validar la predicción generada en cada una de las baterías. En la Figura 42 se observan los resultados obtenidos del escenario 1 en el que se hizo la división de los datos en tres conjuntos: entrenamiento, validación y prueba, no se realiza suavizamiento a los datos obtenidos por medio de sensores y se implementa la red LSTM en el modelo predictivo. En este

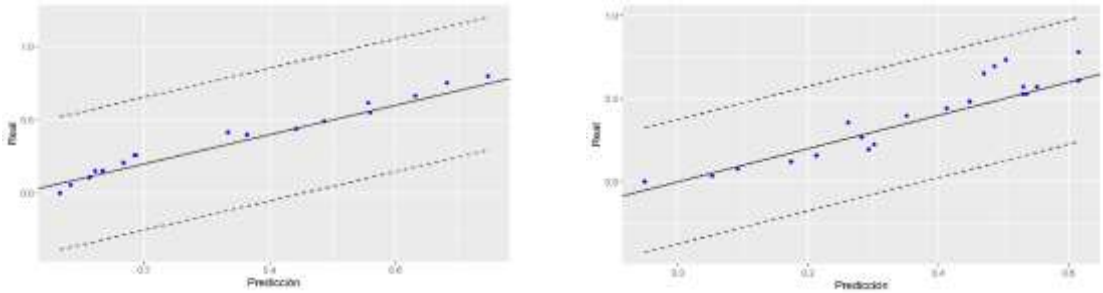
caso, se observa que para la B01 no hay un buen ajuste en la mayoría de las predicciones, donde se observa que la predicción queda por encima al inicio, luego, las predicciones quedan debajo de los valores reales del RUL, es decir, se genera una subestimación, lo que no permite un buen ajuste en la predicción. Así mismo, para la B02, se observa que al iniciar las predicciones no hay un buen ajuste, ya que las predicciones quedan por encima de los valores reales del RUL. Al final de la gráfica también se observa una subestimación para varios de los ciclos de la B02, ya que los puntos aparecen por debajo de la línea.



(a) Batería 1 (b) Batería 2

Figura 42. Resultados escenario 1 con datos en condiciones de tráfico real

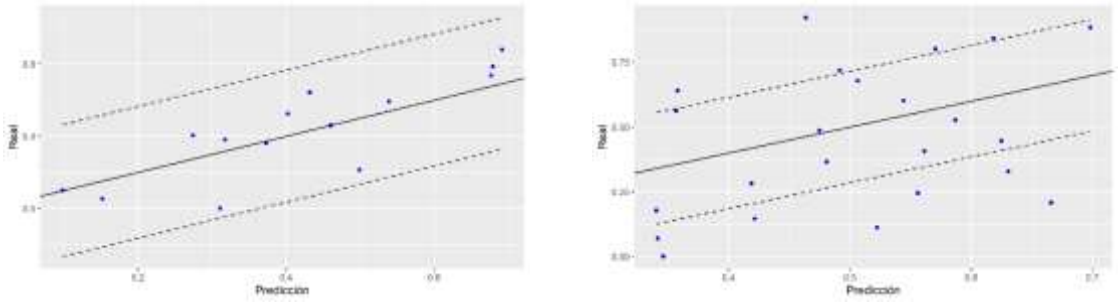
En la Figura 43 se observan los resultados obtenidos del escenario 2 en el que se hizo la división de los datos en tres conjuntos: entrenamiento, validación y prueba, se realiza suavizamiento de los datos obtenidos por medio de sensores a través de la CNN y se implementa la red LSTM en el modelo predictivo. En este caso, se observa que para la B01 tiene un buen ajuste de las predicciones respecto a los valores reales del RUL, con una subestimación en algunos de los valores reales al finalizar la gráfica, sin embargo, la distancia del punto real al predicho es menor en comparación con el escenario anterior. Para la B02, también se observa un buen ajuste de las predicciones respecto a los valores reales, se observa que al iniciar se sobreestiman algunas de las predicciones y al finalizar se subestiman algunas de éstas, sin embargo, el error es menor comparado con los resultados del escenario 1 para ambas baterías.



(a) Batería 1 (b) Batería 2

Figura 43. Resultados escenario 2 con datos en condiciones de tráfico real

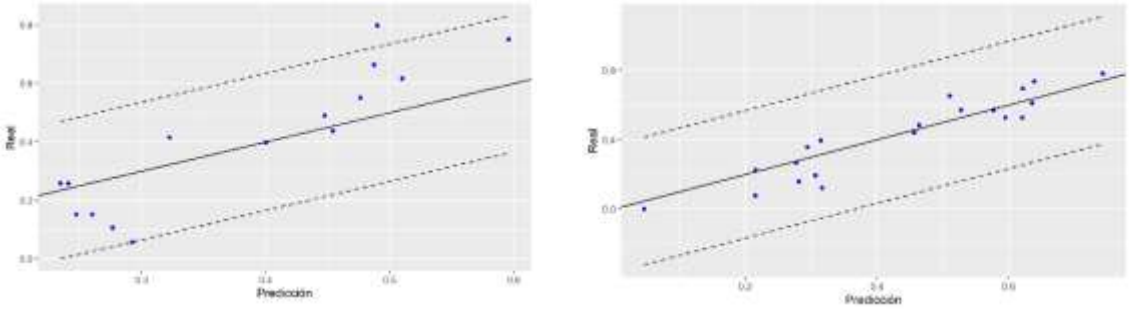
En la Figura 44 se observan los resultados obtenidos del escenario 3 en el que se hizo la división de los datos en cuatro conjuntos: entrenamiento, validación, prueba y degradación. En este caso, no se realiza suavizamiento de los datos obtenidos por medio de sensores y se implementa la red LSTM en el modelo predictivo. De esta forma, se observa que, para la B01, se presenta una buena predicción en algunos puntos, sin embargo, al inicio, se sobreestiman las predicciones, y al finalizar los puntos de la gráfica se subestima la predicción en algunos puntos, ya que los puntos aparecen por debajo de la línea. En el caso de la B02, se observa un comportamiento diferente, ya que no se observa un buen comportamiento de las predicciones versus los valores reales. Varios puntos son sobreestimados y subestimados, lo que genera que este escenario no tenga buenos resultados en cuanto a la predicción.



(a) Batería 1 (b) Batería 2

Figura 44. Resultados escenario 3 con datos en condiciones de tráfico real

En la Figura 45 se observan los resultados obtenidos del escenario 4 en el que se hizo la división de los datos en cuatro conjuntos: entrenamiento, validación, prueba y degradación. Además, se realiza suavizamiento de los datos obtenidos por medio de sensores a través de la CNN y se implementa la red LSTM en el modelo predictivo. En este caso, se observa que, para la B01, se subestiman en general varias de las predicciones, ya que la mayoría de los puntos queda por debajo de la línea. En el caso de la B02, se observa un comportamiento de sobreestimación para los primeros puntos graficados, ya que estos puntos aparecen por encima de la línea, luego, el modelo tiende a mejorar su predicción versus los datos reales del RUL, y en algunos casos, se subestiman las predicciones.



(a) Batería 1 (b) Batería 2

Figura 45. Resultados escenario 4 con datos en condiciones de tráfico real

En la Tabla 34 se observa el resumen de los errores obtenidos en cada una de las predicciones de las baterías. Los errores que se generan en la predicción se miden a partir del RMSE, MAPE, MAE y MSE. En general, se observa que en el escenario 1 se obtiene un menor error de entrenamiento comparado con los demás escenarios, sin embargo, los resultados de los errores con los datos de entrenamiento para el escenario 1 y el escenario 2 son similares. Así mismo, el escenario 2 obtiene una mejor predicción del RUL con los datos de validación y prueba. En estos escenarios se destaca el valor obtenido en cada uno de los errores, destacándose la combinación del modelo de suavizamiento con la CNN con el modelo predictivo a partir de la red LSTM para generar una predicción más precisa del RUL. Así mismo, se destaca la importancia del suavizamiento de los datos obtenidos por medio de sensores, ya que permiten un mejor ajuste en la predicción del RUL.

Además, se resaltan los resultados obtenidos en el escenario 4, con la implementación de la división de los datos en cuatro conjuntos de datos: entrenamiento, validación, prueba y degradación junto con el modelo de suavizamiento con la CNN y el modelo predictivo con la red LSTM. En este escenario en general, se tiene un menor error comparado con el escenario 1 y 3, los cuales, no consideran el suavizamiento de los datos obtenidos por medio de sensores.

