

TECNOLÓGICO NACIONAL DE MEXICO

Instituto Tecnológico de Culiacán



TITULACIÓN INTEGRAL

Opción:

TESIS PROFESIONAL

NOMBRE DEL PROYECTO:

Desarrollo de modelos predictivos para predicción del voltaje de una celda de combustible microbiana de sedimentos.

ALUMNO:

Marco Antonio Aranda Sainz

CARRERA:

Ing. Mecatrónica

Num. De Control:

20170369

Culiacán, Sinaloa 2025



Instituto Tecnológico de Culiacán

DIVISION DE ESTUDIOS PROFESIONALES
ASUNTO: **Autorización de Impresión**

Culiacán, Sinaloa, 12/08/2025 16:26

C. MARCO ANTONIO ARANDA SAINZ
PASANTE(S) DE LA CARRERA DE INGENIERÍA MECATRÓNICA
P R E S E N T E.

De acuerdo con fallo emitido por la Comisión Revisora Integrada por:

Cédula Profesional

Presidente:	Doctorado en Ciencias en Ingeniería Eléctrica Opción en Sistemas Computacionales Héctor Rodríguez Rangel	9972835
Secretario:	Doctorado en Ingeniería y Ciencias Aplicadas (Tecnología Química) Blenda Ramírez Pereda	12152006
Vocal:	Maestría en Informática Aplicada Rogelio Prieto Alvarado	13976653

Y considerando que reúne los requisitos establecidos en la Opción: **TITULACIÓN INTEGRAL: TESIS PROFESIONAL: "DESARROLLO DE MODELOS PREDICTIVOS PARA PREDICCIÓN DEL VOLTAJE DE UNA CELDA DE COMBUSTIBLE MICROBIANA DE SEDIMENTOS"**, damos a usted nuestra autorización para que proceda a imprimir su trabajo profesional.

Sin otro particular, quedo de usted.

ATENTAMENTE
Excelencia en Educación Tecnológica®

LIC. CONCEPCIÓN MENDOZA ROSALES
JEFA DE LA DIVISIÓN DE ESTUDIOS PROFESIONALES

 **Educación**
Secretaría de Educación Pública
INSTITUTO TECNOLÓGICO DE CULIACÁN
DIRECCIÓN



2025
Año de
La Mujer
Indígena

Juan de Dios Bátiz 310 Pte. Col. Guadalupe C.P. 80220
Culiacán, Sinaloa. Tel. 667-454-0100
tecnm.mx www.culiacan.tecnm.mx



SEP
TNM
INSTITUTO TECNOLÓGICO DE CULIACÁN



Desarrollo de modelos predictivos para predicción del voltaje de una celda de combustible microbiana de sedimentos.

TESIS

PRESENTADA ANTE EL DEPARTAMENTO ACADÉMICO DE METAL MECÁNICA

POR:

MARCO ANTONIO ARANDA SAINZ

LICENCIATURA EN INGENIERÍA MECATRÓNICA

DIRECTOR DE TESIS:

DR. HÉCTOR RODRÍGUEZ RANGEL

CODIRECTOR DE TESIS:

MTRO. ROGELIO PRIETO ALVARADO

CULIACÁN, SINALOA

13 de agosto del 2025

Dedicatoria

Dedico este trabajo a mis padres Mirna Selene Sainz Sanchez y Marco Antonio Aranda Quintero, quienes siempre me han apoyado incluso en los momentos más difíciles siendo una brújula en mi vida y siempre me han apoyado en mis decisiones día con día.

Agradecimientos

Agradezco a mis **padres y hermana** por su apoyo incondicional durante mis estudios.

Al **Instituto Tecnológico de Culiacán** y los profesores que me formaron durante mi licenciatura.

A mis compañeros **Jordi Benitez y Kassandra Anaya** por su contribución en este trabajo y amistad.

Al **Dr Héctor Rodríguez** por brindarme su apoyo y guía a través del desarrollo de este trabajo.

Al **Mtro. Rogelio Prieto** por siempre compartirme su experiencia laboral como personal.

A la **Dra Blenda Pereda** por estar siempre a disposición y ofrecerme su apoyo en todo momento.

Resumen

Las celdas de combustible microbianas de sedimentos (CCMS) representan una alternativa prometedora para la producción de energía limpia y sostenible, esto frente a otras fuentes de energías limpias las cuales su fabricación trae consigo desechos y ser invasivas al medio ambiente. En esta tesis se plantea desarrollar el modelado del comportamiento del voltaje de las CCMS a través de herramientas de aprendizaje automático y profundo. Para ello se probaron distintos enfoques de regresión, así como pronóstico, utilizando K-Vecinos Más Cercanos (KNN), XGBoost y redes neuronales LSTM. Los datos utilizados fueron preprocesados por técnicas de imputación, normalización y detección de valores atípicos, los cuales se utilizaron para probar distintas configuraciones de hiperparámetros en los modelos de regresión, así como distintas ventanas en el caso de los modelos de pronóstico. Los resultados mostraron que el modelo LSTM obtuvo el mejor desempeño predictivo al lograr mimetizar correctamente el funcionamiento del voltaje de las CCMS. Este trabajo de investigación contribuye en el avance de la optimización de estos sistemas electroquímicos, demostrando la capacidad de algoritmos de aprendizaje automático como una herramienta para el estudio y diseño de CCMS con mayor eficiencia energética.

Palabras Clave

- Inteligencia artificial
- Aprendizaje automático
- Aprendizaje profundo
- Optimización de hiperparámetros
- Celdas de combustible microbianas de sedimentos
- Series de tiempo

Índice general

Índice de figuras	X
Índice de tablas	XI
1. Introducción	1
1.1. Definición del problema	3
1.2. Hipótesis	4
1.3. Objetivos	5
1.4. Preguntas de investigación	5
1.5. Justificación	6
1.6. Estructura de la tesis	7
2. Marco Teórico	9
2.1. Celdas de Combustible Microbiano de Sedimentos	9
2.2. Sistemas dinámicos y su modelado	11
2.3. Principios de la inteligencia artificial	11
2.4. La IA conexionista	12
2.5. Aprendizaje automático	13
2.6. Algoritmos de regresión lineal y polinómica	14
2.7. Algoritmo k-vecinos más cercanos	15
2.8. Algoritmo árbol de decisión	16
2.9. Algoritmo bosque aleatorio	17
2.10. Algoritmo XGBoost	19
2.11. Aprendizaje profundo	19
2.12. Red neuronal recurrente	20
2.13. Red de memoria a corto-largo plazo	20
2.14. Redes neuronales convolucionales	22
2.15. Métricas de evaluación	22
2.16. Series temporales	23
2.17. Valores atípicos	23
2.18. Limpieza de datos	24
3. Estado del arte	26
3.1. Modelado del rendimiento de CCMS: del enfoque tradicional a los algoritmos de aprendizaje profundo	26

3.2.	Evolución del modelado de sistemas dinámicos	27
3.3.	Aplicación de modelos de aprendizaje automático para la predicción del rendimiento de una CCM	28
3.4.	Aplicación de modelos de aprendizaje automático para la predicción del rendimiento de una CCM	28
3.5.	Identificación de la brecha de conocimiento	29
4.	Metodología	30
4.1.	Obtención y preparación de datos	32
4.2.	Desarrollo y validación de modelos	34
4.2.1.	KNN para regresión	34
4.2.2.	XGBoost para regresión	35
4.2.3.	LSTM para pronóstico	36
4.2.4.	XGBoost para pronóstico	37
4.3.	Evaluación de modelos obtenidos	38
4.3.1.	Evaluación del modelo KNN para regresión	38
4.3.2.	Evaluación del modelo XGBoost para regresión	38
4.3.3.	Evaluación del modelo LSTM	39
4.3.4.	Evaluación del modelo XGBoost pronóstico	40
5.	Resultados	42
5.1.	Resultados del procesamiento de datos	42
5.2.	Resultados modelo KNN	45
5.3.	Resultados modelo XGboost	46
5.4.	Resultados modelo LSTM	47
5.5.	Resultados modelo XGBoost pronóstico	49
6.	Discusión de resultados	50
6.1.	Caracterización de los resultados de cada modelo obtenido	50
6.2.	Comparación entre los modelos obtenidos	54
6.3.	Comparación con otros modelos previos	55
7.	Conclusiones	56
7.1.	Conclusión	56
7.2.	Limitaciones y retos	56
7.3.	Aportaciones	57
7.4.	Trabajos a futuro	57
	Bibliografía	59

Índice de figuras

2.1.	Ejemplo de CCMS generado con ayuda de inteligencia artificial (OpenAI), basado en Feregrino-Rivas et al. (2023).	10
2.2.	Ejemplo de regresión lineal y polinómica. Elaboración propia de Marco Aranda, basada en Vinuesa, J. (2019).	15
2.3.	Ejemplo ilustrativo de algoritmo k-vecinos más cercanos. Creación propia, basada en James et al. (2023).	16
2.4.	Ejemplo de árbol de decisiones. Creación de Marco Aranda, basada en Roth (2016).	17
2.5.	Ejemplo de Random Forest. Creación de Marco Aranda basado en Sánchez-Reyes y Morales-Palma (2020).	18
2.6.	Diagrama de una unidad LSTM. Creación de Marco Aranda, basado en MatWorks,sf.	21
2.7.	Serie de tiempo de temperatura de una CCMS. Hecha por Marco Aranda.	23
4.1.	Metodología general para el desarrollo del proyecto. Hecha por Marco Aranda.	31
4.2.	Imagen representativa del procedimiento de descarga de archivos.	32
4.3.	Variables de interés obtenidas desde el CSV dado por la interfaz.	33
4.4.	Metodología de evaluación de los modelos de regresión	39
4.5.	Metodología de evaluación del modelo LSTM.	40
4.6.	Metodología de la evaluación del modelo XGBoost para pronóstico.	41
5.1.	Datos previos a primer filtrado.	42
5.2.	Datos seguidos del primer filtrado.	43
5.3.	Tabla comparativa, resultados de imputación y media móvil	44

Índice de tablas

2.1. Técnicas comunes para el tratamiento de problemas en los datos	25
5.1. Resultados durante entrenamiento de KNN con el escalador MinMaxScaler .	45
5.2. Resultados durante entrenamiento de KNN con el escalador StandardScaler .	45
5.3. Resultados durante entrenamiento de KNN con el escalador RobustScaler . .	46
5.4. Resultados de validación mediante el nuevo dataset para los 3 escaladores utilizados	46
5.5. Resultados durante entrenamiento de XGBoost con el escalador MinMaxScaler	46
5.6. Resultados durante entrenamiento de XGBoost con el escalador RobustScaler	46
5.7. Resultados durante entrenamiento de XGBoost con el escalador StandardScaler	47
5.8. Resultados de validación mediante el nuevo dataset para los 3 escaladores utilizados	47
5.9. Resultados de prueba del conjunto [14,1] durante el entrenamiento para los 3 escaladores	47
5.10. Resultados de prueba del conjunto [4,1] durante el entrenamiento para los 3 escaladores	48
5.11. Resultados de validación del conjunto [14,1] mediante el nuevo dataset para los 3 escaladores	48
5.12. Resultados de validación del conjunto [4,1] mediante el nuevo dataset para los 3 escaladores	48
5.13. Resultados de pruebas de los distintos conjuntos de ventanas durante el en- trenamiento	49
5.14. Resultados de validación de los distintos conjuntos de ventanas durante el entrenamiento	49
5.15. Resultados de validación de los distintos conjuntos de ventanas mediante el nuevo dataset	49
6.1. Métricas de evaluación del modelo KNN durante la validación con el <i>dataset</i> nuevo	50
6.2. Configuración del modelo KNN	51
6.3. Métricas de evaluación del modelo XGBoost durante la validación con el <i>dataset</i> nuevo	51
6.4. Configuración del modelo XGBoost	51
6.5. Métricas de evaluación del modelo LSTM durante la validación con el <i>data-</i> <i>set</i> nuevo	52

6.6.	Configuración del modelo LSTM	52
6.7.	Métricas de evaluación del modelo XGBoost pronóstico durante la validación con el <i>dataset</i> nuevo	53
6.8.	Configuración del modelo XGBoost pronóstico	53
6.9.	Comparativa de variables de entrada en el entrenamiento y ventanas de los modelos LSTM y XGBoost para pronóstico	53
6.10.	Comparativa de resultados de evaluación de los distintos modelos	54

Capítulo 1

Introducción

La energía eléctrica es un recurso fundamental para la sociedad actual, ya que es requerida prácticamente para todas las actividades cotidianas, desde la conservación de alimentos hasta el transporte de productos y la manufacturación de estos. Sin embargo, debido a la creciente demanda de dicho recurso, el ser humano se ha vuelto dependiente de recursos no renovables como lo son el petróleo, gas natural y carbón. Frente a esta situación de alta demanda y consumo de recursos que producen daños irreversibles al medio ambiente, la transición hacia fuentes de energía renovables se ha convertido en una necesidad para suplir la demanda que incrementa año con año, además de mitigar el daño al medio ambiente causado por el consumo de fuentes no renovables. En esta perspectiva de cambio, las Celdas de Combustible Microbianas de sedimentos (CCMS) surgen como una propuesta innovadora para lograr esta transición hacia fuentes de energía limpias y sostenibles.

Las CCMS son sistemas bioelectroquímicos con la capacidad de generar energía eléctrica a partir del metabolismo de microorganismos que habitan en entornos de agua. Estas celdas están construidas por cuatro elementos principales: el ánodo, el cátodo, el sedimento y el electrolito. El ánodo es un electrodo el cual se inserta en el sedimento (rico en materia orgánica), además que el cátodo se sitúa a la misma altura límite del electrolito. En este ambiente, los microorganismos oxidan la materia orgánica concentrada en el sedimento, liberando electrones que son transferidos al cátodo a través de un circuito externo (normalmente elaborado a partir de alambre de acero inoxidable), generando así electricidad. (Feregrino-Rivas et al., 2023, Lee et al., 2015). Esta generación de energía es realizada mediante los procesos exoelectrogénicos que ocurren naturalmente en el sedimento a través de los microorganismos

que habitan en él, donde la materia orgánica actúa como sustrato prácticamente inagotable en el caso de ambientes acuáticos naturales.(Hubenova et al., 2024).

El estudio de las CCMS ha tomado fuerza en los últimos años debido a su potencial como fuente de energía limpia y no invasiva. Si bien, la energía generada por estas celdas es relativamente baja (alrededor de 700-800mv en condiciones de sequía con electrolito de lluvia), la capacidad de operar de forma continua y autónoma de las CCMS las vuelve una herramienta de interés para su uso en sensores de bajo consumo en zonas remotas o de difícil acceso. Además, las CCMS permiten la remoción de contaminantes en los sedimentos, lo que aporta beneficios adicionales a los ecosistemas donde se implementan; esto las diferencia de las celdas de combustible microbianas (CCM) las cuales no poseen dicha capacidad (Rahimnejad et al., 2015).

Entre las principales ventajas de las CCMS se encuentra su bajo costo de fabricación, esto es gracias a que sus materiales no son complejos ni requieren de procesos industriales para su elaboración, lo que presenta una clara ventaja en el ámbito de aprovechamiento de recursos en comparación con otras fuentes de energía renovables, como lo son los paneles solares (Rahimnejad et al., 2015). De igual forma, su incorporación al entorno no es invasiva, lo que favorece su integración a entornos naturales sin la necesidad de modificar el entorno para su aplicación. Además, las CCMS, poseen la capacidad de escalabilidad mediante arreglos de múltiples celdas para ampliar su capacidad de producción energética. Por otro lado, investigaciones recientes han demostrado su capacidad de producir biohidrógeno, lo cual amplía sus aplicaciones en el contexto de transición energética global.

Sin embargo, la implementación a gran escala de estas celdas presenta varios problemas. Uno de los más importantes es su baja eficiencia energética, la cual se encuentra relacionada principalmente con la escasa concentración de materia orgánica en los sedimentos (entre 0.4 % y 2.2 %) lo cual restringe la transferencia de materia en el ánodo (Lee et al., 2015). Además, el comportamiento dinámico de estos sistemas dificulta su modelado preciso, tanto con métodos clásicos como con técnicas modernas como el aprendizaje automático (Hossain et al., 2023). Otra problemática importante es la obtención de datos confiables, esto debido a que los métodos tradicionales de muestreo manual son propensos a errores humanos, además de una frecuencia de muestreo algo limitada.

Para enfrentar dichos problemas mencionados, este trabajo plantea el desarrollo de un modelo de aprendizaje automático capaz de mimetizar el comportamiento del voltaje de una CCMS. La base de datos que será utilizada se ha obtenido mediante un sistema automatizado de monitoreo y control previamente implementado, lo cual mejora la precisión y cantidad de datos disponibles, en comparación con otros trabajos. En adición, se aplicarán técnicas de limpieza y procesamiento de datos para garantizar la calidad de estos, para el entrenamiento de distintos modelos.

Para la realización de este trabajo, se llevó a cabo dos proyectos en paralelo a este, ambos enfocados en las CCMS; el primero desarrollado por Kassandra Velázquez Anaya en su tesis “sistema de monitoreo automatizado de la producción de energía de una celda de combustible microbiana de sedimentos”, donde como su título lo dice, desarrolló un sistema para la obtención de las variables de interés para este proyecto. Este estudio fue clave para la posible realización de este trabajo ya que fue el que permitió la adquisición de datos para desarrollar esta tesis. Además, el segundo trabajo elaborado en paralelo fue el de Jordi Ariel Benítez Ibarra el cual desarrolló su tesis con el título “Desarrollo e implementación de un sistema de control para regulación de temperatura y nivel de agua por goteo en una celda de combustible microbiana de sedimentos” el cual permitió mantener de forma constante la cantidad de electrolito, así como la temperatura de la CCMS estudiada para mejorar la estabilidad del sistema y con ello el estudio de la celda.

