



Series de tiempo y machine learning aplicado al comportamiento de variables eléctricas en un banco de baterías

**Trabajo de Título para optar al título profesional de
Ingeniero Civil Matemático**

Felipe Andrés Córdova Miranda

Profesor Guía

Cristian Meza

Instituto de Ingeniería Matemática

Profesora Co-Guía

Natalia Bahamonde

Pontificia Universidad Católica de Valparaíso

Integrantes de la comisión evaluadora:

Karine Bertin

Héctor Olivero

INSTITUTO DE INGENIERÍA MATEMÁTICA
VALPARAÍSO, DICIEMBRE 2025

Resumen

En este trabajo de título se propone un enfoque para la predicción de fallas en un banco de baterías de plomo-ácido a partir de datos históricos de variables eléctricas comprendidas en el período de abril 2024 a abril 2025, las cuales son: voltaje total, voltaje promedio de celdas, equilibrio, corriente y temperatura promedio. El objetivo central es mostrar empíricamente los resultados de la comparación entre el desempeño de modelos clásicos de series de tiempo (Holt y Holt-Winters) frente a modelos de aprendizaje automático (redes de memoria a corto y largo plazo) para pronosticar el comportamiento de las señales y detectar, con antelación, eventos de fallo. Para esto, en primer lugar, se realiza un cuidadoso preprocesamiento: sincronización de señales a un muestreo común (mismo inicio y paso temporal), limpieza de ruidos y outliers, e imputación de datos faltantes. Como línea base, se presentan y ajustan modelos de Holt-Winters, midiendo errores de pronóstico, con particular énfasis en la raíz cuadrada del error cuadrático medio (RMSE). Se usa un paso temporal de 3 horas, estableciendo umbrales de falla, así como de funcionamiento normal/óptimo para la clasificación de los datos en cada una de las variables. Paralelamente, se desarrolla un modelo de machine learning, en específico de red neuronal artificial, llamado Long Short Term Memory (LSTM), entrenado de modo secuencial con validación temporal simple (entrenamiento: abril-diciembre 2024; validación: enero-abril 2025). La evaluación incluye la comparación de los modelos mediante métricas de testeo estadísticas. Con ello, el estudio aporta evidencia cuantitativa sobre la conveniencia del uso de algoritmos de aprendizaje automático en aplicaciones reales de ingeniería, y en particular, en nuestro contexto de la industria minera, eléctrica y de energía.

Agradecimientos

En primer lugar, agradecer a Dios por la oportunidad de poder estudiar la carrera que siempre soñé. Desde pequeño, en educación básica y media me imaginaba encontrando formas de aplicar las matemáticas en la vida real y derribar los mitos acerca de su no uso en la vida cotidiana. Y lo pude lograr más de allá de mis expectativas en este proceso.

Gracias a todos mis profesores y profesoras, que realizan un arduo trabajo de formación profesional, con dedicación y vocación, considerando además el factor social que involucra trabajar con personas con diferentes historias y realidades, cualidad en la cual la Universidad de Valparaíso destaca profundamente. Pase por dos universidades antes y de ninguna me sentí parte. Sin su apoyo y ayuda incondicional dada mis circunstancias personales, no hubiese sido capaz de tener una formación profesional de calidad.

Gracias a mi hija Amelia Córdova, quien con solo su existencia me motivó a ser mejor cada día, tanto personal como profesionalmente. Ella fue mi sostén, el motivo más fuerte para no rendirme cada semestre pese a las dificultades y seguir adelante, por nuestro futuro, bienestar, crecimiento y estabilidad. A mi familia, sobre todo a mi madre, Soledad Miranda, quien estuvo apoyándome en cada año de la carrera, esforzándose más de lo que sus capacidades y fuerzas le permitían debido a su edad. Gran parte de este logro se lo lleva ella y es un honor poder dedicarle este trabajo. A mi padre, René Córdova, por siempre apoyarme, sustentarme y estar ahí en los momentos más importantes.

Como no nombrar a esa familia de la vida que uno mismo elige, los amigos y amigas. Raúl Carrasco, Simón Rivera, Risiery Aguirre, Víctor Gallardo, Daniel Mancilla, Jonathan Ruiz, Maite Fuentes, Javier Andrades, Johnny Huincahue, entre otros. A todos aquellos que fueron parte y estuvieron presentes en este proceso. Sin todos ellos/as, el desarrollo de este trabajo no hubiese sido posible. Un ser humano requiere tiempos de recreación para que la mente pueda desarrollarse de manera sana. Esto me permitió seguir adelante y me dió fuerzas cada día, sobre todo en los momentos más difíciles del proceso.

Índice general

Resumen	1
Agradecimientos	3
Capítulo 1. Preliminares	6
1.1 Introducción y motivación	6
1.2 Contexto y formulación del problema	7
1.3 Adquisición y presentación de los datos	8
1.4 Objetivos	9
1.4.1 Objetivo general	10
1.4.2 Objetivos específicos	10
1.5 Hipótesis de trabajo	10
Capítulo 2. Marco Teórico	11
2.1 Series de tiempo	11
2.1.1 Clasificación de series temporales	12
2.1.2 Componentes de una serie temporal	12
2.2 Métodos para el tratamiento de los datos	13
2.2.1 Tratamiento de Outliers	14
2.2.2 Tratamiento de datos faltantes (filtro de Kalman)	16
2.3 Modelos clásicos	20
2.3.1 Modelos aditivos y multiplicativos	20
2.3.2 Métodos Ingenuos (Holt - Winters)	21
2.3.3 Identificación de modelos: autocorrelación y autocorrelación parcial	22
2.4 Modelos de machine learning	23
2.4.1 Redes neuronales artificiales	24
2.4.2 Tipos de neuronas	25
2.4.3 Redes Neuronales Recurrentes (RNN)	27
2.4.4 Redes Long Short-Term Memory (LSTM)	29

2.4.5	Arquitectura LSTM	29
2.4.6	Análisis del flujo de gradiente en LSTM	30
2.4.7	Aplicación a predicción de series temporales	31
2.5	Métricas de error	32
Capítulo 3.	Metodología	35
3.1	Pre-procesamiento de los datos	35
3.1.1	Análisis exploratorio y estudio descriptivo de datos	35
3.1.2	Tratamiento de datos faltantes: Métodos heurísticos	37
3.1.3	Tratamiento de outliers	38
3.1.4	Tratamiento de datos faltantes: Filtro de Kalman	40
3.1.5	Extracción de características complementarias	41
3.2	Tests de estacionariedad: ADF y KPSS	42
3.2.1	Test ADF (Dickey–Fuller Aumentado)	42
3.2.2	Test KPSS	43
3.2.3	Interpretación conjunta y protocolo práctico	43
Capítulo 4.	Resultados y análisis	45
4.1	Gráficos de ACF y PACF	45
4.2	Gráficos de predicciones	50
4.2.1	Predicciones con tratamiento heurístico	50
4.2.2	Resultados métricas de error	55
4.2.3	Predicciones post filtro de Kalman	56
4.2.4	Comparación de rendimiento entre modelos	59
4.3	Automatización y puesta en marcha	63
Capítulo 5.	Conclusiones	66
Anexos		73
Capítulo A.	Capítulo 1	73
A.1	Rangos de normalidad y falla	73
Capítulo B.	Capítulo 2	75
B.1	Modelos Autorregresivos y de Medias Móviles	75
B.2	Modelos ARMA y ARIMA	77
Capítulo C.	Capítulo 3	78
C.1	Gráfico Distancia de Mahalanobis	78

C.2	Gráficos de variables crudas y ordenadas	78
C.3	Código R: Imputación con Filtro de Kalman	81
C.4	Características complementarias	87
C.4.1	Gráficos	87
C.4.2	Estacionariedad	90
Capítulo D.	Capítulo 4	92
D.1	Código fuente y notebooks	92
D.2	Gráficos de predicciones LSTM (con tratamiento heurístico)	92

Capítulo 1

Preliminares

1.1. Introducción y motivación

El mantenimiento predictivo de sistemas de almacenamiento de energía es un área clave en el funcionamiento de instalaciones industriales y de energía. Si no se tratan estos aspectos, se pueden desencadenar eventos que dificulten la eficiencia operativa y en peores escenarios, que puedan parar la producción por completo. En particular, los bancos de baterías de plomo-ácido, compuestos por múltiples celdas de aproximadamente 2 V en serie, son ampliamente usados en respaldo energético y subestaciones eléctricas. Su comportamiento dinámico, reflejado en señales como voltaje total, voltaje por celda, corriente de carga/descarga y temperatura, encierra información valiosa sobre el estado de salud (SoH) y la aparición incipiente de fallas (sulfatación, pérdida de capacidad, cortocircuito interno). Este trabajo se centra en el análisis de un año de datos históricos (abril 2024–abril 2025) de un banco de 60 celdas, con el fin de desarrollar y comparar modelos matemáticos y de aprendizaje automático capaces de anticipar eventos de falla con suficiente antelación para planificar acciones de mantenimiento preventivo y predictivo.

Abordar este problema es crucial para la gestión eficiente de recursos en industrias energéticas y mineras, donde los bancos de baterías soportan sistemas de respaldo y estabilidad de la red eléctrica. Una detección preventiva de fallas puede reducir hasta un 30 % los costos de mantenimiento correctivo y minimizar tiempos de inactividad no planificados [1].

Desde el punto de vista teórico, la comparación rigurosa entre modelos clásicos y modelos de machine learning en un caso real, aporta evidencia sobre la aplicabilidad y límites de cada enfoque, reforzando las competencias matemáticas y estadísticas de la disciplina.

El carácter secuencial de los datos operativos en un banco de baterías convierte el análisis de series temporales en una herramienta clave para transformar información cruda en beneficios tangibles: reducir tiempos de inactividad, optimizar reemplazos y prolongar la vida útil de los equipos. Además, este trabajo persigue la adquisición de experiencia práctica en el análisis teórico, el procesamiento y la ingeniería de datos, consolidando las competencias necesarias para proyectos de gran escala. Desde la perspectiva de la Ingeniería Civil Matemática, el desafío es doble; por un lado, requiere profundizar en la teoría de series de tiempo y en arquitecturas de redes neuronales recurrentes para capturar patrones complejos; por otro, exige validar esos modelos frente a métodos clásicos (Holt–Winters, ARIMA) mediante métricas cuantitativas y pruebas estadísticas rigurosas. El resultado esperado no se limita a demostrar si un modelo LSTM (Long-Short Term Memory) mejora la predicción y la detección temprana de fallas, sino también a generar lineamientos prácticos para integrar estas herramientas en sistemas de adquisición, control y supervisión de datos (SCADA/IoT), cumpliendo con los estándares de precisión y aplicabilidad propios de un proyecto de ingeniería de alto nivel.

1.2. Contexto y formulación del problema

En la actualidad, para los operadores de bancos de baterías de plomo-ácido, las señales eléctricas y térmicas, en particular el voltaje total, el voltaje promedio de celdas, el equilibrio (balancing), la corriente y la temperatura, constituyen las únicas mediciones históricas disponibles de forma rutinaria. Estas variables capturan aspectos complementarios del estado electroquímico y operativo: el voltaje total y promedio reflejan la tensión global y la homogeneidad entre celdas; la señal de equilibrio indica desbalance interno entre celdas; la corriente evidencia condiciones de carga/descarga que aceleran degradación; y la temperatura influye fuertemente en la cinética y la vida útil. Sin embargo, se carece de herramientas cuantitativas que conviertan estos datos en alertas tempranas de falla. Esta carencia provoca que las intervenciones sean reactivas (tras la aparición de síntomas evidentes o fallas) en lugar de preventivas, generando costos adicionales por paradas imprevistas y reemplazos de emergencia. El estado deseado es contar con un sistema capaz de anticipar, con suficiente antelación, la degradación o el fallo inminente del banco, aprovechando los datos existentes. Por lo tanto, modelar estas señales de manera rigurosa es vital para un servicio robusto y permitirá identificar patrones tempranos de degradación y construir alertas tempranas de fallo con la infraestructura de monitoreo existente. El análisis se realizará en función de estas cinco variables centrales, considerando como horizonte de predicción de enero a abril de 2025 y para puesta en producción en tiempo real, desde una semana, hasta un mes.

Esto último, significa que el modelo debe funcionar de manera local en un computador y hacer predicciones diarias con los datos actualizados de las variables.

De aquí, es que nace la pregunta de investigación de este trabajo: ¿Puede un modelo de aprendizaje automático (o de red neuronal), entrenado sobre datos de voltaje, corriente, temperatura y equilibrio, anticipar eventos de falla en un banco de baterías de plomo-ácido con menor error de predicción y mayor tiempo de aviso que los métodos clásicos de series de tiempo (Holt-Winters o ARIMA)?

El proyecto fue viable dado que se tenía un año completo de datos de alta frecuencia, provistos por la empresa Vértice Consultores Spa. Se contó con capacidad computacional suficiente para entrenar modelos en Python y se tenía experiencia previa en machine learning. Los recursos incluyen bibliotecas estándar (statsmodels, TensorFlow/PyTorch) y acceso a entornos de cómputo con GPU (Graphics Processing Unit). Algunas implementaciones se hicieron usando el software RStudio. La estructura propuesta (preprocesamiento, partición temporal, implementación y evaluación) fue factible dentro del cronograma de la tesis.

1.3. Adquisición y presentación de los datos

El conjunto de datos proviene de un banco de baterías de una subestación principal de una mina de cobre (S/E TTE-7 RESERVAS NORTE). Estos bancos, son elementos diseñados para almacenar corriente directa (DC) que se pueda usar en el momento que sea necesario. En el ámbito industrial, un banco de baterías es un sistema de almacenamiento de energía eléctrica, conformado por varias baterías conectadas. Estos bancos de baterías se utilizan para garantizar la continuidad operativa de equipos críticos y procesos industriales, además ayudan a estabilizar el suministro eléctrico durante cortes de energía, fallas en el suministro eléctrico o demanda energética de alto consumo. Existen varios tipos de bancos de baterías de acuerdo a sus características, el uso y rendimiento que se necesite. En este caso, el conjunto de datos proviene de baterías de tipo plomo-ácido. El banco está compuesto de 60 celdas de este tipo, donde cada una de ellas es de 2V. Las variables de estudio consideradas en la base de datos son las siguientes:

1. Voltaje Total (DC): Registra el nivel de voltaje en corriente continua del banco de baterías. Considerando el número de celdas y lo que aporta cada una, en condiciones normales de funcionamiento óptimo, este valor debería estar entre 120V y 132V.
2. Equilibrio (Balancing): Indicador interno de calidad de equilibrio, estimado a partir de los

voltajes de cada celda. Se expresa en porcentaje, donde valores cercano a 100 % indican que la mayoría de las celdas tienen voltajes similares, por el contrario, si aumentan las diferencias entre voltajes por celda, entonces el porcentaje de equilibrio disminuye. Este proceso de balanceo se produce para evitar que algunas se sobrecarguen o se descarguen más rápido que otras, tratando de igualar los voltajes por celdas. Esto es fundamental para preservar la vida útil del banco completo, mantener su capacidad efectiva y prevenir fallas prematuras. Evita el riesgo de sobrecargas, calentamientos o fallas de respaldo.

3. Voltaje Promedio Celdas: Es la media del voltaje de todas las celdas individuales que conforman el banco. Sirve para evaluar si las celdas están en equilibrio eléctrico, el estado de carga general del banco y si hay alguna celda defectuosa o deteriorada que baja el promedio.
4. Corriente: Es el flujo instantáneo de corriente continua que entra o sale del banco de baterías. Corriente positiva (+) significa que el banco se está cargando, mientras que Corriente negativa (-) significa que el banco se está descargando. Se mide con sensores de corriente tipo shunt, sensores Hall u ópticos, y la lectura se muestra en amperios (A).
5. Temperatura Promedio Celdas: Parámetro clave para evaluar su estado de salud, rendimiento y seguridad. Permite anticipar fallas, extender la vida útil del banco y prevenir riesgos térmicos. Es la media aritmética de la temperatura de todas las celdas del banco. Se mide con sensores integrados en el sistema de monitoreo, ubicados sobre o cerca de las celdas.

Cada variable representa una magnitud física específica y se mide en unidades propias. Bajo condiciones normales de operación, estas variables fluctúan dentro de rangos bien definidos, lo que permite establecer umbrales de normalidad y detectar desviaciones asociadas a estados de falla. Los detalles específicos de cada variable y sus rangos se presentan en el Anexo [A.1](#).

1.4. Objetivos

Este trabajo busca construir un proceso reproducible y robusto para limpiar, imputar y predecir señales eléctricas de un banco de baterías. Se combinan técnicas estadísticas (detección de outliers y filtro de Kalman) con modelos de aprendizaje profundo (LSTM) para obtener predicciones útiles y confiables del comportamiento dinámico de cada serie.

1.4.1. Objetivo general

El objetivo general es ajustar métodos clásicos y de aprendizaje automático a través de datos históricos de variables eléctricas, usando Python y RStudio para preparación y predicción en series temporales, y así anticipar fallas en un banco de baterías de una mina de cobre.

1.4.2. Objetivos específicos

Los objetivos específicos guiarán el diseño, implementación y evaluación de modelos para la predicción y detección anticipada de fallas en el banco de baterías, comparando enfoques clásicos y de aprendizaje automático, asegurando un preprocesamiento y validación temporal coherente con la naturaleza de las series. A continuación se presentan:

- OE 1: Contextualizar, realizar un análisis exploratorio exhaustivo y el preprocesamiento de datos a través de la detección y tratamiento de outliers, selección de ventanas e imputación de datos faltantes (heurísticos y filtro de Kalman) para fundamentar la construcción de conjuntos de datos de entrada apropiados a los modelos.
- OE 2: Definir y ajustar modelos estadísticos clásicos (suavizamiento exponencial simple y doble) usando herramientas teóricas y computacionales para tener una línea base interpretable en la predicción de las series.
- OE 3: Definir y configurar una familia de modelos de aprendizaje automático basada en redes neuronales artificiales LSTM, especificando arquitectura, función de pérdida y estrategia de validación temporal para ser aplicados a predicción en series temporales.
- OE 4: Entrenar y comparar los modelos mediante la evaluación del desempeño con métricas cuantitativas (RMSE, MAE, MAPE) y análisis cualitativos (predicción vs. observado) para determinar las configuraciones más efectivas para la anticipación de fallas.

1.5. Hipótesis de trabajo

Los modelos de aprendizaje automático alcanzan un desempeño predictivo superior, en base a métricas de rendimiento, que los modelos clásicos para predicción de series temporales en nuestro contexto. Además, un adecuado preprocesamiento reduce los errores en las predicciones.

Capítulo 2

Marco Teórico

2.1. Series de tiempo

Básicamente, una serie temporal es una secuencia de datos de un proceso natural (o un experimento) observado a lo largo del tiempo. Por lo tanto, está ordenada en el tiempo y no debemos intercambiar las posiciones de dos valores de la serie temporal. Existen numerosos ejemplos de series temporales en nuestra vida cotidiana. Por ejemplo, el peso corporal medido cada domingo forma una serie temporal; el precio de cierre diario de las acciones de una empresa cotizada también configura una serie temporal. El primer paso en cualquier análisis de series temporales es siempre representar gráficamente los datos registrados y examinarlos cuidadosamente. En un sistema de coordenadas, el índice de tiempo t se encuentra en el eje horizontal y el valor observado en el vertical. De esta manera, los datos de la serie temporal se representan gráficamente en función del tiempo. Este tipo de gráfico se conoce como gráfico de series temporales o gráfico de tiempo. Es fácil de representar y muy útil para el análisis de series temporales. Muestra el patrón de evolución de una serie temporal. Este patrón incluye la tendencia y/o la estacionalidad (periodo) de la serie temporal.

Definición 1 (Serie Temporal). *Una secuencia ordenada de variables aleatorias $\{Y_t\}$ se denomina serie temporal, donde t es el índice temporal y $t \in T = \{\dots, -n, \dots, -1, 0, 1, \dots, n, \dots\}$. Los valores registrados de la serie temporal se denominan realización (o muestra u observación) de la serie temporal.*

Esta definición es formal y general, describe el objeto matemático sin especificar nada sobre la estructura probabilística interna del proceso (como si hay dependencia, estacionariedad, etc.). A veces también se denomina *proceso estocástico (discreto)*. Cabe destacar, que para posterior

modelamiento, los valores que se registran deben estar en intervalos de tiempo regulares, es decir, con el mismo paso temporal. En un contexto claro, una muestra o subconjunto de la serie temporal también puede denominarse simplemente *serie temporal* y, a menudo, se denota por $\{y_{1:n}\} = \{y_t; t = 1, 2, \dots, n\}$, ya que el número de observaciones es siempre finito.

2.1.1. Clasificación de series temporales

Una de las clasificaciones más importantes en el análisis de series temporales es la distinción entre procesos estacionarios y no estacionarios. La estacionariedad es una propiedad estadística crucial que simplifica enormemente el modelado, ya que implica que las propiedades del proceso no cambian con el tiempo.

Definición 2 (Estacionariedad). *Debido a que esta condición es difícil de verificar en la práctica, se suele utilizar una noción más débil: la estacionariedad en sentido débil (o de segundo orden). Una serie es débilmente estacionaria si cumple tres condiciones:*

1. La media del proceso es constante en el tiempo: $\mathbb{E}[Y_t] = \mu$ para todo t .
2. La varianza del proceso es finita y constante en el tiempo: $\mathbb{V}(Y_t) = \sigma^2 < \infty$ para todo t .
3. La autocovarianza entre dos puntos del tiempo, Y_t y Y_s , solo depende de la distancia temporal entre ellos ($|t - s|$), y no de su posición absoluta en el tiempo: $\text{Cov}(Y_t, Y_s) = \gamma(|t - s|)$.

En la práctica, cuando se habla de “estacionariedad”, generalmente se refiere a la estacionariedad débil [2].

Las series no estacionarias, por otro lado, son aquellas cuyas propiedades estadísticas cambian con el tiempo. Típicamente, presentan tendencias o estacionalidad, lo que requiere transformaciones como la diferenciación para poder ser modeladas apropiadamente.

2.1.2. Componentes de una serie temporal

En general, una serie temporal puede descomponerse en cuatro componentes principales: tendencia, estacionalidad, ciclos y ruido, que sumados o multiplicados, dan lugar a la serie observada y explican su comportamiento [3]. La tendencia corresponde al movimiento de largo plazo de la serie y refleja el crecimiento, decrecimiento o estancamiento del fenómeno en el tiempo, pudiendo adoptar una forma lineal o no lineal, y se denota habitualmente por T_t . La estacionalidad describe

las fluctuaciones periódicas y predecibles que se repiten en intervalos fijos de tiempo, como días, meses o trimestres, y que suelen estar asociadas a factores recurrentes tales como el clima, las vacaciones o los ciclos productivos. Los ciclos representan variaciones de mediano a largo plazo que, a diferencia de la estacionalidad, no presentan una periodicidad fija y cuya duración puede variar, estando frecuentemente vinculados a ciclos económicos o empresariales. Finalmente, el ruido o componente irregular corresponde a las variaciones aleatorias e impredecibles que permanecen una vez extraídas las componentes anteriores, y se modela usualmente como un ruido blanco, es decir, una secuencia de variables aleatorias con media cero, varianza constante y ausencia de autocorrelación.

Una vez definido y estudiado lo que es una serie temporal, junto con sus clasificaciones y componentes, se hace indispensable definir métodos para la preparación de los datos en estudio, dado que estamos en un contexto real, en el cual pueden ocurrir errores en las mediciones o en el traspaso de la información. En la siguiente sección se definen métodos para el tratamiento de los datos.

2.2. Métodos para el tratamiento de los datos

El modelamiento matemático a partir de un conjunto de datos, es un proceso que comienza desde el acceso a la información (datos) hasta la puesta en acción, que es una transformación de los datos en resultados inferidos y caracterizan el comportamiento dinámico de los sistemas, luego estos sirven para la toma de decisiones en determinados procesos. En la práctica, los datos muy pocas veces vienen ordenados, limpios y estructurados, por lo que es necesario dotarse de herramientas para poder manejar datos que no vengan con esas características. Existen diferentes métodos de tratamientos según las necesidades que requiera el problema. En este caso, se presentarán métodos para el tratamiento de valores atípicos (outliers), tanto univariados como multivariados y métodos para el tratamiento de datos faltantes o perdidos.

Los métodos heurísticos son técnicas o estrategias de solución utilizadas para encontrar soluciones buenas o aceptables a problemas complejos, especialmente cuando no es factible o práctico encontrar la solución óptima exacta debido a la complejidad o tiempo requerido. Aunque es sencillo y rápido, no siempre garantiza la mejor solución global, pero funciona muy bien en muchos problemas prácticos y puede ser el punto de partida del análisis en la investigación. Por otro lado, existen métodos más sofisticados y robustos para el tratamiento de datos, que en nuestro caso, pueden mejorar considerablemente las predicciones de los modelos matemáticos. En las

siguientes subsecciones se abordarán algunos procesos para el tratamiento de outliers y datos faltantes, considerando tanto métodos heurísticos como métodos estadísticos más robustos.

2.2.1. Tratamiento de Outliers

La presencia de valores atípicos (outliers) en series temporales de monitoreo industrial como las de un banco de baterías es un fenómeno común y de gran importancia. Un outlier puede ser el resultado de dos causas principales:

1. Errores de Medición: Fallos transitorios en un sensor, interferencia electromagnética o errores en la transmisión de datos.
2. Eventos Anómalos Reales: Comportamientos genuinos pero infrecuentes del sistema, que a menudo son precursores de una falla operativa (por ejemplo, un aumento de temperatura súbito o una caída de voltaje inesperada).

Por lo tanto, la correcta identificación de estos puntos es fundamental no solo para la limpieza de datos previa al modelado, sino también como una herramienta de diagnóstico en sí misma.

Existen diversos métodos para la detección y el tratamiento de los valores atípicos, en este trabajo se estudian métodos univariados y multivariados, evaluando su factibilidad y optimalidad para identificar anomalías que pueden indicar errores de sensor o el inicio de una falla en el sistema.

Métodos univariados

Estos métodos tratan cada serie temporal (Voltaje, Corriente, etc.) de forma independiente, es decir, se analiza cada variable por separado. Son útiles para detectar anomalías evidentes en una única señal.

El método del Rango Intercuartílico (IQR) es una regla no paramétrica para la detección de valores atípicos que se basa en la medida robusta de dispersión:

$$IQR = Q_3 - Q_1$$

donde Q_1 y Q_3 son los cuartiles 1 (percentil 25) y 3 (percentil 75), respectivamente. Al usar cuantiles (mediana/cuartiles) en lugar de la media y la desviación estándar, el IQR es poco sensible

a la presencia de valores extremos y, por lo tanto, constituye una primera estrategia de limpieza robusta [4, 5].

A partir de esta medida de dispersión, un punto y_i se considera un outlier si se encuentra fuera del intervalo definido por los límites inferior y superior extendidos, esto es,

$$y_i < Q1 - 1,5 \times IQR \quad \text{o} \quad y_i > Q3 + 1,5 \times IQR$$

Es excelente para una primera limpieza de datos. Puede detectar rápidamente valores extremos en variables como ‘Temperatura Promedio’ o caídas inexplicables en ‘Equilibrio’ que son claramente errores de sensor. Su limitación, mas bien, viene en que no tiene en cuenta la naturaleza temporal de los datos; un valor puede ser anómalo en el contexto de su “vecindario” temporal aunque no lo sea globalmente. Es por esto que es necesario, buscar más herramientas para perfeccionar el modelado.