Tabla 34

Errores obtenidos en cada uno de los escenarios y baterías en condiciones de tráfico real

Escenario	Error	Entrenamiento	Validación	Prueba
1	RMSE	**0.0106**	0.0440	0.1510
	MAPE	**6.81**	19.7	39.9
	MAE	**0.0088**	0.0391	0.1270
	MSE	**0.0001**	0.0019	0.0229
2	RMSE	0.0152	**0.0216**	**0.0939**
	MAPE	7.12	**13.8**	**20.4**
	MAE	0.0093	**0.0195**	**0.0683**
	MSE	0.0002	**0.0005**	**0.0088**
3	RMSE	0.0233	0.1760	0.1710
	MAPE	10.0	45.9	34.7
	MAE	0.0194	0.2180	0.1300
	MSE	0.0005	0.0678	0.0291
4	RMSE	0.0178	0.0568	0.1280
	MAPE	10.2	9.55	24.1
	MAE	0.0154	0.0568	0.0907
	MSE	0.0003	0.0032	0.0163

En general para el conjunto de datos en condiciones controladas y en condiciones de tráfico real, se observa la importancia de realizar el suavizamiento a los datos obtenidos por medio de sensores, lo que permite generar datos suavizados como entrada para que el modelo predictivo pueda realizar sus predicciones de

forma más precisa. La suavización permite eliminar el ruido obtenido en la medición por medio del sensor y generar una medida que permita capturar el comportamiento de la variable. Esto permite disminuir los errores obtenidos en el entrenamiento, prueba y validación del modelo entre un 10% y 15%.

Así mismo, se destaca la importancia de la identificación de los datos de degradación en ambos conjuntos de datos, para el conjunto de datos en condiciones controladas, se observa que este escenario, el que considera la división de los datos en cuatro conjuntos: entrenamiento, validación, prueba y degradación y el modelo con la CNN-LSTM, es el que permite obtener mejores resultados en comparación con los demás escenarios, debido a la cantidad de ciclos de funcionamiento de las baterías. Sin embargo, para el caso del conjunto de datos en condiciones de tráfico real, se observa que da un mejor resultado considerando tres conjuntos de datos: entrenamiento, validación y prueba y el modelo con la CNN-LSTM, esto puede deberse a la cantidad de datos de degradación obtenidos en el análisis de los datos. Esto resalta la importancia de obtener más datos en condiciones de tráfico real que permitan realizar un análisis de más datos para aplicar la metodología.

Así mismo, se observa un comportamiento más estable para las variables obtenidas en el conjunto de datos en condiciones controladas, en comparación con los datos en condiciones de tráfico real. Se observa que la metodología propuesta permite generar una predicción precisa para ambos casos en los que se tienen conjuntos de datos en condiciones diferentes. Se destaca la importancia de generar datos en condiciones de tráfico real que permitan evaluar la generalización de los modelos predictivos propuestos.

7. CONCLUSIONES

El modelo predictivo propuesto incorpora aspectos complejos y específicos en la metodología utilizada para su desarrollo. Se realiza un proceso de suavizado a los datos adquiridos a través de sensores por medio de una CNN, lo que permite eliminar el ruido presente en los datos obtenidos a partir de sensores. Los resultados obtenidos en los dos casos de estudio, un caso en condiciones controladas y otro en condiciones de tráfico real, demuestran que, al realizar el suavizado de los datos obtenidos por medio de sensores, el rendimiento del modelo predictivo mejora significativamente utilizando una red LSTM de múltiples entradas a una sola salida.

Además, mediante la identificación de los datos de degradación, se logra una mejor precisión en la predicción del RUL, como se observó en el escenario 4 para las baterías en condiciones controladas. Con esta evaluación, se obtiene que, en ambos casos, la metodología propuesta muestra un mejor rendimiento en la predicción del RUL, minimizando los errores y obteniendo un RMSE inferior al 0.04 en los datos en condiciones controladas y 0.095 en los datos en condiciones de tráfico real.

La metodología propuesta se implementó en el paquete `cnnlstm`, desarrollado en lenguaje de programación R, lo que permite usar el modelo propuesto en diferentes casos con datos bajo condiciones similares. Este paquete ofrece diversas funciones para generar el modelo de suavizamiento mediante la CNN, tanto para las variables obtenidas a través de sensores como para aquellas variables que presenten ruido en sus mediciones. También permite la implementación del modelo predictivo mediante la red LSTM. Además, el paquete ofrece la flexibilidad de usar las funciones de forma independiente o en conjunto para generar el modelo CNN-LSTM, según las necesidades de cada caso de estudio.

Así mismo, el paquete proporciona funciones que permiten la optimización de los hiperparámetros de la arquitectura, la compilación y el entrenamiento de los dos modelos, el modelo de suavizamiento con la CNN y el modelo predictivo con la red LSTM. Esta optimización tiene como objetivo determinar los valores de los hiperparámetros que minimizan los errores en los conjuntos de datos y permitan obtener predicciones más precisas.

Según los resultados presentados en los dos casos analizados, bajo condiciones controladas y condiciones de tráfico real, se observa que, para las métricas de desempeño como RMSE, MAPE, MAE, los modelos híbridos tienen un mejor desempeño en comparación con el modelo predictivo que implementa sólo un tipo de ANN. Estos modelos híbridos permiten combinar las diferentes características de cada una de las metodologías para lograr una mejor precisión de la predicción del RUL. De esta forma, se resalta el potencial de las redes neuronales en los modelos predictivos para capturar el comportamiento del RUL de un componente en un EV.

Para validar el modelo propuesto, se usó un conjunto de datos en condiciones controladas que ha sido ampliamente utilizado en la literatura para la predicción del RUL de las baterías. En este escenario, los resultados obtenidos revelaron errores más bajos en comparación con los resultados reportados en la literatura existente. Estos hallazgos respaldan de manera contundente la eficacia y precisión de la metodología propuesta en la predicción del RUL. Los resultados mejorados obtenidos con la metodología propuesta proporcionan una sólida evidencia de los beneficios que ésta ofrece en términos de precisión de la predicción del RUL de las baterías para EV de dos ruedas.

Además, los resultados validan la contribución significativa que la metodología propuesta puede realizar tanto en la investigación académica como en la aplicación práctica. Al ofrecer una mejora sustancial en la precisión de la predicción del RUL, se abre un amplio abanico de oportunidades para su implementación en diversos sectores industriales que dependen del monitoreo constante y que mezcla datos provenientes de sensores con datos registrados para el control de las pruebas realizadas. La metodología propuesta se presenta como una herramienta que puede ayudar a optimizar la gestión y el mantenimiento de las baterías, lo que a su vez conlleva beneficios en términos de rendimiento, costos y seguridad.

La predicción del RUL de una batería en un EV de dos ruedas en condiciones de tráfico real es un escenario que ha sido poco explorado en la literatura científica. No obstante, este escenario requiere una gran importancia, ya que permite determinar las variables clave que deben ser monitoreadas y analizadas para comprender el comportamiento de degradación de las baterías a lo largo del tiempo en condiciones de tráfico real. Los resultados obtenidos mediante la implementación del modelo propuesto en este escenario demuestran la efectividad y relevancia de la metodología propuesta.

Además, se observa que el modelo propuesto logra una precisión notable en la predicción del RUL de las baterías en condiciones de tráfico real, lo que demuestra su robustez y capacidad para adaptarse a entornos dinámicos y variables. La aplicación exitosa del modelo en este escenario subraya su potencial para contribuir significativamente al desarrollo y la implementación de estrategias de gestión de baterías más eficientes y sostenibles en el ámbito de la movilidad eléctrica.

De igual forma, con la realización de las pruebas en condiciones de tráfico real, se obtiene la posibilidad de formar una base de datos que contribuya al análisis de EV de dos ruedas. Así mismo, el conocimiento adquirido a través de la predicción precisa del RUL en condiciones de tráfico real puede ser extrapolado a otros tipos de EV, lo que amplía aún más la relevancia y las posibles aplicaciones de la metodología propuesta.

En resumen, la exploración de la predicción del RUL en baterías de EV de dos ruedas en condiciones de tráfico real mediante el modelo propuesto representa un avance significativo en el campo de la gestión de baterías. Los resultados obtenidos no solo validan la efectividad del modelo en un escenario desafiante y poco explorado, sino que también abren nuevas perspectivas para mejorar la eficiencia y la confiabilidad de las baterías en EV en condiciones de tráfico real.

8. REFERENCIAS

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., … Zheng, X. (2015). *TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems*. https://research.google/pubs/pub45166/

Abdelaty, H., Al-Obaidi, A., Mohamed, M., & Farag, H. E. Z. (2021). Machine learning prediction models for battery-electric bus energy consumption in transit. *Transportation Research Part D: Transport and Environment*, *96*, 102868. https://doi.org/10.1016/J.TRD.2021.102868

Aggarwal, C. C. (2018). Neural Networks and Deep Learning. *Neural Networks and Deep Learning*. https://doi.org/10.1007/978-3-319-94463-0

Ajanovic, A., Haas, R., & Schrödl, M. (2021). On the Historical Development and Future Prospects of Various Types of Electric Mobility. *Energies 2021, Vol. 14, Page 1070*, *14*(4), 1070. https://doi.org/10.3390/EN14041070

Ardeshiri, R. R., Balagopal, B., Alsabbagh, A., Ma, C., & Chow, M. Y. (2020). Machine Learning Approaches in Battery Management Systems: State of the Art: Remaining useful life and fault detection. *Proceedings - 2020 2nd IEEE International Conference on Industrial Electronics for Sustainable Energy Systems, IESES 2020*, 61–66. https://doi.org/10.1109/IESES45645.2020.9210642

Bergmeir, C., & Benítez, J. M. (2012). Neural Networks in R Using the Stuttgart Neural Network Simulator: RSNNS. *Journal of Statistical Software*, *46*(7), 1–26. https://doi.org/10.18637/JSS.V046.I07