En resumen, este trabajo busca desarrollar y ofrecer una herramienta que contribuya en la investigación de las CCMS y con esto aportar conocimiento sobre las CCMS, para facilitar su integración como fuente de energía en entornos reales.

1.1. Definición del problema

Durante los últimos años ha incrementado la crisis energética global y la urgencia de ralentizar el cambio climático; debido a esto, han surgido iniciativas como la Agenda 2030 para el Desarrollo Sostenible, adoptada por las Naciones Unidas, 2015, septiembre, que establece varios objetivos, entre ellos, el ODS 7: Energía asequible y no contaminante. Dicho objetivo busca impulsar el uso de energías renovables accesibles para todos. Este pensamiento sigue

en pie, como señala la Organización Internacional del Trabajo (OIT), 2022, donde el cambio climático ha impulsado a reducir el uso de combustibles fósiles y, con ello, el desarrollo de nuevas fuentes de energía renovables.

Debido a la búsqueda de estas energías limpias, las Celdas de Combustible Microbianas de Sedimentos (CCMS) han crecido como una propuesta viable capaz de generar electricidad, con cualidades positivas como su bajo impacto ambiental, ser poco invasivas y su potencial para aplicaciones en zonas remotas.

Además, la escasez de *datasets* disponibles limita la capacidad de entrenar modelos de aprendizaje automático que posean la capacidad de generalizar el sistema. Esto se debe a que los pocos datos, al ser recopilados de forma manual, presentan variabilidad caótica, así como ruido y errores de medición; por lo cual es necesario aplicar técnicas de limpieza, filtrado y suavizado de datos para asegurar la calidad de la información.

Por consecuencia, la predicción del voltaje generado por una CCMS resulta ser algo compleja y poco fiable utilizando métodos tradicionales; esto impide determinar con precisión las condiciones óptimas para la máxima eficiencia energética sin experimentación empírica extensa. Además, es difícil anticipar cómo se comportará la celda ante diferentes condiciones, lo cual produce un obstáculo para la adopción de estos sistemas como una fuente de energía.

Actualmente, existe una carencia de modelos predictivos que sean capaces de capturar correctamente la complejidad bioelectroquímica y el funcionamiento no lineal que rige la generación de voltaje de las celdas de combustible microbianas de sedimentos. Por ende, el problema que aborda esta tesis es la ausencia de un modelo predictivo robusto y confiable que permita determinar el voltaje de salida de las celdas mencionadas y, con este, tener una herramienta para mejorar la eficiencia de estos sistemas.

1.2. Hipótesis

El uso de algoritmos de aprendizaje automático con los valores de entrada temperatura, salinidad, conductividad y tiempo, correlacionados con la variable de voltaje a predecir, permitirá la construcción de un modelo capaz de predecir el voltaje de una CCMS con alta precisión.

1.3. Objetivos

Desarrollar algoritmos de predicción basados en modelos de aprendizaje automático, para predecir el voltaje generado por una celda de combustible microbiana de sedimentos, a partir de datos adquiridos por un sistema de monitoreo y control automatizado de las variables fisicoquímicas de interés.

Objetivos específicos:

- Obtener un conjunto de datos de las variables temperatura, conductividad, salinidad, tiempo y voltaje, procedentes de un sistema automatizado de monitoreo y control de una CCMS para su preprocesamiento y limpieza para su utilización en modelos de aprendizaje supervisado.
- Preprocesar y limpiar el conjunto de datos obtenidos a través del sistema automatizado para su uso en el desarrollo de los diversos algoritmos de aprendizaje supervisado.
- Desarrollo y validación del rendimiento de diversos algoritmos de aprendizaje supervisado, mediante el uso de métricas de evaluación estadísticas.
- Seleccionar el modelo con mayor rendimiento predictivo considerando los resultados obtenidos de las métricas de evaluación y las predicciones generadas sobre la CCMS.
- Validar el modelo seleccionado a través del *dataset* de prueba para determinar su capacidad de aplicación mediante técnicas de computación, garantizando su utilidad para predecir el voltaje de la CCMS.

1.4. Preguntas de investigación

En este apartado se formulan las preguntas de investigación que orientan el desarrollo de este proyecto, sirviendo como guía para alcanzar los objetivos planteados.

- ¿Cuál es el modelo con mayor precisión entre modelos de aprendizaje supervisado para predecir el voltaje generado por una celda de combustible microbiana de sedimentos?

- ¿Cuál es la combinación de parámetros de entrada para obtener el modelo con mejor evaluación?
- ¿Cuál es la mejor arquitectura para obtener el mejor modelo predictivo con mejor evaluación?
- ¿Cuál es la precisión y capacidad de generalizar del mejor seleccionado después de validarlo con datos no vistos durante su entrenamiento?

1.5. Justificación

El uso de energías renovables ha sido un tema de gran interés, esto debido al incremento de las temperaturas año con año, causado por el empleo de energías no renovables y el cambio climático; si bien existen varios tipos de energías renovables, no están exentas de su ciclo de vida, ya que requieren de mantenimiento constante, como es el caso de los paneles solares. Además de su alto coste de mantenimiento, también requieren de condiciones puntuales para su instalación, lo cual trae consigo una posible invasión al entorno natural en algunos casos. Es por esto último que, para casos donde se requiere una fuente de energía tanto limpia como no invasiva, las celdas de combustible microbianas se han presentado como una opción viable debido a su tamaño reducido, así como a una estructura adaptable al entorno.

Sin embargo, el aprovechamiento de estas fuentes limpias plantea un reto debido a su complejidad inherente al ser orgánicas, ya que su energía, a diferencia de las fuentes convencionales, no es lineal debido a procesos orgánicos que se desarrollan dentro de estas; estos procesos, por ende, afectan el rendimiento energético de estas fuentes, lo cual vuelve difícil predecir mediante métodos tradicionales, lo que obstaculiza una implementación confiable en sistemas reales.

Debido a las limitaciones que presentan los métodos de predicción tradicionales, en esta tesis se propone el desarrollo y validación de un modelo predictivo para el voltaje de una celda de combustible microbiana de sedimentos, la cual es un subtipo de las primeras celdas mencionadas. Por consecuencia, el desarrollo de dicho modelo permitirá reducir la investigación empírica y, con esto, disminuir tanto costos como tiempo en el desarrollo de estas

tecnologías, además de profundizar en el entendimiento de estos sistemas complejos, los cuales recaen en procesos orgánicos que rigen la producción de la energía eléctrica.

En consecuencia, con el desarrollo de este trabajo se proporciona una herramienta para la comprensión e implementación eficiente de estos sistemas de celdas de combustible microbianas, lo cual impulsará la transición hacia el uso de energías renovables.

1.6. Estructura de la tesis

Este trabajo está dividido en 7 capítulos organizados de la siguiente manera:

- Capítulo 1- Introducción: Se define el problema, la hipótesis, el objetivo, los objetivos específicos, las preguntas de investigación y la justificación de este trabajo.
- Capítulo 2- Marco teórico: Se presentan los conceptos clave sobre las CCMS, como su principio básico de funcionamiento, además de los principios Inteligencia artificial, así como los conocimientos que se requirieron para elaborar ese trabajo, desde aprendizaje automático supervisado y aprendizaje profundo hasta técnicas de limpieza de datos.
- Capítulo 3- Estado del arte: se presentan diversos trabajos que se han desarrollado previamente y que se encuentran relacionados a las CCMS, dando sus resultados obtenidos y haciendo una comparación entre las limitantes que tuvieron y los cambios que propone este trabajo.
- Capítulo 4- Metodología: Se muestra el proceso que se llevó a cabo para la realización de este trabajo, desde la limpieza de datos, así como las pruebas como los distintos modelos y la evaluación de cada uno.
- Capítulo 5- Resultados: Se presentan los resultados tras el procesado de los datos, así como los obtenidos de cada modelo tras el entrenamiento la validación de estos.

- Capítulo 6- Discusión de resultados: se explica los resultados obtenidos de cada modelo y sus causas, además de su comparación entre estos y trabajos previos.
- Capítulo 7- Conclusión: Se presenta un resumen del mejor modelo obtenido en base al criterio del autor, así como las aportaciones que presenta a la ciencia y propuestas para trabajos futuros en base a los resultados obtenidos en este proyecto.

Capítulo 2

Marco Teórico

En este capítulo se presentan los conceptos claves y temas requeridos para la comprensión de este trabajo. Este apartado aborda conocimientos básicos para el entendimiento de las CCMS, así como la comprensión de la evolución de la Inteligencia Artificial, así como los distintos algoritmos que se han desarrollado a través del tiempo y las técnicas de limpieza de datos requeridas para el funcionamiento óptimo de estos modelos.

2.1. Celdas de Combustible Microbiano de Sedimentos

Una Celda de Combustible Microbiana de Sedimentos (CCMS) es un sistema bioelectroquímico con la capacidad de producir energía eléctrica debido a las reacciones exoelectrogenéticas que producen los microorganismos. Esta generación de energía es debido a que, al ser la celda un biorreactor electroquímico en donde suceden reacciones entre el sustrato (oxidación) y los microorganismos, esto genera electricidad producto del metabolismo microbiano. Dicho procedimiento se puede observar de forma clara en la siguiente Figura 2.1 (Feregrino-Rivas et al., 2023). La producción de esta energía se logra a través de los microorganismos que aprovechan los recursos orgánicos presentes en los sedimentos. Dichos actuadores presentes en el sedimento catabolizan el sustrato orgánico, liberando electrones y protones. Los electrones viajan al cátodo a través de un circuito externo como se observa en la Figura 2.1. En el cátodo, los electrones reaccionan con el electrolito completando el circuito y así generando electricidad.

Otra parte importante de estas celdas es el sustrato, este se encuentra compuesto de car-

bono orgánico contenido en el sedimento, cuyo contenido es utilizado como “alimento” para los microorganismos (Feregrino-Rivas et al., 2023).

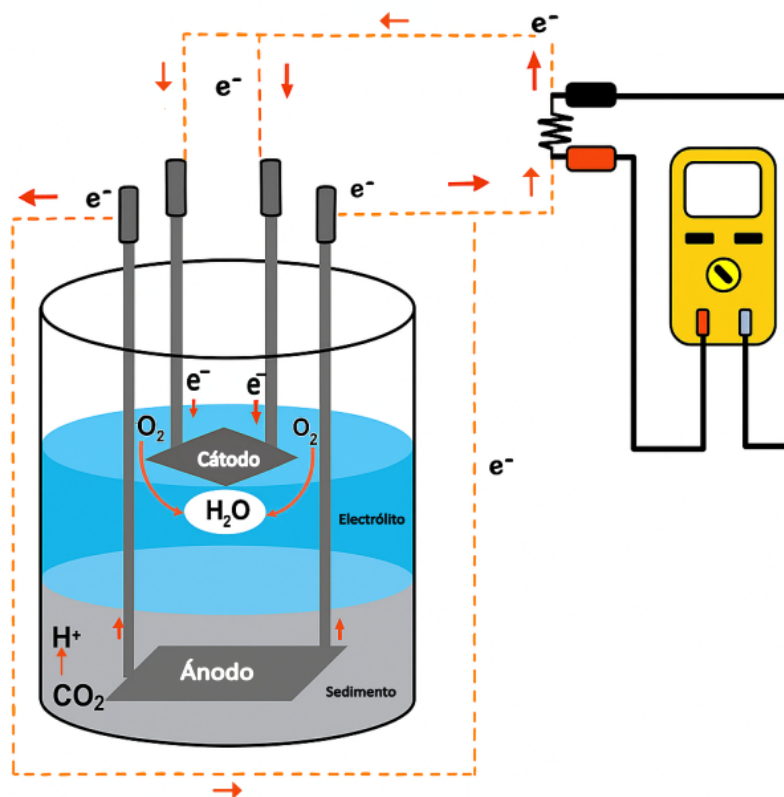


Figura 2.1: Ejemplo de CCMS generado con ayuda de inteligencia artificial (OpenAI), basado en Feregrino-Rivas et al. (2023).

En adición a lo anterior, es necesario saber que los elementos que componen estas celdas consiste principalmente en cuatro partes, el ánodo, donde se generarán los electrones a través de las reacciones que producirán los microorganismos que lo colonizarán; el cátodo, donde se recibirán los electrones; el sedimento, donde se encuentran la materia orgánica necesaria para los procesos metabólicos de los actuadores del sistema; el electrolito, el cual permite el flujo de los iones algo clave para cerrar el circuito (Feregrino-Rivas et al., 2023).

Estos componentes son claves para el funcionamiento correcto de estas celdas, las cuales pueden utilizarse a través de distintos arreglos para mejorar su capacidad de producción energía, donde con una sola celda es capaz de producir entre 700 y 800 milivoltios de acuerdo con el estudio de Feregrino-Rivas et al., 2023.

Debido a que las CCMS son sistemas bioelectroquímicos, en los cuales se encuentra una delimitada cantidad de microorganismos en sus procesos y poseen la capacidad de almacenar energía, se les considera sistemas de tipo dinámicos (Kofman, 2012).

2.2. Sistemas dinámicos y su modelado

Un sistema dinámico se le considera a un conjunto de elementos el cual poseen la capacidad de almacenar energía o algún tipo de información, de dicha forma se les puede atribuir el hecho de tener memoria. Dentro de un sistema de este tipo se encuentran dos tipos de magnitudes que caracterizan el comportamiento de este, uno de ellos son los parámetros, los cuales son magnitudes constantes las cuales pueden ser en el caso de una CCMS, su volumen, la masa del sedimento, o incluso las dimensiones del cátodo y ánodo; por otro lado, se encuentran las variables, las cuales son estados que cambian con el tiempo, como lo puede ser el voltaje producido, o la salinidad y conductividad de dicha CCMS (Kofman, 2012)

Un modelo es una mimetización de forma simplificada del comportamiento de sistema, estos se realizan con el objetivo de poder experimentar de forma controlada mediante simulaciones del sistema, sin la necesidad de realizar dichos experimentos de forma física. Este tipo de prácticas ahorran tanto tiempo como dinero, al no depender de eventos reales para poner a prueba el sistema estudiado. Para la realización del modelado de un sistema existen dos formas, la primera mediante principios tanto físicos como analíticos, así como mediante la experimentación. Estos modelos se representan mediante ecuaciones matemáticas implementadas computacionalmente (Kofman, 2012)

2.3. Principios de la inteligencia artificial

En el año 1956, surgió el concepto de inteligencia artificial como lo conocemos ahora, donde se definió como un campo de estudio en una conferencia en Dartmouth, y en ese mismo año, dos de los participantes de dicha conferencia, Alan Newell y Herbert Simon, publicaron el primer modelo de inteligencia artificial llamado “Logic Theory Machine”. Este último consistió en que el programa podría construir expresiones cada vez más complejas a

partir de operaciones simples (Abeliuk & Gutiérrez, 2021).

La presentación de este algoritmo presentó la idea de un enfoque heurístico, el cual consiste en buscar la solución a un problema de forma empírica, buscando atajos que permitan reducir las tareas para llegar a un resultado más rápido (Abeliuk & Gutiérrez, 2021).

A partir del planteamiento de dicho enfoque, todas las tareas de resolución de problemas implicaron la toma de decisiones a partir de modelos en forma de árboles de decisiones, cuyas ramas se siguen para llegar a un resultado. Al desarrollo de estos algoritmos se le terminó por dar el nombre de inteligencia artificial simbólica, la cual consiste en condiciones y resultados (Abeliuk y Gutiérrez, 2021).

Sin embargo, una limitación de la IA simbólica es creer que la mente es un lenguaje de pensamiento sujeto a reglas y que todos los procesos que puedan llegar a suceder en la realidad siempre puedan ser las mismas condiciones. Por esto último, surgió paralelamente otro pensamiento con respecto a la IA (Carabantes López, 2015).

2.4. La IA conexionista

El pensamiento de cómo funciona la mente humana se ha visto desde distintas perspectivas. Por un lado, la IA simbólica ve la mente humana como una computadora que procesa símbolos. Por otro lado, la IA conexionista modela sistemas imitando el cerebro humano, el cual posee redes neuronales (Abeliuk y Gutiérrez, 2021).

De acuerdo con Abeliuk & Gutiérrez, 2021, en el año 1958 Frank Rosenblatt propuso el concepto de perceptrón, el cual en conjunto genera lo que es conocido hoy como redes neuronales. Esto se basó en la neurona de McCulloch-Pitts, la cual adquiría experiencia mediante un coeficiente llamado peso, que se le asignaba a la entrada de cada neurona.

Sin embargo, debido al bajo desarrollo de tecnologías especializadas para este tipo de tareas, el desarrollo científico se vio limitado. Esta situación cambió a principios de los 2000 con el desarrollo exponencial de las computadoras domésticas e industriales. Desde entonces, el desarrollo de estas tecnologías ha crecido rápidamente en los últimos años (Abeliuk & Gutiérrez, 2021).

2.5. Aprendizaje automático

Según López Peláez, 2023, el aprendizaje automático es una rama de la inteligencia artificial. A diferencia de las redes neuronales representadas en la IA conexionista o la lógica simbólica en la IA simbólica, este consiste en entrenar un modelo de tal forma que no requiera acceder a una base de datos para llegar a una resolución. Este entrenamiento parte de la idea de que, al aprender ciertos comportamientos con base en patrones, el modelo puede llegar a realizar conclusiones por sí mismo. Sin embargo, estas conclusiones no siempre son acertadas, y es por esto por lo que existen dos tipos de aprendizajes automáticos: el supervisado y el no supervisado.