El Método de la Desviación Estándar (Z-score) es un método paramétrico que asume que los datos siguen aproximadamente una distribución gaussiana. Se calculan la media (μ) y la desviación estándar (σ) de la serie [6].

Un punto y_i se considera un outlier si su Z-score es mayor que un umbral (típicamente 2 o 3):

$$Z_i = \frac{|y_i - \mu|}{\sigma} > 3$$

Si bien, es simple de implementar, su validez depende de la distribución de los datos. Podría funcionar para variables como Voltaje Total, pero sería menos fiable para Corriente, que puede tener una distribución asimétrica. Además, la media y la desviación estándar son muy sensibles a la presencia de los outliers que se intentan detectar, lo que puede “enmascarar” algunos puntos anómalos.

Métodos multivariados

Estos métodos son útiles para nuestro caso de estudio, ya que consideran la relación entre las variables. Un valor puede ser normal de forma aislada, pero anómalo en el contexto de los otros valores en ese mismo instante de tiempo.

La Distancia de Mahalanobis es la generalización multivariada del Z-score. Donde se mide

la distancia de un punto multivariado $(y_1, y_2, \dots, y_p)$ al centro de la nube de datos (el vector de medias), teniendo en cuenta la covarianza (y por tanto la correlación) entre las variables [7, 8, 9]. Los puntos con una distancia de Mahalanobis muy grande son considerados outliers. Puede detectar escenarios físicamente anómalos, como por ejemplo un Voltaje Total alto durante una fuerte corriente de descarga, aunque ninguno de los dos valores por separado sea un outlier univariado. Su limitación es que asume que los datos siguen una distribución elíptica (ej. normal multivariada) y puede ser sensible a la presencia de outliers en el cálculo de la matriz de covarianza.

2.2.2. Tratamiento de datos faltantes (filtro de Kalman)

El filtro de Kalman es un algoritmo recursivo de estimación óptima diseñado para inferir el estado no observable de un sistema dinámico a partir de mediciones ruidosas. Formulado originalmente por Rudolf E. Kalman (1960) [10], este método combina información proveniente de un modelo dinámico (predicción) con observaciones disponibles (actualización), produciendo una estimación del estado que minimiza el error cuadrático medio bajo supuestos de linealidad y ruido gaussiano. Debido a su eficiencia computacional y capacidad para integrarse en sistemas en tiempo real, el filtro de Kalman se ha convertido en una herramienta fundamental en áreas como control automático, navegación, robótica y procesamiento de señales [11, 12].

Filtro de Kalman discreto

Se considera un modelo de sistema dinámico que incorpora fuentes de ruido más flexibles y la presencia explícita de señales de control. En particular, se permite que el ruido de proceso posea correlaciones y actúe multiplicado por una matriz determinista G_k , y que la evolución del sistema esté influida por un término de control conocido $\Gamma_k u_k$. Este vector de control constituye la única entrada determinista del sistema, dado que incluso el estado inicial x_0 se modela como una variable aleatoria para reflejar la incertidumbre inherente a su valor verdadero [10, 13].

Bajo este marco, la evolución del sistema se describe mediante el siguiente conjunto de ecuaciones [14].

1. Modelo de la dinámica:

$$x_{k+1} = F_k x_k + \Gamma_k u_k + G_k w_k, \quad k \geq 0, \quad (2.1)$$

donde x_k es el estado del sistema, u_k es la entrada de control conocida, w_k representa el ruido de proceso y F_k , Γ_k y G_k son matrices deterministas de dimensiones apropiadas.

2. Modelo de la medición:

$$z_k = H_k x_k + v_k, \quad k \geq 0, \quad (2.2)$$

donde z_k denota el vector de observaciones disponibles y v_k corresponde al ruido de medición, típicamente modelado como un proceso aleatorio con media cero.

Todas las hipótesis que suponemos quedan aquí recogidas:

1. Del estado inicial x_0 se conoce su media $\mathbb{E}(x_0) = \bar{x}_0$ y su matriz de covarianzas

$$P_{x_0} = \text{Cov}(x_0).$$

2. De los ruidos w_k y v_k se conocen sus medias $\mathbb{E}(w_k) = 0$ y $\mathbb{E}(v_k) = 0$, y la matriz de covarianzas de su distribución conjunta

$$\text{Cov} \begin{pmatrix} w_k \\ v_l \end{pmatrix} = \begin{pmatrix} P_{w_k, w_l} & P_{w_k, v_l} \\ P_{v_k, w_l} & P_{v_k, v_l} \end{pmatrix} = \delta_{k,l} \begin{pmatrix} Q_k & S_k \\ S_k^\top & R_k \end{pmatrix},$$

donde $\delta_{k,l}$ es la delta de Kronecker, $Q_k := \text{Cov}(w_k)$, $R_k := \text{Cov}(v_k)$ y $S_k := \text{Cov}(w_k, v_k)$.

3. R_k es invertible para todo k .

4. x_0 es independiente de (w_k, v_k) para todo $k \geq 0$.

5. La matriz $P_{\tilde{z}_k}$ es la matriz de covarianzas de la innovación

$$\tilde{z}_k := z_k - \mathbb{E}_\ell(z_k | Z_{k-1}),$$

esto es,

$$P_{\tilde{z}_k} := \text{Cov}(\tilde{z}_k) = \mathbb{E} \left[\tilde{z}_k \tilde{z}_k^\top \right],$$

invertible. Y se consideran las observaciones

$$Z_k = (z_0^\top, \dots, z_k^\top)^\top$$

hasta el instante k , con $Z_{-1} = \emptyset$.

Teorema 2.1 (Algoritmo de Kalman). Sea $\hat{x}_{k|k}$ el mejor estimador afín de mínima varianza (BAMVE, por sus siglas en inglés) del estado para el sistema (2.1)–(2.2). Entonces $\hat{x}_{k|k}$ puede calcularse recursivamente mediante el siguiente procedimiento [14]:

1. Inicialización:

$$\hat{x}_{0|0} := \mathbb{E}(x_0) = \bar{x}_0, \quad P_{0|0} := P_{x_0}.$$

2. Etapa de predicción:

$$\hat{x}_{k+1|k} = (F_k - G_k S_k R_k^{-1} H_k) \hat{x}_{k|k} + G_k S_k R_k^{-1} z_k + \Gamma_k u_k, \quad (2.3)$$

$$P_{k+1|k} = (F_k - G_k S_k R_k^{-1} H_k) P_{k|k} (F_k - G_k S_k R_k^{-1} H_k)^\top + G_k (Q_k - S_k R_k^{-1} S_k^\top) G_k^\top. \quad (2.4)$$

donde $P_{k|k}$ y $P_{k+1|k}$ denotan, respectivamente, la matriz de covarianzas del error de estimación a posteriori y a priori. Formalmente,

$$P_{k|k} := \text{Cov}(x_k - \hat{x}_{k|k}), \quad P_{k+1|k} := \text{Cov}(x_{k+1} - \hat{x}_{k+1|k})$$

3. Etapa de estimación (actualización):

$$K_{k+1} = P_{k+1|k} H_{k+1}^\top (H_{k+1} P_{k+1|k} H_{k+1}^\top + R_{k+1})^{-1}, \quad (2.5)$$

$$\hat{x}_{k+1|k+1} = \hat{x}_{k+1|k} + K_{k+1} (z_{k+1} - H_{k+1} \hat{x}_{k+1|k}), \quad (2.6)$$

$$P_{k+1|k+1} = (I - K_{k+1} H_{k+1}) P_{k+1|k}. \quad (2.7)$$

La matriz K_{k+1} se denomina ganancia de Kalman.

La expresión 2.3 corresponde al valor esperado de x_{k+1} condicionado a las observaciones disponibles hasta el instante k , mientras que 2.4 describe la propagación de la incertidumbre asociada a dicha estimación. La etapa de actualización se fundamenta en el modelo de medición. La predicción *a priori* del estado se corrige a partir de la diferencia entre la observación real z_{k+1} y la observación predicha por el modelo, $H_{k+1} \hat{x}_{k+1|k}$. La ganancia de Kalman K_{k+1} se obtiene minimizando la matriz de covarianzas del error de estimación y determina el peso asignado a la nueva medición en función de la incertidumbre asociada tanto a la predicción como al ruido de medida. De este modo, el algoritmo de Kalman combina recursivamente el modelo dinámico y el modelo de medición para obtener, en cada instante, el estimador afín de mínima varianza (BAMVE) del estado del sistema. Esta es la idea central que relaciona las ecuaciones 2.3 - 2.7 con

las ecuaciones 2.1 y 2.2.

Ahora bien, la imputación de datos faltantes mediante el filtro de Kalman se aborda formulando primero la serie temporal como un modelo en espacio de estados (ecuaciones de transición y observación) que capture la dinámica temporal relevante (por ejemplo, nivel y tendencia). Las observaciones ausentes se marcan como no observadas y, durante el recorrido recursivo, la etapa de actualización se omite en esos instantes debido a la inexistencia de una medición z_k . En consecuencia, la estimación del estado se propaga únicamente mediante la etapa de predicción hasta la siguiente observación disponible. Una vez ajustados los parámetros del modelo (por ejemplo por máxima verosimilitud) se aplica el suavizador de Kalman para obtener estimaciones retrospectivas de los estados y de las observaciones. Las predicciones suavizadas resultantes se usan como imputaciones puntuales en los tiempos con datos faltantes. Este enfoque tiene la ventaja de integrar explícitamente un modelo dinámico y la propagación de incertidumbre temporal, de manejar de forma natural huecos de distinta longitud y de permitir validación mediante pruebas de eliminación artificial para evaluar error de imputación. Cabe observar que algunas implementaciones prácticas devuelven únicamente la serie imputada, por lo que la obtención explícita de varianzas o covarianzas del suavizador puede requerir el uso de bibliotecas o rutinas que muestren las matrices de covarianza cuando se desea cuantificar la incertidumbre asociada a las imputaciones [15, 16].

Los métodos de preparación de datos (Método del Rango Intercuartílico, Método Z-Score, Distancia de Mahalanobis, métodos heurísticos y filtro de Kalman) definidos en esta sección cumplen funciones complementarias orientadas a mejorar la calidad estadística y la utilidad predictiva de las series, al permitir la detección y el tratamiento de valores atípicos, la imputación de observaciones faltantes y la reducción de la variabilidad espuria, produciendo series más coherentes con las hipótesis de los modelos estadísticos y más adecuadas para tareas de estimación y predicción. Por su parte, el Filtro de Kalman proporciona un mecanismo óptimo de estimación en presencia de ruido estocástico, permitiendo suavizar la señal, reconstruir estados no observados (faltantes) y manejar de manera explícita la incertidumbre mediante un modelo dinámico lineal con ruido gaussiano. En conjunto, estos procedimientos permiten depurar, estabilizar y estructurar adecuadamente la información, facilitando análisis inferenciales más robustos y modelos predictivos mejor calibrados.

Una vez presentada la teoría previa a la modelización, en las siguientes secciones se presentan las definiciones de los modelos clásicos y de aprendizaje automático.

2.3. Modelos clásicos

Los modelos más básicos, se construyen separando los componentes de una serie de tiempo, es decir, interesa estimar qué parte del valor de la variable observada Y puede atribuirse a la tendencia, a la variación estacional, a la variación cíclica y a la variación irregular (Ruido).

2.3.1. Modelos aditivos y multiplicativos

Los modelos de descomposición clásica son los más básicos y permiten representar una serie temporal como la combinación de sus componentes fundamentales: tendencia (T), estacionalidad (E), ciclo (C) y un término irregular (I). Esta formulación ha sido ampliamente utilizada en la literatura clásica de series temporales [17, 18] y es expuesta en detalle en [3].

Definición 3 (Descomposición aditiva). Sea $(Y_t)_{t \in \mathbb{Z}}$ una serie temporal observada. Una descomposición aditiva expresa cada observación como la suma de cuatro componentes:

$$Y_t = T_t + E_t + C_t + I_t,$$

donde

- T_t es la tendencia en el tiempo t (componente de largo plazo),
- E_t es la estacionalidad periódica de periodo m (variación ligada a estaciones del año, mes, día, etc.),
- C_t es la componente cíclica (fluctuaciones de frecuencia y duración superiores a la estacionalidad, no necesariamente periódicas),
- I_t es la componente irregular o ruido (comportamiento impredecible, usualmente de media cero).

Definición 4 (Descomposición multiplicativa). La descomposición multiplicativa representa la serie como el producto de las mismas cuatro componentes:

$$Y_t = T_t \cdot E_t \cdot C_t \cdot I_t.$$

En este caso, las componentes suelen exigirse estrictamente positivas sin perder generalidad. Esto permite obtener una relación práctica importante, que es que la transformación logarítmica convierte el modelo multiplicativo en uno aditivo:

$$\log Y_t = \log T_t + \log E_t + \log C_t + \log I_t,$$

lo cual facilita estimación y análisis.

Definición 5 (Modelos combinados o híbridos). *Se denominan modelos combinados o híbridos a las formulaciones en las que algunas componentes interactúan multiplicativa y otras aditivamente. Ejemplos típicos son*

$$Y_t = (T_t \cdot E_t \cdot C_t) + I_t,$$

$$Y_t = T_t + (E_t \cdot C_t \cdot I_t).$$

La elección entre un modelo aditivo, multiplicativo o híbrido debe guiarse por la naturaleza de la variación observada (por ejemplo, si la amplitud de la estacionalidad crece con la tendencia, un modelo multiplicativo suele ser más adecuado).

Para garantizar la identificabilidad y facilitar la estimación, es habitual imponer condiciones adicionales sobre las componentes. En modelos aditivos con periodo estacional m : $\sum_{j=0}^{m-1} E_{t+j} = 0$ para cada bloque estacional. En modelos multiplicativos: la media geométrica de los factores estacionales sobre un periodo suele normalizarse a 1 (equivalente a que la media de $\log E_t$ en un periodo sea 0).

2.3.2. Métodos Ingenuos (Holt - Winters)

Estos son métodos de predicción a corto plazo y suponen que el fenómeno bajo estudio solo depende de los valores pasados. Se considera un peso mayor a los valores recientes de la serie, de manera que los coeficientes de ponderación decrecen de manera exponencial a medida que el tiempo se aleja. Los pesos de los valores pasados se calculan a partir de los coeficientes asociados al método [3].

Definición 6 (Suavizamiento exponencial simple (SES)). *Sea $(Y_t)_{t \in \mathbb{Z}}$ una serie temporal observada. El nivel suavizado L_t y el pronóstico a horizonte $h \geq 1$ se definen recursivamente por*

$$L_t = \alpha Y_t + (1 - \alpha) L_{t-1}, \quad (2.8)$$

$$\hat{Y}_{t+h|t} = L_t, \quad (2.9)$$

donde $\alpha \in (0, 1]$ es el parámetro de suavizamiento.

La elección de α puede realizarse minimizando la suma de cuadrados de los errores $\sum_t (Y_t - \hat{Y}_{t|t-1})^2$ o por máxima verosimilitud bajo supuestos gaussianos [3].

Definición 7 (Suavizamiento exponencial de Holt (doble)). *Para series con tendencia local se definen el nivel L_t y la tendencia T_t mediante*

$$L_t = \alpha Y_t + (1 - \alpha)(L_{t-1} + T_{t-1}), \quad (2.10)$$

$$T_t = \beta^*(L_t - L_{t-1}) + (1 - \beta^*)T_{t-1}, \quad (2.11)$$

$$\hat{Y}_{t+h|t} = L_t + h T_t, \quad h \geq 1, \quad (2.12)$$

con $\alpha, \beta^* \in (0, 1]$ *parámetros de suavizamiento.*

Definición 8 (Método de Holt–Winters aditivo). *Sea $(Y_t)_{t \in \mathbb{Z}}$ una serie temporal con estacionalidad de período m . El método de Holt–Winters aditivo define el nivel L_t , la tendencia T_t y el componente estacional S_t mediante las siguientes ecuaciones de suavizamiento:*

$$L_t = \alpha(Y_t - S_{t-m}) + (1 - \alpha)(L_{t-1} + T_{t-1}), \quad (2.13)$$

$$T_t = \beta^*(L_t - L_{t-1}) + (1 - \beta^*)T_{t-1}, \quad (2.14)$$

$$S_t = \gamma(Y_t - L_{t-1} - T_{t-1}) + (1 - \gamma)S_{t-m}, \quad (2.15)$$

$$\hat{Y}_{t+h|t} = L_t + hT_t + S_{t-m+h}, \quad h \geq 1, \quad (2.16)$$

donde $\alpha, \beta^*, \gamma \in (0, 1]$ *son los parámetros de suavizamiento y m es el período estacional.*

Este método extiende Holt incorporando una dinámica estacional explícita, adecuada para series con tendencia y estacionalidad aproximadamente aditivas.

Si bien los modelos autorregresivos y de medias móviles (AR, MA, ARMA y ARIMA) no se aplican en este estudio debido a que no resultan adecuados para la naturaleza de las series analizadas, se mencionan a lo largo del texto con fines descriptivos y comparativos. Una presentación más detallada de sus formulaciones y principales características puede encontrarse en el Anexo B.

2.3.3. Identificación de modelos: autocorrelación y autocorrelación parcial

La metodología de Box-Jenkins utiliza dos herramientas gráficas clave para identificar los órdenes p y q de un modelo ARIMA a partir de los datos de una serie estacionaria [2, 18].

Definición 9 (Función de autocorrelación — ACF). Sea $(Y_t)_{t \in \mathbb{Z}}$ un proceso estocástico débilmente estacionario con media constante μ . La función de autocovarianza en el rezago h se define como

$$\gamma(h) = \text{Cov}(Y_t, Y_{t-h}) = \mathbb{E}[(Y_t - \mu)(Y_{t-h} - \mu)],$$

y en particular

$$\gamma(0) = \text{Var}(Y_t).$$

La función de autocorrelación (ACF) se define como la autocovarianza normalizada:

$$\rho(h) = \frac{\gamma(h)}{\gamma(0)}, \quad h \in \mathbb{Z}.$$

El comportamiento de $\rho(h)$ es útil para la identificación de modelos $\text{MA}(q)$, ya que para estos modelos la ACF se trunca aproximadamente después del rezago q .

Definición 10 (Función de autocorrelación parcial — PACF). Sea (Y_t) un proceso débilmente estacionario. La función de autocorrelación parcial en el rezago h se define como

$$r(h) = \text{Corr}(Y_t, Y_{t-h} \mid Y_{t-1}, Y_{t-2}, \dots, Y_{t-h+1}),$$

es decir, es la correlación entre Y_t y Y_{t-h} una vez controlado el efecto lineal de los rezagos intermedios.

La PACF es especialmente útil para identificar modelos $\text{AR}(p)$, pues para estos procesos $r(h)$ se trunca aproximadamente después del rezago p .

Por ejemplo, un $\text{ARMA}(p, q)$ suele presentar una autocorrelación $\rho(h)$ con caída exponencial al igual que la autocorrelación parcial $r(h)$.

2.4. Modelos de machine learning

En el ámbito del aprendizaje automático (machine learning), los algoritmos pueden considerarse simultáneamente modelos predictivos y métodos de aprendizaje [19, 20]. Esto se debe a que, por una parte, definen una estructura funcional capaz de aproximar la relación entre las observaciones pasadas y futuras, es decir, un modelo de predicción; y por otra, incorporan el procedimiento mediante el cual ajustan sus parámetros, constituyendo así su propio método de

aprendizaje.

En contraste, los enfoques clásicos de series de tiempo, como ARIMA o Holt–Winters, establecen de manera explícita una formulación del modelo (por ejemplo, la estructura autoregresiva o de suavizamiento exponencial) [2, 3, 18] y un método de estimación separado (como la estimación por máxima verosimilitud o mínimos cuadrados). Esta diferencia refleja un cambio conceptual fundamental entre la modelación estadística tradicional, en la que se parte de supuestos teóricos sobre el proceso generador de los datos, y los enfoques modernos basados en aprendizaje automático, donde la estructura del modelo y su entrenamiento se integran dentro de un mismo algoritmo orientado a la optimización predictiva.

En este trabajo se estudió y aplicó en específico, un modelo de aprendizaje automático, en particular, de redes neuronales artificiales, llamadas redes de memoria a corto y largo plazo (LSTM, por sus siglas en inglés) [21]. Para comprender este modelo, en primera instancia es necesario introducir los conceptos de redes neuronales artificiales de manera general, y en específico, las redes neuronales recurrentes [20, 22].

2.4.1. Redes neuronales artificiales

Una red neuronal artificial (RNA) es un modelo computacional inspirado en la estructura y funcionamiento del cerebro humano. Está compuesta por unidades de procesamiento interconectadas llamadas neuronas o nodos, organizadas en capas. La arquitectura más básica es el perceptrón multicapa (MLP), que consta de una capa de entrada, una o más capas ocultas y una capa de salida. Cada neurona en una capa recibe entradas de las neuronas de la capa anterior, las pondera, suma, y aplica una función de activación no lineal (como Sigmoide, ReLU o Tanh) para producir su salida. En la Figura 2.1 se muestra la representación habitual de una neurona artificial básica. El aprendizaje de los pesos y del sesgo en redes modernas se realiza mediante el algoritmo de retropropagación, que consiste en ajustar iterativamente los pesos siguiendo la dirección opuesta al gradiente de una función de pérdida que mide el error del modelo [20, 23]. La diferenciabilidad (o diferenciabilidad casi en todas partes) de la función de activación es esencial para que este procedimiento pueda calcular gradientes y actualizar los parámetros correctamente.

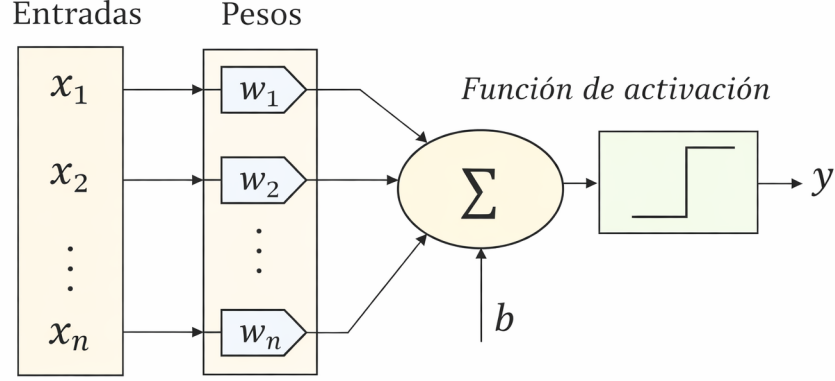


Figura 2.1. Esquema de una neurona artificial básica.

En este trabajo, la serie temporal se modela como un proceso estocástico $\{Y_t\}_{t \in \mathbb{Z}}$. Las realizaciones observadas de dicho proceso, denotadas por y_t , se utilizan como entradas del modelo de red neuronal y se representan por x_t para mantener la notación estándar en aprendizaje profundo. Así, en este caso, se cumple $x_t = y_t$.

2.4.2. Tipos de neuronas

Una *neurona artificial* es una unidad computacional que mapea un conjunto de entradas a una salida mediante una combinación lineal ponderada de las entradas, un término de sesgo y la aplicación de una función de activación $f : \mathbb{R} \rightarrow \mathbb{R}$ no lineal que introduce capacidad de modelado no lineal en la neurona y que, en redes entrenadas mediante descenso de gradiente, se supone diferenciable o diferenciable casi en todas partes. Formalmente, una neurona puede representarse como [24]:

$$y = f \left(\sum_{i=1}^n w_i x_i + b \right),$$

donde x_i son las entradas, w_i los pesos, b el sesgo y f la función de activación. Este esquema básico constituye el bloque elemental de las redes neuronales y su estudio remonta a trabajos clásicos sobre modelos formales de neuronas [25], y a la formalización del perceptrón por Rosenblatt [24]. El aprendizaje de los pesos en redes modernas se realiza típicamente por descenso de gradiente y retropropagación del error [23], lo que requiere que las funciones de activación sean diferenciables o casi diferenciables en la práctica [20].

1. Perceptrón

El *perceptrón* es el modelo histórico de neurona binaria propuesto por Rosenblatt [24]. En este modelo la función de activación es una función escalón (unit step), por lo que la salida es binaria:

$$f(z) = \begin{cases} 1, & z \geq 0, \\ 0, & z < 0, \end{cases} \quad \text{con} \quad z = \sum_{i=1}^n w_i x_i + b.$$

El perceptrón ajusta sus pesos mediante una regla de aprendizaje basada en el error cometido en cada iteración (perceptron learning rule) y puede resolver problemas linealmente separables, es decir, aquellos en los que existe una línea, plano o hiperplano que separa perfectamente las distintas clases (como las operaciones AND y OR). No obstante, como se señaló en trabajos posteriores, el perceptrón no puede resolver problemas no linealmente separables (por ejemplo XOR), lo que motivó el desarrollo de redes multicapa (MLP) y funciones de activación diferenciables [20, 26].

2. Neurona sigmoidea (Logística)

La función sigmoide logística es una función suave y acotada definida por

$$\sigma(z) = \frac{1}{1 + e^{-z}},$$

con $\sigma(z) \in (0, 1)$ y derivada $\sigma'(z) = \sigma(z)(1 - \sigma(z))$. Al ser diferenciable, facilita el entrenamiento por retropropagación y descenso de gradiente, ya que pequeños cambios en los pesos provocan pequeños cambios en la salida. Aun así, la sigmoide puede sufrir el problema del *desvanecimiento del gradiente* cuando las entradas son de gran magnitud, fenómeno que ocurre durante la retropropagación cuando los gradientes se vuelven progresivamente muy pequeños al propagarse hacia las capas iniciales de la red, dificultando la actualización efectiva de los pesos, especialmente en arquitecturas profundas. Por lo que su uso en capas internas de redes profundas ha sido en muchos casos reemplazado por funciones con mejor comportamiento numérico [20].

3. Tangente hiperbólica ($\tanh$)

La tangente hiperbólica se define como

$$\tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}},$$

y toma valores en $(-1, 1)$. Su derivada es $1 - \tanh^2(z)$. Al estar centrada en cero, $\tanh$ suele presentar comportamiento numérico superior frente a la sigmoide en muchas arquitecturas, ya que puede acelerar la convergencia al producir activaciones con media cercana a cero. Por ello es habitual su uso en RNN clásicas y en componentes internas de celdas LSTM (por ejemplo, como función para el *candidate state*) [20, 21].

4. ReLU (Rectified Linear Unit)

La unidad lineal rectificada (ReLU) se define por

$$\text{ReLU}(z) = \max(0, z).$$

Es computacionalmente eficiente y mitiga en gran medida el problema del desvanecimiento del gradiente en redes profundas, favoreciendo la convergencia en tareas de aprendizaje profundo [27]. Su principal inconveniente práctico es el fenómeno de *neuronas muertas* (“dying ReLU”), donde unidades quedan fijadas en salida 0 y dejan de aprender; por ello se emplean variantes como Leaky ReLU o Parametric ReLU para mantener un gradiente pequeño cuando $z < 0$ [28].

