

**Universidad De Los Andes**



**Facultad De Ingeniería**

**Departamento De Ingeniería Mecánica**

**Proyecto Individual En Ingeniería Mecánica**

**Análisis Y Predicción De Radiación En Sistemas Fotovoltaicos Haciendo Uso De  
Machine Learning**

**Autor**

**Alejandro Herrera Jiménez**

**a.herrera20@uniandes.edu.co**

**201915788**

**Asesor**

**Andrés Leonardo Gonzáles Mancera, Ph.D.**

**angonzal@uniandes.edu.co**

**Profesor Asociado**

**Bogota – Junio 2023**



**Análisis Y Predicción De Radiación En Sistemas Fotovoltaicos Haciendo Uso De  
Machine Learning**

**Alejandro Herrera Jiménez**  
**a.herrera20@uniandes.edu.co**  
**201915788**

**Proyecto individual En Ingeniería Mecánica**

**Asesor**  
**Andrés Leonardo Gonzáles Mancera, Ph.D.**  
**angonzal@uniandes.edu.co**  
**Profesor Asociado**

**Universidad De Los Andes**  
**Facultad De Ingeniería**  
**Departamento De Ingeniería Mecánica**  
**Bogotá – Junio 2023**

# **AGRADECIMIENTOS**

En primer lugar, quiero agradecer a mi familia por su apoyo incondicional a lo largo de todo el proceso. Su amor, paciencia y comprensión fueron fundamentales para superar los desafíos y mantenerme motivado. Agradezco a mis padres, hermanos y seres queridos por su constante apoyo y aliento.

También quiero expresar mi profundo agradecimiento al profesor asesor, Andrés Leonardo Gonzáles Mancera, por su invaluable orientación y asesoramiento a lo largo de este proyecto. Su experiencia, conocimientos y dedicación fueron fundamentales para orientar mis esfuerzos en la dirección correcta y alcanzar resultados satisfactorios. Su guía y comentarios constructivos fueron de gran ayuda en cada etapa del proyecto.

También quiero agradecer a la institución educativa por brindarme los recursos necesarios para llevar a cabo este proyecto y por proporcionarme un entorno propicio para aprender y crecer académicamente.

## RESUMEN

Este proyecto se enfocó en desarrollar un modelo de predicción de radiación solar utilizando una red neuronal LSTM. El objetivo principal fue predecir la radiación solar generada por una planta, lo cual es crucial para la planificación y operación eficiente de instalaciones solares. El desarrollo de un modelo de predicción de radiación solar tiene aplicaciones importantes en la industria de la energía solar. Conocer con antelación la radiación solar esperada permite optimizar la generación de energía y tomar decisiones informadas sobre la operación de las plantas solares. Además, un modelo preciso de predicción de radiación solar puede contribuir a la planificación a largo plazo de proyectos solares y a la integración eficiente de la energía solar en la red eléctrica. En este sentido, este proyecto busca aportar al avance en el campo de la energía renovable y proporcionar herramientas útiles para mejorar la eficiencia y confiabilidad de las plantas solares.

Para lograr esto, se realizaron varias etapas. En primer lugar, se llevó a cabo el preprocesamiento de los datos, que incluyó la limpieza de datos faltantes y la transformación de variables para su posterior análisis. Luego, se seleccionó y optimizó el modelo de red neuronal LSTM, explorando diferentes arquitecturas y hiperparámetros. La búsqueda de la arquitectura óptima se realizó utilizando técnicas como la búsqueda bayesiana, lo que permitió encontrar la combinación ideal de capas LSTM, número de neuronas, dropout y learning rate. Se evaluaron múltiples combinaciones para encontrar la arquitectura que proporcionara la mejor aproximación de los datos reales.

Una vez entrenado el modelo, se evaluó su desempeño utilizando un conjunto de datos de prueba. Se realizaron visualizaciones para comparar los valores reales con las predicciones del modelo, y se calculó la correlación entre ellos. Los resultados mostraron que el modelo logró una alta correlación ( $R = 0.983$ ) entre los datos reales y las predicciones, lo que indica una buena aproximación. Sin embargo, se identificó que algunas predicciones se desviaron en las horas nocturnas, donde la radiación es cercana a cero.

**Palabras Clave:** Machine Learning, Deep Learning, Redes neuronales, LSTM, Energías Renovables, Plantas Fotovoltaicas, Radiación solar, Búsqueda Bayesiana.