La principal diferencia entre el aprendizaje supervisado y no supervisado radica en qué datos recibe. El primero necesita recibir datos etiquetados (con nombres), mientras que el segundo tipo no requiere de esto. Los modelos supervisados se centran en aprender las relaciones entre las entradas y las salidas, mientras que el no supervisado se utiliza para encontrar relaciones en los datos sin un orden definido. Un ejemplo de la aplicación de estos últimos modelos es el caso de Facebook, que te cataloga en un grupo de usuarios basándose en tus búsquedas y reacciones sin necesidad de definir dicho grupo como tal (Russell & Norvig, 2021).

De acuerdo con Russell y Norvig (2021), algunos algoritmos de aprendizaje supervisado son:

- Regresión lineal
- Regresión polinómica
- K-vecinos más cercanos
- Árboles de decisión

Por el otro lado, según con base en James et al. (2023), algunos algoritmos de aprendizaje no supervisado son

- Agrupación en clústeres mediante el modelo K-means
- Agrupamiento jerárquico (o Clústeres jerárquicos’')
- Mínimos cuadrados parciales

2.6. Algoritmos de regresión lineal y polinómica

Estos son un subconjunto de los algoritmos supervisados, los algoritmos de regresión permiten predecir una variable a partir de una entrada cuantitativa, esto es la regresión lineal. Para el caso de la regresión polinómica, también consiste en la predicción de una variable a partir de una entrada; sin embargo, con relaciones no lineales mediante el uso de funciones complejas, esto se presenta cuando la relación entre la entrada y la salida no puede representarse linealmente. En el caso de estos algoritmos se puede extrapolar a múltiples entradas que se correlacionan con una salida (de igual forma, variables cuantitativas) (Vinuesa, 2019).

Como se observa en la Figura 2.2, en la gráfica de la izquierda se tiene una regresión lineal, donde la línea verde representa las predicciones que tratan de ajustarse a los valores reales; sin embargo, en el caso de rangos de datos poco constantes este tipo de modelos no logra una predicción tan certera. Este caso es distinto para la segunda gráfica donde se tiene un ejemplo de regresión polinómica, donde la función del modelo permite un mejor seguimiento entre la predicción y los valores reales, ya que trabaja mejor con valores no constantes.

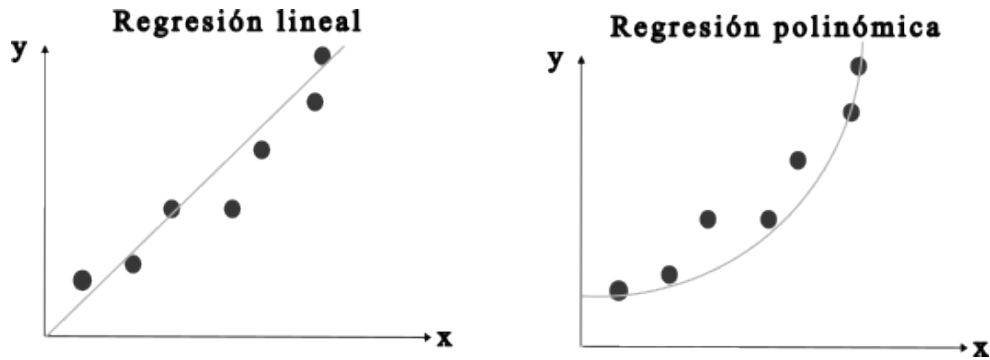


Figura 2.2: Ejemplo de regresión lineal y polinómica. Elaboración propia de Marco Aranda, basada en Vinuesa, J. (2019).

2.7. Algoritmo k-vecinos más cercanos

También conocido como KNN, por sus siglas en inglés *K-Nearest neighbors*, es un modelo no paramétrico utilizado mayormente en problemas de clasificación, aun así, también se puede utilizar para predicción. Su funcionamiento consiste en analizar la cercanía entre puntos dentro de un espacio de características, comparando un punto nuevo con puntos ya existentes. Al calcular las distancias Manhattan, euclidiana, etc.; Este algoritmo determina qué puntos están más cerca del nuevo generado, asignándole una etiqueta más común entre sus vecinos más cercanos. Esta metodología divide el espacio de decisiones en regiones definidas por la proximidad que existe entre los datos (James et al., 2023).

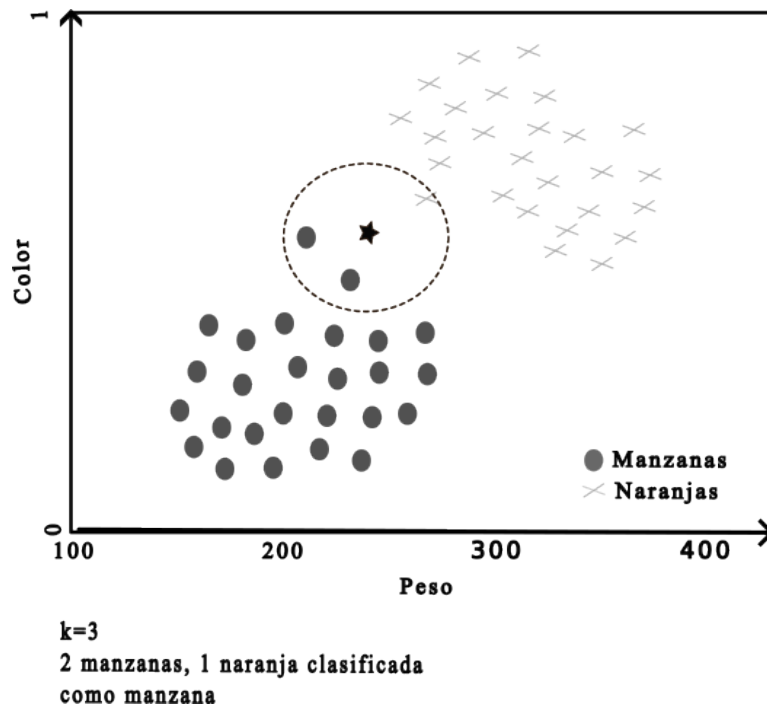


Figura 2.3: Ejemplo ilustrativo de algoritmo k-vecinos más cercanos. Creación propia, basada en James et al. (2023).

La Figura 2.3 muestra gráficamente el funcionamiento de un algoritmo k-vecinos más cercanos, para el caso de clasificación de frutas, donde se tienen dos características: peso (eje x) y color (eje y). La estrella negra representa una fruta con características conocidas (peso y color), sin embargo, se desconoce a qué clasificación pertenece, es ahí donde entra en función el algoritmo; donde se dibuja el círculo punteado y a partir de la cantidad de vecinos dentro de este, se determinará a qué clasificación pertenece. Para el caso de esta gráfica de los 3 vecinos más cercanos, 2 son manzanas y una es naranja, por lo cual el elemento de características conocidas se le clasificará como una manzana.

2.8. Algoritmo árbol de decisión

Los árboles de decisión son algoritmos de aprendizaje supervisados utilizados mayormente en tareas de clasificación y regresión, su funcionamiento consiste en construir una estructura jerárquica en forma de un árbol; dicho árbol está compuesto por una raíz, de la cual parten nodos internos que representan ideas basadas en condiciones, seguido de ramas

que representan posibles decisiones y hojas que simbolizan las salidas o predicciones finales del modelo (Russell & Norvig, 2021).

El nombre “árbol de decisión” como se observa en la Figura 2.4 que es un fragmento de un árbol de decisiones, es una estructura jerárquica donde cada división corresponde a una elección realizada por el modelo a partir de las condicionales generadas a través de las ramas y que dan como resultado un conjunto de datos segmentados (Russell & Norvig, 2021).

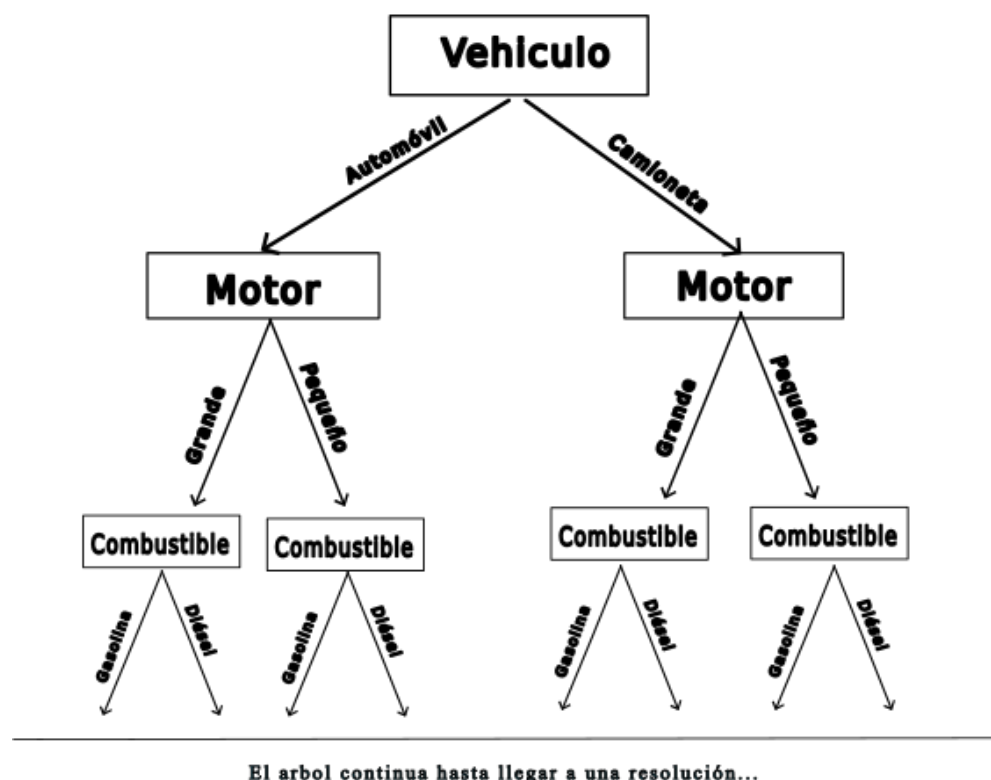


Figura 2.4: Ejemplo de árbol de decisiones. Creación de Marco Aranda, basada en Roth (2016).

2.9. Algoritmo bosque aleatorio

Conocido mayormente en inglés como *Random Forest*, es un algoritmo de aprendizaje supervisado que se basa en el uso de múltiples árboles de decisión para resolver problemas de clasificación y regresión. A diferencia del modelo de árbol de decisión donde solo se genera un árbol, en *Random Forest* se generan múltiples árboles (de ahí su nombre) donde

cada árbol es alimentado por un subconjunto de datos diferente. En tareas de clasificación, el resultado final se lleva a cabo mediante una votación entre todos los árboles; mientras que para tareas de regresión se utiliza el promedio de la suma de los resultados (James et al., 2023).

Una ventaja que presenta sobre otros modelos es que reduce el sobre ajuste, ya que, al ser múltiples árboles, el resultado no depende de uno solo, sino del resultado de todos. Esta lógica se puede observar en la siguiente ilustración, donde los árboles trabajan en paralelo con subconjuntos de datos diferentes (Kroese et al., 2024).

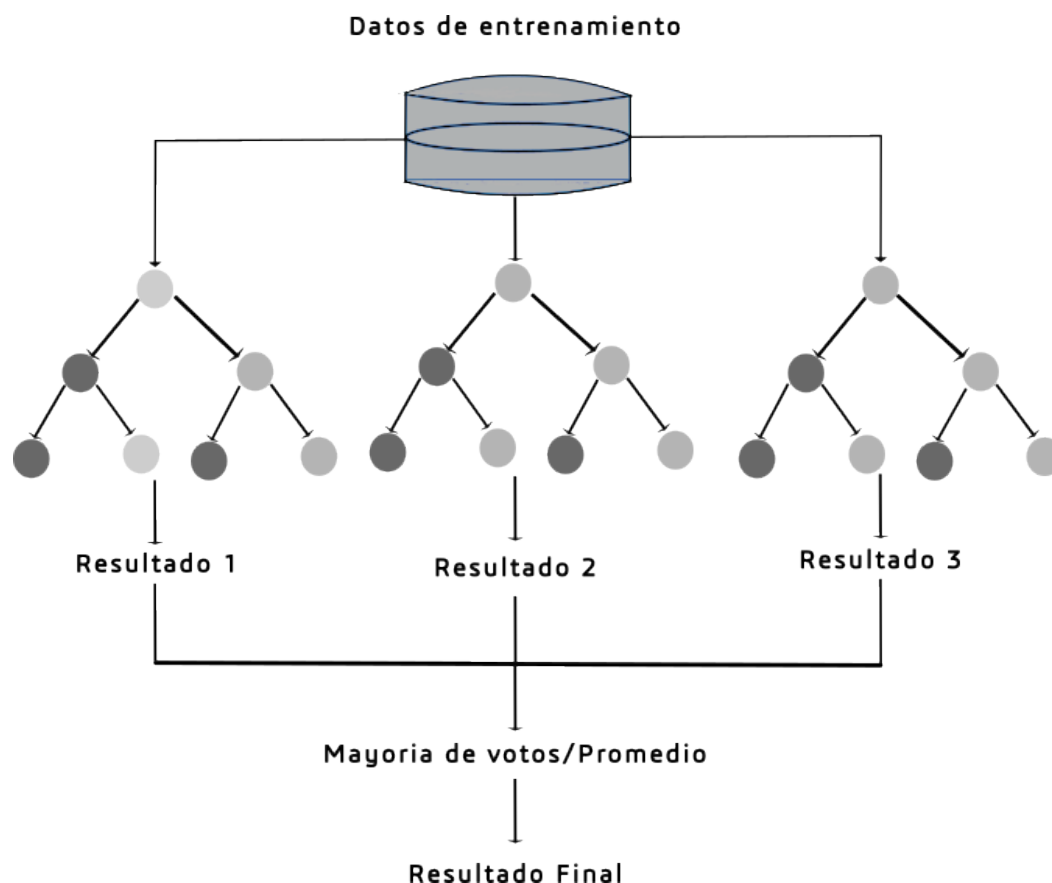


Figura 2.5: Ejemplo de Random Forest. Creación de Marco Aranda basado en Sánchez-Reyes y Morales-Palma (2020).

Además, este modelo es una extensión del método *bagging*, se emplea para reducir la varianza dentro de un conjunto de datos. Gracias a que cada árbol es entrenado con datos y características aleatorias, esto permite que los árboles que genere el modelo no estén correlacionados y así aumentar la estimación precisa de las predicciones (James et al., 2023).

2.10. Algoritmo XGBoost

De acuerdo con Russell y Norvig (2021), este es un algoritmo de aprendizaje supervisado que se basa en árboles de decisión. Si bien es similar a *Random Forest*, su característica principal es que los árboles que se generan son mejorados por la técnica de “potencia gradiente”, este es un método que mejora el rendimiento a través del descenso de gradiente.

A diferencia de otros algoritmos basados en árboles, XGBoost opera de manera secuencial y aditiva, esto quiere decir que construye aprendices débiles o modelos de forma progresiva, donde los siguientes árboles arreglan los errores que tuvieron los previos (Russell & Norvig, 2021).

Según Russell & Norvig, 2021, este modelo trabaja construyendo un árbol que genera las raíces y seguido de este se crean nuevos, los cuales con cada nueva iteración estos se construyen a partir de los errores residuales del modelo acumulado hasta el momento. El algoritmo calcula las diferencias entre el valor predicho y el real, con el cual construye una función de pérdida la cual se busca ir reduciendo con el paso de las iteraciones; esto logra un modelo que con el paso del tiempo se vuelve más preciso.

2.11. Aprendizaje profundo

El aprendizaje profundo es una rama del aprendizaje automático que se basa en redes neuronales con múltiples capas, diseñadas para procesar datos de forma similar al cerebro humano. Estas redes aprenden a partir de grandes volúmenes de información, ajustando pesos y sesgos para generar salidas precisas. A diferencia de otros enfoques, los modelos de aprendizaje profundo pueden operar sin necesidad de datos previamente etiquetados, por lo cual pueden trabajar en entornos no supervisados. Un elemento clave de este tipo de aprendizaje es la retropropagación, un proceso que permite medir el error entre las predicciones y los valores reales, y luego ajustar los pesos y sesgos retrocediendo a través de las capas del modelo, desde una salida hasta la entrada (Gómez, 2025)

2.12. Red neuronal recurrente

Conocida como RNN, este es un modelo de redes neuronales entrenadas con datos secuenciales en el tiempo. A diferencia de otros algoritmos basados en redes neuronales, las RNN se caracterizan por su capacidad de “memorizar”, ya que utilizan información previa para calcular valores futuros (Russell & Norvig, 2021).

Esto quiere decir que este tipo de red depende mucho de los datos anteriores para predecir los nuevos, este modelo, por ende, sirve para tareas de pronóstico, además estos modelos generan el mismo peso en cada capa de la red.

De acuerdo con James et al., 2023, algunas funciones de activación usadas en estos algoritmos son las siguientes:

- Función sigmoidea, esta permite interpretar la salida como una probabilidad o para controlar las puertas que determinan si se retiene la información o se olvida.
- Función Tanh, esta función entrega resultados alrededor del eje 0, lo que corresponde a un margen de corrección y con ello, mejores resultados.
- Función ReLU, este puede entregar ajustes de pesos algo descontrolados por lo cual, su versión base no es la mejor.