En la implementación y diseño de redes conviene justificar la elección de la función de activación de cada capa en función de la arquitectura (profundidad, tipo de tarea) y de consideraciones numéricas; por ejemplo, en redes convolucionales modernas, ReLU y sus variantes son la elección estándar en capas ocultas, mientras que funciones como sigmoide o $\tanh$ siguen siendo útiles en salidas o en compuertas internas de arquitecturas recurrentes [20, 21, 27].

2.4.3. Redes Neuronales Recurrentes (RNN)

Las redes neuronales recurrentes (RNN) constituyen una familia de modelos de aprendizaje profundo diseñados para procesar datos secuenciales, como series de tiempo, texto o señales, en los que el orden temporal de las observaciones es fundamental [20]. En las redes neuronales feedforward (de propagación hacia adelante) la información fluye únicamente desde la entrada hacia la salida, sin considerar datos previos. En contraste, las RNN incorporan conexiones o bucles recurrentes que permiten a la red mantener un estado oculto h_t , capaz de retener información del pasado. Esto les permite modelar dependencias temporales y relaciones entre elementos de una secuencia [29, 30].

La ecuación de actualización de una red neuronal recurrente (RNN) está dada por:

$$h_t = f(W_h h_{t-1} + W_x x_t + b), \quad (2.17)$$

donde cada término representa:

- h_t : estado oculto en el tiempo t , que actúa como la memoria de la red y resume la información procesada hasta ese instante.
- h_{t-1} : estado oculto del tiempo anterior, encargado de transportar información histórica.
- x_t : vector de entrada en el tiempo t , correspondiente al dato observado en ese instante (por ejemplo, un valor de una serie de tiempo o una palabra codificada).
- W_h : matriz de pesos asociada al estado oculto previo h_{t-1} , que determina cómo la memoria pasada influye en el nuevo estado.
- W_x : matriz de pesos asociada a la entrada actual x_t , responsable de incorporar la nueva información al estado oculto.
- b : vector de sesgo (*bias*) añadido antes de la activación, que permite desplazar la transformación lineal.
- $f(\cdot)$: función de activación no lineal, comúnmente $\tanh$ o ReLU, que permite dotar al modelo de capacidad de representación no lineal y estabilidad numérica [20].

En consecuencia, la salida del modelo en un tiempo dado depende tanto de la información de entrada actual como del historial acumulado en memoria de la serie.

No obstante, las RNN en su forma estándar presentan problemas para capturar dependencias temporales a largo plazo, principalmente debido a los problemas de desvanecimiento y explosión del gradiente durante el proceso de entrenamiento [20]. Estas limitaciones motivaron el desarrollo de arquitecturas recurrentes mejoradas, como las redes de memoria a corto y largo plazo (LSTM), las cuales introducen mecanismos de compuertas para regular el flujo de información y preservar dependencias relevantes a lo largo del tiempo [21]. La ecuación 2.17 corresponde a la forma más simple de una RNN. Arquitecturas más avanzadas, como las redes LSTM, reemplazan esta actualización directa por un conjunto de ecuaciones internas basadas en mecanismos de compuertas, manteniendo la dependencia funcional entre h_t , h_{t-1} y la entrada x_t .

2.4.4. Redes Long Short-Term Memory (LSTM)

Las redes Long Short-Term Memory (LSTM) son un tipo avanzado de RNN que resuelven el problema de desvanecimiento del gradiente mediante una arquitectura de memoria más compleja [20, 21, 31]. La formulación original fue propuesta por Hochreiter y Schmidhuber (1997) y posteriormente fue refinada, entre otros, por Gers et al. (2000) con la introducción explícita de la "puerta de olvido" para mejorar la capacidad de olvido selectivo en presencia de secuencias largas [21, 31].

La clave de la LSTM es la celda de memoria (c_t), un estado interno que actúa como una cinta transportadora de información a lo largo de la secuencia. El flujo de información hacia y desde esta celda es controlado por tres "puertas" (gates) principales [20, 22]:

1. Puerta de Olvido (Forget Gate): Decide qué información de la celda de memoria anterior (c_{t-1}) se debe descartar.
2. Puerta de Entrada (Input Gate): Decide qué nueva información de la entrada actual (x_t) se debe almacenar en la celda de memoria.
3. Puerta de Salida (Output Gate): Decide qué parte de la celda de memoria actualizada (c_t) se debe usar para generar la salida de la celda en el paso actual (h_t).

Esta arquitectura permite a las LSTM aprender, recordar patrones y dependencias a lo largo de secuencias muy largas, lo que las hace extremadamente efectivas para el modelado de series temporales complejas y multivariantes [20, 21, 32].

En las siguientes subsecciones se desarrolla de manera formal la estructura matemática de las redes neuronales LSTM, su entrenamiento mediante retropropagación a través del tiempo (BPTT, por sus siglas en inglés, Backpropagation Through Time), y las propiedades que explican su capacidad para modelar dependencias de largo plazo en series temporales.

2.4.5. Arquitectura LSTM

La celda LSTM introduce un estado de celda c_t que se actualiza por mecanismos multiplicativos (puertas) para permitir flujo de gradiente con menos atenuación. Una formulación canónica de la celda LSTM corresponde a la versión estándar del modelo, ampliamente utilizada en la literatura y que sirve como referencia para sus distintas variantes [21, 31, 32], la cual es:

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i), \quad (2.18)$$

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f), \quad (2.19)$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o), \quad (2.20)$$

$$\tilde{c}_t = \tanh(W_c x_t + U_c h_{t-1} + b_c), \quad (2.21)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t, \quad (2.22)$$

$$h_t = o_t \odot \tanh(c_t) \quad (2.23)$$

donde $\odot$ denota el producto elemento a elemento (producto de Hadamard). Cada puerta i_t, f_t, o_t toma valores en $(0, 1)$ por la función sigmoideal σ , y el producto elemento a elemento con c_{t-1} permite que el estado celular se propague sin pasar por una transformación lineal completa que destruya la información. En esta notación, x_t representa la entrada en el instante t , construida a partir de la realización observada de la serie temporal, mientras que h_{t-1} denota el estado oculto previo. Las matrices $W_\bullet$ corresponden a los pesos asociados a la entrada x_t , mientras que las matrices $U_\bullet$ modelan la influencia del estado oculto previo h_{t-1} . En una RNN simple, la influencia del estado oculto previo se modela mediante una única matriz de pesos W_h . En las arquitecturas LSTM, este rol se generaliza mediante distintas matrices $U_\bullet$, asociadas a cada una de las compuertas, lo que permite un control más fino del flujo de información temporal [20]. En conjunto, las ecuaciones 2.18 - 2.23 definen el mecanismo mediante el cual una celda LSTM actualiza su estado interno regulando de forma controlada la información que se conserva del pasado y la que se incorpora en cada instante de tiempo.

2.4.6. Análisis del flujo de gradiente en LSTM

Considérese la derivada del estado celular final respecto a un estado anterior:

$$\frac{\partial c_t}{\partial c_{t-1}} = \text{diag}(f_t) + \text{términos dependientes de } \frac{\partial f_t}{\partial c_{t-1}},$$

La operación $\text{diag}(f_t)$ denota la matriz diagonal construida a partir del vector de compuerta de

olvido f_t [21, 32]. Si $f_t = (f_t^{(1)}, f_t^{(2)}, \dots, f_t^{(d)}) \in (0, 1)^d$, entonces

$$\text{diag}(f_t) = \begin{pmatrix} f_t^{(1)} & 0 & \dots & 0 \\ 0 & f_t^{(2)} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & f_t^{(d)} \end{pmatrix}.$$

La notación $(0, 1)^d$ denota el producto cartesiano del intervalo abierto $(0, 1)$ consigo mismo d veces, es decir, el conjunto de todos los vectores en $\mathbb{R}^d$ cuyas componentes toman valores estrictamente entre 0 y 1:

$$(0, 1)^d = \{x \in \mathbb{R}^d \mid 0 < x_i < 1 \text{ para todo } i = 1, \dots, d\}.$$

En el contexto de las redes LSTM, esta notación refleja que cada componente de los vectores de compuerta f_t , i_t y o_t actúa como un coeficiente de ponderación que regula, de forma independiente, el flujo de información en cada dimensión del estado interno.

Esta representación permite interpretar la actualización del estado de celda como una transformación aproximadamente lineal del estado previo,

$$c_t = \text{diag}(f_t) c_{t-1} + i_t \odot \tilde{c}_t,$$

donde cada componente del estado celular es escalada de manera independiente por el valor correspondiente de la compuerta de olvido. En particular, si las entradas de f_t son cercanas a 1, la derivada parcial es cercana a la identidad y la información (y por ende el gradiente) puede fluir a través de muchos pasos sin decaer exponencialmente, es decir, la información asociada a dicha componente se preserva. Esta propiedad explica la capacidad de las redes LSTM para mantener dependencias de largo plazo y mitigar el problema del desvanecimiento del gradiente [21, 31].

2.4.7. Aplicación a predicción de series temporales

En problemas de pronóstico univariado o multivariado es habitual emplear esquemas de entrenamiento y evaluación que respeten la naturaleza temporal de los datos. En particular, se utilizan ventanas deslizantes, donde la red recibe como entrada un conjunto de observaciones pasadas consecutivas y produce una o varias predicciones futuras, ya sea a un paso o en múltiples pasos. Este enfoque permite modelar la dependencia temporal de forma directa. En casos que involucran secuencias largas, puede recurrirse a arquitecturas que conservan el estado interno entre bloques consecutivos de datos, lo que favorece la captura de dependencias de largo plazo, aunque exige un

control cuidadoso del reinicio de dichos estados al finalizar cada serie. Finalmente, la evaluación del desempeño se realiza mediante esquemas de validación temporal, como particiones cronológicas o evaluaciones progresivas en el tiempo, evitando la mezcla aleatoria de observaciones y previniendo el uso de información futura durante el entrenamiento, con el fin de obtener estimaciones realistas de la capacidad predictiva del modelo [3, 20, 21].

2.5. Métricas de error

Para evaluar y comparar el rendimiento de los modelos de pronóstico, se utilizan diversas métricas de error. Las más comunes son [3]:

Definición 11 (Error cuadrático medio — MSE). Sean $Y_1, \dots, Y_n$ los valores observados e $\hat{Y}_1, \dots, \hat{Y}_n$ los valores pronosticados. El error cuadrático medio (Mean Squared Error) se define como

$$MSE = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2. \quad (2.24)$$

Este criterio penaliza más fuertemente los errores grandes debido al cuadrado de las diferencias.

Definición 12 (Raíz del error cuadrático medio — RMSE). La raíz del error cuadrático medio (Root Mean Squared Error) se define como

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2}. \quad (2.25)$$

donde $Y_1, \dots, Y_n$ son los valores observados, $\hat{Y}_1, \dots, \hat{Y}_n$ los valores pronosticados y n el número de observaciones.

Al tomar la raíz cuadrada del MSE, el RMSE queda expresado en las mismas unidades que la variable original, facilitando su interpretación.

Definición 13 (Error absoluto medio — MAE). El error absoluto medio (Mean Absolute Error) se define por

$$MAE = \frac{1}{n} \sum_{i=1}^n |Y_i - \hat{Y}_i|. \quad (2.26)$$

donde Y_i es el valor real, $\hat{Y}_i$ es el valor pronosticado y n es el número de observaciones [3].

Este indicador expresa el error medio en las mismas unidades de la variable observada y penaliza de forma lineal las desviaciones absolutas, lo que facilita su interpretación práctica frente a medidas que ponderan en exceso los errores grandes (por ejemplo, el MSE).

Definición 14 (Error porcentual absoluto medio — MAPE). *El error porcentual absoluto medio (Mean Absolute Percentage Error) se define por*

$$MAPE = \frac{100}{n} \sum_{i=1}^n \left| \frac{Y_i - \hat{Y}_i}{Y_i} \right|. \quad (2.27)$$

donde Y_i es el valor real, $\hat{Y}_i$ es el valor pronosticado y n es el número de observaciones [3].

Este indicador expresa el error en términos porcentuales, siendo útil para comparar el rendimiento de modelos en series con distintas escalas.

El RMSE se selecciona como criterio central debido a su capacidad para cuantificar de manera directa la magnitud promedio del error entre los valores observados y los valores pronosticados, manteniéndose en las mismas unidades de la variable original. Esta característica facilita su interpretación dentro del contexto físico de las señales eléctricas. Además, el RMSE penaliza con mayor intensidad los errores grandes, lo cual es especialmente relevante en sistemas sensibles como los bancos de baterías, donde desviaciones significativas pueden asociarse a fallas incipientes, comportamientos anómalos o condiciones operativas críticas. Por lo tanto, su uso permite priorizar modelos que no solo sean precisos en promedio, sino también más consistentes en la predicción de variaciones abruptas.

En este capítulo se han revisado los conceptos y herramientas fundamentales necesarios para abordar la modelación y predicción de series temporales en el estudio: las características intrínsecas de las series (tendencia, estacionalidad, autocorrelación y estacionariedad), las aproximaciones clásicas de modelado (Holt y Holt-Winters), alternativas basadas en machine learning (redes recurrentes LSTM), y las técnicas para el tratamiento de anomalías y datos faltantes. Cada familia de métodos aporta ventajas y limitaciones bien definidas. Por su parte, los modelos clásicos de suavizamiento exponencial son transparentes (es decir, su funcionamiento puede entenderse y explicarse directamente a partir de sus parámetros y estructura) e interpretables, ya que sus componentes (nivel, tendencia y estacionalidad) tienen una interpretación directa y permiten comprender el comportamiento de la serie. Sin embargo, su capacidad para modelar dinámicas no lineales o cambios estructurales es limitada. En contraste, los métodos basados en machine learning, como las redes LSTM, poseen una alta capacidad de representación y permiten capturar

relaciones no lineales y dependencias de largo plazo, a costa de una menor interpretabilidad, mayores requerimientos de datos y un mayor riesgo de sobreajuste. Por su parte, las técnicas de tratamiento de outliers e imputación de datos faltantes influyen de manera decisiva en la calidad del modelado, ya que una aplicación inadecuada puede introducir sesgos y afectar la validez de las inferencias y predicciones.

Las implicancias prácticas de este marco teórico influyen directamente en la metodología del trabajo: la identificación de estacionariedad y dependencia temporal orientará la elección de modelos de referencia (por ej. Holt-Winters o ARIMA) como línea base. La posible presencia de relaciones no lineales sugiere el uso de LSTM como modelo alternativo. La detección y tratamiento de outliers (IQR, distancia de Mahalanobis) y la gestión de datos faltantes (heurísticos simples versus estimación por filtro de Kalman en un enfoque de espacio de estados) deben especificarse y justificarse antes del ajuste de modelos.

A partir de estas conclusiones, en el capítulo de metodología se describirán: (i) el flujo de preprocesamiento (detección y manejo de outliers, criterios y orden de imputación), (ii) la especificación y calibración de los modelos candidatos (modelos base y modelos avanzados), y (iii) el esquema de validación temporal y las métricas empleadas para la comparación. Con esto se establece un puente claro entre la fundamentación teórica y las decisiones empíricas que se implementarán en los capítulos siguientes.

Capítulo 3

Metodología

3.1. Pre-procesamiento de los datos

Esta parte es fundamental y conlleva la mayoría de las veces un gran trabajo de preparación de los datos para luego poder realizar estudios más profundos sobre ellos. El tratamiento de los datos incluyó la consolidación de archivos CSV mensuales en un único conjunto ordenado temporalmente, se depura mediante la eliminación de columnas irrelevantes, se filtran (extraen) por separado las variables y reordenan cronológicamente las observaciones de cada una. Esta etapa garantiza la integridad y coherencia del registro antes del análisis. En este contexto, la integridad del registro se refiere a la preservación completa y correcta de las observaciones originales, asegurando que no existan pérdidas, duplicaciones ni errores estructurales en los datos consolidados. Por su parte, la coherencia del registro implica que las observaciones se encuentren correctamente alineadas en el tiempo, respeten el orden cronológico y mantengan una consistencia lógica entre las distintas variables analizadas.

3.1.1. Análisis exploratorio y estudio descriptivo de datos

De la variable Voltaje Total se tienen 48.019 registros, de Equilibrio 35.314, de Voltaje Promedio Celdas 14.827, de Corriente 61.253, y de Temperatura Promedio 16.394 registros, lo que suma un conjunto total de 175.807 datos. Los gráficos de las series de tiempo en bruto se incluyen en el Anexo C.2, mientras que en la Tabla 3.1 se resumen las estadísticas descriptivas de las mismas.

Estadístico	Voltaje Total (V)	Equilibrio (%)	Voltaje Prom. (V)	Corriente (A)	Temp. Prom. (°C)
cantidad	48019	35314	14827	61 253	16394
media	426.908	93.34	2.119	232.413	28.54
des. est.	19548.77	6.873	0.102	17139.83	4.711
mínimo	0.00	0.000	0.000	-1.89	0.000
25 %	124.31	93.18	2.019	0.66	27.40
50 %	129.54	94.31	2.156	0.71	28.30
75 %	133.41	96.18	2.198	0.75	29.20
máximo	$1,37562 \times 10^6$	97.77	2.291	$1,4405 \times 10^6$	217.3

Tabla 3.1. Estadísticos descriptivos de variables eléctricas en estudio en bruto.

En este punto, ya se tienen los datos separados por variable y ordenados en el tiempo. Cabe destacar que la frecuencia de cada observación por variable tiene una frecuencia aproximada de 5 minutos, pero varía en cada caso. Ahora bien, para poder continuar hacia una investigación predictiva mediante series de tiempo, se necesita convertir cada variable de estudio en una serie temporal con el mismo paso temporal. Esto es la base para la validez de los modelos, la consistencia e interpretación de los resultados. Además, se cuentan los datos nulos existentes o que no son números (NaN: Not a Number) y se tiene que si bien, no se presentan datos NaN, hay datos faltantes o perdidos, por lo que se debe realizar un tratamiento de imputación de datos para poder continuar con un modelamiento en nuestro contexto, así como también un tratamiento de outliers (valores atípicos).

Se proponen dos caminos diferentes para el tratamiento de los datos:

Camino 1: (i) realizar una imputación a través de métodos heurísticos; (ii) realizar el tratamiento de outliers; (iii) realizar predicciones con modelos de Holt-Winters y LSTM; (iv) comparar resultados y concluir la superioridad del rendimiento del modelo LSTM.

Camino 2: (i) realizar tratamiento de outliers; (ii) realizar imputación de datos faltantes vía filtro de Kalman; (iii) rehacer predicciones con series post aplicación del filtro de Kalman; (iv) comparar rendimiento del modelo LSTM a través de métricas de error con los diferentes tratamientos de datos faltantes.

3.1.2. Tratamiento de datos faltantes: Métodos heurísticos

Estos métodos son razonables como punto de partida para luego realizar una comparación de los resultados con los obtenidos al trabajar con métodos más sofisticados.

Dado que la frecuencia inicial de los datos varía entre 5 y 10 minutos y se tienen distintas cantidades de datos por variable con diferentes pasos temporales, en primera instancia, se realiza el estudio sobre series temporales construidas considerando un mismo paso temporal de 3 horas, además de alinearlas en el tiempo; es decir, hacer que comiencen y terminen exactamente en la misma fecha y hora. Para cada variable, se construyen 2 series temporales; una con el valor mínimo/máximo en cada paso de tiempo y otra considerando la mediana/promedio en este intervalo de tiempo (3 hrs.). La regla de elección de este valor es de tipo heurístico y recae en el sentido de lo que representa cada variable. Por ejemplo, en las series de voltaje promedio por celda y equilibrio, tiene sentido estudiar cuando se producen los valores mínimos, pues estos valores pueden ser indicativos de fallas, mientras que para la variable temperatura promedio, tiene más sentido estudiar cuando se producen los valores máximos, pues estos indican rangos de anomalías. Si en algún intervalo de tiempo no se registran datos, se reemplazan por el valor mínimo/máximo entre su anterior valor y el siguiente. Si el siguiente es un valor perdido, toma el valor del anterior paso.

En síntesis, los datos fueron regularizados a intervalos de 3 horas mediante remuestreo a partir del mínimo, la mediana o el máximo (dependiendo de cada caso). Las ventanas sin observaciones fueron imputadas utilizando reglas heurísticas basadas en el valor mínimo o máximo de los puntos adyacentes (y la mediana global), dependiendo del tipo de variable y su comportamiento esperado. Además, se realiza un alineamiento temporal de los datos, los cuales comienzan exactamente con fecha 6 de abril del 2024 a las 18 : 00 hrs. y terminan el 22 de abril del 2025 a las 12 : 00 hrs. Con estos antecedentes, se decide trabajar con las series construidas a partir de la mediana para las series de Voltaje Total, Voltaje Promedio y Corriente; mientras que para la serie Equilibrio se trabaja con la serie construida a partir del mínimo en cada intervalo; y finalmente, la serie Temperatura Promedio construida a partir del máximo, ya que estas reglas logran rescatar el comportamiento de cada variable, así como los valores que se desean estudiar. A continuación, se presentan los resultados de este proceso a través del resumen con los estadísticos descriptivos de las cinco variables en la Tabla 3.2.

Estadístico	Voltaje Total (V)	Voltaje Prom. (V)	Equilibrio (%)	Corriente (A)	Temp. Prom. (°C)
cantidad	3047	3047	3047	3047	3047
media	129,36	2,15	90,65	0,69	28,71
des. est.	4,60	0,14	12,42	0,31	10,44
mínimo	110,04	0,00	0,00	- 1,79	25,50
25 %	126,89	2,11	91,89	0,68	26,70
50 %	130,71	2,18	93,38	0,71	28,10
75 %	133,58	2,23	94,39	0,73	29,00
máximo	136,04	2,27	97,42	6,36	217,30

Tabla 3.2. Resumen estadístico de las variables de estudio tras el remuestreo a tres horas y la imputación heurística de datos faltantes.

Al realizar la construcción de las series mediante las reglas mencionadas anteriormente a intervalos fijos de 3 horas, se reduce el tamaño de cada serie. Dada esta metodología, cabe destacar que en las series donde se usó la mediana como valor por ventana temporal, se produce implícitamente una eliminación de valores extremos y es por esto que cambian algunos estadísticos. Sin embargo, en la Tabla 3.2, se observa, por ejemplo, un valor negativo de corriente (-1.79 A), el cual puede significar en la práctica que el banco de baterías se está descargando, es decir, está entregando energía, o bien, puede ser un error de medición. De la misma manera, el máximo valor de la variable temperatura promedio (217.30 °C) supone un dato anómalo que debe ser revisado.

El siguiente paso es el tratamiento de valores atípicos (outliers) para terminar con esta etapa de pre-procesamiento. Ésto se realiza a través de diferentes métodos, tanto univariados como multivariados, los cuáles se presentan en la siguiente subsección.

3.1.3. Tratamiento de outliers

Se utiliza el método IQR para la detección y posterior tratamiento de valores atípicos ya que constituye una técnica no paramétrica, robusta y poco sensible a valores extremos, lo que lo hace especialmente adecuado para la estructura de nuestros datos. En particular, las series analizadas presentan fluctuaciones irregulares, distribuciones potencialmente no gaussianas y presencia de observaciones atípicas, las cuales en su mayoría son producto de errores de medición.

A diferencia de métodos paramétricos como el Z-score, cuyo rendimiento depende fuertemente del supuesto de normalidad, el método IQR no requiere ninguna hipótesis específica sobre la distribución de los datos. Esto es relevante en nuestro caso, ya que la presencia de asimetrías, colas pesadas o episodios aislados de variabilidad puede sesgar estimaciones basadas en la media y la desviación estándar.

En consecuencia, el uso del método IQR resulta apropiado como primera etapa de detección y limpieza de datos atípicos, contribuyendo a mejorar la calidad del conjunto de datos antes de aplicar los modelos de predicción. En la Figura 3.1 se presentan los gráficos, luego de la detección, eliminación de outliers y posterior reemplazo por la media global de cada serie.

Es importante señalar, dado que existen registros de solo dos fallas en el período estudiado, entonces se aplicó el proceso anterior a todo el período, menos en el mes de junio de 2024 para no quitar los eventos de fallas reales de los cuales se tiene conocimiento.

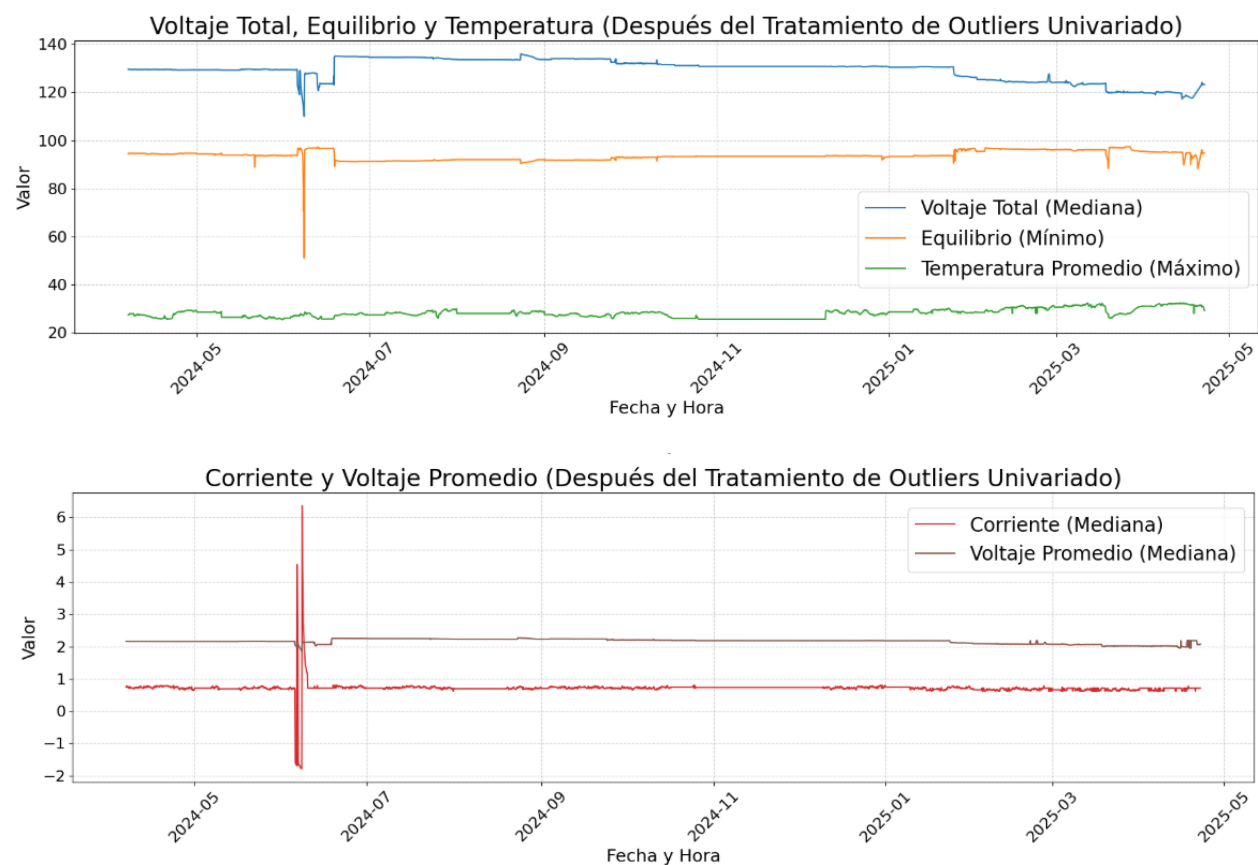


Figura 3.1. Gráficos de series superpuestas luego de tratamiento de outliers.