Berk, R. A. (2016). *Statistical Learning from a Regression Perspective*. Springer International Publishing. https://doi.org/10.1007/978-3-319-44048-4

Burzyński, D., & Kasprzyk, L. (2021). A novel method for the modeling of the state of health of lithium-ion cells using machine learning for practical applications. *Knowledge-Based Systems*, *219*, 106900. https://doi.org/10.1016/J.KNOSYS.2021.106900

Chen, J. C., Chen, T. L., Liu, W. J., Cheng, C. C., & Li, M. G. (2021). Combining empirical mode decomposition and deep recurrent neural networks for predictive maintenance of lithium-ion battery. *Advanced Engineering Informatics*, *50*, 101405. https://doi.org/10.1016/J.AEI.2021.101405

Chen, T., Kou, Q., He, T., & Acharya, A. (2020, September 11). *MXNet: A Flexible and Efficient Machine Learning Library for Heterogeneous Distributed Sys-tems*. Cran R. https://github.com/anirudhacharya

Chen, Z., Shi, N., Ji, Y., Niu, M., & Wang, Y. (2021a). Lithium-ion batteries remaining useful life prediction based on BLS-RVM. *Energy*, *234*, 121269. https://doi.org/10.1016/J.ENERGY.2021.121269

Chen, Z., Shi, N., Ji, Y., Niu, M., & Wang, Y. (2021b). Lithium-ion batteries remaining useful life prediction based on BLS-RVM. *Energy*, *234*, 121269. https://doi.org/10.1016/J.ENERGY.2021.121269

Chen, Z., Xue, Q., Wu, Y., Shen, S., Zhang, Y., & Shen, J. (2020). Capacity prediction and validation of lithium-ion batteries based on long short-term memory recurrent neural network. *IEEE Access*, *8*, 172783–172798. https://doi.org/10.1109/ACCESS.2020.3025766

Chien, Y. H., Hsieh, I. Y. L., & Chang, T. H. (2023). Beyond personal vehicles: How electrifying scooters will help achieve climate mitigation goals in Taiwan. *Energy Strategy Reviews*, *45*, 101056. https://doi.org/10.1016/J.ESR.2023.101056

Chollet, F. (2015). *Keras*. GitHub Repository. https://github.com/fchollet/keras

Chollet, F., & Allaire, J. J. (2016). Deep Learning in R. In *R-bloggers* (Vol. 1, Issue 7080). https://www.r-bloggers.com/deep-learning-in-r-2/

Dinov, I. D. (2018). Data science and predictive analytics: Biomedical and health applications using R. *Data Science and Predictive Analytics: Biomedical and Health Applications Using R*, 1–832. https://doi.org/10.1007/978-3-319-72347-1/COVER

Drees, M., Friedrich, C. M., Hinton, G., Salakhutdinov, R., & Rasmussen, C. E. (2016). *darch: Package for Deep Architectures and Restricted Boltzmann Machines*. Neural Computation. https://doi.org/10.1162/NECO.2006.18.7.1527

Eddelbuettel, D., & François, R. (2011). Rcpp: Seamless R and C++ Integration. *Journal of Statistical Software*, *40*(8), 1–18. https://doi.org/10.18637/JSS.V040.I08

Elgeldawi, E., Sayed, A., Galal, A. R., & Zaki, A. M. (2021). Hyperparameter Tuning for Machine Learning Algorithms Used for Arabic Sentiment Analysis. *Informatics 2021, Vol. 8, Page 79*, *8*(4), 79. https://doi.org/10.3390/INFORMATICS8040079

Elwert, T., Goldmann, D., Römer, F., Buchert, M., Merz, C., Schueler, D., & Sutter, J. (2015). Current Developments and Challenges in the Recycling of Key Components of (Hybrid) Electric Vehicles. *Recycling 2016, Vol. 1, Pages 25-60*, *1*(1), 25–60. https://doi.org/10.3390/RECYCLING1010025

Fan, J., Fan, J., Liu, F., Qu, J., & Li, R. (2019). *A Novel Machine Learning Method Based Approach for Li-Ion Battery Prognostic and Health Management*. IEEE Access. https://doi.org/10.1109/ACCESS.2019.2947843

Gao, R., Wang, L., Teti, R., Dornfeld, D., Kumara, S., Mori, M., & Helu, M. (2015). Cloud-enabled prognosis for manufacturing. *CIRP Annals*, *64*(2), 749–772. https://doi.org/10.1016/J.CIRP.2015.05.011

Ge, M. F., Liu, Y., Jiang, X., & Liu, J. (2021). A review on state of health estimations and remaining useful life prognostics of lithium-ion batteries. *Measurement*, *174*, 109057. https://doi.org/10.1016/J.MEASUREMENT.2021.109057

Gers, F. A., Schmidhuber, J., & Cummins, F. (1999). Learning to forget: Continual prediction with LSTM. *IEE Conference Publication*, *2*(470), 850–855. https://doi.org/10.1049/CP:19991218

Ghatak, A. (2019). Deep learning with R. *Deep Learning with R*, 1–245. https://doi.org/10.1007/978-981-13-5850-0/COVER

Goodfellow, I., Bengio, Y., & Aaron, C. (2016). *Deep Learning* (MIT Press). https://www.deeplearningbook.org/

Gou, B., Xu, Y., & Feng, X. (2021). An Ensemble Learning-Based Data-Driven Method for Online State-of-Health Estimation of Lithium-Ion Batteries. *IEEE Transactions on Transportation Electrification*, *7*(2), 422–436. https://doi.org/10.1109/TTE.2020.3029295

Grzegorz, K. (2015). *FCNN4R: Fast Compressed Neural Networks for R*. R Package. https://mran.microsoft.com/snapshot/2016-02-26/web/packages/FCNN4R/citation.html

Guedes Soares, C. (2002). Reliability engineering and risk analysis: a practical guide. *Reliability Engineering & System Safety*, *77*(2), 207–208. https://doi.org/10.1016/S0951-8320(02)00008-X

Günther, F., & Fritsch, S. (2010). neuralnet: Training of Neural Networks. *The R Journal*, *2*.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning*. https://doi.org/10.1007/978-0-387-84858-7

Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. R. (2012). *Improving neural networks by preventing co-adaptation of feature detectors*. http://arxiv.org/abs/1207.0580

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, *9*(8), 1735–1780. https://doi.org/10.1162/NECO.1997.9.8.1735

Hong, J., Wang, Z., & Yao, Y. (2019). Fault prognosis of battery system based on accurate voltage abnormity prognosis using long short-term memory neural networks. *Applied Energy*, *251*, 113381. https://doi.org/10.1016/J.APENERGY.2019.113381

Hosen, M. S., Youssef, R., Kalogiannis, T., Van Mierlo, J., & Berecibar, M. (2021). Battery cycle life study through relaxation and forecasting the lifetime via machine learning. *Journal of Energy Storage*, *40*, 102726. https://doi.org/10.1016/J.EST.2021.102726

Hothorn, T. (2022). *CRAN Task View: Machine Learning & Statistical Learning*. https://CRAN.R-project.org/view=MachineLearning

Hu, J., Morais, H., Sousa, T., & Lind, M. (2016). Hu, J., Morais, H., Sousa, T., & Lind, M. (2016). Electric vehicle fleet management in smart grids: A review of services, optimization and control aspects. Renewable and Sustainable Energy Reviews, 56, 1207–1226. https://doi.org/10.1016/J.RSER.2015.12.014. *Renewable and Sustainable Energy Reviews*, *56*, 1207–1226. https://doi.org/10.1016/J.RSER.2015.12.014

Hu, X., Deng, Z., Lin, X., Xie, Y., & Teodorescu, R. (2021). Research directions for next-generation battery management solutions in automotive applications. *Renewable and Sustainable Energy Reviews*, *152*, 111695. https://doi.org/10.1016/J.RSER.2021.111695

Hu, X., Feng, F., Liu, K., Zhang, L., Xie, J., & Liu, B. (2019). State estimation for advanced battery management: Key challenges and future trends. *Renewable and Sustainable Energy Reviews*, *114*, 109334. https://doi.org/10.1016/J.RSER.2019.109334

Hu, X., Xu, L., Lin, X., & Pecht, M. (2020). Battery Lifetime Prognostics. *Joule*, *4*(2), 310–346. https://doi.org/10.1016/J.JOULE.2019.11.018

Hu, X., Yang, X., Feng, F., Liu, K., & Lin, X. (2021). A Particle Filter and Long Short-Term Memory Fusion Technique for Lithium-Ion Battery Remaining Useful Life Prediction. *Journal of Dynamic Systems, Measurement and Control, Transactions of the ASME*, *143*(6). https://doi.org/10.1115/1.4049234

Humphrey, G. B., Maier, H. R., Wu, W., Mount, N. J., Dandy, G. C., Abrahart, R. J., & Dawson, C. W. (2017). Improved validation framework and R-package for artificial neural network models. *Environmental Modelling & Software*, *92*, 82–106. https://doi.org/10.1016/J.ENVSOFT.2017.01.023