2.13. Red de memoria a corto-largo plazo

Conocidas como LSTM por su nombre en inglés (*Long-Short Term Memory*), es un modelo de aprendizaje profundo basado en las redes neuronales recurrentes, como se mencionó estos modelos poseen la capacidad de “memorizar”, ya que utilizan información previa para calcular valores futuros, lo que los hace muy útiles para tareas como pronóstico. Sin embargo, para aplicaciones reales, las RNN se encuentran limitadas para aprender relaciones a largo plazo. Dicha limitación es causada debido al aplicar retropropagación se produce un desvanecimiento, lo que reduce el margen de los pesos a muy bajos o altos, esto último reduce la eficiencia de los modelos RNN para aplicaciones prácticas (Russell & Norvig, 2021).

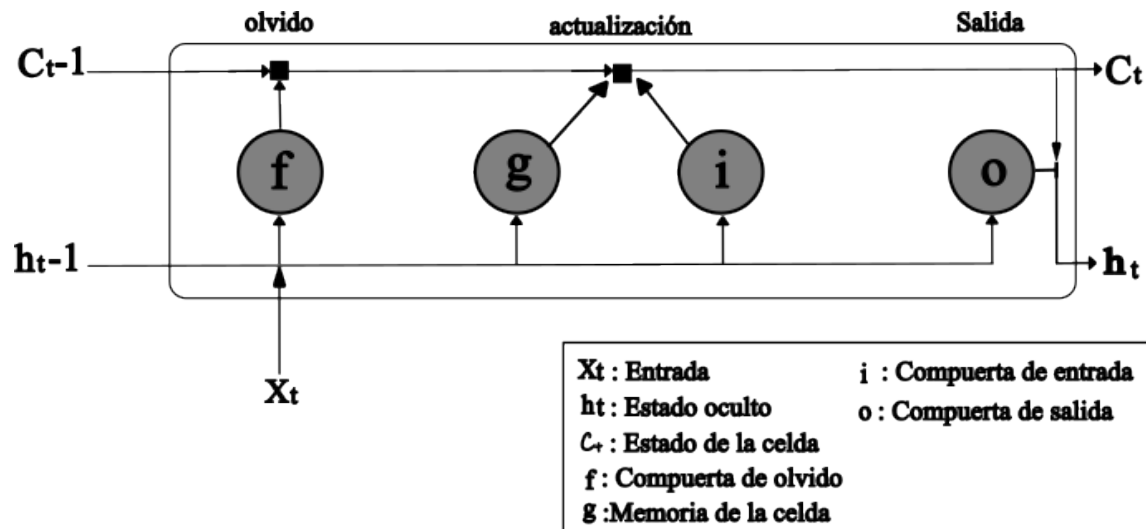


Figura 2.6: Diagrama de una unidad LSTM. Creación de Marco Aranda, basado en MatWorks,sf.

En la Figura 2.6 se observa un diagrama representativo del flujo de datos a través de una unidad LSTM, estos son las bases de este tipo de RNN, donde además del estado oculto de las tradicionales, las LSTM poseen una celda de memoria, además de una puerta de olvido, así como sus entradas y salida. Las puertas adicionales que poseen este modelo le permiten aprender con mayor eficacia las relaciones de datos a largos plazos. Los pesos de la entrada permiten controlar cuando ingresa un nuevo valor a la unidad LSTM, así como las compuertas de olvido y salida, controlan cuanto permanece un valor dentro del sistema, además de determinar hasta cuando es útil ese valor para calcular la activación de salida (Russell & Norvig, 2021).

En pocas palabras, el modelo LSTM, posee la capacidad de determinar hasta qué punto un valor es útil para el aprendizaje y cuando deshacerse de él mediante la compuerta de olvido e ingresar un nuevo dato a través de la compuerta de entrada, esto es de gran utilidad para predicciones a largo plazo donde se requiere determinar estrechas relaciones entre los valores.

Por otro lado, una desventaja de este modelo es que se requiere un poder de cómputo considerable para alcanzar un buen desempeño; Sin embargo, gracias las actuales tarjetas gráficas, el poder computacional ha aumentado considerablemente en los últimos años, lo que ha traído consigo aplicaciones más poderosas para este tipo de modelos (Russell & Norvig, 2021).

2.14. Redes neuronales convolucionales

Este tipo de algoritmo se le conoce mayormente como CCN, estas se utilizan comúnmente para tareas de clasificación y visión artificial. Este modelo aprovecha el álgebra lineal y la multiplicación de matrices para encontrar relaciones dentro de imágenes. Este tipo de tareas se realizan gracias a que tiene 3 principales capas, las cuales de acuerdo con Russell & Norvig, 2021, son:

- Capa convolucional, aquí se utilizan los valores de entrada, así como su mapa de características y se realizan la mayoría de los cálculos.
- Capa convolucional adicional, esta separa los componentes de los valores de entrada y crea una jerarquía de características.
- Capa de agrupamiento, en esta se reduce el número de entrada, esto para reducir la complejidad.
- Capa totalmente conectada, en esta se conectan los nodos de las capas ocultas en una sola, además se realiza la clasificación.

2.15. Métricas de evaluación

Dentro de las métricas de evaluación para modelos de aprendizaje supervisado, una de las más usadas es la R^2 , conocida como coeficiente de determinación. Esta función permite cuantificar qué porcentaje de la varianza del objetivo a predecir es explicada por el algoritmo. Si esta métrica tiene el valor de 1, esto significa que el modelo explica el 100 % de la varianza observada; sin embargo, esto también puede significar un sobreentrenamiento, ya que no tendría flexibilidad suficiente. Por otro lado, un valor como 0 significa que el modelo no está tomando en cuenta los valores de entrada y no posee la capacidad de predecir la varianza del modelo (scikit-learn developers, n.d.).

Además, de acuerdo con Hodson, 2022, el error absoluto medio (MAE) mide la magnitud media del error, al ser absoluto el resultado siempre será positivo, Sin embargo, al obviar el

signo del error esto puede causar problemas a la hora de evaluar. Es por esto por lo que se usa en conjunto con el error cuadrático promedio (MSE), dicho parámetro posee la ventaja de al no ser absoluto, se puede utilizar para varios procesos matemáticos a diferencia del MAE (Chai & Draxler, 2014).

2.16. Series temporales

De acuerdo con Russell y Norvig (2021), una serie de tiempo es un grupo de datos ordenados en relación con el momento que fueron capturados, esto puede ser a través de un tiempo definido; En la siguiente Figura 2.7 se observa una serie de tiempo de voltaje a través del tiempo, con un intervalo de un minuto, este es un ejemplo claro de una serie de tiempo.

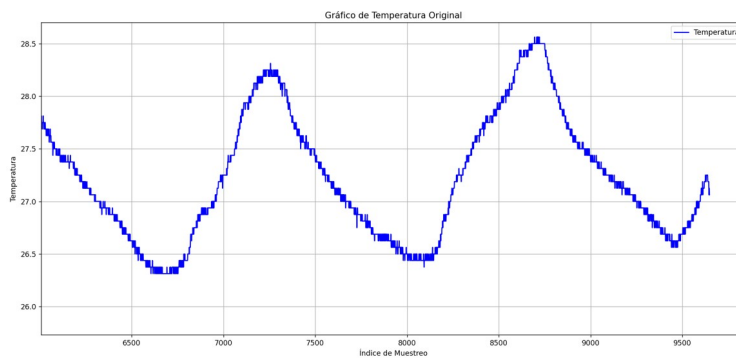


Figura 2.7: Serie de tiempo de temperatura de una CCMS. Hecha por Marco Aranda.

Estas gráficas son de gran utilidad para detectar errores de muestreo, además si bien se puede utilizar cualquier intervalo de tiempo para representarlas, es necesario que todas las mediciones tengan el mismo intervalo, en el caso de que algún intervalo no tenga muestra, es necesario sustituirla para mantener la tendencia original (Russell & Norvig, 2021).

2.17. Valores atípicos

De acuerdo con Cousineau y Chartier (2010), cuando en una serie temporal no se puede explicar la razón de un valor (que no siga la tendencia) se le denomina valor atípico (conocidos en inglés como Outliers). La generación de estos valores se puede determinar de diversas formas, desde la visualización manual, así como el uso de parámetros estadísticos

como intersección por cuartiles.

Existen diversos tipos de valores atípicos, entre estos están los de tipo aditivo, donde los valores son muy altos o bajos sin ser esperados, esto solo sucede en muestra, mientras que las demás continúan con la tendencia regular. Por otro lado, también existen los de tipo innovador, donde la muestra irregular afecta las siguientes a esta (Kim et al., 2022).

2.18. Limpieza de datos

La limpieza de datos es parte clave del entrenamiento de modelos de aprendizaje automatizado, esto debido a que se requiere que los datos con los que se alimenten sean lo más óptimos posibles, para esto es necesario corregir incongruencias por ejemplo valores atípicos (fuera del rango medio). Las ventajas de la limpieza de datos es que el rendimiento del modelo mejora, además de que se reduce la incongruencia entre los datos.

Este proceso de limpieza se puede llevar a cabo de diversas formas, pero el primer paso en caso en todas es examinar el dataset para encontrar valores faltantes, valores fuera de rango o inusuales. Para esto se puede realizar visualmente al usar técnicas como gráficas de series de tiempo, así como histogramas. Seguido de realizar dichas observaciones, se pueden aplicar diferentes metodologías, una de ellas es la imputación, la cual consiste en corregir información faltante o reemplazar valores fuera de rango con procesos de estimación, dicha técnica posee múltiples variantes, una de ellas, por ejemplo, es la imputación por regresión donde los valores faltantes se obtienen a través de un modelo de predicción (Montgomery et al., 2012).

Otra herramienta para realizar la limpieza de datos es el uso de la media móvil, esta es utilizada para cuando existen valores atípicos en una serie de tiempo, al aplicarla, se toman los valores anteriores y seguidos del valor atípico, y se genera un promedio de estos el cual toma el lugar de dicho valor, esta función regularmente se hace con un movimiento de 3 pasos adelante y 3 atrás. Además, la media móvil se puede aplicar hasta dos veces; sin embargo, una tercera aplicación no cambiará los valores (Montgomery et al., 2012).

Para el caso de valores con errores de escritura en toda una fila de datos, ya sea ruidos generados por fallas de lectura en los sensores, estos se pueden ya sea eliminar o reemplazar

con los métodos anteriormente mencionados; Sin embargo, estos pueden ser sustituidos por NA (No Aplica) cuando es estrictamente necesario conservarlos.

Tabla 2.1: Técnicas comunes para el tratamiento de problemas en los datos

Tipo de problema	Acciones recomendadas
Ausencia de datos	Eliminar registros incompletos o aplicar técnicas de imputación para estimar los valores perdidos
Valores incorrectos o atípicos	Revisar manualmente los datos anómalos para corregirlos o descartarlos si es necesario
Diferencias en los formatos o códigos	Unificar la codificación siguiendo un criterio estándar para mantener coherencia
Información de contexto insuficiente	Verificar campos dudosos y consultar la fuente original para aclarar su interpretación

Adaptado libremente a partir del enfoque de preparación de datos propuesto en la guía CRISP-DM de IBM SPSS Modeler.

En la tabla 2.1 se observa que existen distintas soluciones para distintos casos, estos procesos son muy importantes debido a que, en el caso de no realizar una correcta limpieza, el algoritmo puede aprender comportamientos erróneos, los cuales arrastrará hasta la validación.

Además de estas técnicas, existen las funciones de autocorrelación y autocorrelación parcial, las cuales asocian los valores pasados de las series actuales y pasadas para indicar cuáles podrían ser los valores a un futuro, con estos datos generados se puede generalizar el orden de los procesos en un modelo externo. La función de autocorrelación (FAS), capturan la relación entre los valores que se encuentran a k intervalos de distancia en dichas series; por otro lado, la función de autocorrelación parcial (FAP), relaciona los valores que se encuentran a k intervalos de distancia en una serie, considerando los valores de intervalos intermedios (López et al., 2015).

Capítulo 3

Estado del arte

En este capítulo tiene como objetivo principal recopilar de la literatura científica los resultados de trabajos previos relacionados con el modelo de las CCMS y la predicción de su voltaje. Se analizará los resultados obtenidos a través de distintas técnicas de modelado, así como las áreas de oportunidad que se presentan y como este trabajo puede abordar dicha problemática desde una perspectiva innovadora que permita generar resultados superiores a los de trabajos previos.

Finalmente, a partir de la evaluación de los trabajos críticos, se delimitará la brecha de conocimiento que justifica la investigación actual, sentando las bases de la metodología y los objetivos de este trabajo.

3.1. Modelado del rendimiento de CCMS: del enfoque tradicional a los algoritmos de aprendizaje profundo

Como se explicó en el capítulo del marco teórico, las CCMS son sistemas con la capacidad de producir energía de forma limpia gracias a microorganismos que viven en ella y son los causantes de la producción de energía; sin embargo, debido a que su producción de energía es dependiente de procesos biológicos, esta última no es lineal, lo que dificulta su modelado mediante enfoques tradicionales (Feregrino-Rivas et al., 2023).

3.2. Evolución del modelado de sistemas dinámicos

Si bien el estudio de las CCMS ha sido de gran interés en las últimas décadas, varias investigaciones se han quedado cortas en el aspecto del modelado de estos sistemas, por ejemplo, en el caso de Jiménez-Escamilla et al., 2018, donde se centran en la evaluación experimental de un sistema de una CCM con distintos electrodos y condiciones. Si bien en dicha investigación, se logró realizar un modelo matemático basado en ecuaciones diferenciales, donde se obtuvo una ecuación para el crecimiento de biomasa de la celda, además del modelado del consumo de glucosa (elemento utilizado durante la experimentación) para la formación de productos metabólicos, así como la derivación de la corriente teórica a partir del modelo. Estos modelos se obtuvieron de forma analítica, siendo comparados con datos experimentales.

Sin embargo, a pesar de haber logrado un coeficiente de determinación aceptable respecto a las estimaciones de los modelos y los valores reales, existieron divergencias al inicio de los procesos, ya que en la realidad estos sistemas dinámicos no siempre tienen una formación instantánea. Además, dichos modelos obtenidos, no poseen la capacidad de considerar retardos, por lo cual, en el caso de efectos acumulativos, o situaciones inesperadas, los modelos no tienen la capacidad de seguir los datos reales (Jiménez-Escamilla et al., 2018).

Otra investigación correlacionada al ámbito de este trabajo fue el de Romero Ramayo, 2021, el propuso una forma de modelar el comportamiento de una CCM determinando una función de transferencia obtenida a través de espectroscopía de impedancia electroquímica. El modelo, de orden reducido, buscaba describir cómo respondía la celda cuando se le aplicaba una corriente, basándose en cómo varía su impedancia en el tiempo y la frecuencia. Para ello, se midió el voltaje generado por la celda frente a ligeras perturbaciones de corriente. Aunque el modelo logró representar bastante bien el comportamiento en ciertas condiciones, especialmente a frecuencias bajas y medias, tuvo dificultades para ajustarse a lo que ocurre a frecuencias más altas. Además, no logró captar los comportamientos más complejos del sistema, lo que limita su uso como modelo predictivo en situaciones reales.

Estos casos revelan que los métodos tradicionales utilizados para el modelado de estos sistemas dinámicos presentan limitaciones para seguir las tendencias reales, debido a que

dichos sistemas no trabajan bajo las mismas condiciones en la práctica, además de no poseer una linealidad.

3.3. Aplicación de modelos de aprendizaje automático para la predicción del rendimiento de una CCM

Debido a lo difícil que es modelar estos procesos de forma convencional, en un estudio reciente Hossain et al., 2023, se ha demostrado la viabilidad de utilizar métodos basados en algoritmos de aprendizaje supervisado enfocados en aprendizaje profundo. Donde se desarrolló un modelo híbrido mediante el algoritmo **Super Learner**, el cual permite mezclar varios modelos base, entre ellos Bosque Aleatorio, así como potenciador de gradiente, regresión por vectores de soporte y vecinos más cercanos.

Los resultados obtenidos mediante este método fueron un $R^2=0.995$ y MSE de 0.032, el cual superó significativamente los modelos individuales. Durante el estudio que se realizó, se utilizaron como variables de entrada la temperatura, el pH, la concentración del sustrato, además del tipo de inóculo. Estas variables fueron utilizadas para la predicción del voltaje, en el caso de este estudio, se utilizó una variante de las CCM, pero sin sedimentos (Hossain et al., 2023).

3.4. Aplicación de modelos de aprendizaje automático para la predicción del rendimiento de una CCM

En adición a lo anterior, estudios como el de Feregrino-Rivas et al., 2023 utilizaron como variables de interés la conductividad, salinidad, temperatura y oxígeno disuelto, esto para el análisis de dos CCMS, una utilizando como electrolito agua de mar y otra agua de río. Los resultados que se obtuvieron fueron que la producción de voltaje es influenciada por la temporada en la que se encuentra, ya sea temporada de lluvia o sequía, ya que sedimentos obtenidos durante la temporada de sequía obtuvieron mayor producción eléctrica, comparándolos con los obtenidos en épocas de lluvia.