# TABLA DE CONTENIDO

AGRADECIMIENTOS.....	4
RESUMEN.....	5
TABLA DE CONTENIDO.....	6
LISTA DE ILUSTRACIONES.....	8
LISTA DE TABLAS.....	8
1. INTRODUCCIÓN .....	9
1.1 Energías Renovables.....	9
1.2 Machine Learning.....	12
2. OBJETIVOS.....	14
2.1. Objetivo General.....	14
2.2. Objetivos específicos.....	14
3. MARCO TEORICO.....	15
3.1. Energías Renovables.....	15
3.1.1. Energía solar .....	15
3.1.2. Radiación solar .....	15
3.1.3. Sistemas fotovoltaicos .....	16
3.1.4. Paneles fotovoltaicos .....	16
3.1.5. Humedad relativa.....	16
3.1.6. Temperatura ambiente.....	16
3.1.7. Parámetros medidos .....	17
3.2. Machine Learning.....	17
3.2.1. Deep Learning.....	17
3.2.2. Redes neuronales .....	17
3.2.3. Redes neuronales Recurrentes (RNN) .....	18
3.2.4. Long-Short Term Memory (LSTM).....	19
3.2.5. Dropout .....	21
3.2.6. Hiperparámetros.....	22
3.2.6.1. Número y tamaño de capas .....	22

3.2.6.2.	Función de activación.....	22
3.2.6.3.	Learning rate .....	23
3.2.6.4.	Optimizador.....	23
3.2.6.5.	Epoch.....	23
3.2.6.6.	Batch Size.....	23
3.2.6.7.	Funciones de coste .....	23
3.2.7.	Optimización Bayesiana .....	24
4.	METODOLOGIA .....	24
4.1.	Selección de datos .....	25
4.2.	Preprocesamiento.....	27
4.3.	Selección del modelo.....	29
4.4.	Entrenamiento del modelo.....	32
4.5.	Evaluación del modelo .....	32
4.6.	Mejoramiento del modelo.....	35
4.7.	Implementación del modelo .....	36
5.	RESULTADOS.....	36
6.	DISCUSION.....	37
7.	CONCLUSIONES .....	38
8.	RECOMENDACIONES .....	39
	REFERENCIAS .....	40
	ANEXOS.....	43

LISTA DE ILUSTRACIONES

Ilustración 1¿Cómo funcionan las plantas fotovoltaicas? ..... 11

Ilustración 2Proceso del Machine Learning ..... 13

Ilustración 3Arquitectura de una red neuronal ..... 18

Ilustración 4. Diagrama de grafo cíclico de una RNN (Charniak, 2019) ..... 19

Ilustración 5Estructura celda LSTM (Codificando Bits, 2019)..... 21

Ilustración 6Dropout aplicado a una red neuronal (Yadav, 2023). ..... 22

Ilustración 7Software VCOM Universidad de los Andes..... 26

Ilustración 8 Código preprocesamiento de datos..... 28

Ilustración 9 Arquitectura red neuronal (¿Qué es una Red Neuronal? | Codificando Bits, s. f.). ..... 29

Ilustración 10 Gráfico de comparación de datos reales vs predichos. .... 33

Ilustración 11 Gráfico de correlación entre datos reales vs predichos. .... 34

Ilustración 12 Implementación de la predicción a futuro de la radiación ..... 36

LISTA DE TABLAS

Tabla 1 Datos Seleccionados ..... 27



# 1. INTRODUCCIÓN

## 1.1 Energías Renovables

A lo largo de la historia, la creatividad humana ha dado lugar a una amplia gama de métodos para la generación de energía. Un ejemplo destacado es el invento de la máquina de vapor en el siglo XVIII. Este ingenioso artefacto, alimentado por una fuente de calor, desencadenaba un cambio de fase en el agua, transformando la energía térmica en energía potencial o incluso en energía eléctrica. Este hito marcó el inicio de la revolución industrial y sentó las bases para innumerables avances tecnológicos hasta nuestros días, trayendo así nuevas formas de generar energía potencial o eléctrica (Ruiz Molano & García Martínez, s. f.).

En la actualidad, en Colombia se cuentan con dos fuentes principales de generación de energía: los recursos hídricos, los cuales representan aproximadamente el 63.7% de la producción total de energía en el país, y los recursos térmicos, como el gas, carbón, fueloil y combustóleo, que contribuyen con un 31.5% (Como se genera la electricidad | Enel Colombia, s. f.). Sin embargo, es importante destacar que estas fuentes de generación de energía generan una cantidad significativa de contaminantes, lo cual resulta perjudicial para el medio ambiente. Por consiguiente, resulta imperativo buscar alternativas más respetuosas con el entorno natural.