Huotari, M., Arora, S., Malhi, A., & Främling, K. (2021). Comparing seven methods for state-of-health time series prediction for the lithium-ion battery packs of forklifts. *Applied Soft Computing*, *111*, 107670. https://doi.org/10.1016/J.ASOC.2021.107670

Ji, Y., Chen, Z., Shen, Y., Yang, K., Wang, Y., & Cui, J. (2021). An RUL prediction approach for lithium-ion battery based on SADE-MESN. *Applied Soft Computing*, *104*, 107195. https://doi.org/10.1016/J.ASOC.2021.107195

Jiang, Y., Zeng, W., Chen, L., & Xin, Y. (2019). Lithium-Ion Battery RUL Indirect Prediction Based on GAAA-ELM. *Proceedings - 2018 International Conference on Sensing, Diagnostics, Prognostics, and Control, SDPC 2018*, 415–420. https://doi.org/10.1109/SDPC.2018.8664829

Jo, S., Jung, S., & Roh, T. (2021). Battery State-of-Health Estimation Using Machine Learning and Preprocessing with Relative State-of-Charge. *Energies 2021, Vol. 14, Page 7206*, *14*(21), 7206. https://doi.org/10.3390/EN14217206

Jorge, I., Samet, A., Mesbahi, T., & Bone, R. (2020). New ANN results on a major benchmark for the prediction of RUL of Lithium Ion batteries in electric vehicles. *Proceedings - 19th IEEE International Conference on Machine Learning and Applications, ICMLA 2020*, 1246–1253. https://doi.org/10.1109/ICMLA51294.2020.00196

Kara, A. (2021). A data-driven approach based on deep neural networks for lithium-ion battery prognostics. *Neural Computing and Applications 2021*, 1–14. https://doi.org/10.1007/S00521-021-05976-X

Khumprom, P., & Yodo, N. (2019). A Data-Driven Predictive Prognostic Model for Lithium-ion Batteries based on a Deep Learning Algorithm. *Energies 2019, Vol. 12, Page 660*, *12*(4), 660. https://doi.org/10.3390/EN12040660

Kuhn, M. (2008). Building Predictive Models in R Using the caret Package. *Journal of Statistical Software*, *28*(5), 1–26. https://doi.org/10.18637/JSS.V028.I05

Kuhn, M., & Johnson, K. (2013). Applied predictive modeling. *Applied Predictive Modeling*, 1–600. https://doi.org/10.1007/978-1-4614-6849-3

Kuhn, M., & Johnson, K. (2019). *Feature Engineering and Selection: A Practical Approach for Predictive Models*. http://www.feat.engineering/important-concepts.html

Kutkina, O., & Feuerriegel, S. (2016). Deep Learning in R. *R-Bloggers*, *1*(7080). https://www.manning.com/books/deep-learning-with-r

Landry, M., Bartz, A., Aiello, S., Eckstrand, E., Fu, A., & Aboyoun, P. (2018). *Machine Learning with R and H2O*. http://h2o.ai/resources/

Lee, C.-J., Kim, B.-K., Kwon, M.-K., Nam, K., & Kang, S.-W. (2021). Real-Time Prediction of Capacity Fade and Remaining Useful Life of Lithium-Ion Batteries Based on Charge/Discharge Characteristics. *Electronics 2021, Vol. 10, Page 846*, *10*(7), 846. https://doi.org/10.3390/ELECTRONICS10070846

Li, W., Jiao, Z., Du, L., Fan, W., & Zhu, Y. (2019). An indirect RUL prognosis for lithium-ion battery under vibration stress using Elman neural network. *International Journal of Hydrogen Energy*, *44*(23), 12270–12276. https://doi.org/10.1016/J.IJHYDENE.2019.03.101

Li, X., Liu, J., Bai, M., Li, J., Li, X., Yan, P., & Yu, D. (2021). An LSTM based method for stage performance degradation early warning with consideration of time-series information. *Energy*, *226*, 120398. https://doi.org/10.1016/J.ENERGY.2021.120398

Li, X., Zhang, L., Wang, Z., & Dong, P. (2019). Remaining useful life prediction for lithium-ion batteries based on a hybrid model combining the long short-term memory and Elman neural networks. *Journal of Energy Storage*, *21*, 510–518. https://doi.org/10.1016/J.EST.2018.12.011

Listou Ellefsen, A., Bjørlykhaug, E., Æsøy, V., Ushakov, S., & Zhang, H. (2019). Remaining useful life predictions for turbofan engine degradation using semi-supervised deep architecture. *Reliability Engineering & System Safety*, *183*, 240–251. https://doi.org/10.1016/J.RESS.2018.11.027

Liu, H. C., Cheng, C. H., Chun-Yen, W., Kung-Ming, J., Chen, Y. G., Chung, C., Chih-Chien, S., & Yong-Kai, L. (2013). Micro controller unit based motor control system using field oriented control algorithm. *ICICS 2013 - Conference Guide of the 9th International Conference on Information, Communications and Signal Processing*. https://doi.org/10.1109/ICICS.2013.6782806

Liu, K., Shang, Y., Ouyang, Q., & Widanage, W. D. (2021). A Data-Driven Approach with Uncertainty Quantification for Predicting Future Capacities and Remaining Useful Life of Lithium-ion Battery. *IEEE Transactions on Industrial Electronics*, *68*(4), 3170–3180. https://doi.org/10.1109/TIE.2020.2973876

Liu, Y., Zhao, G., & Peng, X. (2019). Deep learning prognostics for lithium-ion battery based on ensembled long short-term memory networks. *IEEE Access*, *7*, 155130–155142. https://doi.org/10.1109/ACCESS.2019.2937798

Machedon-Pisu, M., & Borza, P. N. (2019). Are Personal Electric Vehicles Sustainable? A Hybrid E-Bike Case Study. *Sustainability 2020, Vol. 12, Page 32*, *12*(1), 32. https://doi.org/10.3390/SU12010032

Mao, W., He, J., Tang, J., & Li, Y. (2018). Predicting remaining useful life of rolling bearings based on deep feature representation and long short-term memory neural network. *Advances in Mechanical Engineering*, *10*(12). https://doi.org/10.1177/1687814018817184/ASSET/IMAGES/LARGE/10.1177_1687814018817184-FIG2.JPEG

Marcos, J. T., Godina, R., Spengler, T. S., Scheller, C., & Carvalho, H. (2021). Sources of uncertainty in the Closed-loop supply chain of lithium-ion batteries for electric vehicles. *Cleaner Logistics and Supply Chain*, 100006. https://doi.org/10.1016/J.CLSCN.2021.100006

Mayer, B. L. (2019, December 20). *deep: A Neural Networks Framework*. Cran R. https://cran.r-project.org/web/packages/deep/index.html

Mejia, J., Avelar-Sosa, L., Mederos, B., Ramírez, E. S., & Díaz Roman, J. D. (2021). Prediction of time series using an analysis filter bank of LSTM units. *Computers & Industrial Engineering*, *157*, 107371. https://doi.org/10.1016/J.CIE.2021.107371

Nanaki, E. A. (2021). Electric vehicle capitals – case studies. *Electric Vehicles for Smart Cities*, 181–247. https://doi.org/10.1016/B978-0-12-815801-2.00003-4

Nasiriany, S., Thomas, G., Wang, W., & Yang, A. (2019). *A Comprehensive Guide to Machine Learning*. http://www.eecs189.org/

Ozkok, F. O., & Celik, M. (2022). A hybrid CNN-LSTM model for high resolution melting curve classification. *Biomedical Signal Processing and Control*, *71*, 103168. https://doi.org/10.1016/J.BSPC.2021.103168

Pan, D., Li, H., & Song, Y. (2020). A Comparative Study of Particle Filters and its Variants in Lithium-ion Battery SOH Estimation. *International Conference on Sensing, Measurement and Data Analytics in the Era of Artificial Intelligence, ICSMD 2020 - Proceedings*, 198–203. https://doi.org/10.1109/ICSMD50554.2020.9261654

Park, K., Choi, Y., Choi, W. J., Ryu, H. Y., & Kim, H. (2020). LSTM-Based Battery Remaining Useful Life Prediction with Multi-Channel Charging Profiles. *IEEE Access*, *8*, 20786–20798. https://doi.org/10.1109/ACCESS.2020.2968939

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury Google, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Xamla, A. K., Yang, E., Devito, Z., Raison Nabla, M., Tejani, A., Chilamkurthy, S., Ai, Q., Steiner, B., … Chintala, S. (2019). PyTorch: An Imperative Style, High-Performance Deep Learning Library. *In Advances in Neural Information Processing Systems*, *32*, 8024–8035. https://proceedings.neurips.cc/paper_files/paper/2019/file/bdbca288fee7f92f2bfa9f7012727740-Paper.pdf

Pawlus, Michael., & Devine, Rodger. (2020). *Hands-On Deep Learning with R*. Packt Publishing.

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., & Duchesnay, É. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830. http://scikit-learn.sourceforge.net.