Además, se identificó que tanto la salinidad como la conductividad poseen una relación positiva con la producción de voltaje, siendo más significativa para los sustratos de río en comparación a la de mar. También menciona que la variación de temperatura afecta negativamente la producción de energía (Feregrino-Rivas et al., 2023)

3.5. Identificación de la brecha de conocimiento

En conclusión, en base a las investigaciones previas, se observa que existe un desarrollo limitado en el modelado de estos sistemas a través de algoritmos de aprendizaje supervisado. Además, aunque varios estudios han caracterizado las relaciones de las variables correlacionadas con la producción de voltaje de una CCMS y una CCM, como el de Feregrino-Rivas et al., 2023, estos estudios no profundizan lo suficiente en el modelado de estos sistemas para poder realizar una mimetización de ellos. En contraste, el estudio de Hossain et al., 2023, realizó exitosamente el modelado de este sistema a través del método de regresión, demuestra que es viable el modelado de estos sistemas a través de técnicas avanzadas como las que se plantean en este trabajo. No obstante, aún existen otros métodos para realizar dicha tarea por explorar, lo cual constituye el objetivo de esta investigación.

Capítulo 4

Metodología

A continuación, se presenta la metodología diseñada para el desarrollo de este proyecto. Esta inicia con la obtención del conjunto de datos a través de un sistema de monitoreo y control automatizado previamente desarrollado, además de la preparación de los datos para su uso en distintos modelos de aprendizaje supervisado para la predicción del voltaje de una CCMS, hasta obtener el mejor modelo predictivo.

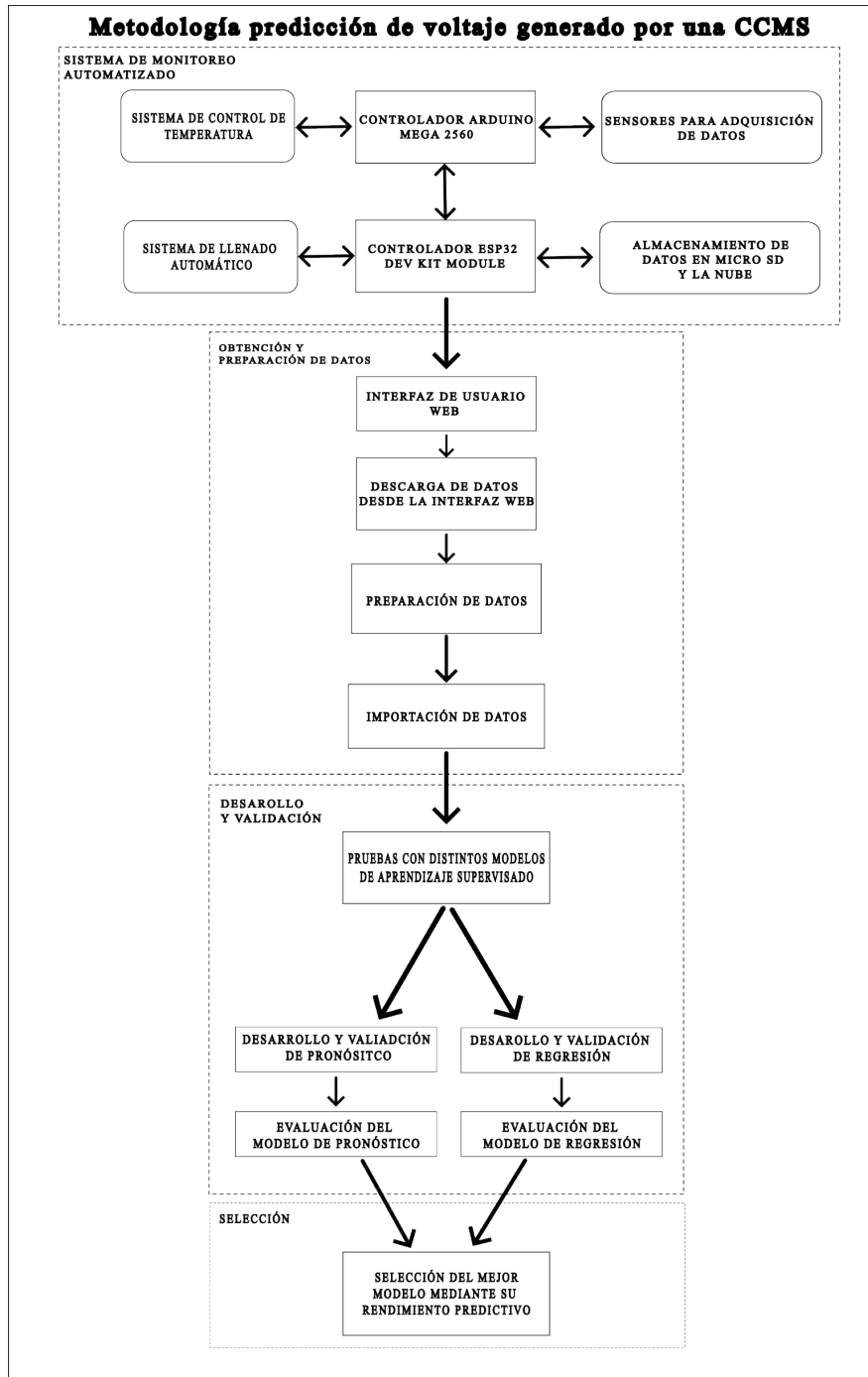


Figura 4.1: Metodología general para el desarrollo del proyecto. Hecha por Marco Aranda.

En la Figura 4.1 se visualiza el punto de partida de este proyecto es la etapa de obtención y preparación de datos, donde se limpian los datos para ser importados y realizar las múltiples pruebas con distintos modelos de aprendizaje supervisado los cuales se aplicarán para la tarea de regresión, así como pronóstico, además de la respectiva validación de cada modelo. Una vez los mejores modelos sean validados, se realizará la selección del mejor modelo con base en su capacidad predictiva, evaluada a través de métricas de evaluación como MSE, MAE y R^2 , las cuales son el estándar de evaluación de modelos predictivos.

4.1. Obtención y preparación de datos

La obtención de datos se realizó a través de la interfaz generada de un trabajo previo donde se recopilan los datos tanto físicamente como en la nube. Para este trabajo se utilizaron datos recopilados en esta última, a través de la plataforma gratuita IO Adafruit. Como se observa en la siguiente Figura 4.2, se realizó la descarga de los datos a través del siguiente apartado que se encuentra en el campo "Feeds".

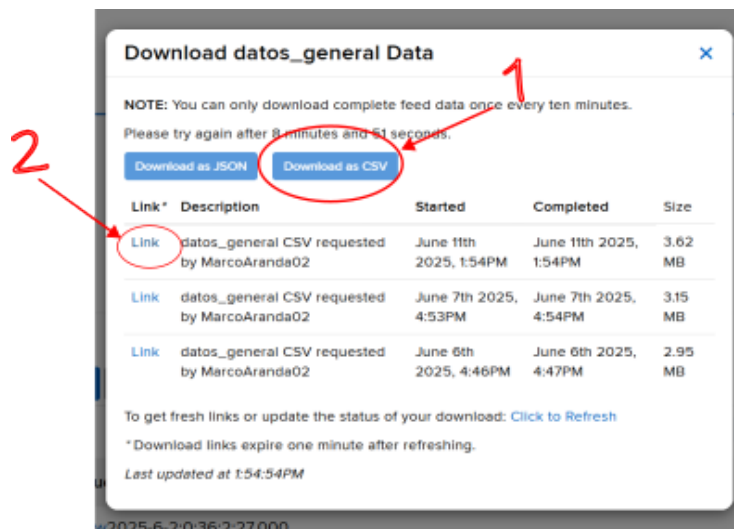


Figura 4.2: Imagen representativa del procedimiento de descarga de archivos.

A través de la Figura 4.2 se muestra que, el primer paso fue escoger el formato de descarga, en este caso fue CSV, y seguido de esto se dio clic en el enlace de descarga. Una vez descargados los datos, se determinaron las columnas de interés para el entrenamiento:

	Fecha	Hora	Temp.1			Conducti.	Salinidad		Voltaje
22370	0FWYSQQWB9G2VQY0BJSYE0MYD,2025-6-1	05:51:33	28.875	28.638	0.237	656.558	919.181	0.465	1.827
22371	0FWYSQTYKAH8AG1WG1H0X226D,2025-6-1	05:51:43	28.938	28.638	0.3	642.63	899.683	0.455	1.799
22372	0FWYSQY1K5K0E4TV1Y2EN8PE1D,2025-6-1	05:51:54	28.938	28.638	0.3	642.63	899.683	0.455	1.799
22373	0FWYSR13S999912GV1P33CB1D,2025-6-1	05:52:04	28.875	28.638	0.237	649.598	909.438	0.46	2.388
22374	0FWYSR45TXW0EWCRCQXYWW0PD4,2025-6-1	05:52:14	28.875	28.7	0.175	642.63	899.683	0.455	1.342
22375	0FWYSR6ZGSX0DHTQSCDZP1ZREY,2025-6-1	05:52:23	28.938	28.638	0.3	642.63	899.683	0.455	2.52
22376	0FWYSRA166BZVXTGVE7ZKVBC,2025-6-1	05:52:33	28.875	28.7	0.175	642.63	899.683	0.455	1.966
22377	0FWYSR0731Q6MPYQ6K5JXDHDEN,2025-6-1	05:52:53	28.938	28.7	0.237	670.453	938.634	0.475	1.287
22378	0FWYSRK96NMA81PDGCGX8RAK29,2025-6-1	05:53:03	28.938	28.7	0.237	649.598	909.438	0.46	2.041
22379	0FWYSRPC85XRC15C1EF5CARH04,2025-6-1	05:53:13	28.938	28.7	0.237	642.63	899.683	0.455	1.717
22380	0FWYSRSEG4P8NJ4P4FW44H0SN,2025-6-1	05:53:23	28.938	28.7	0.237	649.598	909.438	0.46	1.931
22381	0FWYSRWGJE8B66B4I8Y0B17ZMY,2025-6-1	05:53:33	28.875	28.7	0.175	642.63	899.683	0.455	1.966
22382	0FWYSRZKNF47H0AHWMK14V5R0K,2025-6-1	05:53:44	28.875	28.7	0.175	649.598	909.438	0.46	1.799
22383	0FWYSS2NZ47E6ANRD1PC0A0KPM,2025-6-1	05:53:54	28.938	28.7	0.237	642.63	899.683	0.455	1.931
22384	0FWYSSSF3WK9R2FDCX3X9CCZ0,2025-6-1	05:54:03	28.875	28.638	0.237	663.509	928.913	0.47	1.799
22385	0FWYSS8H8X36ZHR2D9TXJXUC,2025-6-1	05:54:13	28.875	28.638	0.237	642.63	899.683	0.455	1.827
22386	0FWYSSSQ2M96SW0HYBZJKNPFEX,2025-6-1	05:54:33	28.938	28.7	0.237	649.598	909.438	0.46	1.799
22387	0FWYSSHS67ZGZNFQHR201PRFY,2025-6-1	05:54:43	28.875	28.7	0.175	642.63	899.683	0.455	1.799
22388	0FWYSSMVA7QNBK3A47C46ST09,2025-6-1	05:54:53	28.875	28.638	0.237	642.63	899.683	0.455	1.53
22389	0FWYSSQYRVZ0FA77CV4JQDM5A0,2025-6-1	05:55:03	28.875	28.7	0.175	642.63	899.683	0.455	2.228
22390	0FWYSSV1K53RRRA8F7HBM5EYS0M,2025-6-1	05:55:13	28.875	28.638	0.237	663.509	928.913	0.47	1.966
22391	0FWYSS3NWRG0H5J9BXV0YAZ0X,2025-6-1	05:55:23	28.938	28.7	0.237	635.654	889.915	0.45	1.931
22392	0FWYST15TQZ58TK1KNHY0XPKWA,2025-6-1	05:55:34	28.875	28.7	0.175	642.63	899.683	0.455	2.361
22393	0FWYST3Z14T9JAH3CRSD/WZD4,2025-6-1	05:55:43	28.938	28.638	0.3	649.598	909.438	0.46	1.799
22394	0FWYST716M204YNRP2258XJHP4,2025-6-1	05:55:53	28.938	28.7	0.237	642.63	899.683	0.455	1.744

Figura 4.3: Variables de interés obtenidas desde el CSV dado por la interfaz.

Mediante la Figura 4.3 se observa las variables de interés capturadas desde el CSV dado por la interfaz fueron Fecha, Hora, Temperatura 1, Conductividad, Salinidad y Voltaje (Figura 4.3). Una vez seleccionadas estas variables se les agregaron sus títulos correspondientes en el archivo CSV para someterlas a un primer filtrado realizado por un script en Python.

Tras el primer filtrado realizado a través del script de Python denominado “SegundoFiltro”, se conservaron solamente las columnas de interés, además de que se modificó el formato de la variable “Fecha”.

Una vez realizado el filtrado de columnas excedentes en el dataset de interés se procedió a la identificación de valores atípicos; esto debido a que existieron cortes de monitoreo, por lo cual se observaron valores muy fuera de rango. Por ello, se realizó un filtrado a través del método de imputación. Además, las variables de interés se redujeron a Tiempo, Temperatura (Temperatura1), Conductividad, Salinidad y Voltaje. Para el caso de la salinidad, así como la conductividad, se les aplicó una media móvil después de la imputación con un tamaño de ventana de 20.

Los resultados de dicho filtrado se guardaron en un nuevo *dataset* listo para su uso en el desarrollo de los modelos.

4.2. Desarrollo y validación de modelos

Durante esta etapa se explica el desarrollo de los modelos durante su entrenamiento, así como su validación durante este, además de los conjuntos utilizados para la prueba de cada algoritmo utilizado.

4.2.1. KNN para regresión

Para el desarrollo de las predicciones de voltaje a través del algoritmo KNN, se alimentó este modelo con las variables Tiempo, Temperatura, Salinidad, Conductividad como entradas; y como salida la variable voltaje, este algoritmo se le denominó como KNN2205.

El modelo se entrenó con el *dataset* obtenido durante el posprocesamiento conocido como “datos_imputados_y_suavizados_2”, esto se realizó así para tener un punto de partida igual entre los distintos modelos de predicción que se usaron en este trabajo y reducir la cantidad de variaciones durante los entrenamientos. Después de haber tanto entradas como salidas, este algoritmo tuvo un preprocesamiento de la variable tiempo para convertirlo en un formato aceptable para el modelo

Realizada dicha etapa, se integró un filtrado de valores atípicos, así como el graficado de estos a través de *boxplots*. Sin embargo, al haberse realizado estos filtrados en el proceso de posprocesamiento del *dataset*, tanto la función de filtrado de valores atípicos como una de eliminación de filas con columnas faltantes, quedó como una opción al no ser necesaria su aplicación de nuevo.

Después de esos filtrados opcionales, se probaron distintos escaladores. Para una mayor flexibilidad se implementó una función con múltiples opciones: estas fueron *StandardScaler*, *RobustScaler*, *MinMaxScaler*, y se probó cada uno. Una consideración que se tomó durante este proceso de escalado de los datos fue realizar un escalado para cada columna y no todo en conjunto, esto aplicado tanto para las variables de entrada como para la de salida.

Posteriormente, se probaron distintos parámetros para el modelo. Esto se realizó mediante la herramienta *GridSearchCV* perteneciente a *Scikit-Learn*, la cual busca los hiperparámetros con mejores resultados con base en una métrica de evaluación específica. Para el caso de ambos modelos se utilizó MSE, además de una validación cruzada de 5 pliegues. Esta etapa

ayuda a reducir el sobreajuste del modelo y permite una mejor generalización de los datos. Los valores probados mediante GridSearchCV para el modelo anteriormente mencionado fueron número de vecinos: [3, 6, 9, 15, 20, 40, 50, 70, 100].

Se procedió a entrenar el modelo con los distintos conjuntos de hiperparámetros, utilizando el 70 % de los datos previamente designados como conjunto de entrenamiento (X_{train} , y_{train}). Además, posteriormente se realizó la prueba con el 20 % de la misma forma, así como la validación con el 10 % restante. Durante estos procedimientos, se calcularon los valores de MSE, MAE y R^2 tanto para el entrenamiento como para los valores de prueba y validación, y se guardó el modelo con los mejores resultados bajo el nombre “modeloknn”.

4.2.2. XGBoost para regresión

En adición al modelo “KNN2205” para regresión, también se desarrolló un segundo mediante XGBoost para la misma tarea, denominado “Xgboostmodelo714”. De igual forma fue entrenado con el *dataset* “datos_imputados_y_suavizados_2”, y con variables de entrada Tiempo, Temperatura, Salinidad, Conductividad y como salida el voltaje. En similitud con el modelo “KNN2205”, este segundo modelo también se le añadió un filtrado de eliminación de filas cuando existieran columnas faltantes. Posteriormente, se realizaron pruebas para la asignación del escalador con *StandardScaler*, *RobustScaler* y *MinMaxScaler*.

Posteriormente, se procedió a la etapa de la división de los datos, donde se utilizó la distribución 80-20 con aleatorización, designándose el 80 % para el entrenamiento y el 20 % restante para las pruebas del modelo. Con respecto a la designación de hiperparámetros, se probaron los siguientes conjuntos: número de estimadores [200, 400, 600, 800, 1400, 2200, 2000], máxima profundidad [7, 9], tasa de aprendizaje [0.1, 0.15, 2], muestras utilizadas por árbol [0.6, 0.8, 0.1]. También, para el caso de este modelo, se utilizó GridSearchCV para encontrar los mejores hiperparámetros.

Una vez seleccionado el mejor conjunto de hiperparámetros y entrenado el modelo, este fue guardado con el nombre de “modelo_xgboost_mejorado_28abril”, destacando, de igual forma, que se creó un escalador para cada columna, tanto para las entradas como para la salida. Además, tras el entrenamiento se utilizó el *dataset* de prueba guardado (20 % del total) y se calcularon las métricas de evaluación MSE, MAE y R^2 para determinar los resultados

tanto de entrenamiento como de prueba.