Las hidroeléctricas desempeñan un papel fundamental en la generación de energía, proporcionando una fuente renovable y confiable de electricidad. Sin embargo, según investigaciones realizadas, se han identificado nueve problemas ambientales principales asociados con las represas hidroeléctricas. Estos problemas incluyen la ocupación de tierras destinadas a uso residencial y actividades agrícolas, la pérdida de vegetación, la erosión del suelo, la contaminación del agua, la destrucción de la vegetación y la pérdida total de suelos en los lugares de construcción, el impacto en las comunidades humanas cercanas, la afectación a la reproducción y desove de la fauna de agua dulce, la interrupción de los procesos migratorios de los peces, la deforestación, la acumulación de sedimentos en los embalses, la contribución al cambio climático y la posible relación con la actividad sísmica local (La Marca et al., 2018). Estos problemas ambientales resaltan la necesidad de abordar los impactos negativos de las represas hidroeléctricas en el medio ambiente y fomentar el desarrollo de alternativas más sostenibles y respetuosas con el entorno.

Por otro lado, la producción de energía mediante recursos térmicos también causa daños al medio ambiente, ya que genera emisiones de gases tóxicos, como dióxido de carbono,

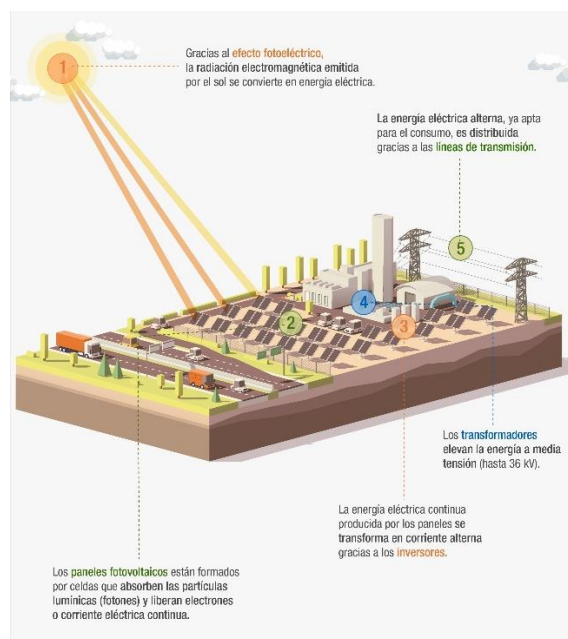
dióxido de nitrógeno y azufre. Estas emisiones contribuyen al aumento del efecto invernadero y a la degradación de la capa atmosférica. Como resultado, en los últimos años se han propuesto sistemas de almacenamiento y conversión de energía que aprovechan el potencial de la energía solar (Jiménez Martínez et al., 2016). Estas soluciones buscan contrarrestar los impactos ambientales negativos al ofrecer una alternativa más limpia y sostenible.

En el contexto actual, las energías renovables desempeñan un papel fundamental en la búsqueda de alternativas más sostenibles y respetuosas con el medio ambiente. Estas fuentes de energía se derivan de fuentes naturales que se reponen de manera más rápida de lo que se consumen, lo que las convierte en opciones viables y prometedoras para abordar la problemática del agotamiento de los recursos y la mitigación de la crisis provocada por el cambio climático. Entre las diversas formas de energías renovables, se destacan la energía solar, eólica, hidráulica, geotérmica, oceánica, bioenergía, entre otras. La energía solar se obtiene a partir de la radiación solar, aprovechando la luz del sol para generar electricidad o calor. Por otro lado, la energía eólica se obtiene del viento, utilizando aerogeneradores que transforman la energía cinética del viento en energía eléctrica. La energía hidráulica, por su parte, aprovecha el movimiento del agua, ya sea de ríos o de mares, para generar energía eléctrica mediante el uso de turbinas. La energía geotérmica se obtiene del calor del interior de la Tierra, aprovechando el vapor o el agua caliente para generar electricidad o calor. La energía oceánica se deriva de las mareas, corrientes marinas y diferencias de temperatura en el agua. La bioenergía, por su parte, se produce a partir de la biomasa, como los residuos agrícolas, forestales o urbanos, que se convierten en combustible para generar energía (United Nations, s. f.).