Se observa una dinámica simultánea de Voltaje Total y Equilibrio. Valores bajos de Voltaje Total coinciden con valores bajos de Equilibrio. Además, momentos de descarga (corriente negativa) se reflejan en caídas de voltaje.

Luego, se aplicó la distancia de Mahalanobis para detectar valores atípicos de manera multivariada, sin embargo, los únicos valores detectados, fueron justamente los que se encontraban en el mes de junio de 2024 y por ende no se eliminan ni reemplazan, mas bien, se conservan (ver Anexo C.1).

3.1.4. Tratamiento de datos faltantes: Filtro de Kalman

Por otro lado, de manera paralela se realizó un preprocesamiento diferente en orden y métodos. A las series remuestreadas a intervalos de 3 horas, se les aplicó en primera instancia un tratamiento de outliers mediante el método IQR y se eliminaron estas observaciones, dejándolas como datos faltantes. De esta manera, luego se puede aplicar el filtro de Kalman para imputación de datos faltantes, sin tener valores atípicos que puedan sesgar la estimación de los parámetros y por ende, se evita producir imputaciones y predicciones de baja calidad. Así, eliminar o marcar los outliers antes de la imputación reduce la influencia de valores extremos en la estimación dinámica.

A partir del marco teórico expuesto en la sección previa, la imputación de datos faltantes se implementa mediante una aplicación univariada del filtro de Kalman, realizada de forma independiente para cada una de las series temporales disponibles. En la práctica, se siguió el siguiente flujo reproducible en R: lectura del archivo CSV, conversión explícita a objeto serie de tiempo, diagnóstico de datos faltantes y finalmente imputación mediante un modelo estructural en espacio de estados, que emplea internamente estimación por máxima verosimilitud sobre el log-verosímil del filtro de Kalman para ajustar los parámetros del modelo y producir las imputaciones por suavizado. En cada caso se plantea un modelo lineal en espacio de estados que captura la dinámica temporal propia de la serie (componente de nivel y, cuando corresponde, de tendencia), y la función en R que devuelve la serie imputada utilizando estimaciones suavizadas $\hat{z}_{k|T}$; no obstante, dicha función no proporciona directamente las varianzas de las imputaciones ni las covarianzas del suavizamiento en su salida estándar, por lo que la cuantificación explícita de la incertidumbre asociada requiere el uso de implementaciones que devuelvan las matrices de covarianza (por ejemplo, `d1m` o `KFAS`) o procedimientos adicionales. Esta estrategia aprovecha la capacidad del filtro de Kalman para modelar la dinámica temporal y producir imputaciones coherentes en presencia de datos incompletos, facilitando análisis inferenciales y predictivos posteriores [15, 16]. Para más

detalles de este procedimiento ver código R en Anexo [C.3](#)

3.1.5. Extracción de características complementarias

Dentro del marco metodológico, se propone la idea de extraer características complementarias de las series, por ejemplo, pendientes de voltaje y corriente, resistencia interna aparente ($\Delta V/\Delta I$) y/o dimensión fractal para enriquecer la información de entrada. Las características complementarias condensan información física y dinámica de la batería que no aparece claramente en las series originales, y por ende, que los modelos podrían tardar más en inferir por sí mismos. Además, mejoran interpretabilidad, aumentan la potencia predictiva de modelos de machine learning, permiten detección temprana de degradación y reducen la sensibilidad al ruido cuando se diseñan apropiadamente.

Las pendientes/derivadas de las señales, miden la rapidez de cambio de las señales, que pueden ser la respuesta a impulsos de carga/descarga. Esto es importante, ya que cambios anómalos en las pendientes indican alteraciones en la dinámica electroquímica o en el control (por ejemplo, pérdida de capacidad activa, estratificación térmica). Por otro lado, la resistencia interna aparente, mide la oposición interna al flujo de corriente; incluye efectos óhmicos, de difusión y polarización. Esto es fundamental, ya que la resistencia interna aumenta con envejecimiento (pérdida de capacidad, sulfación, contactos pobres) y con temperatura/SOC, por lo que es un buen indicador del estado de salud (SoH). Por último, la dimensión fractal sirve como forma de medición de la complejidad y rugosidad temporal de la señal (cuánto “ruido estructurado” o irregularidad tiene la serie). Esto incide en que procesos degradativos y transitorios puedan cambiar la estructura temporal (por ejemplo, señales más “irregulares” cuando aparecen micro-fallas o conmutaciones internas).

Todas estas propuestas servirán como base para dar el siguiente paso de mejora de los modelos y avanzar hacia uno más robusto o multivariado. Se calculan y construyen cada una de estas características complementarias para dejarlas como base para trabajos futuros, las cuales quedan como anexo en este trabajo (ver Anexo [C.4.1](#)).

Para la evaluación de la estacionariedad de estas series se aplicaron de forma complementaria los tests Augmented Dickey–Fuller (ADF) y Kwiatkowski–Phillips–Schmidt–Shin (KPSS) siguiendo el protocolo descrito en la Sección [3.2](#). Las decisiones finales (valores de los estadísticos, p-valores y el veredicto combinado por serie) se presentan en el Anexo [C.4.2](#) (Tabla [C.1](#) y Tabla [C.3](#)).

3.2. Tests de estacionariedad: ADF y KPSS

La evaluación de la estacionariedad constituye un paso fundamental en el análisis de series temporales y condiciona la elección de modelos (por ejemplo, ARIMA vs. modelos alternativos). En este trabajo se emplean de forma complementaria el test de Dickey–Fuller Aumentado (ADF) y el test de Kwiatkowski–Phillips–Schmidt–Shin (KPSS), dado que abordan la hipótesis de estacionariedad desde hipótesis nulas opuestas y, por tanto, su uso conjunto aporta mayor robustez a la decisión previa a la modelación [3, 33, 34].

3.2.1. Test ADF (Dickey–Fuller Aumentado)

El test ADF parte de una especificación autorregresiva aumentada para $\Delta Y_t = Y_t - Y_{t-1}$ con la inclusión opcional de una constante y/o una tendencia determinística [33, 35]:

$$\Delta Y_t = \alpha + \beta t + \gamma Y_{t-1} + \sum_{i=1}^p \delta_i \Delta Y_{t-i} + \varepsilon_t, \quad (3.1)$$

donde t es el índice temporal, p es el número de rezagos incluidos para absorber autocorrelación en los residuos y ε_t es ruido blanco. Las tres especificaciones usuales son:

1. *sin constante ni tendencia* ($\alpha = \beta = 0$),
2. *con constante* ($\alpha \neq 0, \beta = 0$),
3. *con constante y tendencia* ($\alpha \neq 0, \beta \neq 0$).

La hipótesis nula del ADF es

$$H_0 : \gamma = 0 \quad (\text{existencia de raíz unitaria}),$$

mientras que la alternativa es $H_1 : \gamma < 0$ (estacionariedad). El estadístico de prueba es el estadístico t asociado al estimador de γ (usualmente denominado estadístico ADF). Dado que la distribución del estadístico bajo H_0 no es la t estándar, la decisión se toma frente a valores críticos especiales tabulados (o mediante valores- p numéricos) [33].

Para evitar autocorrelación en los residuos y problemas de tamaño del test, el número de rezagos p debe seleccionarse adecuadamente. Dos enfoques frecuentes son [36, 35]: *Reglas empíricas/heurísticas*, esto es, usar un lag máximo razonable en función del tamaño muestral

y seleccionar p por criterios de información u otros procedimientos; y/o *Criterios de información* (AIC, BIC) aplicados al modelo en (3.1). En la práctica se recomienda comprobar residuos (autocorrelación) tras la selección y ajustar p si es necesario [35, 36].

3.2.2. Test KPSS

El test KPSS contrasta la hipótesis nula de estacionariedad (alrededor de una media o de una tendencia determinística) [34]. Se parte de la descomposición:

$$Y_t = \mu_t + \varepsilon_t, \quad \mu_t = \mu_{t-1} + u_t,$$

donde ε_t es estacionario y u_t es un término de paseo aleatorio. La hipótesis nula del KPSS es

$$H_0 : \sigma_u^2 = 0 \quad (\text{serie estacionaria}),$$

y la alternativa implica $\sigma_u^2 > 0$ (no estacionariedad). El estadístico KPSS se construye a partir de la suma acumulada de residuos:

$$\text{KPSS} = \frac{1}{T^2 \hat{\sigma}_u^2} \sum_{t=1}^T S_t^2, \quad S_t = \sum_{i=1}^t \hat{\varepsilon}_i,$$

donde $\hat{\varepsilon}_i$ son residuos de una regresión de Y_t contra la constante (o contra constante y tendencia) y $\hat{\sigma}_u^2$ es una estimación consistente de la varianza de largo plazo del proceso ε_t , típicamente estimada por un estimador consistente de heterocedasticidad y autocorrelación, HAC (por sus siglas en inglés: Heteroskedasticity and Autocorrelation Consistent estimator), por ej. Newey–West/Andrews [34, 37, 38]. El estadístico se compara con valores críticos publicados por KPSS o por tablas posteriores.

La estimación de la varianza de largo plazo requiere elegir un parámetro de truncamiento o bandwidth. Se pueden utilizar reglas automáticas (por ej. basadas en la longitud de la muestra) o procedimientos de selección de banda óptima; en la práctica, paquetes estadísticos implementan reglas por defecto razonables, aunque conviene verificar sensibilidad a ese parámetro [39].

3.2.3. Interpretación conjunta y protocolo práctico

Debido a que ADF y KPSS tienen hipótesis nulas opuestas, su uso conjunto permite una clasificación más informativa. Se adopta el siguiente procedimiento operativo:

1. Elegir especificación apropiada (nivel vs. tendencia) para cada test de acuerdo con la inspección gráfica (media estable o tendencia) y el conocimiento de la variable.
2. Seleccionar parámetros: Para ADF, fijar un max_lag y seleccionar p mediante AIC/BIC o procedimiento secuencial; comprobar que los residuos del modelo no presenten autocorrelación significativa. Para KPSS, estimar la varianza de largo plazo con un estimador HAC y probar robustez frente a variantes del bandwidth.
3. Nivel de significancia: usar $\alpha = 0,05$ como criterio estándar (reportar también resultados a 1 % y 10 % cuando sea pertinente).
4. Decisión combinada:
 - ADF rechaza y KPSS no rechaza, entonces existe evidencia consistente de *estacionariedad*.
 - ADF no rechaza y KPSS rechaza, entonces existe evidencia consistente de *no estacionariedad*.
 - Ambos aceptan o ambos rechazan, entonces el resultado *contradictorio/incierto*: proceder con análisis complementarios (diferenciación, test de ruptura estructural, pruebas alternativas).

El ADF suele tener bajo poder frente a alternativas locales a la raíz unitaria y puede sufrir problemas de tamaño si los rezagos no son adecuados; el KPSS puede ser sensible a la serial correlation en los residuos si la varianza de largo plazo no se estima correctamente [35]. La presencia de rupturas en media o tendencia puede provocar fallos en la inferencia de ambos tests. En tales casos se recomienda aplicar tests que permitan rupturas (por ej. Zivot–Andrews) o incorporar términos de ruptura en el modelo antes de testear [39]. Si existe estacionalidad, considerar tests de raíces unitarias estacionales (por ej. HEGY) o remover la estacionalidad previa al test.

Este enfoque combinado se utiliza en este trabajo para evaluar la estacionariedad de cada variable antes de la etapa de modelación, tal como se resume en la Tabla 4.1.

Capítulo 4

Resultados y análisis

En este capítulo se presentan los resultados derivados de la aplicación de los modelos y métodos definidos en el marco teórico. Se trabajará con las series resampleadas a paso temporal de 3 horas luego del preprocesamiento, es decir, aplicado el tratamiento de outliers y por separado ambos métodos de imputación (heurística y filtro de Kalman). En primer lugar, para cada variable se incluyen los gráficos de autocorrelación (ACF) y autocorrelación parcial (PACF) de las series preprocesadas, seguidos de las gráficas de predicción (valores observados vs. predichos) para los horizontes considerados. Se reportan además las métricas de error más relevantes (por ejemplo, MAE, RMSE y MAPE) y se realiza una comparación cuantitativa y cualitativa del rendimiento entre los enfoques estudiados (modelos estadísticos clásicos y modelos de aprendizaje automático). Complementariamente, se describe el proceso de automatización empleado para la selección y validación de modelos (búsqueda de hiperparámetros, validación cruzada temporal y pipelines reproducibles) y se discuten aspectos prácticos de puesta en marcha: generación automática de pronósticos, integración en flujos de trabajo y monitorización del desempeño en producción. Las implementaciones se desarrollaron en Python y RStudio.

4.1. Gráficos de ACF y PACF

En primer lugar, se muestran los gráficos de autocorrelación (ACF) y autocorrelación parcial (PACF) de las series temporales preprocesadas para cada variable. Estos diagramas permiten evaluar la dependencia temporal y sirven como guía inicial para la identificación de órdenes en modelos ARIMA/ARMA. Las figuras correspondientes se incluyen en las Figuras 4.1 - 4.5.

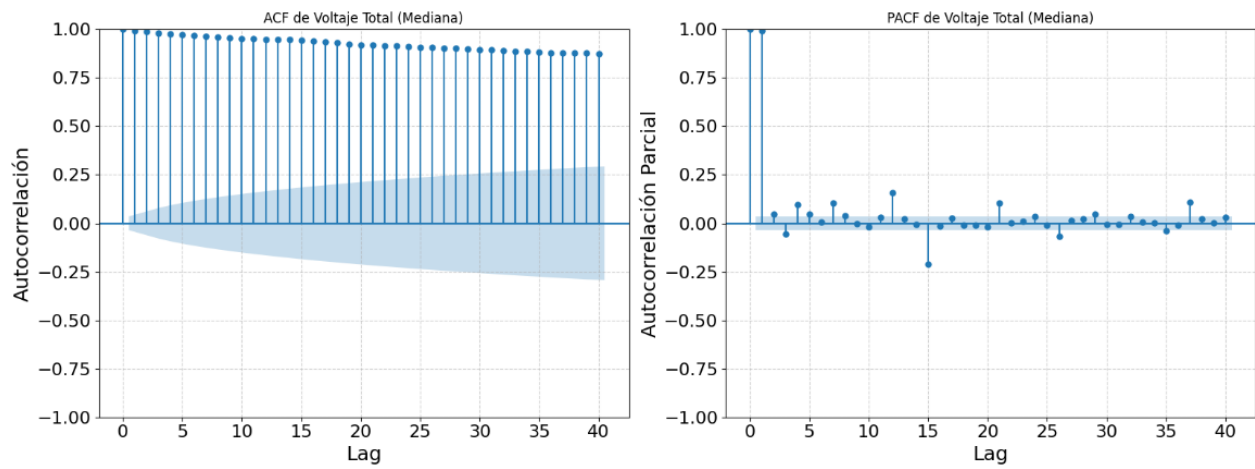


Figura 4.1. ACF y PACF de Serie Voltaje Total.

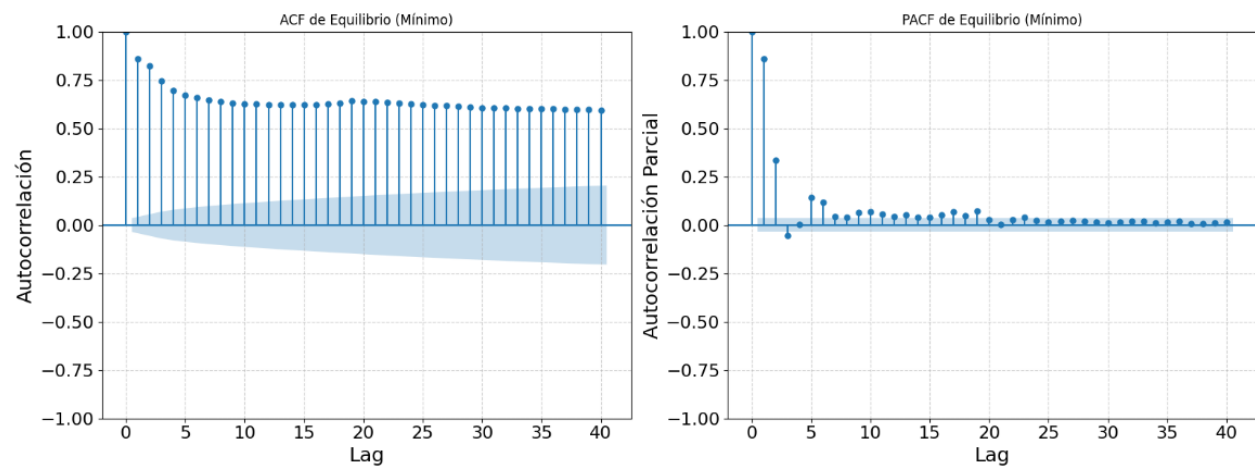


Figura 4.2. ACF y PACF de Serie Equilibrio.

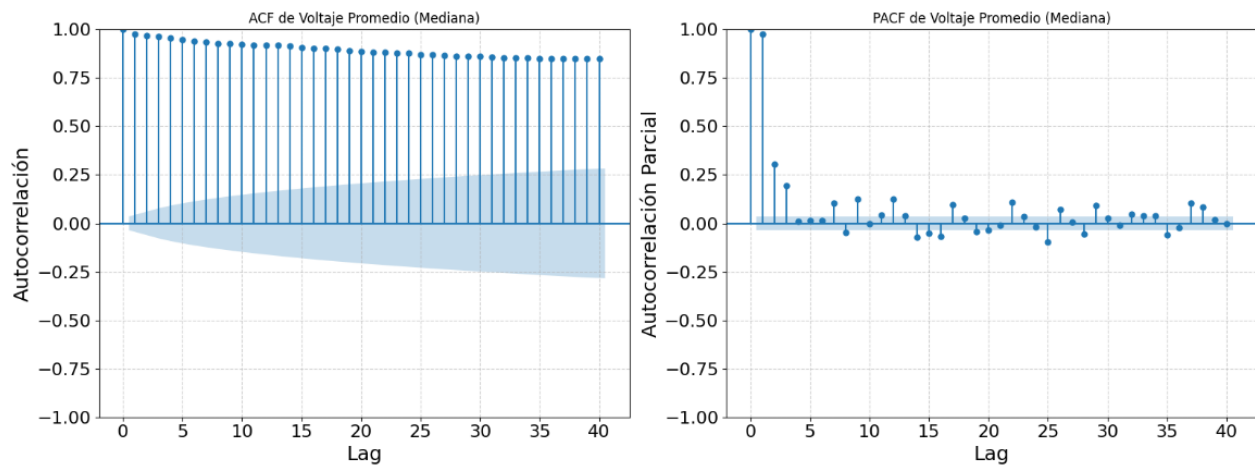


Figura 4.3. ACF y PACF de Serie Voltaje Promedio.

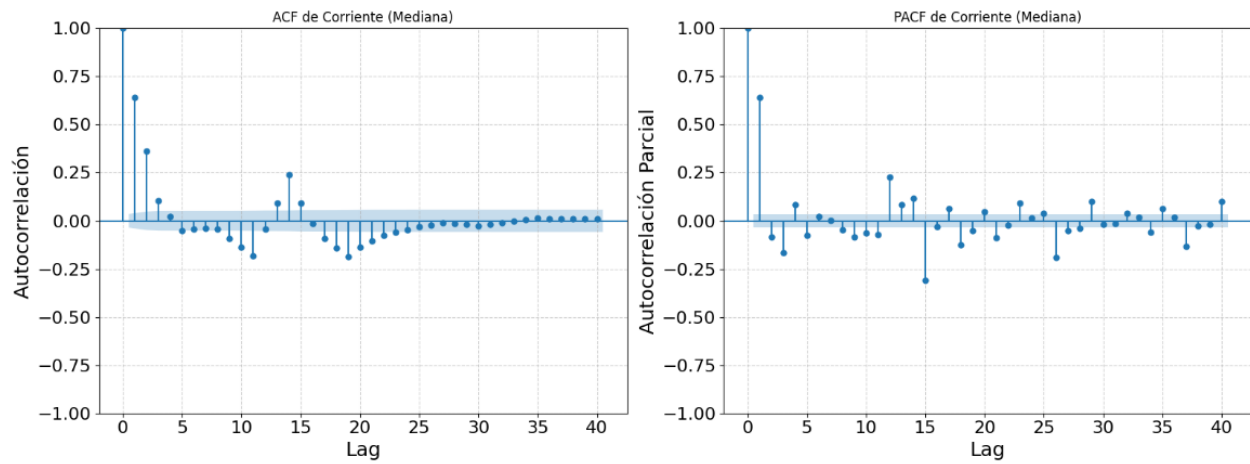


Figura 4.4. ACF y PACF de Serie Corriente.

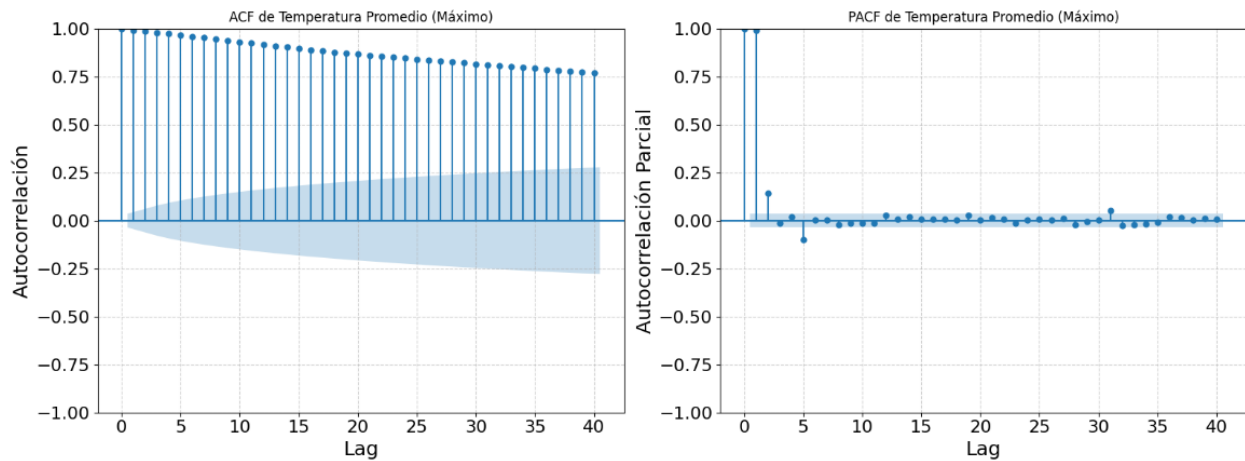


Figura 4.5. ACF y PACF de Serie Temperatura Promedio.

Los gráficos de autocorrelación (ACF) muestran una persistencia elevada en todas las variables analizadas, con coeficientes significativamente distintos de cero incluso para rezagos altos, lo que evidencia un decaimiento lento de la dependencia temporal. Este comportamiento es característico de series no estacionarias y sugiere la presencia de memoria de largo plazo. La única que puede sospecharse podría ser estacionaria es la serie de Corriente, ya que presenta una rápida caída hacia 0 en su gráfica de autocorrelación, sin embargo, presenta algunas fluctuaciones.

Por su parte, los gráficos de autocorrelación parcial (PACF) presentan valores dominantes en el primer rezago y ausencia de un corte claro posterior, lo que descarta una estructura autorregresiva de bajo orden. En conjunto, estos resultados indican la sospecha de que los supuestos de estacionariedad requeridos por modelos clásicos como ARMA no se cumplen. Sin embargo, esto no es suficiente para concluir no estacionariedad de series temporales, se requiere realizar test de hipótesis que sustente esta conclusión.

Para evaluar la estacionariedad de las series analizadas se aplicaron de forma complementaria los tests Augmented Dickey–Fuller (ADF) y Kwiatkowski–Phillips–Schmidt–Shin (KPSS), siguiendo el procedimiento descrito en detalle en el Capítulo de Metodología (Sección 3.2). Dado que ambos tests plantean hipótesis nulas opuestas (el ADF contrasta la existencia de una raíz unitaria y el KPSS la estacionariedad alrededor de una media o tendencia), como se mencionó antes, su uso conjunto ofrece una base más robusta para la decisión sobre transformar o no cada serie antes de la modelación [3, 33, 34], o bien para considerar modelos alternativos. A continuación se presentan los resultados empíricos obtenidos para cada variable y la interpretación combinada de ambos tests.

Serie	ADF	p-ADF	lags	KPSS	p-KPSS	lags	Decisión combinada
Voltaje total	-1.486	0.5405	28	4.4240	0.0100	32	No estacionaria
Equilibrio	-3.226	0.0185	22	3.4782	0.0100	30	Contradictorio / incierto
Temperatura prom.	-3.215	0.0191	4	3.3223	0.0100	32	Contradictorio / incierto
Corriente	-12.347	0.0000	28	0.2607	0.1000	16	Estacionaria
Voltaje prom.	-2.093	0.2474	28	4.1990	0.0100	32	No estacionaria

Tabla 4.1. Resumen de resultados de los tests de estacionariedad (ADF y KPSS).

Los resultados resumidos en la Tabla 4.1 muestran comportamientos heterogéneos entre las series. Por ejemplo, las series de voltaje total y voltaje promedio presentan evidencia consistente de no estacionariedad, mientras que la serie de corriente resulta claramente estacionaria (ADF rechaza raíz unitaria y KPSS no rechaza estacionariedad), como se sospechaba previamente. Las series de equilibrio y temperatura promedio presentan resultados contradictorios entre ADF y KPSS, por lo que su situación requiere una inspección adicional, por ejemplo, diferenciación, comprobación de rupturas estructurales o tests alternativos.

Si bien modelos ARIMA podrían considerarse mediante diferenciación, la fuerte dependencia temporal observada y la ausencia de una estructura parsimoniosa hacen que estos enfoques resulten poco adecuados para describir la dinámica subyacente de las variables, la cual no tienen un patrón claro, dado que las series son complejas y caóticas. Por lo tanto, se requiere el uso de herramientas o modelos de otro tipo para poder realizar predicciones realistas. Estas características motivan el uso de enfoques alternativos capaces de capturar dependencias complejas y no lineales en el tiempo, como las redes neuronales recurrentes del tipo LSTM. Sin embargo, se usa el modelo de Holt y Holt-Winters como punto de partida para luego compararlos con modelos de redes neuronales artificiales LSTM. En la siguiente sección se presentan los resultados obtenidos al aplicar estos métodos a las series, así como las métricas de error en las predicciones para efectos de comparación.