Qiao, J., Liu, X., & Chen, Z. (2020). Prediction of the Remaining Useful Life of Lithium-Ion Batteries Based on Empirical Mode Decomposition and Deep Neural Networks. *IEEE Access*, *8*, 42760–42767. https://doi.org/10.1109/ACCESS.2020.2977429

Qu, J., Liu, F., Ma, Y., & Fan, J. (2019). A Neural-Network-Based Method for RUL Prediction and SOH Monitoring of Lithium-Ion Battery. *IEEE Access*, *7*, 87178–87191. https://doi.org/10.1109/ACCESS.2019.2925468

Quast, B. (2016). *rnn: a Recurrent Neural Network in R*. The Graduate Institute. https://qua.st/rnn-package

Ren, L., Dong, J., Wang, X., Meng, Z., Zhao, L., & Deen, M. J. (2021). A Data-Driven Auto-CNN-LSTM Prediction Model for Lithium-Ion Battery Remaining Useful Life. *IEEE Transactions on Industrial Informatics*, *17*(5), 3478–3487. https://doi.org/10.1109/TII.2020.3008223

Ren, L., Zhao, L., Hong, S., Zhao, S., Wang, H., & Zhang, L. (2018). Remaining Useful Life Prediction for Lithium-Ion Battery: A Deep Learning Approach. *IEEE Access*, *6*, 50587–50598. https://doi.org/10.1109/ACCESS.2018.2858856

Rincon, C. (2023). *cnnlstm*. Github. https://github.com/cjrincon/cnnlstm

Rong, X. (2022). *deepnet: Deep Learning Toolkit in R*. Cran R. https://cran.r-project.org/web/packages/deepnet/index.html

Saha, B., & Goebel, K. (2007). *Battery Data Set*. NASA Ames Research Center. Moffett Field, CA.

Samanta, A., Chowdhuri, S., & Williamson, S. S. (2021). Machine Learning-Based Data-Driven Fault Detection/Diagnosis of Lithium-Ion Battery: A Critical Review. *Electronics 2021, Vol. 10, Page 1309*, *10*(11), 1309. https://doi.org/10.3390/ELECTRONICS10111309

Shenghao, W. (2021). Improved Algorithm Combining Wavelet Transform and Kalman Filter for Estimating SOC of Lithium-ion Battery in Vehicle. *2021 IEEE 2nd International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering, ICBAIE 2021*, 175–179. https://doi.org/10.1109/ICBAIE52039.2021.9390009

Si, X. S., Wang, W., Hu, C. H., & Zhou, D. H. (2011). Remaining useful life estimation – A review on the statistical data driven approaches. *European Journal of Operational Research*, *213*(1), 1–14. https://doi.org/10.1016/J.EJOR.2010.11.018

Skansi, S. (2018). *Introduction to Deep Learning*. https://doi.org/10.1007/978-3-319-73004-2

Spreafico, C., Russo, D., & Rizzi, C. (2017). A state-of-the-art review of FMEA/FMECA including patents. *Computer Science Review*, *25*, 19–28. https://doi.org/10.1016/J.COSREV.2017.05.002

Sultan, L. (2011). Risk analysis method: FMEA/FMECA in the organizations. *International Journal of Basic & Applied Sciences IJBAS-IJENS*, *11*.

Sun, T., Xia, B., Liu, Y., Lai, Y., Zheng, W., Wang, H., Wang, W., & Wang, M. (2019). A Novel Hybrid Prognostic Approach for Remaining Useful Life Estimation of Lithium-Ion Batteries. *Energies 2019, Vol. 12, Page 3678*, *12*(19), 3678. https://doi.org/10.3390/EN12193678

Tong, Z., Miao, J., Tong, S., & Lu, Y. (2021). Early prediction of remaining useful life for Lithium-ion batteries based on a hybrid machine learning method. *Journal of Cleaner Production*, *317*, 128265. https://doi.org/10.1016/J.JCLEPRO.2021.128265

Un-Noor, F., Padmanaban, S., Mihet-Popa, L., Mollah, M. N., & Hossain, E. (2017). A comprehensive study of key electric vehicle (EV) components, technologies, challenges, impacts, and future direction of development. *Energies*, *10*(8). https://doi.org/10.3390/EN10081217

UPME. (2019). *Estrategia Nacional de Movilidad Eléctrica*. UPME. https://www1.upme.gov.co/DemandaEnergetica/ENME.pdf

Ushey, K., Allaire, J., & Tang, Y. (2023). *Reticulate: Interface to "Python."* https://rstudio.github.io/Reticulate/

van Houdt, G., Mosquera, C., & Nápoles, G. (2020). A review on the long short-term memory model. *Artificial Intelligence Review*, *53*(8), 5929–5955. https://doi.org/10.1007/S10462-020-09838-1/TABLES/1

Venables, W. N., & Ripley, B. D. (2002). *Modern Applied Statistics with S* (Springer, Ed.; Fourth Edition). http://www.stats.ox.ac.uk/pub/MASS4/

Wang, J., Wen, G., Yang, S., & Liu, Y. (2019). Remaining Useful Life Estimation in Prognostics Using Deep Bidirectional LSTM Neural Network. *Proceedings - 2018 Prognostics and System Health Management Conference, PHM-Chongqing 2018*, 1037–1042. https://doi.org/10.1109/PHM-CHONGQING.2018.00184

Wang, Z., Feng, G., Zhen, D., Gu, F., & Ball, A. (2021). A review on online state of charge and state of health estimation for lithium-ion batteries in electric vehicles. *Energy Reports*, *7*, 5141–5161. https://doi.org/10.1016/J.EGYR.2021.08.113

Wei, M., Ye, M., Bo Li, J., Wang, Q., & Xu, X. (2020). State of charge estimation of lithium-ion batteries using LSTM and NARX neural networks. *IEEE Access*, *8*, 189236–189245. https://doi.org/10.1109/ACCESS.2020.3031340

Wu, L., Fu, X., & Guan, Y. (2016). Review of the Remaining Useful Life Prognostics of Vehicle Lithium-Ion Batteries Using Data-Driven Methodologies. *Applied Sciences 2016, Vol. 6, Page 166*, *6*(6), 166. https://doi.org/10.3390/APP6060166

Xia, T., Song, Y., Zheng, Y., Pan, E., & Xi, L. (2020). An ensemble framework based on convolutional bi-directional LSTM with multiple time windows for remaining useful life estimation. *Computers in Industry*, *115*, 103182. https://doi.org/10.1016/J.COMPIND.2019.103182

Xing, Y., Ma, E. W. M., Tsui, K. L., & Pecht, M. (2011). Battery Management Systems in Electric and Hybrid Vehicles. *Energies 2011, Vol. 4, Pages 1840-1857*, *4*(11), 1840–1857. https://doi.org/10.3390/EN4111840

Xiong, R. (2019). Battery management algorithm for electric vehicles. *Battery Management Algorithm for Electric Vehicles*, 1–297. https://doi.org/10.1007/978-981-15-0248-4

Xiong, R. (2020). Battery management algorithm for electric vehicles. *Battery Management Algorithm for Electric Vehicles*, 1–297. https://doi.org/10.1007/978-981-15-0248-4

Xiong, R., Li, L., & Tian, J. (2018). Towards a smarter battery management system: A critical review on battery state of health monitoring methods. *Journal of Power Sources*, *405*, 18–29. https://doi.org/10.1016/J.JPOWSOUR.2018.10.019

Xu, M., Wu, W., Zhou, W., Ma, Y., Shi, X., & Li, J. (2020). State of charge estimation of low-speed electric vehicle battery using back propagation neural network. *Proceedings - 2020 IEEE International Symposium on Parallel and Distributed Processing with Applications, 2020 IEEE International Conference on Big Data and Cloud Computing, 2020 IEEE International Symposium on Social Computing and Networking and 2020 IEE*, 1438–1443. https://doi.org/10.1109/ISPA-BDCloud-SocialCom-SustainCom51426.2020.00215

Xue, Z., Zhang, Y., Cheng, C., & Ma, G. (2020). Remaining useful life prediction of lithium-ion batteries with adaptive unscented kalman filter and optimized support vector regression. *Neurocomputing*, *376*, 95–102. https://doi.org/10.1016/J.NEUCOM.2019.09.074

Yang, L., & Shami, A. (2020). On hyperparameter optimization of machine learning algorithms: Theory and practice. *Neurocomputing*, *415*, 295–316. https://doi.org/10.1016/J.NEUCOM.2020.07.061

Yang, R., Xiong, R., Shen, W., & Lin, X. (2021). Extreme Learning Machine-Based Thermal Model for Lithium-Ion Batteries of Electric Vehicles under External Short Circuit. *Engineering*, *7*(3), 395–405. https://doi.org/10.1016/J.ENG.2020.08.015

Yang, Y. (2021). A machine-learning prediction method of lithium-ion battery life based on charge process for different applications. *Applied Energy*, *292*, 116897. https://doi.org/10.1016/J.APENERGY.2021.116897

Zhang, M., Wu, L., & Peng, Z. (2021). The early prediction of lithium-ion battery remaining useful life using a novel Long Short-Term Memory network. *Proceedings of the 16th IEEE Conference on Industrial Electronics and Applications, ICIEA 2021*, 1364–1371. https://doi.org/10.1109/ICIEA51954.2021.9516254