4.2.3. LSTM para pronóstico

El uso del modelo mencionado de este apartado se logró mediante TensorFlow Keras y se propusieron dos conjuntos de ventanas y salidas a predecir: [4,1], [14,1]. También se definió el conjunto de entradas y salida. Las entradas que se utilizaron fueron las variables Temperatura, Salinidad, Conductividad y Voltaje, además como salida el voltaje de la celda.

Otro punto que mencionar es que al igual que en los otros algoritmos, se utilizó el *dataset* “datos_imputados_y_suavizados2” para el entrenamiento de este. Para este modelo, también se realizó pruebas con distintos escaladores de *MinMaxScaler*, *StandardScaler* y *RobustScaler*. Seguido se generó la ventana móvil, esto es muy importante para este tipo de algoritmos donde es necesario a desplazarse a través de los datos en forma secuencial con respecto al tiempo para lograr la tarea de pronóstico correctamente. Después de integrar la ventana, también se agregó un filtro de *outliers* usando desviación estándar, donde al encontrar ventanas con valores faltantes, estas se eliminarían para mejorar la pureza de los datos de entrenamiento.

Seguido de haber generado la función de la ventana, así como la ejecución de su filtrado, se aplicó la distribución 80-20 en el *dataset*, para ser utilizado en el entrenamiento y prueba (80 % y 20 % respectivamente), considerando *Shuffle* en falso debido a que se necesita mantener la secuencia original de estos. Una vez realizada dicha etapa, se procedió con la construcción del modelo, donde se utilizó una estructura de pirámide (disminución sucesiva de neuronas en cada capa).

Durante la construcción del modelo se probaron distintos conjuntos de neuronas para cada capa de la red, para la primera capa se probaron el conjunto [128,140,160,180], para la segunda capa [60,80,100] y para la de tercera [32,40,50]. Además, se probaron los distintos conjuntos de ventanas y valores a predecir que se propusieron para encontrar, a través del entrenamiento y prueba, el número de pasos de predicción con mejor precisión entre los distintos conjuntos. Este procedimiento de determinar el mejor conjunto de neuronas, así como ventana y valores a predecir, se logró a través de las métricas de evaluación MSE, MAE y R^2 obtenidas durante entrenamiento y prueba de los datos restantes.

4.2.4. XGBoost para pronóstico

El modelo XGBoost también fue utilizado para la tarea de pronóstico, donde se alimentó con el *dataset* “datos11-28abrilfiltrados2”, estos datos, como se mencionó con anterioridad, son un derivado del dataset principal “datos_imputados_y_suavizados”. Para el desarrollo de este algoritmo de pronóstico, se tomó en consideración como variables de entrada la Fecha, Hora y voltaje, así como salida este último. Una vez definidas las variables de entrada y su escalamiento, el cual se utilizó *MinMaxScaler*, se generó la función de la ventana móvil, además de la distribución de los datos en el formato 70-20-10, donde 70 % se definió para entrenamiento, 20 % para prueba y un 10 % para validación, donde cada subconjunto mantendría su secuencia original debido a la aplicación de series temporales. Seguido de esto, se definió el modelo y sus hiperparámetros, donde la búsqueda de estos se realizó en conjunto de *RandomizedSearchCV*. También se procedió a experimentar con la ventana móvil, donde se utilizaron los siguientes conjuntos [4,1], [14,1], [12,1] y [10,1], el primer número correspondiente a la ventana y el segundo al número de valores a predecir.

Una vez finalizadas las pruebas de ventanas y hiperparámetros, se evaluaron a través de las métricas MAE, MSE y R^2 , donde se escogió la variante del modelo de pronóstico con mejores resultados y se guardó con el nombre “xgboost_model_pronostico”.

4.3. Evaluación de modelos obtenidos

Durante esta etapa del proyecto, se pusieron a prueba los modelos a través de datos no vistos durante el entrenamiento y la primera validación, esto para verificar que los modelos tuviesen la capacidad de predecir nuevos datos y con esto asegurarse de la capacidad de mimetización de cada modelo.

4.3.1. Evaluación del modelo KNN para regresión

Para esta etapa del proyecto, se extrajeron datos del sistema previamente implementado en la CCMS para validar y evaluar las predicciones generadas por el modelo de regresión KNN2205. Estos datos extraídos para este proceso se denominaron “datos4-6juniofiltrados2”, los cuales estuvieron compuestos de 3723 registros. También se pasó el conjunto por un filtrado manual como los utilizados para el entrenamiento.

Dentro de esta evaluación, se importaron tanto el modelo guardado y sus escaladores (modelo con mejores métricas obtenidas durante el entrenamiento), así como los datos nuevos de prueba, donde se hizo un proceso de cargar de nuevo las variables de entrada, así como definir las de salida, retomando las variables que se utilizaron durante el entrenamiento. Por dicha razón, también se realizó el preprocesado de la variable tiempo dentro de los datos nuevos, así como el escalar las nuevas variables de entrada, así como de salida. Una vez definido todo esto, se procedió a utilizar el modelo cargado para realizar las predicciones normalizadas en base al escalador y con estas realizar el cálculo de las métricas de evaluación MSE, MAE y R^2 , así como la visualización de estas predicciones vs los valores reales.

4.3.2. Evaluación del modelo XGBoost para regresión

Se aplicó el mismo procedimiento de evaluación que KNN, importando tanto el modelo como el escalador de XGBoost y el nuevo *dataset* para la evaluación “datos4-6juniofiltrados2”. Una vez realizada la etapa, los procesos que se realizaron fueron los mismos que para KNN, tanto en la división de datos como el cálculo de las métricas de evaluación y su análisis. La metodología que se llevó a cabo para la evaluación de los dos modelos para regresión se pue-

de observar en la Figura 4.4, donde se observan los pasos de manera general y que fueron compartidos para evaluar los dos algoritmos mencionados.

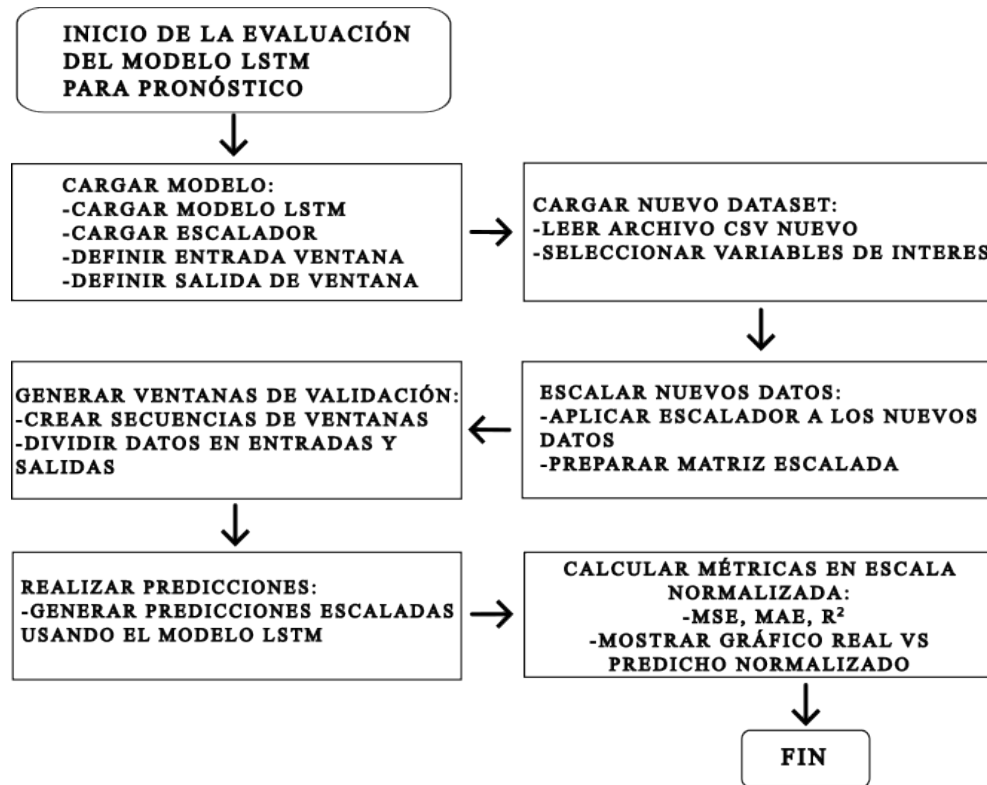


Figura 4.4: Metodología de evaluación de los modelos de regresión

Esta metodología para evaluar tanto el modelo “modelo knn” como el de XGBoost denominado “Xgboostmodelo714 ” se compartió debido a que ambos modelos fueron entrenados con las mismas variables de entrada como de salida, además de que el proceso para la importación, así como integración de estos modelos fueron idénticos para este trabajo en particular.”

4.3.3. Evaluación del modelo LSTM

La evaluación de los resultados de este modelo fue muy similar con respecto al de KNN o XGBoost enfocados en regresión, donde se importó el modelo cargado y su escalador, además que se utilizó como dataset de prueba el denominado “datos4-6juniofiltrados2”; Sin embargo, en este caso el cual fue para pronóstico la diferencia puntual es que los valores utilizados para las predicciones como entrada fueron a través de Temperatura, Salinidad, Conductividad y

voltaje (esto debido al usar series temporales) además de utilizar una ventana móvil definida a través de los resultados obtenidos previamente durante el entrenamiento. El desarrollo de este proceso se puede observar en la siguiente Figura 4.5 donde se observa la metodología llevada a cabo para esta etapa de evaluación.

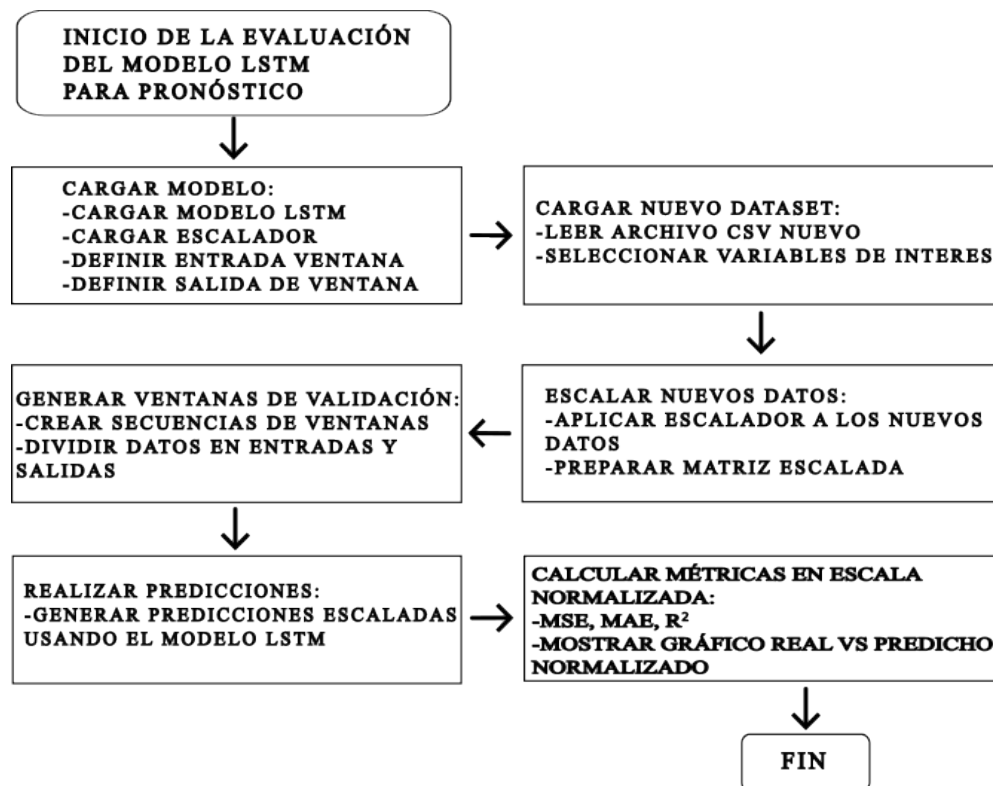


Figura 4.5: Metodología de evaluación del modelo LSTM.

Una vez aplicado el modelo a los nuevos datos a predecir, se evaluaron los resultados mediante MSE, MAE y R^2 utilizando las predicciones normalizadas y con ello visualizándolas a través de un gráfico de los datos predichos contra los reales.

4.3.4. Evaluación del modelo XGBoost pronóstico

Para evaluar este modelo de pronóstico a través de XGBoost, también se utilizó el dataset de validación “datos4-6juniofiltrados2”, además de que se importó el modelo con el nombre “xgboost_model_pronóstico” y sus escaladores tanto para entradas como salida. Como se observa en la Figura 4.6. Una vez importadas estas características, se limpiaron las columnas del nuevo *dataset*, así como la creación de la columna *DateTime* (unión de fecha y hora).

También se definieron las variables de retardo, estas correspondientes a los valores anteriores al que se va a predecir, así como la determinación de este último. Dichos valores se obtuvieron durante los resultados del entrenamiento. Posteriormente se definieron las entradas, así como la salida y se les aplicó una normalización a través de los escaladores importados. Finalmente se aplicó el modelo a los datos de entrada para predecir las salidas, además que se calculó las métricas de evaluación MSE, MAE y R^2 de las predicciones normalizadas, así como su gráfico con respecto a los valores reales normalizados.

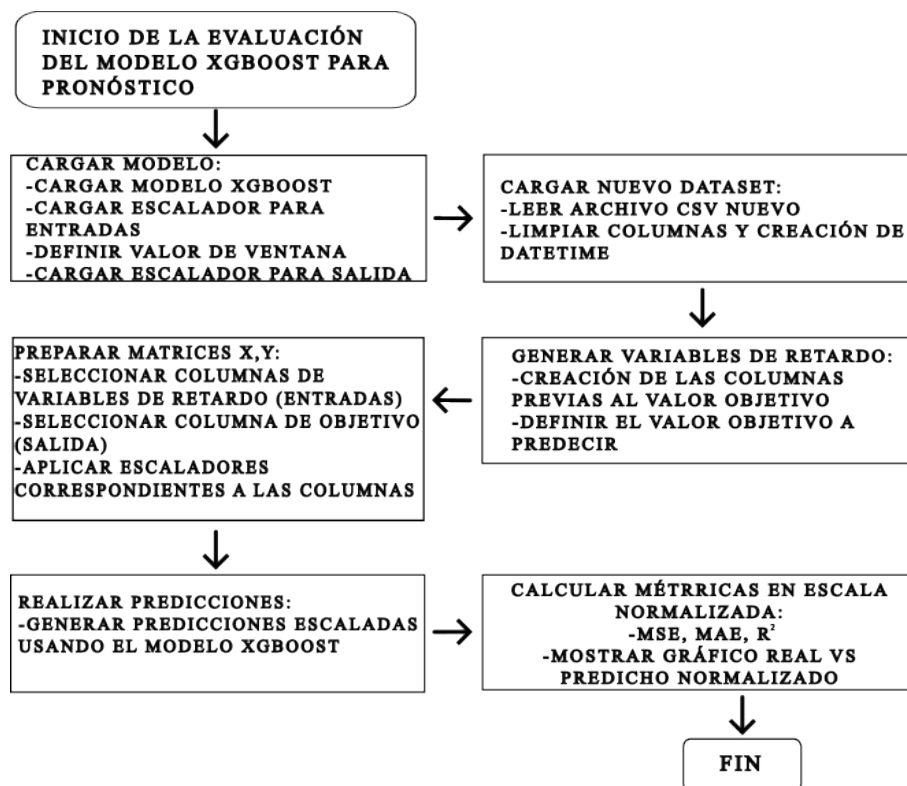


Figura 4.6: Metodología de la evaluación del modelo XGBoost para pronóstico.

Capítulo 5

Resultados

Durante el desarrollo de este trabajo, se limpiaron y procesaron dos conjuntos de datos para su uso en la implementación de un total de 4 modelos predictivos, dos orientados a la tarea de regresión utilizando los algoritmos KNN y XGBoost y los otros enfocados a la tarea de pronóstico, desarrollados mediante LSTM y XGBoost.

A lo largo de la metodología planteada, cada modelo se experimentó con distintas combinaciones de hiperparámetros y configuraciones específicas, seleccionado en cada caso la opción que presentó mejor rendimiento durante el entrenamiento mediante las métricas de evaluación.