4.2. Gráficos de predicciones

4.2.1. Predicciones con tratamiento heurístico

Para cada una de las series ya preprocesadas y tratadas, se ajustaron los modelos clásicos de Holt, Holt-Winters y de redes neuronales LSTM, donde se separó la data; entrenamiento: abril-diciembre 2024; validación: enero-abril 2025. El horizonte de predicción y validación se fija de enero a abril 2025, de manera de poder comparar los valores reales en estas fechas con los predichos por los modelos. Los resultados para Holt y Holt-Winters se muestran en las Figuras 4.6 - 4.10. Además, en las Tablas 4.2 - 4.6 se muestran algunos elementos asociados al ajuste de los modelos en cada caso por variable de estudio, entre ellos, SSE (Suma de Residuos al Cuadrado), AICC (Criterio de Información de Akaike Corregido) y los parámetros estimados, los cuales están definidos en las ecuaciones 2.8 - 2.16.

Modelo	α	β	γ	L_0	T_0	SSE	AICC
Holt (aditivo)	0.9413	0.0000	—	129.6081	0.0005	708.8825	-2385.8994
Holt-Winters (aditivo)	0.9416	0.0000	0.0000	129.6345	0.0005	707.0494	-2375.3195
Holt-Winters (multiplicativo)	0.9415	0.0000	0.0000	129.5340	0.0000	707.4917	-2373.9723

Tabla 4.2. Resumen de ajuste – Voltaje Total (Mediana)

Modelo	α	β	γ	L_0	T_0	SSE	AICC
Holt (aditivo)	0.5952	0.0000	—	94.5606	-0.0006	2578.6669	395.6428
Holt-Winters (aditivo)	0.5956	0.0000	0.0000	94.6364	-0.0006	2569.3992	404.0446
Holt-Winters (multiplicativo)	0.5948	0.0000	0.0000	94.6371	-0.0006	2573.6706	407.6225

Tabla 4.3. Resumen de ajuste – Equilibrio (Mínimo)

Modelo	α	β	γ	L_0	T_0	SSE	AICC
Holt (aditivo)	0.9249	0.0000	—	2.1607	0.000008	0.1946	-20050.0234
Holt-Winters (aditivo)	0.9249	0.0000	0.0000	2.1604	0.000008	0.1942	-20038.4171
Holt-Winters (multiplicativo)	0.9247	0.0000	0.0000	2.1598	0.000008	0.1943	-20037.2976

Tabla 4.4. Resumen de ajuste – Voltaje Promedio (Mediana)

Modelo	α	β	γ	L_0	T_0	SSE	AICC
Holt (aditivo)	0.8710	0.0000	–	0.7569	0.000008	135.9509	-5943.0066
Holt–Winters (aditivo)	0.8710	0.0000	0.0000	0.7679	0.000009	135.4205	-5935.2695

Nota: Holt–Winters (multiplicativo) omitido: serie contiene valores no positivos.

Tabla 4.5. Resumen de ajuste – Corriente (Mediana)

Modelo	α	β	γ	L_0	T_0	SSE	AICC
Holt (aditivo)	1.0000	0.0000	–	27.4994	0.00056	61.2193	-7661.5339
Holt–Winters (aditivo)	1.0000	0.0000	0.0000	27.5323	0.00056	60.9231	-7655.8260
Holt–Winters (multiplicativo)	1.0000	0.0000	0.0000	27.8483	0.00057	60.9203	-7655.9236

Tabla 4.6. Resumen de ajuste – Temperatura Promedio (Máximo)

Los resultados de ajuste muestran un patrón consistente: el esquema de suavizamiento simple de Holt (tendencia aditiva) proporciona, en la práctica, un ajuste tan bueno o mejor que las variantes Holt–Winters para la mayoría de las series consideradas, según el criterio AICC. Específicamente, para *Voltaje Total* las diferencias en SSE entre Holt (708.88) y Holt–Winters (≈ 707) son pequeñas, pero el AICC favorece la parsimonia del modelo Holt; para *Voltaje Promedio* y *Corriente* los valores de SSE son muy reducidos (0.19 y 135.95, respectivamente) y el AICC vuelve a preferir la formulación Holt; en *Temperatura Promedio* los ajustes presentan SSE ≈ 61 y la selección por AICC también favorece Holt. La variable *Equilibrio* exhibe errores acumulados notablemente mayores (SSE ≈ 2.570) y AICC positivos, lo que indica un ajuste relativamente pobre frente a las demás series y sugiere mayor complejidad o ruido intrínseco en esa señal.

Desde la inspección de parámetros, varios rasgos son comunes, los valores de α (suavizado de nivel) tienden a ser altos (0.59–1.00), lo que implica que los modelos otorgan un peso considerable a las observaciones recientes. En todos los ajustes $\beta \approx 0$, indicando que la tendencia estimada es prácticamente nula o que no es identificable con el nivel de información disponible. Asimismo, $\gamma \approx 0$ en las versiones Holt–Winters, lo que sugiere que no se detecta una estacionalidad relevante en la granularidad analizada (paso temporal de 3h). Las condiciones iniciales (L_0 , T_0) reflejan los niveles operativos de cada variable (por ej. $L_0 \approx 129$ para *Voltaje Total*, $L_0 \approx 2,16$ para *Voltaje Promedio*, $L_0 \approx 0,76$ para *Corriente*), y en varios casos T_0 es cercano a cero, coherente con $\beta \approx 0$.

En consecuencia, se procede a mostrar las predicciones generadas por los modelos seleccionados para cada variable.

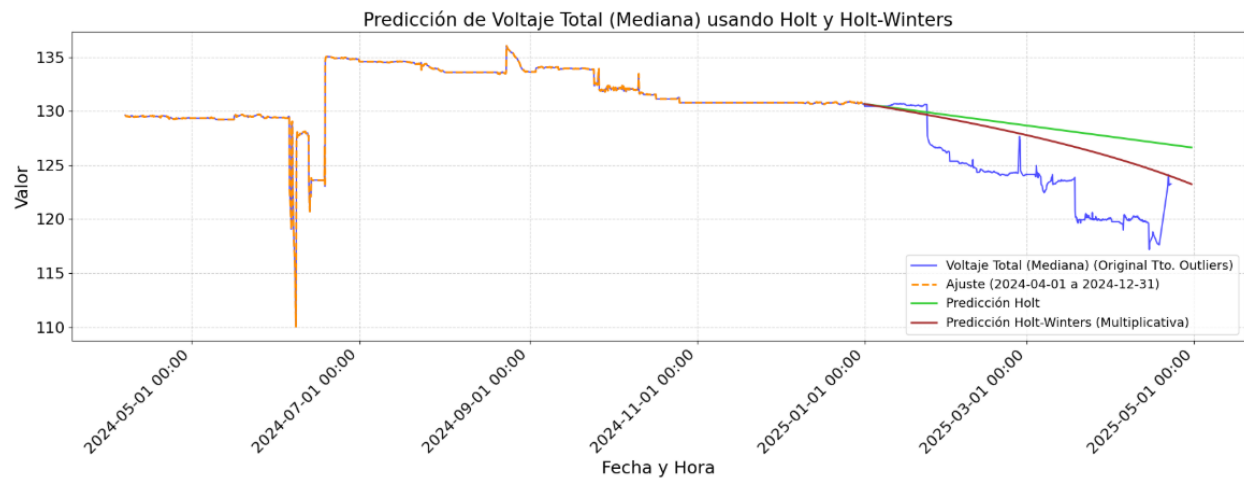


Figura 4.6. Resultados de la aplicación de Métodos de Suavizamiento Exponencial para Serie Voltaje Total.

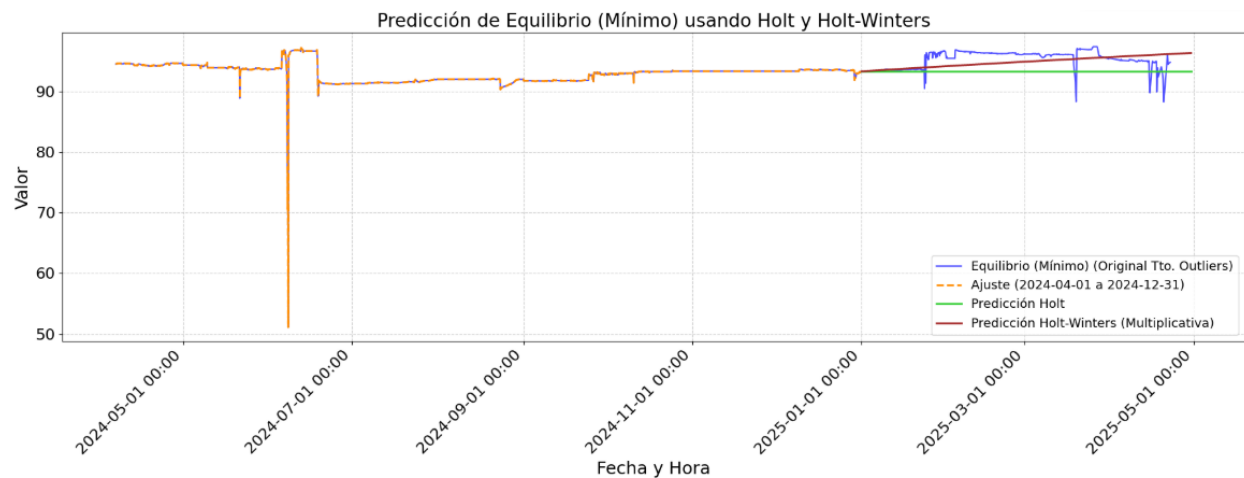


Figura 4.7. Resultados de la aplicación de Métodos de Suavizamiento Exponencial para Serie Equilibrio.

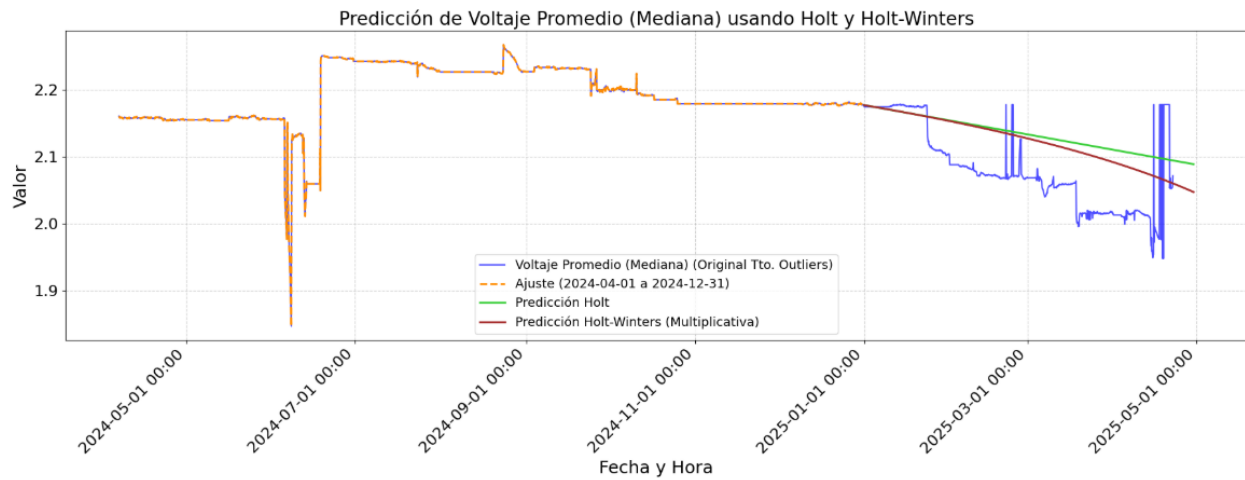


Figura 4.8. Resultados de la aplicación de Métodos de Suavizamiento Exponencial para Serie Voltaje Promedio.

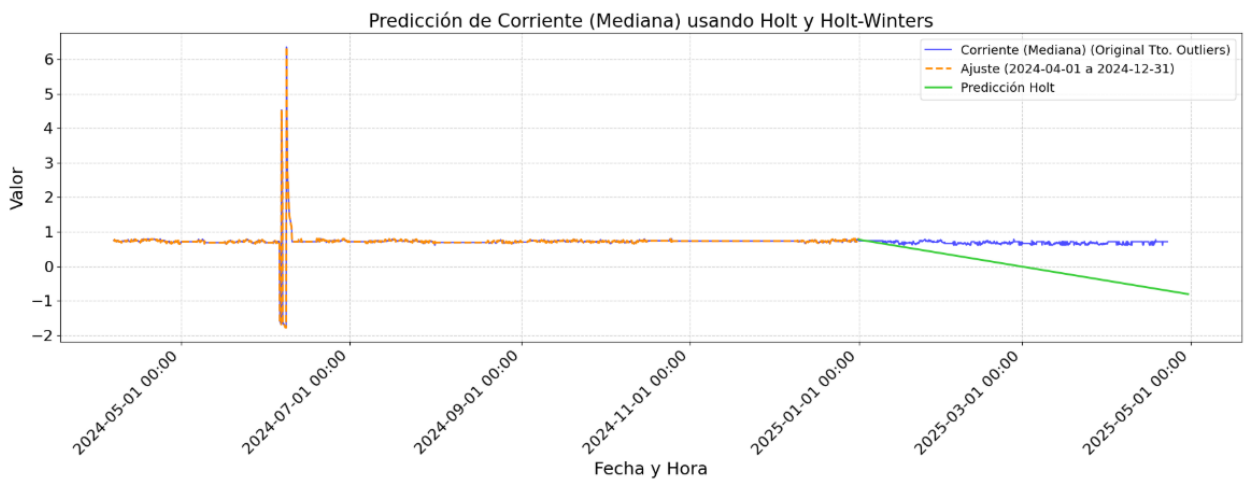


Figura 4.9. Resultados de la aplicación de Métodos de Suavizamiento Exponencial para Serie Corriente.

En esta serie realmente no tiene sentido aplicar el método de Holt-Winters, ya que los valores negativos están fuera del rango de normalidad.

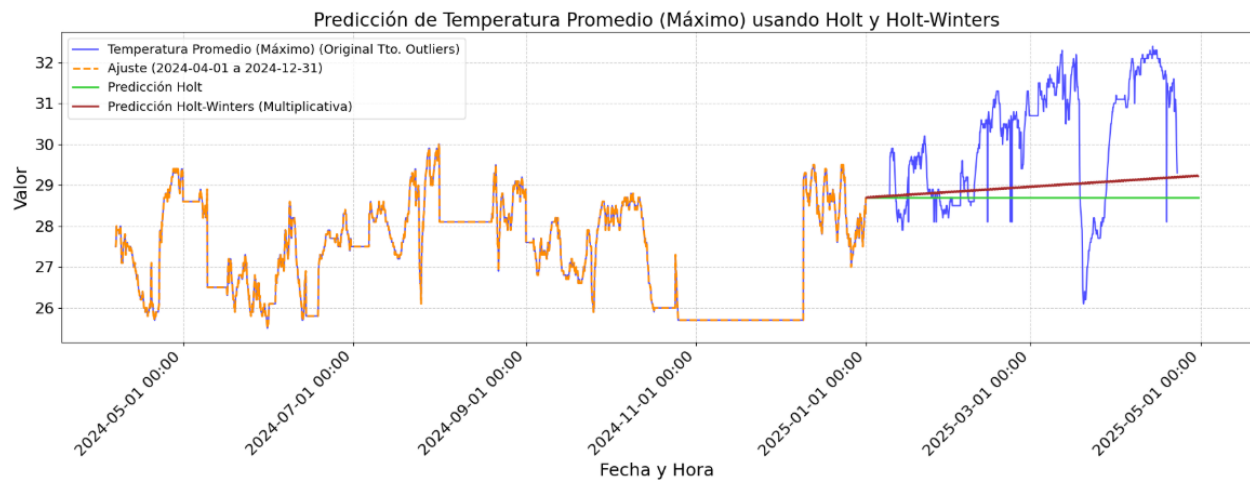


Figura 4.10. Resultados de la aplicación de Métodos de Suavizamiento Exponencial para Serie Temperatura.

Los gráficos, muestran una clara tendencia en las predicciones, lo cual no es interpretable a largo plazo, pues las series realmente no poseen este elemento. Se puede ver que este tipo de modelo, suaviza la curva, lo que no es representativo del comportamiento ruidoso y complejo de las series.

En la Tabla 4.7 se muestran los resultados de las métricas de error, luego de aplicar los métodos de suavizamiento exponencial a las series, en sus versiones aditiva y multiplicativa.

4.2.2. Resultados métricas de error

Variable	Modelo	RMSE	MAE	MAPE (%)
Voltaje Total (Mediana)	Holt	5.2763	4.5456	3.7252
Voltaje Total (Mediana)	Holt–Winters (Aditiva)	4.1258	3.6044	2.9485
Voltaje Total (Mediana)	Holt–Winters (Multiplicativa)	4.0760	3.5624	2.9139
Equilibrio (Mínimo)	Holt	2.4865	2.1770	2.2695
Equilibrio (Mínimo)	Holt–Winters (Aditiva)	1.6533	1.2781	1.3372
Equilibrio (Mínimo)	Holt–Winters (Multiplicativa)	1.5722	1.2105	1.2697
Voltaje Promedio (Mediana)	Holt	0.0706	0.0621	3.0291
Voltaje Promedio (Mediana)	Holt–Winters (Aditiva)	0.1588	0.1371	6.7047
Voltaje Promedio (Mediana)	Holt–Winters (Multiplicativa)	0.0625	0.0551	2.6798
Corriente (Mediana)	Holt	0.7766	0.6543	96.5784
Temperatura Promedio (Máximo)	Holt	1.9394	1.5709	5.0930
Temperatura Promedio (Máximo)	Holt–Winters (Aditiva)	1.7226	1.4233	4.6359
Temperatura Promedio (Máximo)	Holt–Winters (Multiplicativa)	1.7207	1.4220	4.6319

Tabla 4.7. Resumen de métricas de error (RMSE, MAE, MAPE) para cada variable y modelo.

Los estadísticos presentados entre paréntesis en la Tabla 4.7 (mediana, mínimo y máximo) indican la forma en que se construyeron las series durante el proceso de remuestreo a intervalos de 3 horas. En el caso de la serie de corriente, no se aplicó el método de Holt-Winters, ni en su versión aditiva ni multiplicativa. Esto se debe a que, en el enfoque aditivo, la ponderación de valores negativos carece de sentido físico, mientras que en el enfoque multiplicativo la presencia de valores iguales a cero puede anular el modelo. Los resultados obtenidos deben interpretarse a la luz del significado y la definición de cada métrica considerada.

En la Tabla 4.8 se presentan los resultados de las métricas de error para predicciones usando el modelo LSTM con las series tratadas con imputación heurística para comparación. Para mayor detalle, ver gráficos en Anexo D.2.

Es importante mencionar, que hasta este punto, todas las predicciones se han realizado en series imputadas en base a reglas heurísticas, considerando estadísticos representativos de los datos, como el mínimo y el máximo. Sin embargo, estos no son los métodos más sofisticados para la preparación de los datos.

Variable	RMSE	MAE
Voltaje total	0.5593	0.3800
Equilibrio	0.6078	0.2714
Voltaje promedio	0.0207	0.0079
Corriente	0.0323	0.0234
Temperatura promedio	0.3293	0.1348

Tabla 4.8. Métricas de evaluación del modelo LSTM (RMSE y MAE) por variable.

Se considera ahora, un método más robusto para la imputación de datos faltantes, con el objetivo de mejorar el modelamiento de la series y por lo tanto, mejorar las predicciones de los modelos de aprendizaje automático. Este es, el filtro de Kalman, el cuál modela la dinámica de series con ruido gaussiano, por lo que es aplicable a nuestro caso. Si bien, a priori no se sabe la distribución del ruido que tenemos en cada de nuestras señales en las variables eléctricas, se mantiene la aleatoriedad y posteriormente, se prueba el efecto a través de las métricas de rendimiento que se definieron en el Capítulo 2, específicamente en las ecuaciones 2.24 - 2.27.

4.2.3. Predicciones post filtro de Kalman

El modelo LSTM empleado para cada serie consta de una única capa LSTM con 50 unidades seguida de una capa densa de salida con una única neurona. El optimizador utilizado fue Adam y la función de pérdida MSE. El entrenamiento se realizó durante 50 épocas con batch size de 32. El número total de parámetros del modelo es aproximadamente 10 451 (10 400 para la capa LSTM y 51 para la capa densa). Se entrenó un modelo independiente por cada serie y los modelos resultantes se almacenan en un diccionario en memoria.

A continuación, en las figuras 4.11 a la 4.14 se muestran los resultados de las predicciones usando el modelo LSTM para nuestras series tratadas con el filtro de Kalman para la imputación de datos faltantes.

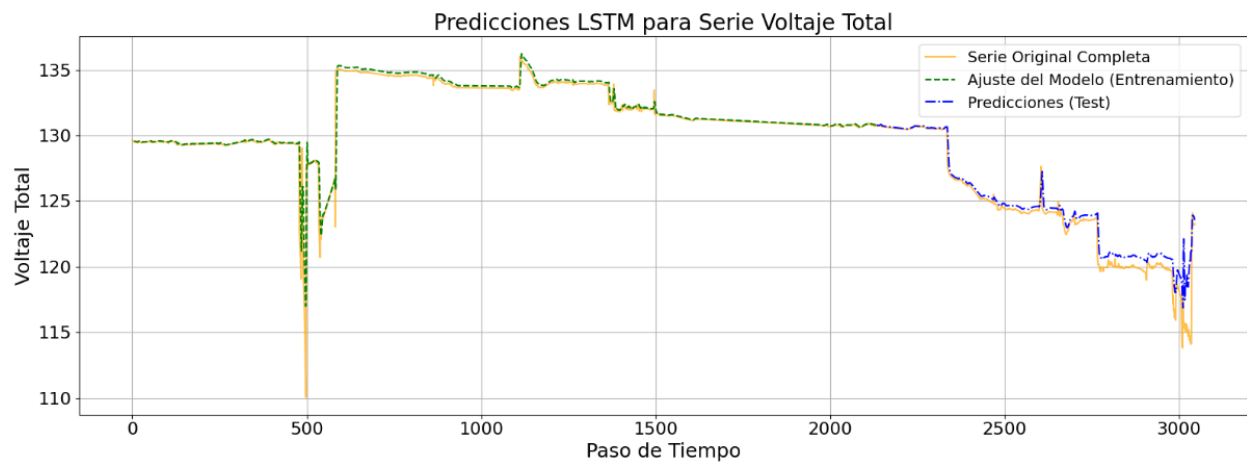


Figura 4.11. Resultados de la aplicación de Modelo LSTM para serie Voltaje Total.

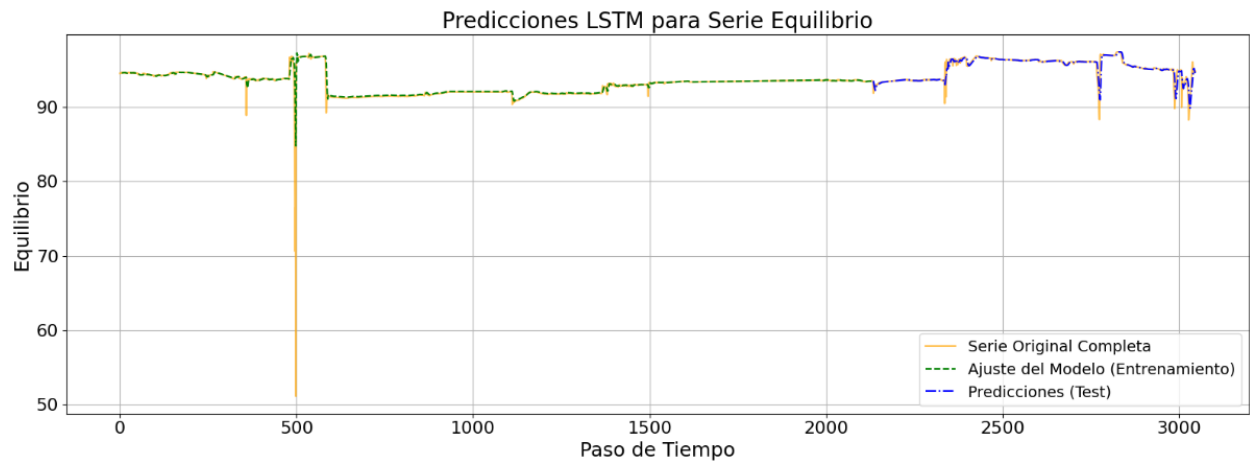


Figura 4.12. Resultados de la aplicación de Modelo LSTM para serie Equilibrio.

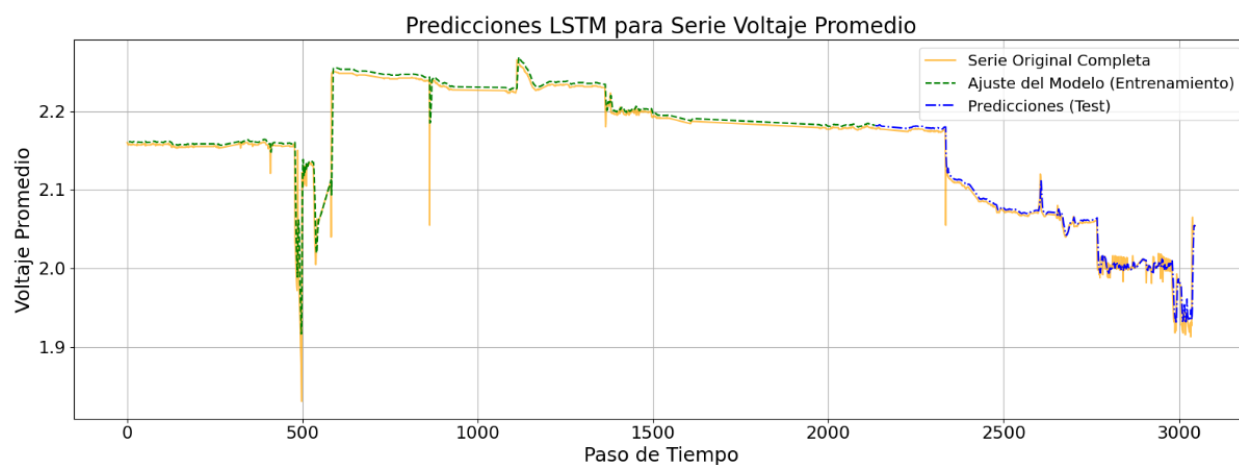


Figura 4.13. Resultados de la aplicación de Modelo LSTM para serie Voltaje Promedio.