Zhang, X., Li, Z., Luo, L., Fan, Y., & Du, Z. (2022). A review on thermal management of lithium-ion batteries for electric vehicles. *Energy*, *238*, 121652. https://doi.org/10.1016/J.ENERGY.2021.121652

Zhang, Y., Xiong, R., He, H., & Pecht, M. G. (2018). Long short-term memory recurrent neural network for remaining useful life prediction of lithium-ion batteries. *IEEE Transactions on Vehicular Technology*, *67*(7), 5695–5705. https://doi.org/10.1109/TVT.2018.2805189

Zhang, Z., Si, X., Hu, C., & Lei, Y. (2018). Degradation data analysis and remaining useful life estimation: A review on Wiener-process-based methods. *European Journal of Operational Research*, *271*(3), 775–796. https://doi.org/10.1016/J.EJOR.2018.02.033

Zhao, Y., Wang, Z., Shen, Z. J. M., & Sun, F. (2021). Data-driven framework for large-scale prediction of charging energy in electric vehicles. *Applied Energy*, *282*, 116175. https://doi.org/10.1016/J.APENERGY.2020.116175

Zhou, Y., Huang, M., Chen, Y., & Tao, Y. (2016). A novel health indicator for on-line lithium-ion batteries remaining useful life prediction. *Journal of Power Sources*, *321*, 1–10. https://doi.org/10.1016/J.JPOWSOUR.2016.04.119

Zhu, J., Tan, T., Wu, L., & Yuan, H. (2019). RUL prediction of lithium-ion battery based on improved DGWO-ELM method in a random discharge rates environment. *IEEE Access*, *7*, 125176–125187. https://doi.org/10.1109/ACCESS.2019.2936822

Zraibi, B., Okar, C., Chaoui, H., & Mansouri, M. (2021). Remaining Useful Life Assessment for Lithium-Ion Batteries Using CNN-LSTM-DNN Hybrid Method. *IEEE Transactions on Vehicular Technology*, *70*(5), 4252–4261. https://doi.org/10.1109/TVT.2021.3071622

9. ANEXOS

Anexo 1. Definiciones y ecuaciones de las métricas de desempeño de modelos predictivos de regresión

En la Tabla 35 se presentan las métricas de desempeño que sirven para realizar la evaluación de un modelo predictivo de regresión, así mismo, se presenta la definición de cada una de estas métricas y se muestran las ecuaciones para calcular cada una de ellas. La notación para el entendimiento de cada una de las ecuaciones es la siguiente: n representa el número de observaciones, i denota el número de la observación, y_i representa el valor real, $\hat{y}_i$ representa el valor predicho (Hosen et al., 2021).

Tabla 35

Métricas de desempeño, definición y ecuación

Métrica	Definición	Ecuación
RMSE	Raíz del error cuadrático medio. Es la raíz del MSE. Un RMSE con valor de cero indica que el modelo predictivo tiene un buen ajuste	$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$
MSE	Error cuadrático medio conocido como pérdida L2. Es la distancia al cuadrado entre los valores reales y predichos. El MSE no puede ser negativo y aumenta exponencialmente cuando hay aumento en el error	$\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$
MAE	Media de los errores absolutos: Es la diferencia absoluta entre el valor real y el valor predicho en cada iteración. Todos los errores se ponderan en la misma escala.	$\frac{1}{n}\sum_{i=1}^{n}\lvert y_i - \hat{y}_i\rvert$
MAPE	Error porcentual absoluto medio. Es el porcentaje que mide el tamaño del error cometido en la predicción.	$\frac{1}{n}\sum_{i=1}^{n}\left\lvert\frac{\hat{y}_i - y_i}{y_i}\right\rvert \; x\,100\%$

Anexo 2. Modelos predictivos computacionales de Python y R

Librerías de Python: En Python existen diferentes librerías que facilitan el proceso de aprendizaje automático y desarrollo de modelos predictivos mediante redes neuronales. Dentro de las librerías para el desarrollo de redes neuronales, se destaca el uso de librerías como `Tensorflow y Keras` que se implementan en Python, sin embargo, también pueden ser usadas en R a través de una interfaz de programación de aplicaciones (API, por sus siglas en inglés: Application Programming Interfaces). En la Tabla 36 se describen las librerías de Python enfocadas en aprendizaje automático con sus respectivos autores.

Tabla 36

Librerías de Python para la implementación de redes neuronales

Paquete de Python	Descripción	Autor
`Tensorflow`	Es un sistema de computación numérica de código abierto o framework de aprendizaje automático que fue desarrollado por Google. Es una librería matemática de bajo nivel que se usa para construir arquitecturas de redes neuronales profundas con capacidades para acelerar el desarrollo de la red (Ghatak, 2019). Esta librería de código libre facilita la creación de modelos de aprendizaje automático, ayudando con la construcción, entrenamiento e implementación de las redes neuronales que realizan la detección de patrones de comportamiento.	(Abadi et al., 2015)
`Keras`	Es una API secuencial de redes neuronales de alto nivel desarrollada para facilitar la construcción y entrenamiento de modelos de aprendizaje profundo. Se usa para la creación rápida de prototipos, la investigación avanzada y la producción (Ghatak, 2019).	(Chollet, 2015)
`PyTorch`	Es una librería desarrollada por Facebook. Es una librería de aprendizaje automático usado para aplicaciones de visión artificial y procesamiento de lenguaje natural.	(Paszke et al., 2019)
`Scikit-learn`	Es una librería gratuita para Python y se define como una herramientas simple y eficiente para el análisis predictivo de datos, es accesible para todos y reutilizable en varios contextos y está en basada en NumPy, SciPy y matplotlib, las cuales son librerías de Python. NumPy sirve para dar soporte en la creación vectores y matrices grandes multidimensionales, junto con una colección de funciones matemáticas de alto nivel para operar con ellas. SciPy proporciona algoritmos para optimización, integración, interpolación, problemas de valores propios, ecuaciones algebraicas, ecuaciones diferenciales, estadísticas y muchas otras clases. Matplotlib es una biblioteca completa para crear visualizaciones estáticas, animadas e interactivas en Python.	(Pedregosa et al., 2011)

Librerías de R: Para el uso de las librerías en R, se dispone de un repositorio llamado CRAN, el cual es una red de servidores que almacenan versiones idénticas y actualizadas de código y documentación para R. Así mismo, se puede acceder a las librerías creadas a partir del repositorio de github, para lo cual, es necesario copiar las líneas de código que permiten la obtención del paquete. En la Tabla 37 se observan las librerías desarrolladas para la implementación de redes neuronales en R (Hothorn, 2022).

Tabla 37

Librerías desarrolladas en R para la implementación de redes neuronales

Paquete de R	Descripción	Autor
`nnet`	Se implementa en las redes neuronales feed-forward con una sola capa oculta y para modelos log-lineales multinomiales.	(Venables & Ripley, 2002)
`neuralnet`	Se usa para el entrenamiento de redes neuronales, utilizando backpropagation, retropropagación resiliente o sin seguimiento de peso o la versión convergente global modificada. El paquete permite configuraciones flexibles a través de la opción personalizada de error y función de activación. Además, se implementa el cálculo de pesos generalizados.	(Günther & Fritsch, 2010)
`h2o`	Implementa una plataforma de aprendizaje automático con funcionalidades de secuencias de comandos R para h20.	(Landry et al., 2018)
`RSNNS`	Ofrece una interfaz para el simulador de red neuronal de Stuttgart.	(Bergmeir & Benítez, 2012)
`deepnet`	Brinda un kit de herramientas de aprendizaje profundo.	(Rong, 2022)
`darch`	Es un paquete para arquitecturas profundas y máquinas Boltzmann restringidas.	(Drees et al., 2016)
`rnn`	Se usa para la implementación de redes neuronales recurrentes.	(Quast, 2016)
`FCNN4R`	Es una interfaz a la biblioteca FCNN que permite redes neuronales ampliables por el usuario.	(Grzegorz, 2015)
`rcppDL`	Se usa para la implementación de métodos básicos de aprendizaje automático con muchas capas de aprendizaje profundo que incluyen codificadores automáticos de ruido, máquinas de Boltzmann restringida y redes profundas.	(Eddelbuettel & François, 2011)
`deepr`	Es un paquete que sirve para agilizar los procesos de entrenamiento, ajuste y predicción para el aprendizaje profundo basado en darch y deepnet.	(Mayer, 2019)
`MXNet`	Este paquete brinda computación de GPU flexible y eficiente y aprendizaje profundo de última generación a R.	(T. Chen et al., 2020)
`Reticulate`	Ofrece una interfaz de Python a R, proporcionando un conjunto de herramientas para la interoperabilidad entre Python y R.	(Ushey et al., 2023)
`caret`	Este paquete de entrenamiento de clasificación y regresión, es un conjunto de funciones que intentan agilizar el proceso de creación de modelos predictivos.	(Kuhn, 2008)

A través de los paquetes como `Reticulate, Tensorflow y Keras` se pueden desarrollar modelos predictivos con redes neuronales. El paquete `Reticulate` funciona como una API que permite la conexión entre Python y R para el uso de las librerías `Tensorflow y keras`, con las cuales, se pueden generar modelos secuenciales con diferentes tipos de capas dependiendo del tipo de modelo a generar. Por ejemplo, para la creación de un modelo predictivo con redes convolucionales permite la creación del modelo secuencial, las capas convolucionales, las capas de *maxpooling*, las capas de aplanamiento conocidas como *flatten*, así como también, la creación de capas densas. Así mismo, para la creación de un modelo predictivo con redes LSTM, las librerías permiten la creación de capas LSTM y capas de abandono conocidas como *dropout*.