5.1. Resultados del procesamiento de datos

Como se mencionó en la metodología de este trabajo, la obtención de datos se realizó a través de la interfaz generada de un trabajo previo donde se recopilan los datos tanto físicamente como en la nube.

```
1 Fecha;Tiempo;Temperatura;;;Conductividad;Salinidad;;Temperatura
2 0FX10PZTQW08N1P7YJFMGB82FD,2025-6-4;00:50:28;27.575;0.425;670.453;938.634;0.475;2.416;6.936;0.768,3098471,2025-06-04 07:01:08 UTC,,,
3 0FX10QHYZVN073EH8PMR5Y77BXC,2025-6-4;00:01:49;28;27.575;0.425;677.388;948.344;0.48;2.306;6.944;0.769,3098471,2025-06-04 07:02:07 UTC,,,
4 0FX10R45ADZBCVAZ1XVCQNIHRVA,2025-6-4;00:02:49;28;27.575;0.425;691.237;967.732;0.49;2.388;6.936;0.769,3098471,2025-06-04 07:03:07 UTC,,,
5 0FX10RPMYSCCDG5S3TSXXK7RD3,2025-6-4;00:03:50;28;27.575;0.425;670.453;938.634;0.475;2.388;6.944;0.768,3098471,2025-06-04 07:04:07 UTC,,,
6 0FX10S8T1NV2ZC2J6GR8KK7TAF,2025-6-4;00:04:49;28;27.575;0.425;670.453;938.634;0.475;1.849;6.936;0.768,3098471,2025-06-04 07:05:07 UTC,,,
7 0FX10SV93EX94FT09SP6QH7JMV,2025-6-4;00:05:50;28;27.575;0.425;670.453;938.634;0.475;2.52;6.936;0.768,3098471,2025-06-04 07:06:07 UTC,,,
8 0FX10TDE6S2MX7FCNDCZ8VQNMW,2025-6-4;00:06:49;28;27.575;0.425;677.388;948.344;0.48;2.361;6.936;0.768,3098471,2025-06-04 07:07:07 UTC,,,
9 0FX10TZM9Q5N2E0CAW7ZZZHHCG,2025-6-4;00:07:49;28;27.575;0.425;663.509;928.913;0.47;2.361;6.936;0.768,3098471,2025-06-04 07:08:07 UTC,,,
10 0FX10VJ35R5F9E8M28REAEFBZ,2025-6-4;00:08:49;28;27.575;0.425;670.453;938.634;0.475;2.146;6.936;0.768,3098471,2025-06-04 07:09:07 UTC,,,
11 0FX10M48BYE4YJFSRH16M9735P,2025-6-4;00:09:49;28;27.575;0.425;677.388;948.344;0.48;2.333;6.936;0.769,3098471,2025-06-04 07:10:07 UTC,,,
12 0FX10MPQEMZH8ZA61R1KA834E4,2025-6-4;00:10:49;28;27.575;0.425;698.15;977.41;0.495;2.333;6.936;0.769,3098471,2025-06-04 07:11:07 UTC,,,
13 0FX10X8XJ41T5CNRCAW69SKZPH,2025-6-4;00:11:49;28;27.575;0.425;677.388;948.344;0.48;2.306;6.951;0.769,3098471,2025-06-04 07:12:07 UTC,,,
```

Figura 5.1: Datos previos a primer filtrado.

```

1  Fecha,Tiempo,Temperatura,Conductividad,Salinidad,Voltaje
2  04/06/2025,00:04:49,28.0,938.634,0.475,0.768
3  04/06/2025,00:05:50,28.0,938.634,0.475,0.768
4  04/06/2025,00:06:49,28.0,948.344,0.48,0.768
5  04/06/2025,00:07:49,28.0,928.913,0.47,0.768
6  04/06/2025,00:08:49,28.0,938.634,0.475,0.768
7  04/06/2025,00:09:49,28.0,948.344,0.48,0.769
8  04/06/2025,00:10:49,28.0,977.41,0.495,0.769
9  04/06/2025,00:11:49,28.0,948.344,0.48,0.769
10 04/06/2025,00:12:48,28.0,958.043,0.485,0.768
11 04/06/2025,00:13:49,28.0,938.634,0.475,0.769
12 04/06/2025,00:14:48,27.938,938.634,0.475,0.768

```

Figura 5.2: Datos seguidos del primer filtrado.

El dataset utilizado para el primer proceso de filtrado se denominó “datos19marzo28abril” el cual se observa un fragmento de este en la Figura 5.1, el cual comprendió, como su nombre lo indica, valores desde el día 19 de marzo hasta el día 28 de abril. Este *dataset* seguido del primer filtrado se guardó como se muestra en la Figura 5.2; en adición a este dataset se guardó otro denominado “datos11-28abrilfiltrados”, el cual fue un derivado del original y mantuvo los datos con menos cortes de monitoreo.

Después del segundo filtrado, los resultados de dicho filtrado se guardaron como un nuevo dataset denominado “datos_imputados_y_suavizados_2” los cuales son los que se utilizaron para el entrenamiento de los modelos experimentados en este trabajo. Este proceso se repitió también para el dataset “datos11-28abrilfiltrados” el cual se conservó con el nombre de “datos11-28abrilfiltrados2”.

Mediante la Figura 5.3 se observa del lado izquierdo las variables Tiempo, Temperatura, Conductividad, Salinidad y Voltaje, previos al proceso de filtrado (en ese orden); del lado derecho se observan los resultados del filtrado. Se constató que después de la imputación algunos picos de caída en los sensores, como en el caso de la variable temperatura, se corrigieron. Sin embargo, en el caso de la salinidad, así como la conductividad, al ser variables que tienen mayor variación y haberse realizado múltiples cortes en las mediciones, no fue posible reducir del todo las caídas en estas variables. Esto trajo consigo la imposibilidad de tener un orden ideal para las series temporales de estas dos variables (importante para aplicaciones como regresión y pronóstico).

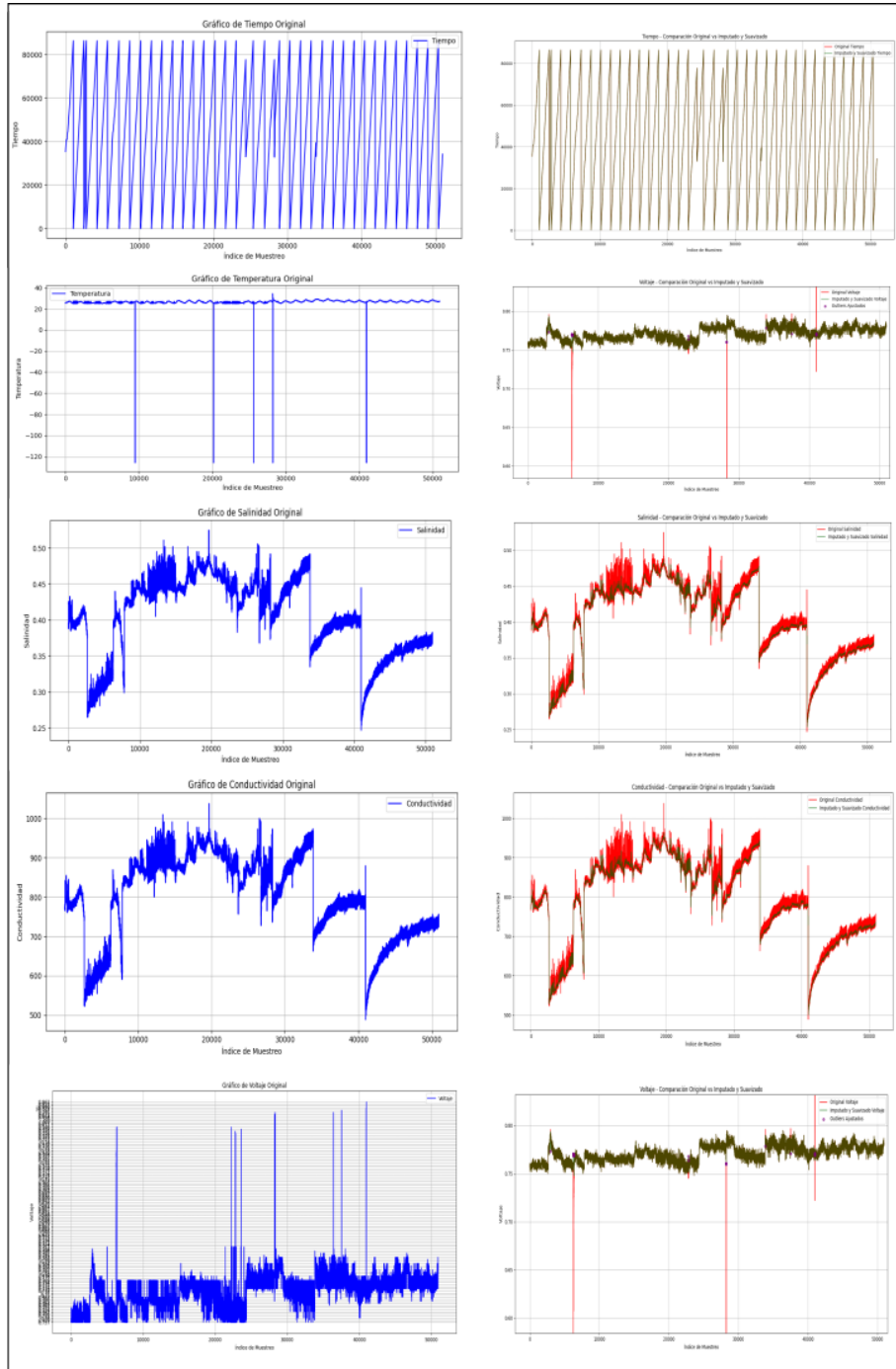


Figura 5.3: Tabla comparativa, resultados de imputación y media móvil .

5.2. Resultados modelo KNN

Para el modelo KNN2205, se obtuvo el mejor conjunto de hiperparámetros a través de GridSearchCV donde dichos resultados fueron un número de vecinos de 6, además de una distancia Manhattan, así como un peso inversamente proporcional a la distancia. Estos resultados obtenidos se pusieron a prueba con distintos escaladores durante el entrenamiento y seguido se aplicó la evaluación a través de nuevos datos, como se mencionó en la metodología. Las primeras 3 tablas que se observan son los resultados durante el entrenamiento con la división de datos 70-20-10 correspondiente a los resultados obtenidos mediante la mejor configuración obtenida. Seguido de esas primeras 3 tablas se encuentran los datos a partir de la evaluación de cada configuración en base al nuevo *dataset* “datos4-6juniofiltrados2”.

Tabla 5.1: Resultados durante entrenamiento de KNN con el escalador MinMaxScaler

KNN: Vecinos: 6 Escalador: MinMaxScaler	MSE	MAE	R2
Entrenamiento	0.0	2×10^{-6}	0.999 976
Prueba	0.000 169	0.008 224	0.809 132
Validación	0.000 199	0.008 334	0.773 564

Tabla 5.2: Resultados durante entrenamiento de KNN con el escalador StandardScaler

KNN: Vecinos: 6 Escalador: StandardScaler	MSE	MAE	R2
Entrenamiento	2.4×10^{-5}	5.4×10^{-5}	0.999 976
Prueba	0.149 088	0.240 007	0.852 538
Validación	0.000 199	0.008 334	0.848 676

Tabla 5.3: Resultados durante entrenamiento de KNN con el escalador RobustScaler

KNN: Vecinos: 6 Escalador: RobustScaler	MSE	MAE	R2
Entrenamiento	2.4×10^{-5}	5.4×10^{-5}	0.999 976
Prueba	0.149 088	0.240 007	0.852 538
Validación	0.152 075	0.242 866	0.848 676

Tabla 5.4: Resultados de validación mediante el nuevo dataset para los 3 escaladores utilizados

KNN	MSE	MAE	R2
MinMaxScaler	0.000 298	0.013 046	-1.400 088
StandardScaler	0.167 494	0.301 869	-1.463 249
RobustScaler	0.374 298	0.452 277	-1.644 405

5.3. Resultados modelo XGboost

En el caso del modelo XGBoost enfocado en regresión, el mejor conjunto de hiperparámetros obtenidos por parte de GridSearchCV fue una tasa de aprendizaje de 0.15, una máxima profundidad de 9, así como un número de estimadores de 200 y una fracción de submuestra de 0.8.

Tabla 5.5: Resultados durante entrenamiento de XGBoost con el escalador MinMaxScaler

XGBoost Escalador: MinMaxScaler	MSE	MAE	R2
Entrenamiento	0.001 403	0.027 887	0.947 620
Prueba	0.002 751	0.037 695	0.897 766

Tabla 5.6: Resultados durante entrenamiento de XGBoost con el escalador RobustScaler

XGBoost Escalador: RobustScaler	MSE	MAE	R2
Entrenamiento	0.021 687	0.109 585	0.947 223
Prueba	0.042 144	0.147 556	0.897 898

Tabla 5.7: Resultados durante entrenamiento de XGBoost con el escalador StandardScaler

XGBoost Escalador: StandardScaler	MSE	MAE	R2
Entrenamiento	0.052 730	0.170 876	0.947 223
Prueba	0.102 469	0.230 083	0.897 898

Tabla 5.8: Resultados de validación mediante el nuevo dataset para los 3 escaladores utilizados

XGBoost	MSE	MAE	R2
MinMaxScaler	0.018 953	0.105 435	-3.275 825
StandardScaler	0.274 290	0.617 574	-3.033 840
RobustScaler	0.666 911	0.396 059	-3.033 840

5.4. Resultados modelo LSTM

Para este modelo durante las pruebas de distintos conjuntos de capas y tamaños de ventana, se determinó que la mejor estructura de la red fue 160 neuronas en la primera, 80 en la segunda y 40 en la tercera, donde a la última (cuarta capa) se le designó una única salida (único valor a predecir). Además, se realizaron pruebas con los distintos conjuntos de ventana [14,1] y [4,1], donde se ejecutó el modelo con la combinación de los dos conjuntos de ventanas y los distintos escaladores; cada modelo se ejecutó con un máximo 120 épocas, con un tiempo de un detenimiento temprano al no mejorar después de las 15 épocas.

Tabla 5.9: Resultados de prueba del conjunto [14,1] durante el entrenamiento para los 3 escaladores

LSTM [14,1]	MSE	MAE	R2
MinMaxScaler	0.000 004	0.0015	0.749 060
StandardScaler	0.000 004	0.0015	0.742 746
RobustScaler	0.000 004	0.0015	0.747 168

Tabla 5.10: Resultados de prueba del conjunto [4,1] durante el entrenamiento para los 3 escaladores

LSTM [4,1]	MSE	MAE	R2
MinMaxScaler	0.000 004	0.0016	0.716 867
StandardScaler	0.000 005	0.0016	0.702 275
RobustScaler	0.000 005	0.0016	0.710 824

Tabla 5.11: Resultados de validación del conjunto [14,1] mediante el nuevo dataset para los 3 escaladores

LSTM [14,1]	MSE	MAE	R2
MinMaxScaler	0.000 392	0.0147	0.911 938
StandardScaler	0.017 165	0.0973	0.896 505
RobustScaler	0.010 132	0.0756	0.851 462

Tabla 5.12: Resultados de validación del conjunto [4,1] mediante el nuevo dataset para los 3 escaladores

LSTM [4,1]	MSE	MAE	R2
MinMaxScaler	0.000 636	0.0195	0.856 586
StandardScaler	0.024 495	0.1193	0.851 975
RobustScaler	0.010 166	0.0807	0.850 620

5.5. Resultados modelo XGBoost pronóstico

Los resultados obtenidos en este modelo a través de RandomizedSearchCV fueron un número de 400 estimadores, así como una máxima profundidad de 5, una fracción de muestra de 0.7 y una tasa de aprendizaje de 0.02. Además, utilizando el escalador MinMaxScaler se probaron los siguientes conjuntos de ventana [4,1], [14,1], [12,1] y [10,1], durante el entrenamiento, así como el proceso de validación con el nuevo dataset.

Tabla 5.13: Resultados de pruebas de los distintos conjuntos de ventanas durante el entrenamiento

XGBoost usando MinMaxScaler	MSE	MAE	R2
[4,1]	0.000 215	0.011 123	0.694 247
[14,1]	0.000 737	0.020 271	0.7243
[12,1]	0.000 211	0.010 886	0.7242
[10,1]	0.000 195	0.010 485	0.7230

Tabla 5.14: Resultados de validación de los distintos conjuntos de ventanas durante el entrenamiento

XGBoost usando MinMaxScaler	MSE	MAE	R2
[4,1]	0.000 201	0.010 693	0.594 556
[14,1]	0.000 661	0.019 193	0.6458
[12,1]	0.000 190	0.010 307	0.6461
[10,1]	0.000 178	0.010 485	0.6396

Tabla 5.15: Resultados de validación de los distintos conjuntos de ventanas mediante el nuevo dataset

XGBoost usando MinMaxScaler	MSE	MAE	R2
[4,1]	0.000 046	0.0052	0.899 342
[14,1]	0.000 178	0.0100	0.898 169
[12,1]	0.000 050	0.0053	0.900 026
[10,1]	0.000 046	0.0051	0.899 439

Capítulo 6

Discusión de resultados

En este apartado se interpretan los resultados obtenidos por cada modelo, y se identifican sus fortalezas, así como debilidades, para determinar cuál de los 4 modelos entrenados y evaluados se presenta como una herramienta óptima para la mimetización de una CCMS.

6.1. Caracterización de los resultados de cada modelo obtenido

Los resultados obtenidos de la evaluación de cada modelo producido durante este trabajo presentaron herramientas útiles para la predicción del voltaje de una CCMS; sin embargo, cada modelo presentó sus fortalezas, así como sus debilidades. En el caso del modelo obtenido mediante el algoritmo KNN, los mejores resultados para este algoritmo a consideración del autor son los siguientes:

Tabla 6.1: Métricas de evaluación del modelo KNN durante la validación con el *dataset* nuevo

MSE	MAE	R2
0.000 298	0.013 046	−1.400 088

Esto se logró a través de la configuración siguiente obtenida durante las pruebas de distintos escaladores, así como hiperparámetros.

Tabla 6.2: Configuración del modelo KNN

Modelo	KNN
Hiperparámetro	Vecinos: 6
Escalador utilizado	MinMaxScaler

Este modelo mencionado, si bien durante el entrenamiento presentó valores muy buenos de ajuste, al igual que durante las pruebas con el resto del dataset original, el modelo parece no poseer la capacidad de ajustarse correctamente a los nuevos datos no vistos. Esto puede deberse a la caoticidad de los datos de entrenamiento (esto se puede observar en la Figura 4.6), los cuales, si bien fueron suavizados, procesos como la imputación y la media móvil no lograron corregir errores de medición causados previamente al desarrollo de este trabajo. Este problema mencionado se presenta principalmente en las variaciones como subidas y bajadas repentinas dentro del voltaje que se desea predecir, mientras que en momentos de asentamiento predice correctamente el objetivo.