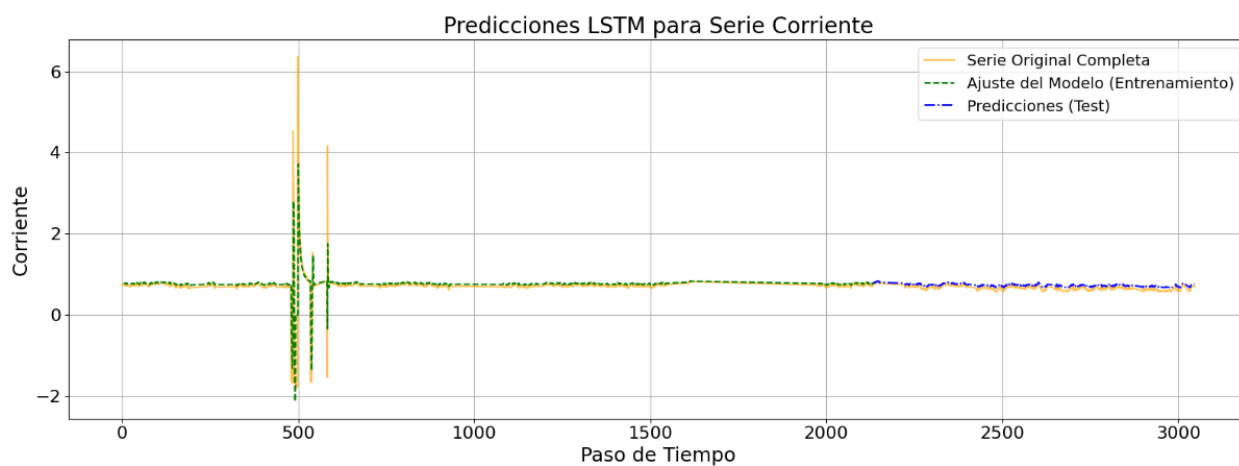


Figura 4.14. Resultados de la aplicación de Modelo LSTM para serie Corriente.

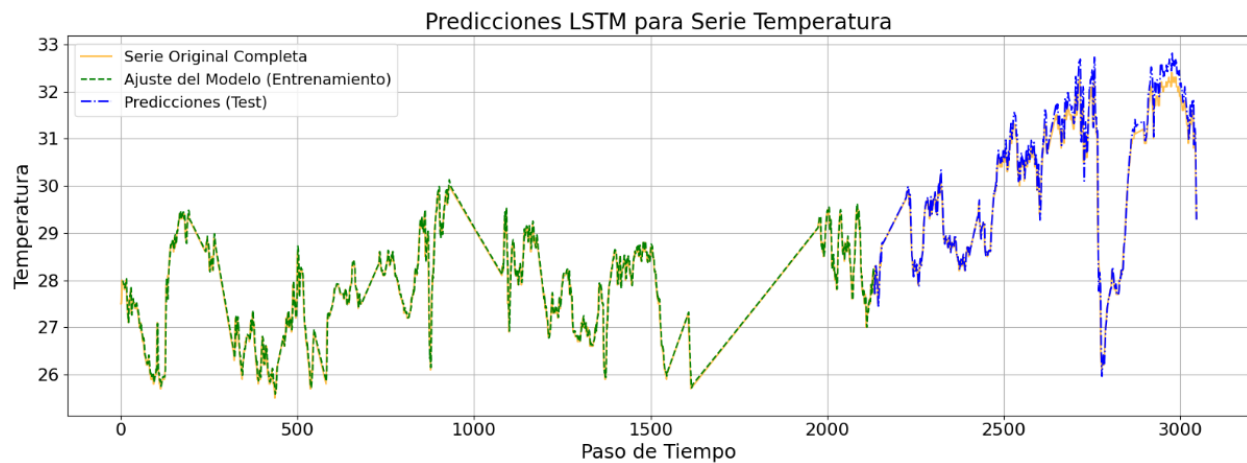


Figura 4.15. Resultados de la aplicación de Modelo LSTM para serie Temperatura Promedio.

Los resultados visuales obtenidos a partir de los cinco modelos LSTM evidencian un desempeño sólido y de alta calidad predictiva. En términos generales, los modelos presentan una elevada fidelidad de ajuste tanto en la etapa de entrenamiento como en la de validación, siendo capaces de reproducir adecuadamente series con comportamiento estable, tendencias suaves y dinámicas más complejas, incluyendo la presencia de anomalías y cambios abruptos, como los observados en la variable corriente. Este comportamiento confirma la robustez del enfoque propuesto y la capacidad de memoria de las redes LSTM para capturar dependencias temporales de corto y largo plazo, incluso en contextos caracterizados por no linealidades y variabilidad irregular.

Como limitación visual recurrente, se observa un leve desfase temporal (lag) de uno o pocos instantes en la respuesta del modelo frente a cambios extremadamente bruscos o discontinuidades repentinas en las series reales, fenómeno común en modelos de predicción basados en ventanas temporales. No obstante, este efecto no compromete de manera significativa la calidad global de las predicciones ni la estabilidad del modelo. En conjunto, los resultados respaldan al modelo LSTM como una herramienta precisa, robusta y adecuada para la modelación y predicción de las series temporales analizadas en este estudio.

4.2.4. Comparación de rendimiento entre modelos

La evaluación del desempeño de los modelos desarrollados constituye un paso esencial para determinar cuál de ellos ofrece las predicciones más precisas y estables en el análisis del comportamiento de las variables eléctricas del banco de baterías. Con este propósito, en esta subsección se presenta una comparación sistemática del rendimiento entre los distintos modelos, empleando

como métrica principal el Root Mean Squared Error (RMSE). Finalmente, se discuten las implicancias de estos resultados en la elección del modelo más adecuado para su implementación en el sistema automatizado de monitoreo y diagnóstico.

Los porcentajes de disminución se calculan como

$$\text{reducción (\%)} = \frac{\text{RMSE}_{\text{antes}} - \text{RMSE}_{\text{actual}}}{\text{RMSE}_{\text{antes}}} \times 100.$$

La Tabla 4.9 muestra la comparación entre los valores de RMSE obtenidos previamente con los modelos clásicos y los valores alcanzados mediante la implementación del modelo LSTM, ambos con tratamiento de datos faltantes por medio del método heurístico realizado. En todos los casos, se observa una disminución significativa del error, lo que evidencia una mejora sustancial en la capacidad predictiva del nuevo modelo. Esta reducción en el RMSE indica que las predicciones generadas por la arquitectura LSTM se ajustan con mayor precisión a las observaciones reales, particularmente en variables que presentan alta variabilidad temporal o dinámicas más complejas, como la corriente, la temperatura y el voltaje total. En particular, destaca la disminución drástica del RMSE en *Voltaje Total* (de 4.0760 a 0.5593, $\approx 86.3\%$ de disminución) y en *Temperatura* (de 1.7207 a 0.3293, $\approx 80.9\%$ de disminución), lo que sugiere que el LSTM captura adecuadamente las dependencias temporales asociadas al comportamiento térmico y al balance general del banco de baterías. Asimismo, para *Corriente* el error se reduce de 0.7766 a 0.0323 (un 95.8% de disminución), indicando que el LSTM es especialmente eficaz modelando dinámicas rápidas y no lineales. En el caso de *Voltaje Promedio*, el RMSE pasa de 0.0625 a 0.0207 ($\approx 66.9\%$ de disminución), mostrando que el modelo es capaz de capturar incluso variaciones sutiles en señales de bajo ruido. Por último, la reducción del RMSE en la variable *Equilibrio* (de 1.5722 a 0.6078, $\approx 61.3\%$ de disminución) indica un ajuste significativamente más preciso, aun cuando esta variable presenta un comportamiento más complejo y menos estable.

Variable	RMSE (Holt-Winters)	RMSE (LSTM)
Voltaje total	4.0760	0.5593
Equilibrio	1.5722	0.6078
Voltaje p.	0.0625	0.0207
Corriente	0.7766	0.0323
Temperatura	1.7207	0.3293

Tabla 4.9. Resultados RMSE con imputación heurística.

Estos porcentajes confirman una mejora notable en el desempeño predictivo del LSTM respecto a los modelos clásicos, siendo especialmente relevante la reducción en variables críticas para la operación del banco de baterías (Corriente y Voltaje Total).

La Tabla 4.10 compara el RMSE obtenido con la imputación heurística y con el filtro de Kalman por variable para el modelo LSTM. El filtro de Kalman ofrece mejoras notables en variables como Temperatura (reducción del RMSE $\approx 50\%$) y también mejora modestamente Voltaje promedio y Corriente ($\approx 24\%$ y $\approx 9\%$ respectivamente). Sin embargo, en Voltaje total y Equilibrio la imputación heurística muestra un RMSE inferior ($\approx 21\%$ y $\approx 5\%$ mejor que Kalman). Estas diferencias sugieren que el Kalman resulta muy eficaz en señales suaves y con dinámica lineal y ruido Gaussiano (por ej. temperatura), mientras que la heurística puede preservar mejor discontinuidades u otros efectos no lineales o cambios de régimen presentes en algunas variables eléctricas. Para validar la significancia de estas diferencias es necesario comparar los errores observación a observación (por ejemplo mediante test emparejados) y analizar la estructura de residuos; además es recomendable ajustar los parámetros de covarianza del filtro (matrices Q y R) o probar variantes robustas/no lineales del filtro si se desea mejorar su rendimiento en aquellas series donde actualmente rinde peor. Sin embargo, dado los resultados, se sugiere trabajar con las series que obtienen las mejores métricas de rendimiento, independiente del método de imputación al llevar el proyecto a producción.

Variable	RMSE (Imputación heurística)	RMSE (Filtro de Kalman)
Voltaje total	0.5593	0.6764
Equilibrio	0.6078	0.6372
Voltaje p.	0.0207	0.0157
Corriente	0.0323	0.0294
Temperatura	0.3293	0.1634

Tabla 4.10. Resultados RMSE imputación heurística vs filtro Kalman para modelo LSTM.

En conjunto, los resultados demuestran que la arquitectura LSTM supera ampliamente a los modelos clásicos empleados previamente en términos de precisión predictiva bajo la métrica RMSE. Su capacidad para modelar dependencias temporales no lineales y capturar patrones complejos en las variables eléctricas del banco de baterías permite obtener predicciones más estables y confiables. Esto valida el uso del enfoque de aprendizaje profundo como herramienta idónea para el seguimiento avanzado del comportamiento del sistema. Finalmente, con esto se valida la hipótesis de investigación de este trabajo, por lo que los modelos de aprendizaje automático proporcionan

métricas de rendimiento mejores que los modelos clásicos para predicción de series temporales en nuestro contexto. Además, un correcto preprocesamiento reduce aún más los errores en las predicciones en la mayoría de las variables.

4.3. Automatización y puesta en marcha

La automatización del proceso de modelamiento constituye una etapa fundamental para garantizar la reproducibilidad, escalabilidad y operación continua de los métodos desarrollados en este estudio. Una vez definidos y validados los modelos estadísticos y de aprendizaje automático, es necesario integrarlos en un flujo sistemático que permita su ejecución periódica, la actualización de sus entradas y la generación automática de resultados. Este proceso asegura que los modelos puedan adaptarse a nuevos datos, mantener su desempeño en el tiempo y operar sin intervención manual significativa.

En las figuras 4.16 y 4.17 se describe la arquitectura implementada para la automatización de los modelos, detallando los componentes encargados de la adquisición y preprocesamiento de datos, la ejecución de los algoritmos de estimación y predicción, la detección temprana de anomalías, y los procedimientos de evaluación continua. Asimismo, se presenta el proceso de puesta en marcha, que incluye la configuración de los entornos de ejecución, la integración con sistemas externos y los protocolos diseñados para el monitoreo y mantenimiento del sistema en producción.

El objetivo de esta etapa es garantizar que los modelos desarrollados no solo funcionen correctamente en un entorno experimental, sino que puedan operar de forma robusta, eficiente y confiable en un contexto real, proporcionando información oportuna para la toma de decisiones y el análisis del comportamiento del banco de baterías.

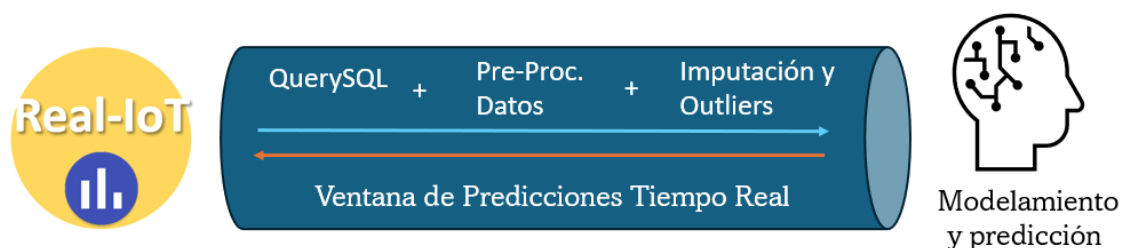


Figura 4.16. Arquitectura General de la implementación

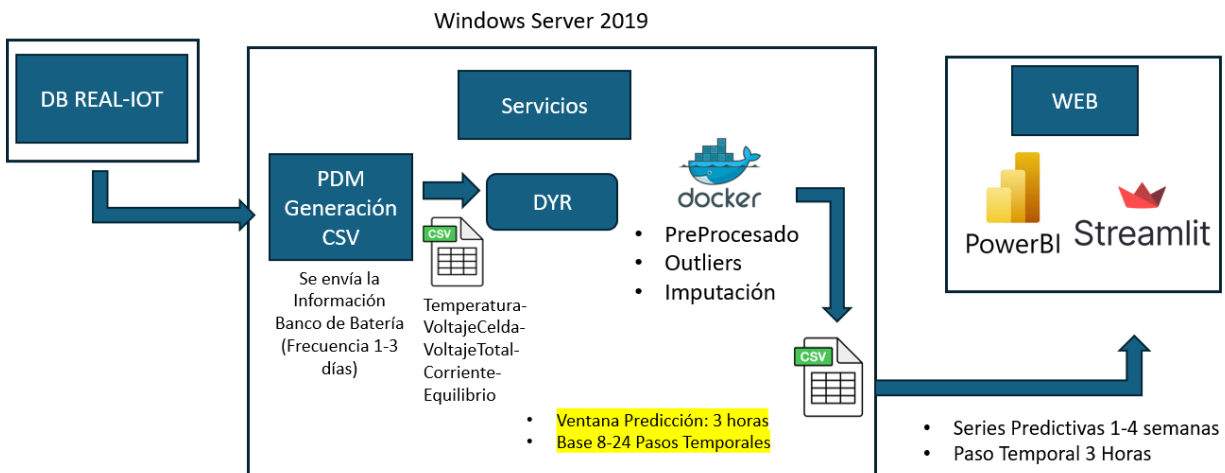


Figura 4.17. Diagrama de flujo ETL

El sistema se inicia con la adquisición de datos desde la base de datos DB REAL-IOT, que concentra la información de telemetría del banco de baterías. Periódicamente (frecuencia de envío entre 1 y 3 días, según la configuración) se exporta un fichero CSV que contiene las señales relevantes para el análisis: temperatura, voltaje por celda, voltaje total, corriente y una medida de equilibrio entre celdas. Esta extracción centraliza y homogeniza las observaciones en un formato tabular que sirve como entrada para el resto del pipeline.

En el servidor principal (Windows Server 2019) se ejecuta el módulo de *PDM — Generación CSV* que prepara los ficheros de entrada y los entrega al componente denominado DYR y a los servicios de procesamiento. A continuación, dentro de un contenedor Docker se implementa la etapa de preprocesado: limpieza, detección y marcado de outliers y la generación de marcas de datos faltantes. Este bloque se encarga también de normalizar y transformar las variables (por ejemplo, suavizado o cálculo de características derivadas) para que los datos queden en la forma requerida por los modelos predictivos.

Tras la limpieza, el pipeline realiza la imputación de valores faltantes utilizando el procedimiento definido (por ejemplo filtros dinámicos o métodos basados en modelos de espacio de estados). La imagen indica que la ventana de predicción y la base de entrada están parametrizadas: se usan ventanas de entrada de entre 8 y 24 pasos temporales y una ventana de predicción operativa de 3 horas (paso temporal de 3 horas). Estas configuraciones permiten que el modelo reciba contexto histórico suficiente y genere predicciones diarias o cada 2-3 días.

La inferencia y los modelos de predicción se ejecutan dentro del mismo entorno Docker para garantizar reproducibilidad y aislamiento. El resultado de la inferencia se volca a un nuevo CSV que contiene las series predictivas generadas; las predicciones pueden cubrir horizontes que van de 1 a 4 semanas, dependiendo de la configuración de salida y del modelo empleado. Este fichero de salida sirve tanto para análisis posterior como para alimentar las capas de visualización y toma de decisiones.

Finalmente, los resultados se exponen a través de la capa web: dashboards y aplicaciones de visualización (Power BI y/o Streamlit) consumen las series predictivas y los ficheros procesados para mostrar informes, alertas y métricas de desempeño. La arquitectura centralizada en Windows Server y el despliegue de componentes en contenedores facilita la automatización del flujo (programación de extracciones, ejecución periódica del preprocesado e inferencia) y el mantenimiento continuo del sistema en producción.

Capítulo 5

Conclusiones

La primera contribución de este trabajo fue la exitosa preparación de un conjunto de datos extenso y detallado. La mezcla y consolidación de datos provenientes de múltiples fuentes, el alineamiento temporal a un paso de 3 horas mediante la elección de estadísticos representativos (mediana para voltaje y corriente, mínimo para equilibrio y temperatura), el tratamiento de datos atípicos (usando una estrategia híbrida de IQR y Distancia de Mahalanobis) y el tratamiento de datos faltantes se debe realizar en este orden. Esta estrategia evita que observaciones extremas contaminen la estimación de parámetros y las imputaciones posteriores. La imputación de datos faltantes se realizó mediante técnicas dinámicas (filtro de Kalman / modelos de espacio de estados), aprovechando la estructura temporal del proceso y proporcionando estimaciones coherentes con la dinámica física del sistema. Esto permitió sentar las bases para un modelado robusto y que se vió reflejado en los resultados de las métricas de rendimiento de las predicciones de los modelos.

Los tests de estacionariedad (ADF y KPSS) indican heterogeneidad en las series: la serie de corriente resulta claramente estacionaria, mientras que voltaje total y voltaje promedio muestran evidencia consistente de no estacionariedad; equilibrio y temperatura arrojan resultados contradictorios entre ADF y KPSS, lo que sugiere necesidad de análisis adicional (diferenciación, detección de rupturas o tests alternativos). Dado este panorama, la superioridad del LSTM se explica por su capacidad para modelar dependencias de largo plazo y no linealidades sin requerir estacionariedad. En tareas de imputación/suavizado, el filtro de Kalman rindió mejor en variables suaves y lineales (temperatura, voltaje promedio y corriente), mientras que una imputación heurística superó al Kalman en Voltaje total y Equilibrio, probablemente debido a no linealidades, picos o rupturas que el Kalman lineal no modeló adecuadamente.

Los modelos de suavizamiento exponencial (Holt y Holt-Winters) sirvieron como una línea

base esencial para evaluar el rendimiento predictivo. Se observó que, en general, el método Holt-Winters (tanto aditivo como multiplicativo, cuando fue aplicable) superó consistentemente al método Holt simple, logrando menores errores en las métricas RMSE y MAPE para la mayoría de las variables analizadas (Voltaje Total, Equilibrio, Voltaje Promedio, Temperatura Promedio). Sin embargo, el error más bajo obtenido en estos modelos, fue notoriamente mejorado por los modelos de redes neuronales artificiales LSTM. De esta manera, se logró probar empíricamente la hipótesis de este trabajo, esto se debe a que las LSTMs son intrínsecamente mejores para capturar patrones secuenciales complejos, dependencias a largo plazo y relaciones no lineales entre múltiples variables, características que están presentes en los datos de un banco de baterías. La capacidad predictiva mejorada de estos modelos se traduce directamente en una mayor anticipación de eventos de falla. Al pronosticar con mayor precisión el comportamiento normal de las señales, las desviaciones significativas que indiquen un posible fallo podrán ser detectadas con mayor antelación, permitiendo un mantenimiento predictivo más eficaz.

El modelo LSTM superó de manera consistente a los métodos clásicos de referencia en las tareas de predicción consideradas. En términos de RMSE, las reducciones relativas observadas frente al mejor modelo Holt–Winters/Holt por variable fueron aproximadamente: Voltaje total 86 %, Equilibrio 61 %, Voltaje promedio 67 %, Temperatura promedio 81 % y Corriente 96 %. Las mejoras en MAE siguen la misma tendencia, confirmando ganancias sustanciales en precisión predictiva y robustez frente a ruido y outliers.

Los resultados y la metodología desarrollada sientan las bases para implementar sistemas de monitoreo y alerta temprana más sofisticados, que podrían reducir costos operativos, minimizar tiempos de inactividad y prolongar la vida útil de los activos críticos. En este sentido, se recomienda emplear LSTM como modelo predictivo preferente para estas series, usando Holt-Winters únicamente como baseline clásico para comparación. Para tareas de imputación o preprocesamiento conviene evaluar caso a caso: ajustar las covarianzas del filtro de Kalman (matrices Q , R), probar variantes robustas/no lineales (EKF/UKF) o combinar enfoques (heurística en transitorios más Kalman en regímenes suaves).

En términos generales, los tiempos de ejecución observados indican que los procedimientos propuestos son factibles para una eventual implementación en tiempo real. Las etapas de preparación de los datos, que incluyen el armado de las series, el tratamiento de valores atípicos y la imputación de datos faltantes, presentan tiempos de cómputo reducidos (minutos de cómputo sobre 3000 observaciones) en relación con el proceso completo, incluso cuando se ejecutan en

un computador personal con procesador convencional. La etapa más costosa en términos de tiempo corresponde al entrenamiento de los modelos de redes neuronales, en CPU suele tardar decenas de minutos a varias horas, mientras que en GPU (Colab con T4 o una GPU local tipo 3060) el mismo entrenamiento se reduce típicamente a unos minutos o decenas de minutos. En ambos casos, una vez entrenado el modelo, el proceso de predicción es rápido y compatible con esquemas de actualización periódica, como los considerados en este trabajo. En consecuencia, el uso de entornos con aceleración gráfica resulta especialmente recomendable para la fase de entrenamiento y ajuste de modelos, mientras que la ejecución del flujo completo de predicción puede realizarse de manera eficiente incluso en infraestructuras computacionales más simples, lo que respalda la viabilidad práctica del enfoque propuesto.

Como trabajos futuros o próximos pasos, se debe explorar el uso de modelos híbridos, es decir, que sean físico estadísticos (por ejemplo, modelos electroquímicos) o bien, que mezclen métodos de redes neuronales multivariantes en combinación con métodos clásicos multivariantes.

Este proyecto ha validado la importancia de un preprocesamiento exhaustivo y ha explorado la aplicación de métodos de series de tiempo y redes neuronales para el análisis de datos eléctricos en un banco de baterías. Si bien los modelos clásicos ofrecen una referencia útil, los resultados y la naturaleza de las LSTMs sugieren que estas últimas tienen un potencial considerablemente mayor para la predicción precisa y la detección temprana de fallas, alineándose con las demandas de la ingeniería moderna para el mantenimiento predictivo y la optimización de sistemas industriales.

Bibliografía

- [1] U.S. Department of Energy, Federal Energy Management Program. *Operations & Maintenance Best Practices Guide: Release 3.0*. Inf. téc. U.S. Department of Energy, 2020. URL: https://www.energy.gov/sites/prod/files/2020/04/f74/omguide_complete_w-eo-disclaimer.pdf.
- [2] George E. P. Box et al. *Time Series Analysis: Forecasting and Control*. 5.^a ed. Hoboken, NJ: John Wiley & Sons, 2015. ISBN: 978-1-118-67502-1.
- [3] Rob J. Hyndman y George Athanasopoulos. *Forecasting: principles and practice*. 2.^a ed. Accedido: 2025-12-28. OTexts, 2018. URL: <https://otexts.com/fpp2/>.
- [4] John W. Tukey. *Exploratory Data Analysis*. Addison-Wesley, 1977.
- [5] Boris Iglewicz y David C. Hoaglin. *How to Detect and Handle Outliers*. ASQC Quality Press (Wiley), 1993.
- [6] Vic Barnett y Toby Lewis. *Outliers in Statistical Data*. 3.^a ed. Wiley, 1994.
- [7] P. C. Mahalanobis. «On the generalised distance in statistics». En: *Proceedings of the National Institute of Sciences of India* 2 (1936), págs. 49-55.
- [8] R. De Maesschalck, D. Jouan-Rimbaud y D. L. Massart. «The Mahalanobis distance». En: *Chemometrics and Intelligent Laboratory Systems* 50 (2000), págs. 1-18.
- [9] Peter J. Rousseeuw y Annick M. Leroy. *Robust Regression and Outlier Detection*. Wiley, 1987.
- [10] Rudolf E. Kalman. «A New Approach to Linear Filtering and Prediction Problems». En: *Journal of Basic Engineering* 82.1 (1960), págs. 35-45. DOI: [10.1115/1.3662552](https://doi.org/10.1115/1.3662552).
- [11] Greg Welch y Gary Bishop. *An Introduction to the Kalman Filter*. Inf. téc. TR 95-041. Tutorial ampliamente usado — disponible en línea. University of North Carolina at Chapel Hill, Department of Computer Science, 1995.

- [12] Dan Simon. *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*. 1.^a ed. Wiley-Interscience, 2006.
- [13] Peter S. Maybeck. *Stochastic Models, Estimation, and Control*. Vol. 1. Academic Press, 1979.
- [14] Jon Ander Alonso Ortiz de Elguea. *El filtro de Kalman*. Trabajo Fin de Grado. Tutor: José María Pérez Izquierdo. Publicado por la Universidad de La Rioja. Distribuido bajo licencia Creative Commons Reconocimiento-NoComercial-SinObraDerivada 3.0 Unported. Disponible en publicaciones.unirioja.es. Logroño, España: Universidad de La Rioja, Facultad de Ciencia y Tecnología, 2023.
- [15] James Durbin y Siem Jan Koopman. *Time Series Analysis by State Space Methods*. 2.^a ed. Oxford: Oxford University Press, 2012.
- [16] Robert H. Shumway y David S. Stoffer. *Time Series Analysis and Its Applications: With R Examples*. 4.^a ed. New York: Springer, 2017.
- [17] Christopher Chatfield. *The Analysis of Time Series: An Introduction*. 6.^a ed. Boca Raton, FL: Chapman & Hall/CRC, 2003. ISBN: 978-1584883173.
- [18] Peter J. Brockwell y Richard A. Davis. *Introduction to Time Series and Forecasting*. 3.^a ed. Springer Texts in Statistics. New York: Springer, 2016. ISBN: 978-3319298542.
- [19] Trevor Hastie, Robert Tibshirani y Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. 2.^a ed. Springer, 2009.
- [20] Ian Goodfellow, Yoshua Bengio y Aaron Courville. *Deep Learning*. Disponible en línea. Cambridge, MA: MIT Press, 2016. ISBN: 978-0262035613. URL: <https://www.deeplearningbook.org/>.
- [21] Sepp Hochreiter y Jürgen Schmidhuber. «Long Short-Term Memory». En: *Neural Computation* 9.8 (1997), págs. 1735-1780. DOI: [10.1162/neco.1997.9.8.1735](https://doi.org/10.1162/neco.1997.9.8.1735).
- [22] Aurélien Géron. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*. 2.^a ed. Práctico, incluye ejemplos de RNN/LSTM con TensorFlow / Keras. Sebastopol, CA: O'Reilly Media, 2019. ISBN: 978-1492032649.
- [23] David E. Rumelhart, Geoffrey E. Hinton y Ronald J. Williams. «Learning internal representations by error propagation». En: *Nature* 323 (1986), págs. 533-536.
- [24] Frank Rosenblatt. «The perceptron: A probabilistic model for information storage and organization in the brain». En: *Psychological Review* 65.6 (1958), págs. 386-408.