Estas librerías también permiten hacer la compilación, entrenamiento y predicción de los modelos predictivos generados a partir de redes neuronales. Para esto, es importante definir la arquitectura y los valores de cada uno de los hiperparámetros necesarios para la compilación y el entrenamiento, los cuales son ajustables y permiten controlar el proceso de entrenamiento del modelo predictivo. Sin embargo, en estos paquetes no se encuentra desarrollado una librería en R que permita la estructuración y división de los datos de entrada para la CNN y la red LSTM.

Anexo 3. Versión de los paquetes implementados en el desarrollo de la metodología propuesta para la predicción con datos controlados y datos en condiciones de tráfico real.

Tabla 38

Versión de paquetes implementados en R

Paquete	Descripción	Versión
readr	Read Rectangular Text Data	2.1.4
dplyr	A Grammar of Data Manipulation	1.1.4
magrittr	A Forward-Pipe Operator for R	2.0.3
corrplot	Visualization of a Correlation Matrix	0.92
tensorflow	R Interface to 'TensorFlow'	2.14.0
keras	R Interface to 'Keras'	2.13.0
qcc	Quality Control Charts	2.7
randomForest	Breiman and Cutler's Random Forests for Classification and Regression	4.7-1.1

Anexo 4. Análisis de importancia de las variables para cada escenario implementando Random Forest

Se realiza un análisis de importancia de las variables a través de un Random Forest con 500 árboles para cada uno de los escenarios y se obtiene que las variables tienen un nivel de importancia significativo para incluir dentro del modelo predictivo. Los resultados de la importancia de las variables se muestran en la Tabla 39 y Tabla 40.

Tabla 39

Importancia de las variables con Radom Forest para el escenario en condiciones controladas

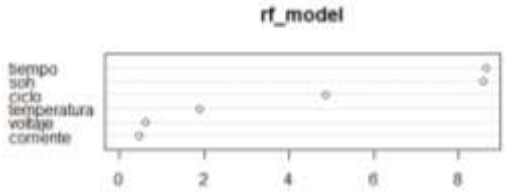
Variable	Importancia (RandomForest)	Gráfico
Tiempo [segundos]	8.85	
SOH [%]	8.37	
Ciclo [conteo]	5.19	
Temperatura [°C]	1.67	
Voltaje [Voltios]	0.52	
Corriente [Amperios]	0.45	

Tabla 40

Importancia de las variables con Radom Forest para el escenario en condiciones de tráfico real

Variable	Importancia (RandomForest)	Gráfico
Ciclo [Conteo]	4,113,907.26	
Voltaje [Voltios]	917,879.97	
Aceleración eje z [m/s^2]	741,356.07	
Tiempo [segundos]	392,785.89	
Temperatura ambiental [°C]	392,600.10	
Aceleración eje y [m/s^2]	339,860.09	
Humedad ambiental [g/m^3]	296,900.03	
Temperatura en un extremo [°C]	248,086.38	
Temperatura en el centro [°C]	200,343.33	
Corriente [Amperios]	196,323.70	
Aceleración eje x [m/s^2]	181,979.32	
Potencia [Watt]	88,022.96	

Anexo 5. Descripción de la bicicleta eléctrica, FMECA, instrumentación y captura de datos experimentales bajo condiciones de tráfico real

En este anexo se describen los pasos que se llevaron a cabo para obtener los datos del EV de dos ruedas implementado en las pruebas bajo condiciones de tráfico real. De esta forma, se describe el EV de dos ruedas usado en la investigación, sus características, componentes y el análisis de modos, efectos y su criticidad – FMECA, realizado para determinar la criticidad de los componentes que deben ser monitoreados para calcular las variables de desempeño asociadas. Así mismo, se presenta la planeación del experimento observacional, la instrumentación de los vehículos para la captura de cada una de las variables y el proceso de toma de datos.

La bicicleta eléctrica se caracteriza por ser una bicicleta Stärker Urban ST, la cual, es una bicicleta que tiene acoplado un EM con potencia nominal de 350 watts (W), el cual brinda asistencia al pedalear para poder avanzar. Tiene una velocidad máxima de 25 km/h y una LIB de 36 voltios con capacidad de 10 amperios hora (Ah) que suministra energía y brinda un recorrido autónomo de 40 km.

En la Figura 46, se muestran los sistemas y funciones del equipo, los cuales, se pueden descomponer en 5 sistemas principales tales como: eléctrico, auxiliar, de frenos, estructural y de tracción. Cada uno de estos sistemas tiene asociados diferentes componentes que le permiten al vehículo funcionar adecuadamente. En el sistema eléctrico se encuentra el controlador, batería, BMS, cargador, tablero de instrumentos y el sensor de asistencia. El sistema auxiliar tiene en cuenta partes del sistema de transmisión como cadena, piñones, cambios y pedales. La bicicleta eléctrica también cuanta con un sistema de frenos. El sistema estructural tiene en cuenta el chasis y el manubrio, y, por último, el EM hace parte del sistema de tracción.

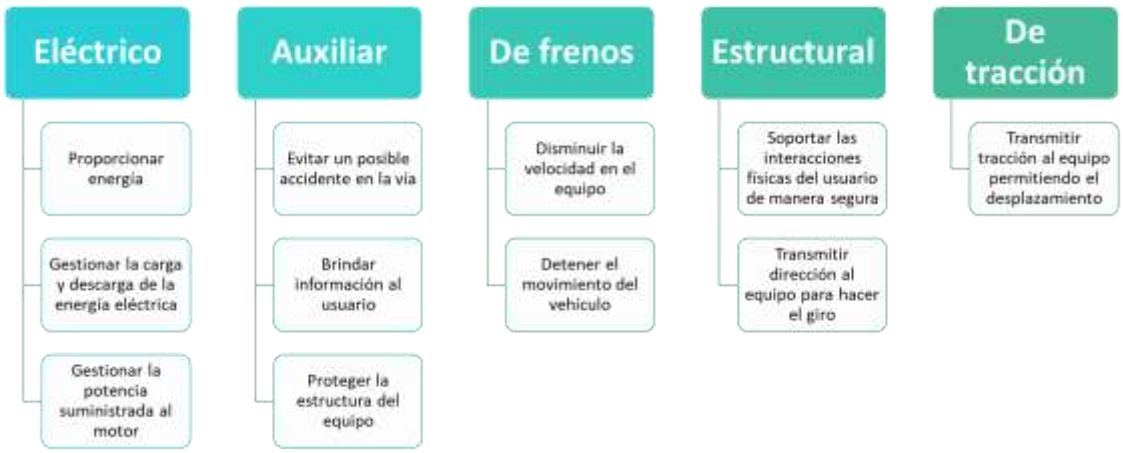


Figura 46. Sistemas y funciones de la bicicleta y patineta eléctrica

Una vez definidos los sistemas de la bicicleta, se definen las fases metodológicas para realizar el FMECA. En este se determinan los modos de falla de cada uno de los componentes de la bicicleta eléctrica a través

de un análisis funcional, la caracterización de los modos de falla de cada uno de los componentes, análisis del efecto de falla para obtener la estimación de la severidad, análisis de la causa de la falla para hacer la estimación de ocurrencia, análisis del método de detección de la falla que permite obtener la estimación de la detección. Así mismo, se realiza un análisis de criticidad para el modo de falla, en el cual se realizan estimaciones para los valores de β, α, λ y T.

El análisis de modos de fallas, efectos y su criticidad conocida como FMECA surgió a partir del análisis de modos de falla y efectos (FMEA, por sus siglas en inglés: *Failure Modes and Effect Analysis*), el cual fue implementado en 1949 por el ejército de los EEUU para estudiar los problemas que podrían surgir por el mal funcionamiento de los sistemas militares (Spreafico et al., 2017). EL FMEA describe las causas inherentes de los eventos que conducen a una falla del sistema, determina sus consecuencias y diseña mecanismos para minimizar su ocurrencia o recurrencia. El FMEA proviene de un nivel o la combinación de niveles de abstracción, que analiza el sistema de funciones, subsistemas y componentes (Guedes Soares, 2002). El FMEA ha tenido varias modificaciones y se ha mejorado con diferentes estándares y publicaciones científicas. Una de las extensiones del FMEA es el FMECA, el cual proporciona un tratamiento extenso de las fallas (Guedes Soares, 2002). El FMECA es una metodología diseñada para identificar posibles modos de falla potenciales para un producto, proceso o diferentes partes de un sistema, los efectos de estas fallas y la definición de estrategias para evitar y/o mitigar las fallas (Sultan, 2011).