Para el modelo XGBoost, los resultados de predicción fueron similares a los del modelo KNN (esto debido a que ambos fueron utilizados para regresión). Las métricas obtenidas tras la evaluación fueron las siguientes:

Tabla 6.3: Métricas de evaluación del modelo XGBoost durante la validación con el *dataset* nuevo

MSE	MAE	R2
0.018 953	0.105 435	-3.275 825

Estos resultados se obtuvieron mediante la siguiente configuración:

Tabla 6.4: Configuración del modelo XGBoost

Modelo	XGBoost
Hiperparámetros	Tasa de aprendizaje: 0.15
	Máxima profundidad: 9
	Número de estimadores: 200
	Fracción de la submuestra: 0.8
Escalador utilizado	MinMaxScaler

Los resultados obtenidos para el modelo XGBoost, el cual también había obtenido buenos

resultados durante el entrenamiento, sufre al igual que el modelo KNN de las consecuencias de ser entrenado con un dataset con variables muy caóticas. Es por esto que el modelo, a pesar de poseer un error promedio cuadrático “bajo” al igual que el modelo anterior, posee una baja capacidad de seguir los nuevos datos. Aun así, como se observó durante el entrenamiento, XGBoost es menos susceptible al sobreajuste en comparación con el modelo KNN, ya que este último varía bastante sus resultados a partir de los escaladores utilizados, esto debido a que el algoritmo depende de las distancias para realizar las predicciones, mientras que XGBoost no.

Con respecto a los modelos de pronóstico, el modelo LSTM presentó resultados con muy buena precisión y capacidad de generalizar en el sistema. Esto se puede observar en los resultados obtenidos mediante la mejor configuración obtenida durante el entrenamiento.

Tabla 6.5: Métricas de evaluación del modelo LSTM durante la validación con el *dataset* nuevo

LSTM [14,1]	MSE	MAE	R2
Escalador: MinMaxScaler	0.000 392	0.0147	0.911 938

Tabla 6.6: Configuración del modelo LSTM

Modelo	LSTM
Hiperparámetros	Primera capa: 160 neuronas Segunda capa: 80 neuronas Capa intermedia: 40 Función de activación: Relu
Escalador utilizado	MinMaxScaler
Tamaño de ventana	14
Paso de salida	1

Como se observa en la tabla de métricas de evaluación, el modelo LSTM logró capturar la dependencia de la serie temporal de las variables correlacionadas al voltaje con alta precisión. El valor de R^2 indica que el modelo posee la capacidad de explicar más del 90 % de la variabilidad del sistema. Estos resultados son muy positivos considerando que modelos basados en regresión no lograron seguir los datos nuevos debido a la caoticidad del entrenamiento. Aun

con dicha limitación por parte del dataset de entrenamiento, el modelo LSTM logró realizar las predicciones con una alta precisión.

Finalmente, el modelo de XGBoost enfocado en pronóstico, al igual que el modelo LSTM, presentó muy buenos resultados como se observa en la siguiente tabla.

Tabla 6.7: Métricas de evaluación del modelo XGBoost pronóstico durante la validación con el *dataset* nuevo

XGBoost usando MinMaxScaler	MSE	MAE	R2
[12,1]	0.000 050	0.0053	0.900 026

Tabla 6.8: Configuración del modelo XGBoost pronóstico

Modelo	XGBoost
Hiperparámetros	Número de estimadores: 400
	Máxima profundidad: 5
	Fracción de la submuestra: 0.7
	Tasa de aprendizaje: 0.02
Escalador utilizado	MinMaxScaler
Tamaño de ventana	12
Paso de salida	1

Los resultados obtenidos mediante este modelo son igual de destacables, ya que poseen errores muy bajos y una capacidad de representar el 90 % de la variabilidad del sistema. Sin embargo, estos resultados son ligeramente inferiores a los obtenidos mediante LSTM. Esto último es posible que se deba tanto a la complejidad de los modelos como a las variables de entrada que recibió cada modelo, ya que, como se puede observar en la siguiente tabla, ambos modelos fueron entrenados con distintas variables de entrada.

Tabla 6.9: Comparativa de variables de entrada en el entrenamiento y ventanas de los modelos LSTM y XGBoost para pronóstico

Modelo	Variables de entrada	Tamaño de ventana
LSTM	Temperatura, Salinidad, Conductividad y voltaje	14
XGBoost pronóstico	Fecha, Hora y voltaje	12

Considerando las características observadas en la tabla anterior, es destacable que un algoritmo de aprendizaje supervisado obtuviera resultados casi igual de buenos que uno de aprendizaje profundo, a pesar de que el primero recibiera una ventana menor y menos variables para el entrenamiento. Esto podría deberse a que los hiperparámetros durante el entrenamiento del modelo XGBoost hayan sido muy finos, de tal forma que se logró un alto nivel predictivo sin tener que recurrir al *overfitting*.

6.2. Comparación entre los modelos obtenidos

Tabla 6.10: Comparativa de resultados de evaluación de los distintos modelos

Modelo	MSE	MAE	R2
KNN	0.000 298	0.013 046	−1.400 088
XGBoost	0.018 953	0.105 435	−3.275 825
LSTM	0.000 392	0.0147	0.911 938
XGBoost pronóstico	0.000 050	0.0053	0.900 026

Como se observa en la tabla anterior, los modelos enfocados en la tarea de pronóstico obtuvieron resultados notablemente superiores en comparación con los de regresión. Esto, como se mencionó durante la caracterización de cada modelo, pudo haber sido influenciado por la caoticidad (alta variabilidad de datos) del *atset* de entrenamiento. Aun así, otra característica importante que deja claro el porqué de los resultados es que el sistema tiene un comportamiento temporal. Esto quiere decir que la variable voltaje está altamente correlacionada con el paso del tiempo y por ello el uso de series de tiempo afectó positivamente la mimetización del sistema.

Además, dentro de los modelos de pronóstico, es necesario destacar que, a pesar de que el modelo XGBoost fue entrenado con una ventana más pequeña (2 valores de retardo menos) y con variables distintas a las utilizadas por el modelo LSTM, obtuvo resultados muy positivos y solo ligeramente inferiores a los del modelo con mayor ventana. Esto deja en claro que, para tareas específicas, los algoritmos de aprendizaje supervisado clásicos siguen siendo herramientas altamente competitivas frente a modelos de aprendizaje profundo, con la capacidad de ofrecer resultados muy cercanos.

6.3. Comparación con otros modelos previos

Como se mencionó en el estado del arte de este trabajo, han existido estudios que han tratado de modelar este tipo de sistemas (principalmente CCM). Aun así, los resultados obtenidos mediante técnicas tradicionales han tenido sus limitaciones, principalmente en la generalización del sistema cuando se trata de subidas o bajadas.

No obstante, han existido trabajos como el de Hossain et al., 2023, que, mediante el uso del algoritmo Super Learner, han logrado resultados considerablemente buenos con un $R^2=0.995$ y un MSE de 0.032. Mientras tanto, los algoritmos LSTM y XGBoost pronóstico realizados en este trabajo presentan un MSE mucho más bajo en comparación con el trabajo de Hossain et al., de hasta 0.000392, y un $R^2=0.911938$. Si bien los modelos desarrollados durante este trabajo tienen un valor más bajo en la representación de la variabilidad del sistema, poseen un error cuadrático promedio mucho menor. En una situación real, un error de 0.18 V en un voltaje normalizado de 0-1 V, como el del primer caso, es considerablemente alto, mientras que en el caso de este trabajo significaría un error de 0.02 V, lo cual es mucho más preciso, considerando que las CCMS producen en el rango de mV.

Capítulo 7

Conclusiones

En este apartado se interpretan los resultados obtenidos por cada modelo, y se identifican sus fortalezas, así como debilidades, para determinar cuál de los 4 modelos entrenados y evaluados se presenta como una herramienta óptima para la mimetización del voltaje de una CCMS.

7.1. Conclusión

La metodología desarrollada durante este trabajo ha permitido realizar dos modelos predictivos con resultados muy positivos y con la clara posibilidad de mimetizar el voltaje de una CCMS. Esto se logró mediante los modelos LSTM y XGBoost enfocados en pronóstico, los cuales presentaron errores cuadráticos promedios inferiores frente a trabajos previos, como se mencionó en la discusión de resultados.

7.2. Limitaciones y retos

Durante este trabajo se realizaron pruebas con distintos modelos. Si bien, durante el entrenamiento todos los modelos alcanzaron resultados aceptables, los modelos de regresión KNN y XGBoost presentaron una negativa capacidad de mimetizar las CCMS. Esto, como se mencionó en las discusiones, fue causado por la caoticidad de los datos utilizados durante el entrenamiento, los cuales, si bien fueron filtrados e imputados, estos procesos no lograron eliminar todos los errores causados durante las mediciones por parte del sistema automatiza-

do de monitoreo y control. Aun así, en trabajos futuros, una vez realizadas las correcciones pertinentes en el sistema automatizado, los nuevos datos deberían ser capaces de ser útiles para un mejor entrenamiento de los modelos de regresión y con esto poder generar modelos predictivos mediante esta tarea.

7.3. Aportaciones

Este trabajo presenta dos herramientas predictivas diseñadas con distintas variables de entrada y con capacidades predictivas muy altas, las cuales podrán ser utilizadas para buscar la optimización de las CCMS en un futuro, para mejorar su eficiencia energética, a través de realizar experimentación sin recurrir a un medio físico para comprender estos sistemas dinámicos. En conclusión, se logró modelar la dinámica de las CCMS a través de algoritmos de aprendizaje supervisado, así como de aprendizaje profundo. Esto último demuestra que la capacidad de técnicas como XGBoost a la hora de realizar el modelado de sistemas es tan competitiva como la de modelos de aprendizaje profundo.

7.4. Trabajos a futuro

Durante el desarrollo de este trabajo se han presentado varias áreas de oportunidad a abordar en un futuro con el objetivo de mejorar la capacidad predictiva, especialmente en modelos de regresión:

- Obtener un conjunto de datos procedente de un sistema automatizado de monitoreo y control de una CCMS para su preprocesamiento y limpieza para su utilización en modelos de aprendizaje supervisado.
- Mejorar la calidad de los datos a través de reducir los errores de medición causados por el sistema automatizado de monitoreo y control.
- Incorporar variables al sistema automatizado como el Oxígeno Disuelto, para mejorar las predicciones, debido a que estudios previos como el de Feregrino-Rivas et al., 2023 definen que están correlacionadas al voltaje.

- Aumentar el tamaño del dataset, a 3 o más meses (el utilizado durante este trabajo se vio comprendido de unas 5 semanas).

Estas mejoras ayudarán a los modelos de regresión a mejorar su generalización del sistema, al reducir el nivel de ruido durante el entrenamiento, además de crear más correlaciones a través de la variable de Oxígeno Disuelto.

Bibliografía

- Abeliuk, A., & Gutiérrez, C. (2021). Historia y evolución de la inteligencia artificial. *Revista de Inteligencia Artificial*, 14-21 (page 12).
- Carabantes López, M. (2015). Cuatro problemas irresolubles de la IA simbólica. *Revista de Filosofía*, 40(1), 81-104. <https://doi.org/10.5209/rev.RESF.2015.v40.n1.48441> (page 12).
- Chai, T., & Draxler, R. R. (2014). Root mean square error (RMSE) or mean absolute error (MAE)? Arguments against avoiding RMSE in the literature. *Geoscientific Model Development*, 7(3), 1247-1250. <https://doi.org/10.5194/gmd-7-1247-2014> (page 23).
- Feregrino-Rivas, M., Ramirez-Pereda, B., Estrada-Godoy, F., Cuesta-Zedeño, L. F., Rochín-Medina, J. J., Bustos-Terrones, Y. A., & Gonzalez-Huitron, V. A. (2023). Performance of a sediment microbial fuel cell for bioenergy production: Comparison of fluvial and marine sediments. *Biomass and Bioenergy*, 168, 106657. <https://doi.org/10.1016/j.biombioe.2022.106657> (pages 1, 9, 10, 26, 28, 29, 57).
- Gómez, P. (2025). Aprendizaje Profundo [Apuntes de curso]. <https://ccc.inaoep.mx/~pgomez/cursos/IC-I/acetatos/ApProfundo.pdf> (page 19).

- Hossain, S. M., Sultana, N., Haji, S., Mufeez, S., Janahi, S., & Ahmed, N. (2023). Modeling of microbial fuel cell power generation using machine learning-based super learner algorithms. *Fuel*, 349, 128646. <https://doi.org/10.1016/j.fuel.2023.128646> (pages 2, 28, 29, 55).
- Hubenova, Y., Bardarov, I., Hubenova, E., & Slavcheva, E. (2024). Sediment microbial fuel cells capable of powering outdoor environmental monitoring sensors. *Sensing and Bio-Sensing Research*, 46, 100695. <https://doi.org/https://doi.org/10.1016/j.sbsr.2024.100695> (page 2).
- James, G., Witten, D., Hastie, T., Tibshirani, R., & Taylor, J. (2023). *An Introduction to Statistical Learning: With Applications in Python*. Springer. (Pages 15, 18, 20).
- Jiménez-Escamilla, M. G., Garibay-Orijel, C., & Borja-Salin, M. A. (2018). Modelo bioquímicamente estructurado para la estimación de la eficiencia de una celda de combustible microbiana. *Revista Internacional de Contaminación Ambiental*, 34(2), 331-345. <https://doi.org/10.20937/RICA.2018.34.02.13> (page 27).
- Kim, E., Yamada, Y., & Okamoto, S. (2022). Detection Error Contaminated by Outliers to Classify Density Profiles Dependent on the Relative Speed Between a MIMO Sensor and a Human Hand. *IEEE Access*, 10, 3201563. <https://doi.org/10.1109/ACCESS.2022.3201563> (page 24).
- Kofman, P. M. (2012). *Materia y Energía: Una Introducción a la Ingeniería [Notas de cátedra]*. https://www.fceia.unr.edu.ar/~kofman/files/eci_MyS_1.pdf (page 11).
- Kroese, D. P., Botev, Z. I., Taimre, T., & Vaisman, R. (2024). *Data Science and Machine Learning: Mathematical and Statistical Methods*. The University of Queensland. <https://people.smp.uq.edu.au/DirkKroese/DSML/DSML.pdf> (page 18).

- Lee, Y.-S., An, J., Kim, B., Park, H.-W., Kim, J.-H., & Chang, I. S. (2015). Increased Power in Sediment Microbial Fuel Cell: Facilitated Mass Transfer via a Water-Layer Anode Embedded in Sediment. *PLoS ONE*, 10(12), e0145430. <https://doi.org/10.1371/journal.pone.0145430> (pages 1, 2).
- López, D., Gelvez Garcia, N. Y., & Herrera, J. (2015). Desarrollo de un Modelo Predictivo para la Estimación del Comportamiento de Variables en una Infraestructura de Red. *Información Tecnológica*, 26(5), 143-154. <https://doi.org/10.4067/S0718-07642015000500018> (page 25).
- López Peláez, A. (2023). Lección inaugural del curso académico 2023-2024: ¿Estamos preparados para la inteligencia artificial? https://www.ubu.es/sites/default/files/portal_page/files/leccion_inaugural_2023-2024.pdf (page 13).
- Montgomery, D. C., Peck, E. A., & Vining, G. G. (2012). *Introduction to Linear Regression Analysis* (5.^a ed.). John Wiley & Sons. (Page 24).
- Naciones Unidas. (2015, septiembre). *La Asamblea General adopta la Agenda 2030 para el Desarrollo Sostenible*. Consultado el 15 de junio de 2025, desde <https://www.un.org/sustainabledevelopment/es/2015/09/la-asamblea-general-adopta-la-agenda-2030-para-el-desarrollo-sostenible/> (page 3).
- Organización Internacional del Trabajo (OIT). (2022). *Informe Mundial sobre el Futuro del Trabajo: Aprendizaje, innovación y transformación en la era digital* (inf. téc.). Organización Internacional del Trabajo (OIT). https://www.ilo.org/sites/default/files/wcmsp5/groups/public/@dgreports/@dcomm/documents/publication/wcms_856649.pdf (page 4).

- Rahimnejad, M., Adhami, A., Darvari, S., Zirepour, A., & Oh, S.-E. (2015). Microbial fuel cell as new technology for bioelectricity generation: A review. *Alexandria Engineering Journal*, 54(3), 745-756. <https://doi.org/https://doi.org/10.1016/j.aej.2015.03.031> (page 2).
- Romero Ramayo, J. (2021). *Modelado dinámico de una celda de combustible microbiana utilizando espectroscopía de impedancia electroquímica* [Tesis de maestría]. Universidad Nacional del Comahue. <https://rdi.uncoma.edu.ar/handle/uncomaid/11084> (page 27).
- Russell, S., & Norvig, P. (2021). *Artificial intelligence: A modern approach* (4th, global). Pearson Education. (Pages 13, 17, 19-23).
- scikit-learn developers. (n.d.). *Model evaluation: Quantification of the quality of predictions*. Consultado el 15 de junio de 2025, desde https://scikit-learn.org/stable/modules/model_evaluation.html (page 22).
- Vinuesa, J. (2019). *Tema 9: Regresión*. Consultado el 15 de junio de 2025, desde https://www.ccg.unam.mx/~vinuesa/R4biosciences/docs/Tema9_regresion.html (page 14).