- [25] Warren S. McCulloch y Walter Pitts. «A logical calculus of the ideas immanent in nervous activity». En: *Bulletin of Mathematical Biophysics* 5 (1943), págs. 115-133.
- [26] Marvin Minsky y Seymour Papert. *Perceptrons: An Introduction to Computational Geometry*. MIT Press, 1969.
- [27] Vinod Nair y Geoffrey E. Hinton. «Rectified Linear Units improve Restricted Boltzmann Machines». En: *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*. 2010, págs. 807-814.
- [28] Andrew L. Maas, Awni Y. Hannun y Andrew Y. Ng. «Rectifier Nonlinearities Improve Neural Network Acoustic Models». En: *Proceedings of ICML Workshop on Deep Learning for Audio, Speech and Language Processing* (2013).
- [29] Simon Haykin. *Neural Networks: A Comprehensive Foundation*. 2.^a ed. Upper Saddle River, NJ: Prentice Hall, 1999. ISBN: 978-0131471399.
- [30] Alex Graves. «Speech recognition with deep recurrent neural networks». En: *2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2013, págs. 6645-6649.
- [31] Felix A. Gers, Jürgen Schmidhuber y Fred Cummins. «Learning to Forget: Continual Prediction with LSTM». En: *Neural Computation* 12.10 (2000), págs. 2451-2471. DOI: [10.1162/089976600300015015](https://doi.org/10.1162/089976600300015015). URL: <https://sferics.idsia.ch/pub/juergen/FgGates-NC.pdf>.
- [32] Klaus Greff et al. «LSTM: A Search Space Odyssey». En: *IEEE Transactions on Neural Networks and Learning Systems* (2017). Revisión exhaustiva sobre variantes y desempeño de LSTM (ver versión arXiv/preprint para detalles).
- [33] David A. Dickey y Wayne A. Fuller. «Distribution of the Estimators for Autoregressive Time Series with a Unit Root». En: *Journal of the American Statistical Association* 74.366 (1979), págs. 427-431. DOI: [10.1080/01621459.1979.10482531](https://doi.org/10.1080/01621459.1979.10482531).
- [34] Denis Kwiatkowski et al. «Testing the Null Hypothesis of Stationarity Against the Alternative of a Unit Root». En: *Journal of Econometrics* 54.1-3 (1992), págs. 159-178. DOI: [10.1016/0304-4076\(92\)90104-Y](https://doi.org/10.1016/0304-4076(92)90104-Y).
- [35] Serena Ng y Pierre Perron. «Lag Length Selection and the Construction of Unit Root Tests with Good Size and Power». En: *Econometrica* 69.6 (2001), págs. 1519-1554.
- [36] G. William Schwert. «Tests for Unit Roots: A Monte Carlo Investigation». En: *Journal of Business & Economic Statistics* 7.2 (1989), págs. 147-159.

- [37] Whitney K. Newey y Kenneth D. West. «A Simple, Positive Semi-definite, Heteroskedasticity and Autocorrelation Consistent Covariance Matrix». En: *Econometrica* 55.3 (1987), págs. 703-708.
- [38] Donald W. K. Andrews. «Heteroskedasticity and Autocorrelation Consistent Covariance Matrix Estimation». En: *Econometrica* 59 (1991), págs. 817-858.
- [39] Eric Zivot y Donald W. K. Andrews. «Further Evidence on the Great Crash, the Oil-Price Shock, and the Unit-Root Hypothesis». En: *Journal of Business & Economic Statistics* 10.3 (1992), págs. 251-270.
- [40] Felipe Lermenda Rodríguez. «Predicción de Demanda Eléctrica Univariada Aplicada al Caso Chileno». Profesora co-referente: Natalia Bahamonde Rozas. Memoria para optar al grado de Magíster en Estadística. Valparaíso, Chile: Pontificia Universidad Católica de Valparaíso, 2022.

Apéndice A

Capítulo 1

A.1. Rangos de normalidad y falla

1. Voltaje Total

- Condiciones normales: Entre 120 V y 132 V.
- Condiciones de falla: Bajo límite inferior (esto puede afectar la capacidad del banco para suministrar energía de manera eficiente) o sobre el límite superior (puede ser señal de sobrecarga, lo que puede dañar las celdas y reducir la vida útil del banco).

2. Equilibrio

- Si todas las celdas están casi iguales, el sistema marca cerca de 100 %.
- Si hay grandes diferencias de voltaje, el balance cae hacia 0 %.

3. Voltaje Promedio

- Bajo 2.05 V: Descarga profunda.
- Entre 2.05-2.14 V: Nivel bajo (requiere revisión de carga o cargador).
- Entre 2.15-2.24 V: Nivel de carga moderada aceptable.
- Entre 2.25-2.30 V: Carga óptima.

Observación: Un voltaje promedio correcto no garantiza que todas las celdas estén bien, por eso es clave compararlo con el balancing (equilibrio). Por ejemplo, Voltaje promedio = 2.25V (óptimo), pero equilibrio = 60 % (algunas celdas están bajas, otras altas := RIESGO)

4. Corriente

- Entre +0.5 A y 5 A: Carga en flotación, banco en mantenimiento de carga normal.
- Entre +5 A y 25 A: Carga rápida, proceso activo de carga.
- Condición de falla: Menos a +0.5 A o sobre 25 A (riesgo crítico), carga forzada o anormal. Sobrecarga.

5. Temperatura Promedio

- Aceptable: Entre 25 y 30 °C.
- Alto Riesgo: Entre 31 y 40 °C.
- Crítico: Superior o igual a 40 °C.

Apéndice B

Capítulo 2

B.1. Modelos Autorregresivos y de Medias Móviles

Los modelos ARIMA (Autorregresivos Integrados de Medias Móviles) son una de las clases más importantes de modelos para el análisis de series temporales estacionarias. Son una combinación de tres componentes [2, 40].

Definición 15 (Modelo autorregresivo de orden p — $AR(p)$). Sea $(Y_t)_{t \in \mathbb{Z}}$ una serie temporal. Un modelo autorregresivo de orden p se define por

$$Y_t = c + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \cdots + \phi_p Y_{t-p} + \epsilon_t,$$

donde $c \in \mathbb{R}$ es una constante, $\phi_1, \dots, \phi_p \in \mathbb{R}$ son los coeficientes autorregresivos y (ϵ_t) es un proceso de ruido blanco con $\mathbb{E}[\epsilon_t] = 0$ y $\text{Var}(\epsilon_t) = \sigma_\epsilon^2$.

Definición 16 (Modelo de media móvil de orden q — $MA(q)$). Un modelo de media móvil de orden q para la serie (Y_t) está dado por

$$Y_t = \mu + \epsilon_t + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + \cdots + \theta_q \epsilon_{t-q},$$

donde $\mu \in \mathbb{R}$ es la media de la serie, $\theta_1, \dots, \theta_q \in \mathbb{R}$ son los coeficientes de media móvil y (ϵ_t) es ruido blanco con $\mathbb{E}[\epsilon_t] = 0$, $\text{Var}(\epsilon_t) = \sigma_\epsilon^2$.

Definición 17 (Modelo autorregresivo de media móvil — $ARMA(p, q)$). El modelo $ARMA(p, q)$ combina los componentes $AR(p)$ y $MA(q)$. Se expresa como

$$Y_t = c + \phi_1 Y_{t-1} + \cdots + \phi_p Y_{t-p} + \epsilon_t + \theta_1 \epsilon_{t-1} + \cdots + \theta_q \epsilon_{t-q},$$

con c, ϕ_i, θ_j definidos como en (15) y (16), y (ϵ_t) ruido blanco.

Definición 18 (Modelo autorregresivo integrado de media móvil — $ARIMA(p, d, q)$). El modelo $ARIMA(p, d, q)$ extiende $ARMA$ a series no estacionarias mediante diferenciación. Si B denota el operador de rezago (lag) $BY_t = Y_{t-1}$ y $\nabla = 1 - B$, entonces Y_t sigue un $ARIMA(p, d, q)$ si la serie diferenciada $\nabla^d Y_t$ satisface un modelo $ARMA(p, q)$, es decir

$$\Phi(B) \nabla^d Y_t = c + \Theta(B) \epsilon_t,$$

donde $\Phi(B) = 1 - \phi_1 B - \dots - \phi_p B^p$ y $\Theta(B) = 1 + \theta_1 B + \dots + \theta_q B^q$, y (ϵ_t) es ruido blanco.

1. Modelo Autorregresivo de orden p (AR(p)): Un modelo $AR(p)$ expresa el valor actual de la serie, Y_t , como una combinación lineal de sus p valores pasados más un término de error aleatorio.

$$Y_t = c + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \dots + \phi_p Y_{t-p} + \epsilon_t$$

donde c es una constante, $\phi_1, \dots, \phi_p$ son los coeficientes autorregresivos, y ϵ_t es un término de ruido blanco.

2. Modelo de Media Móvil de orden q (MA(q)): Un modelo $MA(q)$ expresa el valor actual de la serie, Y_t , como una combinación lineal de los q errores de pronóstico pasados.

$$Y_t = \mu + \epsilon_t + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + \dots + \theta_q \epsilon_{t-q}$$

donde μ es la media de la serie, $\theta_1, \dots, \theta_q$ son los coeficientes de media móvil, y ϵ_t es un término de ruido blanco.

3. Diferenciación: Es una transformación aplicada a una serie temporal cuyo objetivo principal es eliminar la no estacionariedad asociada a la presencia de tendencia. Consiste en reemplazar la serie original por las diferencias entre observaciones consecutivas, de modo que la serie diferenciada de primer orden se define como $\Delta y_t = y_t - y_{t-1}$. En caso de que la no estacionariedad persista, pueden aplicarse diferenciaciones de orden superior. Este procedimiento es fundamental en la modelación clásica de series temporales, particularmente en los modelos $ARIMA$, ya que permite convertir una serie integrada en una serie aproximadamente estacionaria, condición necesaria para el análisis y la inferencia estadística [2, 3].

B.2. Modelos ARMA y ARIMA

1. Modelo Autorregresivo de Media Móvil (ARMA(p,q)): Este modelo combina las características de los modelos AR(p) y MA(q).

$$Y_t = c + \phi_1 Y_{t-1} + \cdots + \phi_p Y_{t-p} + \epsilon_t + \theta_1 \epsilon_{t-1} + \cdots + \theta_q \epsilon_{t-q}$$

2. Modelo Autorregresivo Integrado de Media Móvil (ARIMA(p,d,q)): El modelo ARIMA extiende el modelo ARMA para series de tiempo *no estacionarias*. La letra “I” de “Integrado” se refiere al proceso de diferenciación para hacer la serie estacionaria. Un modelo ARIMA(p,d,q) indica que la serie ha sido diferenciada d veces y que la serie resultante sigue un modelo ARMA(p,q).

En otras palabras, si el operador autorregresivo $\phi(B)$ es de orden p , se toma la d -ésima diferencia y el operador de media móvil $\theta(B)$ es de orden q , decimos que tenemos un modelo ARIMA de orden (p, d, q) , o simplemente un proceso ARIMA(p,d,q).

Apéndice C

Capítulo 3

C.1. Gráfico Distancia de Mahalanobis

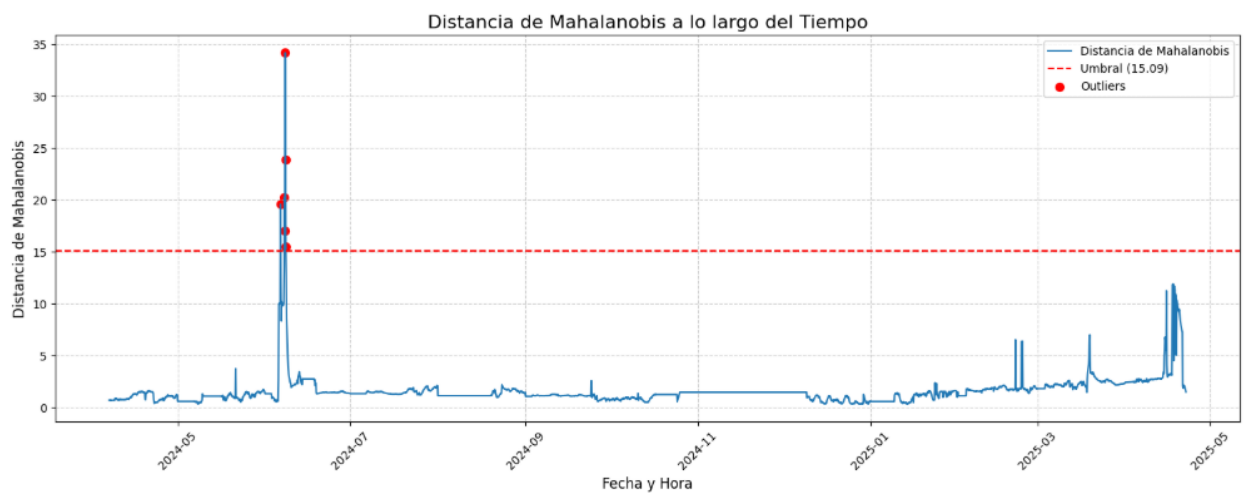


Figura C.1. Distancias de Mahalanobis de las observaciones.

C.2. Gráficos de variables crudas y ordenadas

A continuación, se presentan los primeros gráficos y resúmenes estadísticos de cada variable. Las observaciones están ordenadas en el tiempo sin aún ser tratadas.

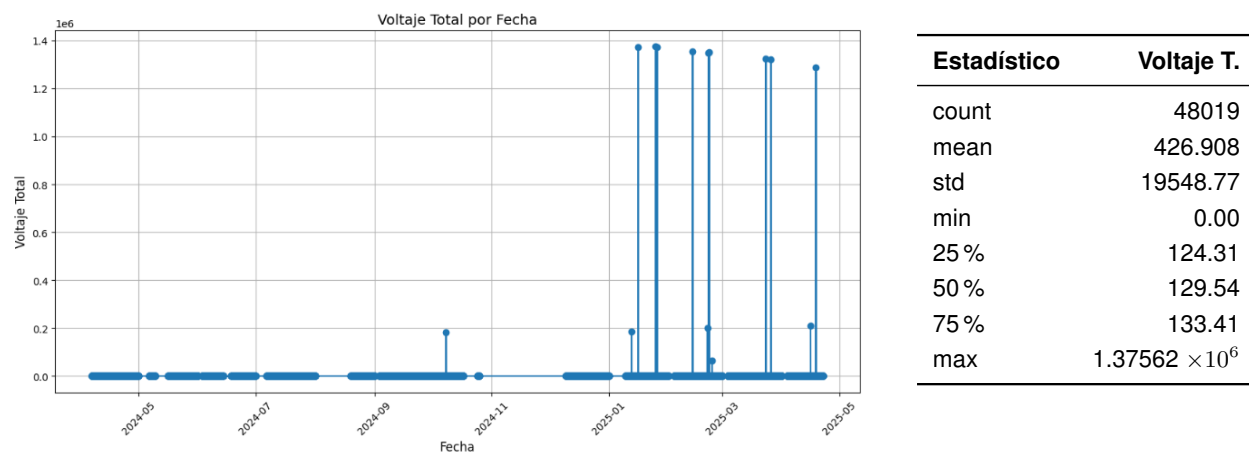


Figura C.2. Gráfico de series de tiempo de la variable Voltaje Total y su resumen estadístico descriptivo.

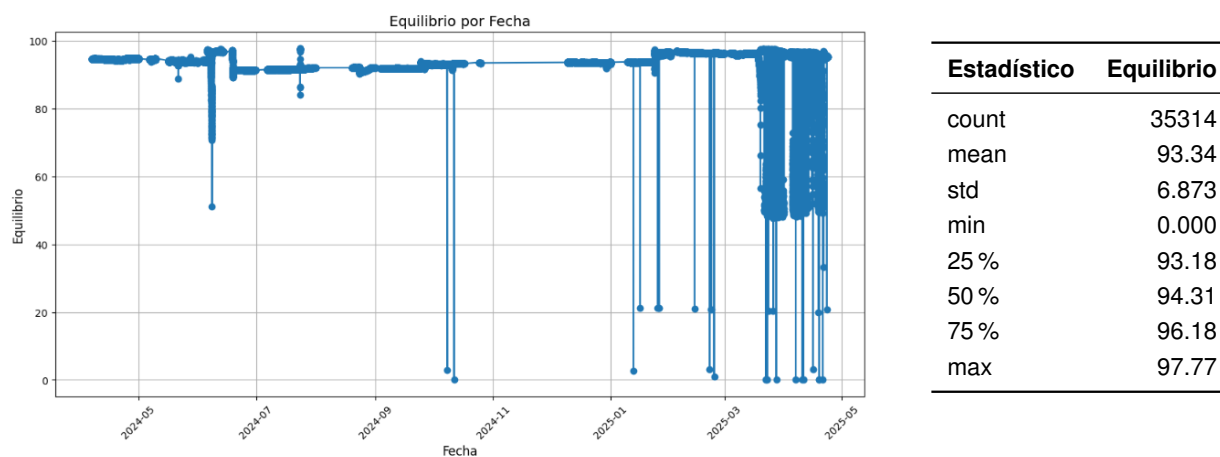


Figura C.3. Gráfico de series de tiempo de la variable Equilibrio y su resumen estadístico descriptivo.

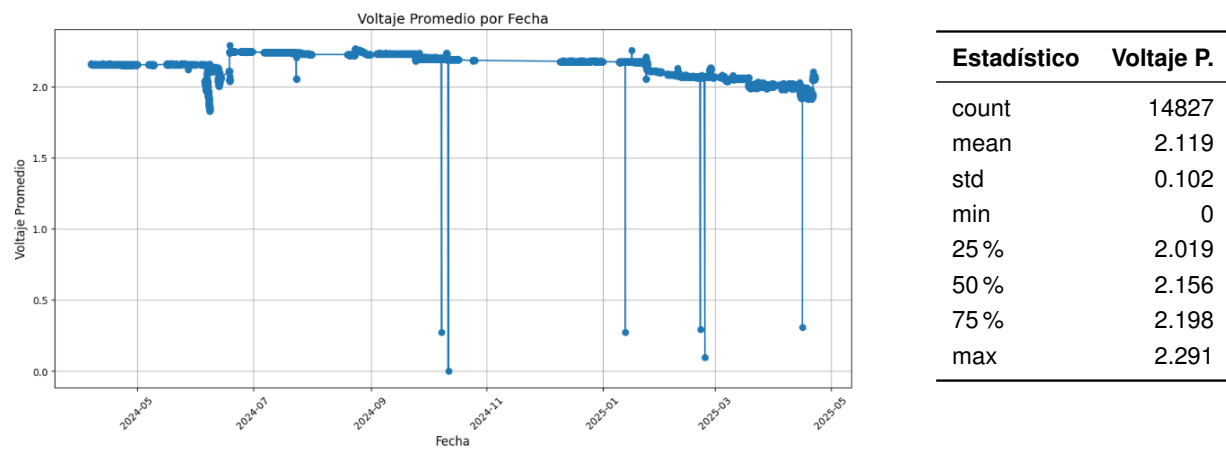


Figura C.4. Gráfico de series de tiempo de la variable Voltaje Promedio y su resumen estadístico descriptivo.

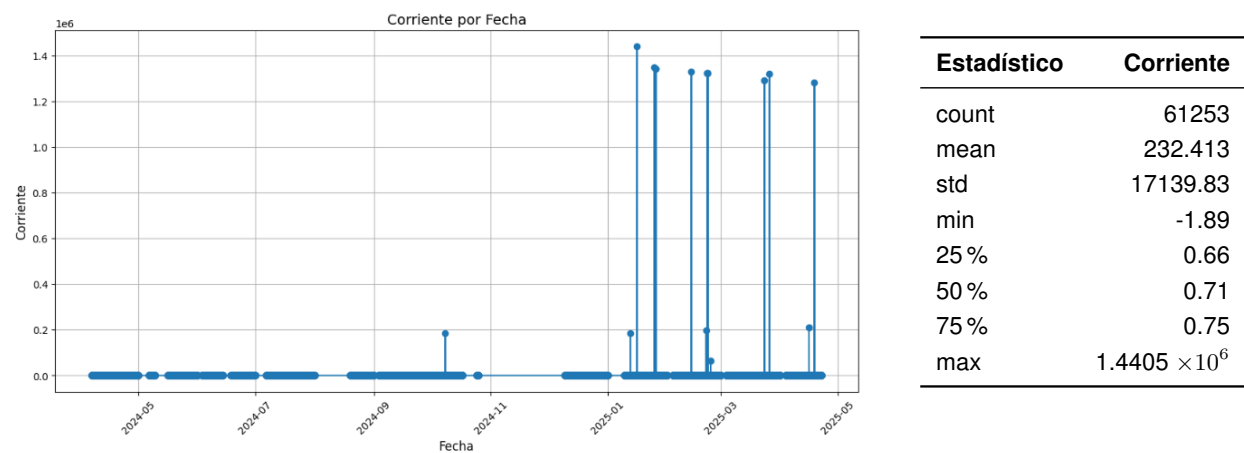


Figura C.5. Gráfico de series de tiempo de la variable Corriente y su resumen estadístico descriptivo.

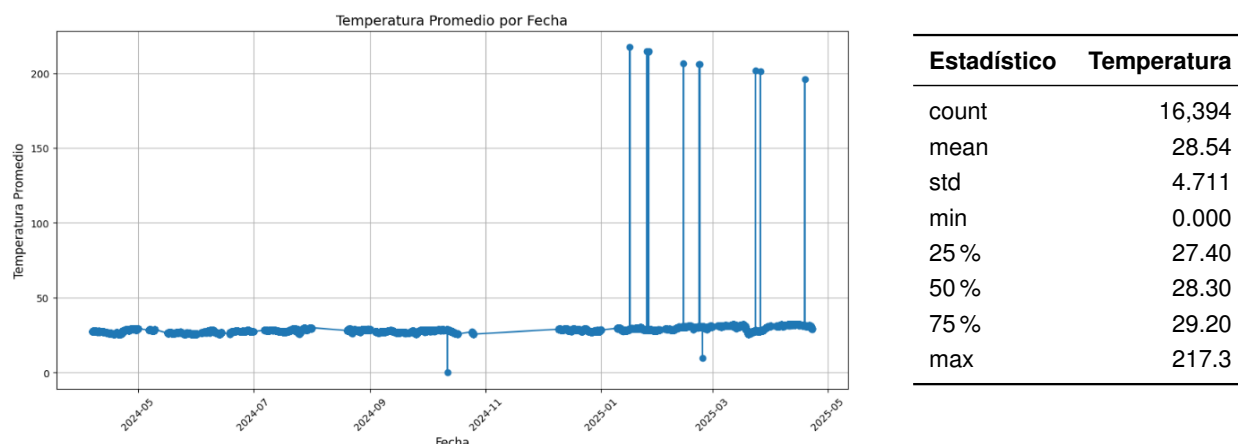


Figura C.6. Gráfico de series de tiempo de la variable Temperatura Promedio y su resumen estadístico descriptivo.

C.3. Código R: Imputación con Filtro de Kalman

A continuación se incluye el script en R utilizado para la imputación de datos faltantes mediante filtro de Kalman (paquete `imputeTS`). El script muestra el flujo general aplicado a las series: lectura CSV, diagnóstico de faltantes, imputación con `na_kalman(...)` y exportación de las series imputadas.

```

1 # =====
2 # Imputación de datos faltantes con Filtro de Kalman
3 # en series temporales leídas desde un archivo CSV
4 # =====
5
6 # 1. Instalación y carga de librerías necesarias
7 install.packages("imputeTS")      # Para imputación con Kalman
8 install.packages("ggplot2")       # Para gráficos
9 install.packages("forecast")      # Para modelos ARIMA opcionales
10
11 library(imputeTS)
12 library(ggplot2)
13 library(forecast)
14
15
16

```

```

17 # 2. Lectura de datos desde un archivo CSV
18 # Asegurase de que el CSV tenga una columna de tiempo y otra con las
    mediciones.
19 # Ejemplo de formato esperado:
20 # tiempo,voltaje
21 # 2024-01-01 00:00:00,3.70
22 # 2024-01-01 00:05:00,3.68
23 # 2024-01-01 00:10:00,NA
24 # ...
25
26 datos <- read.csv("serie_voltaje.csv", stringsAsFactors = FALSE)
27
28 # 3. Conversión a serie temporal
29 # Suponiendo que la columna de interés se llama "voltaje"
30 serie <- ts(datos$voltaje)
31
32 # 4. Diagnóstico de datos faltantes
33 statsNA(serie)
34
35 # 5. Imputación con Filtro de Kalman
36 # Modelo estructural (por defecto)
37 serie_imputada <- na_kalman(serie, model = "StructTS")
38
39 # También se puede probar con ARIMA
40 # serie_imputada <- na_kalman(serie, model = "auto.arima")
41
42 # 6. Comparación de valores originales vs imputados
43 comparacion <- data.frame(
44   Tiempo = 1:length(serie),
45   Original = serie,
46   Imputado = serie_imputada
47 )
48
49 print(head(comparacion, 20)) # Mostrar las primeras filas
50
51 # 7. Visualización
52 ggplot_na_imputations(serie, serie_imputada,
53   legend = TRUE,
54   title = "Imputación de datos faltantes con Filtro de
    Kalman")

```

```

55
56 datos2 <- read.csv("serie_voltaje_3h_filled.csv", stringsAsFactors = FALSE
    )
57 serie2 <- ts(datos2$voltaje)
58 statsNA(serie2)
59
60 comparacion2 <- data.frame(
61   Tiempo = 1:length(serie_imputada),
62   Original = serie2,
63   Imputado = serie_imputada
64 )
65 print(head(comparacion2, 20))
66
67 df <- data.frame(
68   fecha = time(serie_imputada),
69   valor = as.numeric(serie_imputada)
70 )
71 # escribir CSV (base R)
72 write.csv(df, "serie_voltaje_Kalman.csv", row.names = FALSE)
73
74 length(serie_imputada)
75 nrow(datos2)
76 class(datos2$time); head(datos2$time)
77
78
79 # Mismo proceso ahora para variable Equilibrio
80
81 datos_equilibrio <- read.csv("serie_equilibrio_3h.csv", stringsAsFactors =
    FALSE)
82
83 # 3. Conversión a serie temporal
84 serie_equilibrio <- ts(datos_equilibrio$equilibrio)
85
86 # 4. Diagnóstico de datos faltantes
87 statsNA(serie_equilibrio)
88
89 # 5. Imputación con Filtro de Kalman
90 # Modelo estructural (por defecto)
91 equilibrio_imputada <- na_kalman(serie_equilibrio, model = "StructTS")
92