El resultado de un análisis FMECA requiere de dos pasos: El primer paso es el análisis de modos y efectos de falla (FMEA) y el segundo paso es el análisis de criticidad. El FMEA consiste en un grupo sistematizado de actividades destinadas a reconocer y evaluar la falla potencial de un sistema, así como las causas y consecuencias de cada modo de falla. Posteriormente, se identifican acciones y se definen procedimientos de detección que podrían eliminar o reducir la posibilidad de que la falla potencial ocurra (Sultan, 2011). El formato implementado para realizar el FMEA se muestra en la Tabla 43. Para realizar el FMEA, se hace una descripción del sistema y un diagrama que lo represente al nivel de abstracción que se requiera. Para esto, se puede hacer uso del diagrama funcional que ilustra la operación, la interrelación y la dependencia entre los componentes funcionales del sistema. De esta manera, se identifican cada una de las partes que componen el sistema, subsistemas y componentes, éstos se enumeran y se define su función.

Luego se deben listar cada uno de los modos de falla y sus respectivas causas, donde cada modo de falla puede tener diferentes causas. Se define el modo operativo y los efectos de falla que corresponden a las consecuencias de cada modo de falla en la operación. Los efectos, se pueden diferenciar en 3 niveles: local, siguiente nivel de abstracción y efecto final. Los efectos locales muestran el impacto del modo de falla en la operación y funcionalidad del elemento. Los efectos finales pueden ser el resultado de múltiples fallas. Para la columna de método de detección de la falla deben describirse las características de detección para

cada modo de falla, teniendo en cuenta los patrones de comportamiento que pueden indicar la presencia de una falla.

Para las disposiciones compensatorias, se debe tener en cuenta que una falla detectada debe corregirse para eliminar su propagación a todo el sistema y maximizar su confiabilidad. Por esto, se deben identificar las disposiciones que aliviarán el efecto de un mal funcionamiento o falla. Esto incluye: elementos redundantes para una operación continua y segura, dispositivos de seguridad, modos alternativos de operación, como unidades de respaldo y reserva. Cualquier acción que requiera la acción del operador debe describirse claramente. La clase de severidad es usada para proporcionar un indicador cualitativo del peor efecto potencial resultante del modo de falla. Las clases de severidad se muestran en la Tabla 41.

Tabla 41

Niveles de severidad

Efecto	Clasificación	Criterio
Catastrófico	1	Un modo de falla que puede causar la muerte o pérdida completa de la misión
Crítico	2	Un modo de falla que puede causar lesiones graves o una degradación importante del sistema, daños o reducción en el desempeño de la misión
Marginal	3	Una falla que puede causar lesiones menores o degradación en el rendimiento del sistema o de la misión
Menor	4	Una falla que no causa lesiones ni degradación del sistema, pero que puede ocasionar fallas en el sistema y mantenimiento no programado

En el segundo paso del análisis de criticidad (CA), se realiza la combinación de una determinación probabilística de ocurrencia de un modo de falla combinada con el impacto que tiene en el éxito de la misión del sistema (Guedes Soares, 2002). Para realizar el análisis de criticidad, se deben tener en cuenta los siguientes conceptos:

Probabilidad del efecto de falla β: El valor β representa la probabilidad condicional de que ocurra el efecto de falla con la clasificación de criticidad especificada dado que ocurre el modo de falla. En muchos casos, la estimación de β es una cuestión de juicio impulsada en gran medida por la experiencia previa del analista. En la Tabla 42 se presentan los niveles de la probabilidad del efecto de falla.

Tabla 42

Niveles de la probabilidad del efecto de falla β

β	Descripción
1	El sistema ciertamente funcionará mal

0.5	El sistema podrá funcionar incorrectamente
0.1	El sistema raramente fallará
0	El modo de falla no tiene efecto sobre el sistema

Proporción del modo de falla α: Es la probabilidad de que el elemento falle en el modo de falla identificado. Si se enumeran todos los posibles modos de falla de un elemento, la suma de sus valores α correspondientes debe ser igual a 1. Generalmente, los valores de α deben estar disponibles en una fuente de datos, sin embargo, si no están disponibles, los valores pueden ser asignados según la experticia del analista.

Tasa de falla λ: Las estimaciones de λ se pueden obtener de los datos de prueba o campo, o de fuentes genéricas de índices de fallas.

Tiempo de funcionamiento T: Es el tiempo de funcionamiento, en horas o en número de ciclos de funcionamiento del sistema.

El número de criticidad del modo de falla C_r se usa para clasificar cada modo de falla potencial según su ocurrencia y la consecuencia de su efecto. Para una clasificación de gravedad particular, el C_r de un elemento es la suma de los números de criticidad del modo de falla C_m que tienen la misma clasificación de gravedad. En las ecuaciones (11) y (12) se representa C_r y C_m respectivamente.

$$C_r = \sum_{i=1}^{n} (C_m)_i \tag{11}$$

$$C_m = \beta\alpha\lambda T \tag{12}$$

Teniendo en cuenta los pasos que se requieren para realizar el FMECA, se muestra el procedimiento en la Figura 47 para llevar a cabo el mismo a través de una serie de actividades que permite la definición de los sistemas y la identificación de funciones, fallas, efectos y criticidad.

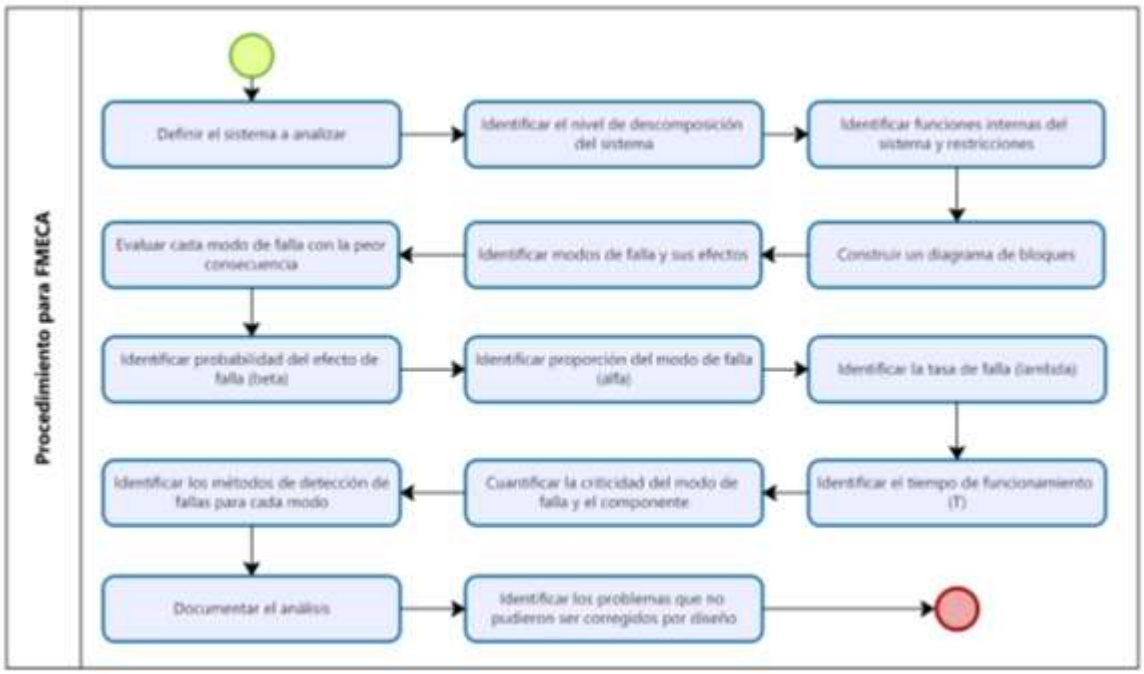


Figura 47. Procedimiento para realizar el FMECA. Fuente: Elaboración propia basada en (Guedes Soares, 2002)

Este tipo de análisis permite determinar los componentes críticos del vehículo que deben ser monitoreados para garantizar la confiabilidad del vehículo, disminuir los costos asociados a mantenimientos correctivos y ofrecer seguridad al usuario. Luego de realizar el análisis FMECA, se determina el RUL del componente para realizar la predicción. Así mismo, se encuentra que las variables tales como: corriente, voltaje, temperatura y potencia de la batería son críticas y es necesario capturar su comportamiento para describir el comportamiento de la batería en condiciones de tráfico real. Por esto, se realiza una planeación para llevar a cabo los experimentos de la bicicleta en condiciones de tráfico real. Para esto, se determinan las condiciones necesarias como peso máximo del conductor según instrucciones del manual de la bicicleta, por lo cual, se fija una persona para realizar las pruebas con un peso promedio de 112 kg. Además, se considera un proceso de instrumentación de la bicicleta, en este proceso se captura por medio de sensores las variables críticas determinadas anteriormente. Las variables obtenidas sin sensores se registran en la bitácora de los experimentos y se integran con la base de datos de cada experimento.

Tabla 43

Formato de la hoja de trabajo FMECA a partir de (Guedes Soares, 2002)

ID	ID ítem	Función	Modos de falla y causas	Modo operativo	Clase de severidad	Probabilidad de falla	Probabilidad de efecto de falla (β)	Proporción del modo de falla (α)	Tasa de falla (λ-p)	Tiempo de funcionamiento (T)	Modo de falla C_m	Ítem C_r	Observación