```

```

93 # 6. Comparación de valores originales vs imputados
94 comparacion3 <- data.frame(
95   Tiempo = 1:length(serie_equilibrio),
96   Original = serie_equilibrio,
97   Imputado = equilibrio_imputada
98 )
99
100 print(head(comparacion3, 20)) # Mostrar las primeras filas
101
102 # 7. Visualización
103 ggplot_na_imputations(serie_equilibrio, equilibrio_imputada,
104                       legend = TRUE,
105                       title = "Imputación de datos faltantes con Filtro de
106                             Kalman")
107
108 df2 <- data.frame(
109   fecha = time(equilibrio_imputada),
110   valor = as.numeric(equilibrio_imputada)
111 )
112 # escribir CSV (base R)
113 write.csv(df2, "serie_equilibrio_Kalman.csv", row.names = FALSE)
114
115 # Mismo proceso ahora para variable Temperatura
116
117 datos_temperatura <- read.csv("serie_temperatura_3h.csv", stringsAsFactors
118                               = FALSE)
119
120 # 3. Conversión a serie temporal
121 serie_temperatura <- ts(datos_temperatura$temperatura)
122
123 # 4. Diagnóstico de datos faltantes
124 statsNA(serie_temperatura)
125
126 # 5. Imputación con Filtro de Kalman
127 # Modelo estructural (por defecto)
128 temperatura_imputada <- na_kalman(serie_temperatura, model = "StructTS")
129
130 # También se puede probar con ARIMA
131 # serie_imputada <- na_kalman(serie, model = "auto.arima")

```

```

131
132 # 6. Comparación de valores originales vs imputados
133 comparacion4 <- data.frame(
134   Tiempo = 1:length(serie_temperatura),
135   Original = serie_temperatura,
136   Imputado = temperatura_imputada
137 )
138
139 print(head(comparacion4, 20)) # Mostrar las primeras filas
140
141 # 7. Visualización
142 ggplot_na_imputations(serie_temperatura, temperatura_imputada,
143                       legend = TRUE,
144                       title = "Imputación de datos faltantes con Filtro de
145                               Kalman")
146
147 df3 <- data.frame(
148   fecha = time(temperatura_imputada),
149   valor = as.numeric(temperatura_imputada)
150 )
151 # escribir CSV (base R)
152 write.csv(df3, "serie_temperatura_Kalman.csv", row.names = FALSE)
153
154 # Mismo proceso ahora para variable Corriente
155
156 datos_corriente <- read.csv("serie_corriente_3h.csv", stringsAsFactors =
157   FALSE)
158
159 # 3. Conversión a serie temporal
160 serie_corriente <- ts(datos_corriente$corriente)
161
162 # 4. Diagnóstico de datos faltantes
163 statsNA(serie_corriente)
164
165 # 5. Imputación con Filtro de Kalman
166 # Modelo estructural (por defecto)
167 corriente_imputada <- na_kalman(serie_corriente, model = "StructTS")
168

```

```

169 # 6. Comparación de valores originales vs imputados
170 comparacion5 <- data.frame(
171   Tiempo = 1:length(serie_corriente),
172   Original = serie_corriente,
173   Imputado = corriente_imputada
174 )
175
176 print(head(comparacion5, 20)) # Mostrar las primeras filas
177
178 # 7. Visualización
179 ggplot_na_imputations(serie_corriente, corriente_imputada,
180                       legend = TRUE,
181                       title = "Imputación de datos faltantes con Filtro de
182                               Kalman")
183
184 df4 <- data.frame(
185   fecha = time(corriente_imputada),
186   valor = as.numeric(corriente_imputada)
187 )
188 # escribir CSV (base R)
189 write.csv(df4, "serie_corriente_Kalman.csv", row.names = FALSE)
190
191 # Mismo proceso ahora para variable Voltaje Total
192
193 datos_voltaje_total <- read.csv("serie_voltaje_total_3h.csv",
194                                stringsAsFactors = FALSE)
195
196 # 3. Conversión a serie temporal
197 serie_voltaje_total <- ts(datos_voltaje_total$voltaje_total)
198
199 # 4. Diagnóstico de datos faltantes
200 statsNA(serie_voltaje_total)
201
202 # 5. Imputación con Filtro de Kalman
203 # Modelo estructural (por defecto)
204 voltaje_total_imputada <- na_kalman(serie_voltaje_total, model = "StructTS
205 ")

```

```

206 # 6. Comparación de valores originales vs imputados
207 comparacion6 <- data.frame(
208   Tiempo = 1:length(serie_voltaje_total),
209   Original = serie_voltaje_total,
210   Imputado = voltaje_total_imputada
211 )
212
213 print(head(comparacion6, 20)) # Mostrar las primeras filas
214
215 # 7. Visualización
216 ggplot_na_imputations(serie_voltaje_total, voltaje_total_imputada,
217                       legend = TRUE,
218                       title = "Imputación de datos faltantes con Filtro de
219                               Kalman")
220
221 df5 <- data.frame(
222   fecha = time(voltaje_total_imputada),
223   valor = as.numeric(voltaje_total_imputada)
224 )
225 # escribir CSV (base R)
write.csv(df5, "serie_voltaje_total_Kalman.csv", row.names = FALSE)

```

Listing C.1: Imputación de datos faltantes con Filtro de Kalman (R).

C.4. Características complementarias

C.4.1. Gráficos

En los gráficos [C.7](#), [C.8](#) y [C.9](#) se muestran los resultados de las series de características complementarias propuestas en la metodología.

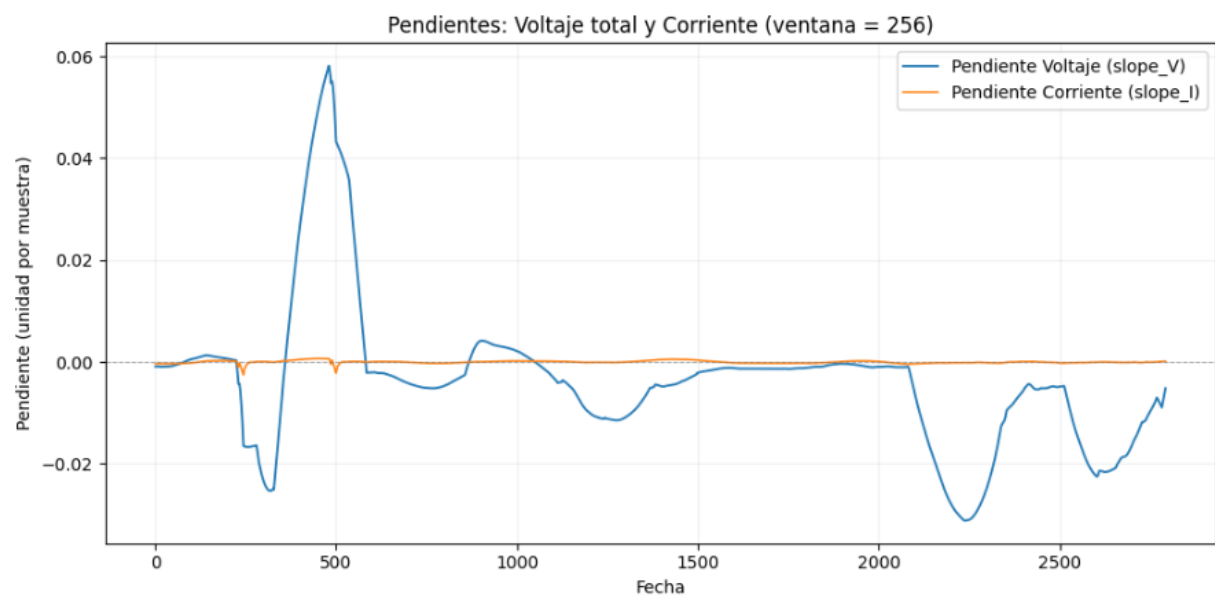


Figura C.7. Resultados de pendientes de voltaje o corriente.

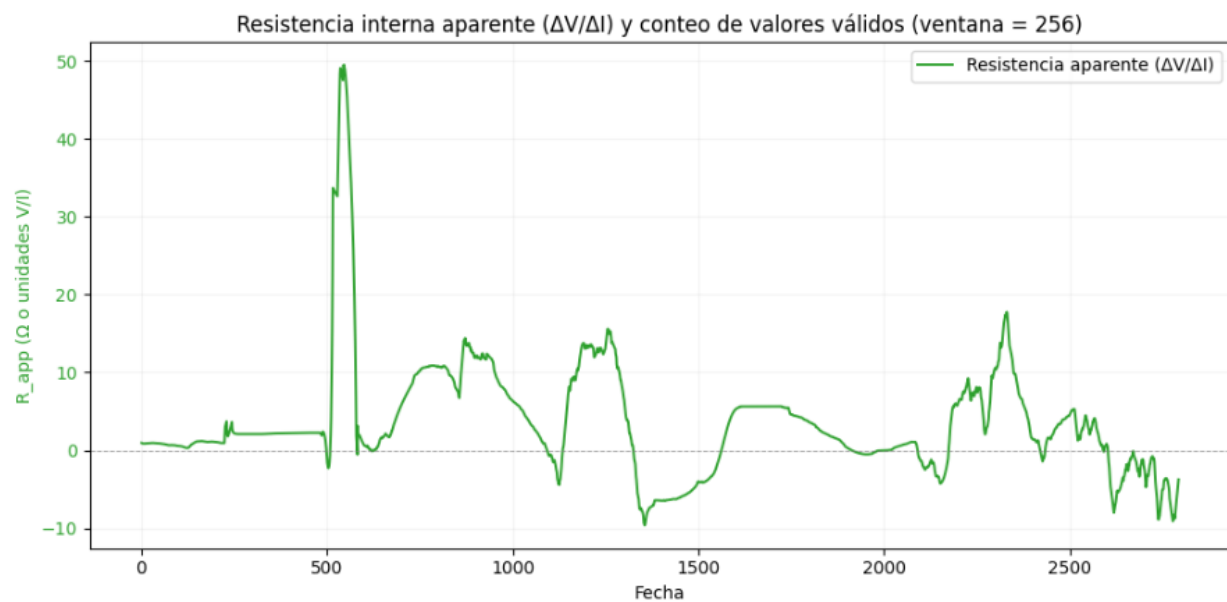


Figura C.8. Resultados de resistencia interna aparente.

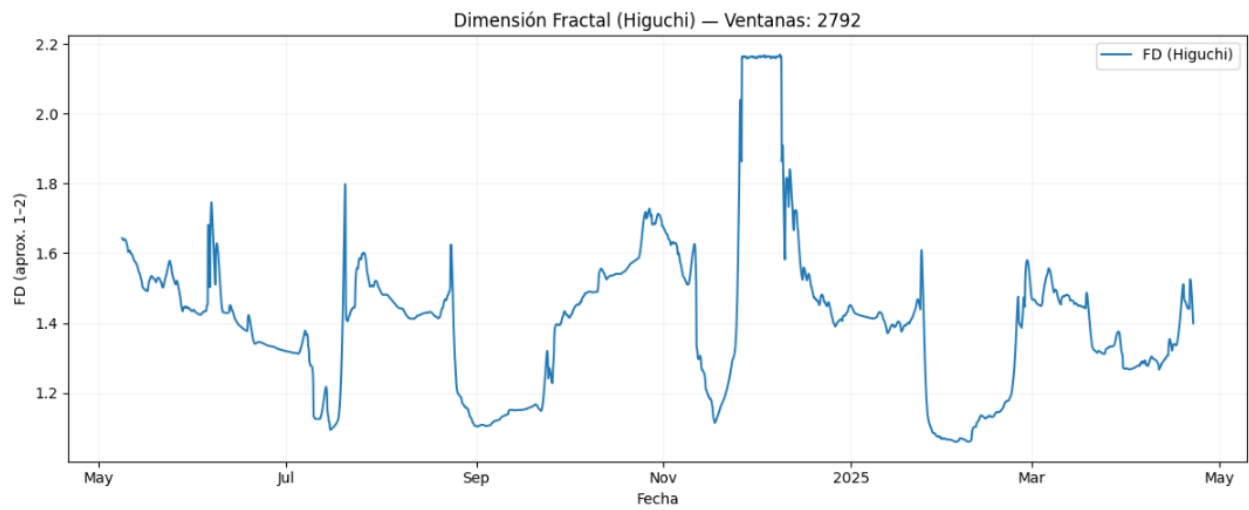


Figura C.9. Resultados de dimensión fractal usando método de Higuchi para serie Voltaje Total.

C.4.2. Estacionariedad

Usando Python, se evalúa la estacionariedad de las series de características complementarias. Los resultados para las series de tiempo de pendientes de voltaje, corriente y resistencia aparente, se muestran en la tabla C.1 y C.2.

Serie	ADF stat.	p-ADF	KPSS stat.	p-KPSS	<i>nlags</i> (KPSS)	Decisión combinada
<i>slope_V</i>	-3.7323	0.0037	2.2138	0.0100	32	Contradictorio / Incierto
<i>slope_I</i>	-4.3348	0.0004	0.4768	0.0469	auto	Contradictorio / Incierto
<i>R_app</i>	-3.6391	0.0051	0.2421	0.1000	auto	Estacionaria (consistente)

Tabla C.1. Resumen de tests de estacionariedad (ADF y KPSS)

Test	1 %	5 %	10 %	Observación
ADF (estadístico)	-3.4327	-2.8626	-2.5673	comparar ADF stat. con estos umbrales niveles: 1 %, 5 %, 10 % (regresión='c')
KPSS (estadístico)	0.7390	0.4630	0.3470	

Tabla C.2. Valores críticos reportados para ADF y KPSS

La serie *slope_V* (pendientes de voltaje) presenta un estadístico ADF de $-3,7323$ con un valor- $p = 0,0037$, lo que implica el rechazo de la hipótesis nula de raíz unitaria y, por tanto, evidencia de estacionariedad según el test ADF. Sin embargo, el test KPSS entrega un estadístico de $2,2138$ con un valor- $p = 0,01$, lo que conduce al rechazo de la hipótesis nula de estacionariedad. En consecuencia, los resultados para esta serie son contradictorios.

De manera análoga, la serie *slope_I* (pendientes de corriente) presenta un estadístico ADF de $-4,3348$ con un valor- $p = 0,0004$, indicando estacionariedad según el test ADF. No obstante, el estadístico KPSS es $0,4768$ con un valor- $p = 0,0469$, inferior al nivel de significancia $\alpha = 0,05$, lo que implica el rechazo de la hipótesis nula de estacionariedad y conduce nuevamente a un resultado contradictorio. Cabe destacar que durante la ejecución del test ADF se generó una advertencia de interpolación, indicando que el estadístico calculado se encuentra fuera del rango de valores tabulados y que el valor- p real es aún menor que el reportado, reforzando la evidencia contra la hipótesis nula de raíz unitaria.

Por último, la serie *R_app* (resistencia interna aparente) presenta un estadístico ADF de $-3,6391$ con un valor- $p = 0,0051$ y un estadístico KPSS de $0,2421$ con un valor- $p = 0,10$. En este caso, ambos tests entregan resultados coherentes con estacionariedad, dado que el ADF rechaza la hipótesis de raíz unitaria y el KPSS no rechaza la hipótesis nula de estacionariedad, constituyendo

un resultado consistente.

Serie	ADF stat.	p-ADF	KPSS stat.	p-KPSS	Decisión combinada
fd	-3.4336	0.0099	0.3563	0.0960	Estacionaria (consistente)

Tabla C.3. Resumen de tests de estacionariedad para Dimensión Fractal (FD)

La Tabla C.3 muestra que la serie correspondiente a la dimensión fractal (*fd*) presenta un estadístico ADF de $-3,4336$ con un valor- $p = 0,0099$, lo que permite rechazar la hipótesis nula de raíz unitaria y sugiere estacionariedad según el test ADF. De forma consistente, el test KPSS arroja un estadístico de $0,3563$ con un valor- $p = 0,0960$, por lo que no se rechaza la hipótesis nula de estacionariedad al nivel de significancia $\alpha = 0,05$. En consecuencia, ambos tests entregan evidencia concordante de que la serie *fd* puede considerarse estacionaria, lo que la hace adecuada para su uso directo en etapas posteriores de modelación sin requerir diferenciación adicional.

Apéndice D

Capítulo 4

D.1. Código fuente y notebooks

Los códigos desarrollados en este trabajo fueron implementados mediante notebooks de Google Colab, los cuales incluyen celdas de texto explicativo y celdas de código Python. El conjunto completo de notebooks se encuentra disponible en el siguiente repositorio GitHub:

<https://github.com/felipec-coder/tesisprediccionesseries>

La versión citada corresponde a la release v1.0. Los notebooks pueden ser visualizados directamente en GitHub o ejecutados en Google Colab.

D.2. Gráficos de predicciones LSTM (con tratamiento heurístico)

Las Figuras D.1 - D.5 muestran los gráficos con los resultados de las predicciones del modelo LSTM para todas las series remuestreadas a 3 horas con tratamiento de datos faltantes tipo heurístico.

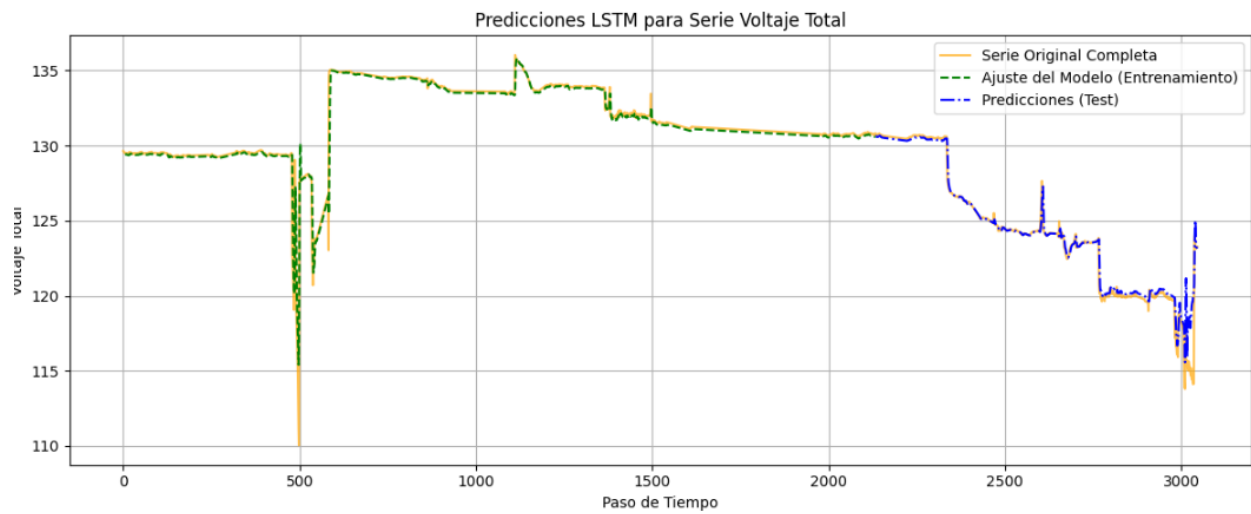


Figura D.1. Resultados de la aplicación de Modelo LSTM para serie Voltaje Total.

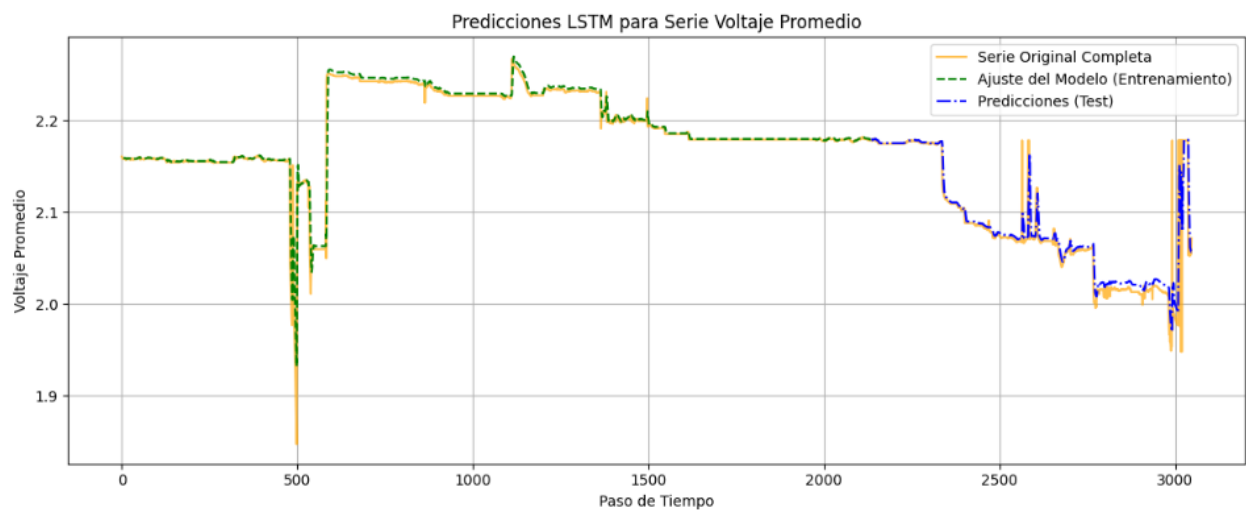


Figura D.2. Resultados de la aplicación de Modelo LSTM para serie Voltaje Promedio.

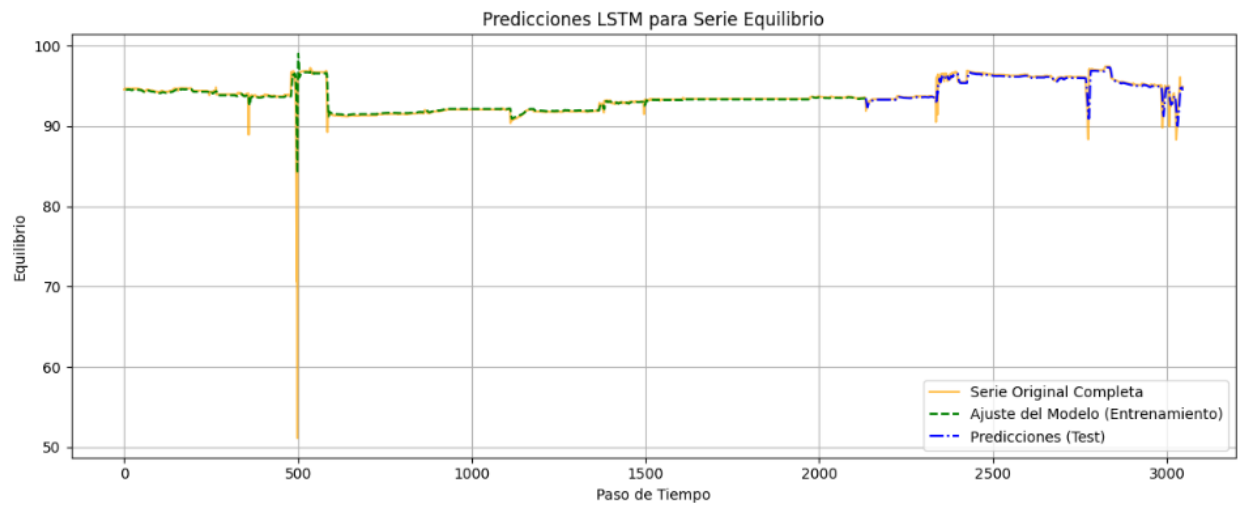


Figura D.3. Resultados de la aplicación de Modelo LSTM para serie Equilibrio.

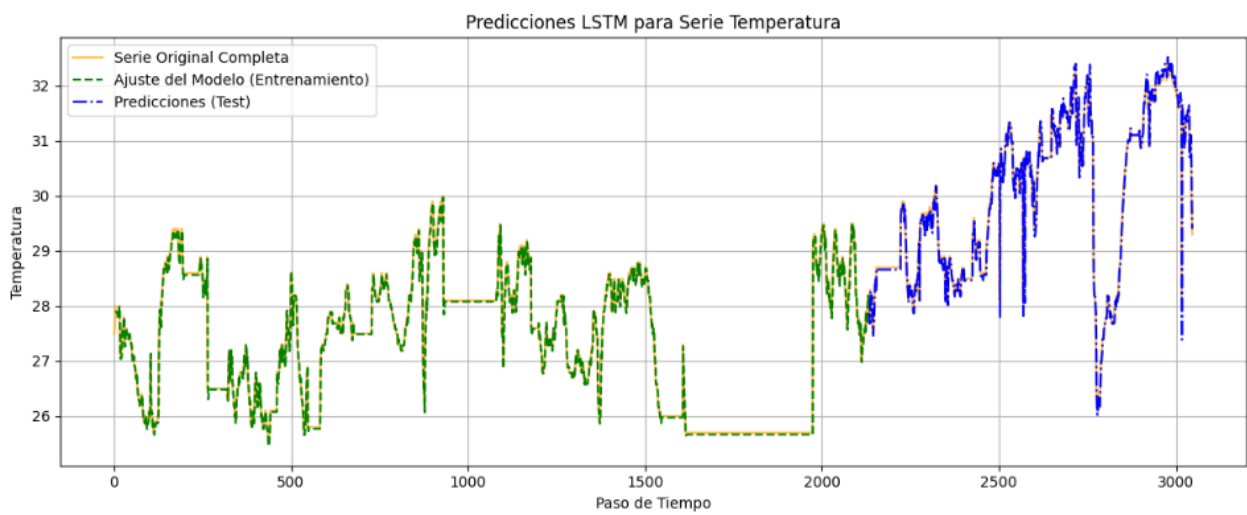


Figura D.4. Resultados de la aplicación de Modelo LSTM para serie Temperatura Promedio.

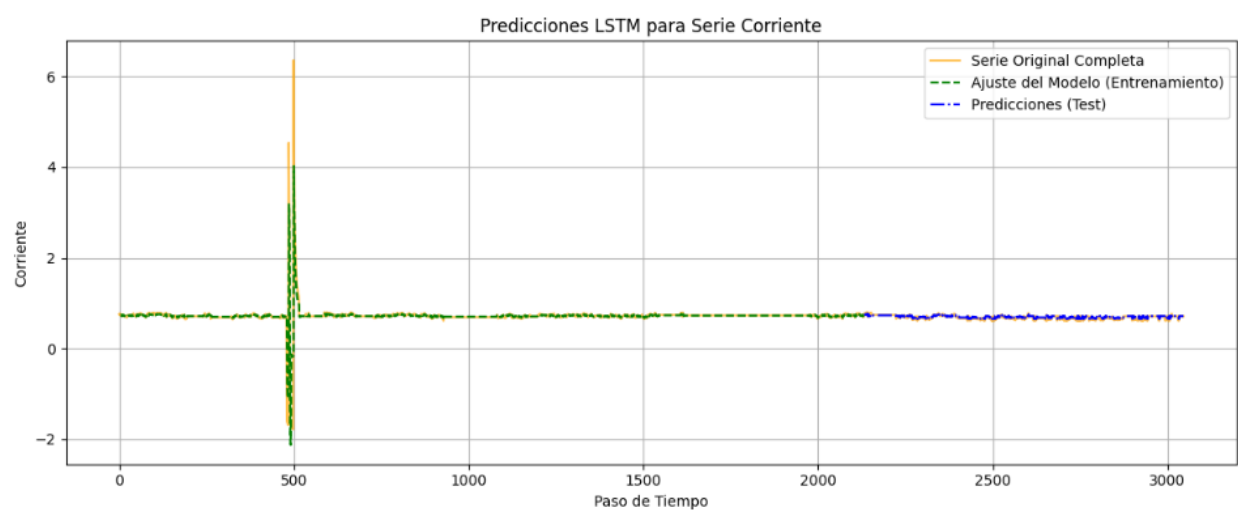


Figura D.5. Resultados de la aplicación de Modelo LSTM para serie Corriente.