Al abordar los sistemas de energía solar, es fundamental destacar la importancia de la tecnología fotovoltaica. Esta tecnología, basada en materiales y dispositivos diseñados para convertir la energía solar en energía eléctrica, se ha convertido en una opción ampliamente utilizada y prometedora para la generación de energía renovable. En el proceso de conversión, los rayos solares, compuestos por fotones cargados de energía, inciden en una celda fotovoltaica, donde pueden ser reflejados o absorbidos. Cuando los fotones son absorbidos, transfieren su energía a los electrones de los átomos de la celda, permitiendo que escapen de su posición inicial y generen una corriente eléctrica en un circuito conectado (Droege, 2017). Esta tecnología presenta numerosas ventajas, como su capacidad para generar electricidad de forma silenciosa y sin emisiones contaminantes, así como su modularidad y escalabilidad. La tecnología fotovoltaica es clave en la transición hacia un modelo energético más sostenible y en la mitigación de los impactos ambientales de las fuentes de energía convencionales.

Las plantas fotovoltaicas han surgido como resultado de esta tecnología. En la ilustración 1 se ilustra el proceso de funcionamiento de estas plantas. En primer lugar, los paneles solares absorben la radiación solar. En segundo lugar, los paneles convierten esa radiación en energía eléctrica de corriente continua. En tercer lugar, los inversores transforman la corriente continua en corriente alterna. En cuarto lugar, los transformadores elevan la energía para obtener una tensión adecuada para su uso. Finalmente, la corriente se distribuye a través de las líneas de transmisión para su consumo regular. Este proceso permite aprovechar de

manera eficiente la energía solar y contribuye a la generación de electricidad de manera sostenible.



*Ilustración 1 ¿Cómo funcionan las plantas fotovoltaicas?*

La recolección de datos generados por los sistemas fotovoltaicos desempeña un papel crucial en el análisis de su eficiencia y flujos de energía, lo que a su vez permite identificar oportunidades de mejora y realizar cambios para optimizar su rendimiento. En este sentido, la norma IEC 61724-1 se establece como un marco de referencia fundamental, ya que describe los equipos, métodos y terminología necesarios para el monitoreo y análisis de sistemas fotovoltaicos. Dentro de los lineamientos establecidos por esta norma, se considera la utilización de sensores especializados que capturan los datos relevantes de los parámetros medidos en los sistemas fotovoltaicos. Estos sensores permiten recopilar información precisa sobre variables como la radiación solar incidente, la temperatura, la corriente y el voltaje generados, entre otros aspectos clave. Además, se enfatiza la importancia de la adquisición de datos precisa y confiable, para garantizar que la información recopilada sea representativa de las condiciones reales de operación.

Además, en la vanguardia de la investigación en sistemas fotovoltaicos se ha explorado el uso de técnicas de aprendizaje automático (machine learning) para realizar predicciones futuras basadas en los datos recopilados. Al analizar patrones y correlaciones en los datos históricos, los algoritmos de machine learning pueden generar modelos predictivos que estiman el rendimiento y los flujos de energía esperados en condiciones futuras. Esto resulta muy útil para la planificación y toma de decisiones estratégicas, permitiendo a los gestores de sistemas fotovoltaicos anticipar posibles escenarios y optimizar la operación de manera proactiva. Además, la predicción precisa de datos futuros contribuye a mejorar la eficiencia,

reducir costos y maximizar la producción de energía solar. Así, se fortalece el papel de las energías renovables en la transición hacia un futuro sostenible y libre de emisiones.

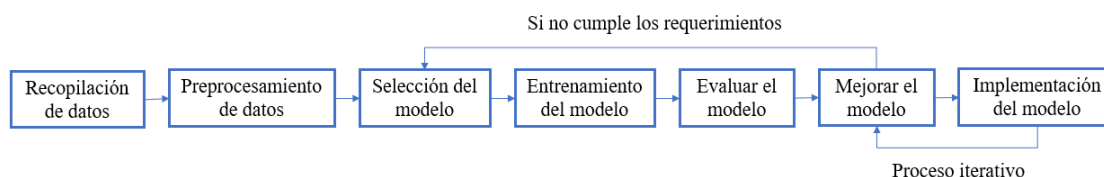
## 1.2 Machine Learning

El aprendizaje automático, también conocido como machine learning, es un campo de estudio de la inteligencia artificial que ha adquirido una importancia fundamental en diversos sectores, incluido el de los sistemas fotovoltaicos. Esta disciplina permite a investigadores y profesionales aprovechar el poder de algoritmos y modelos computacionales para analizar grandes volúmenes de datos y descubrir patrones, correlaciones y tendencias. En el contexto de los sistemas fotovoltaicos, el machine learning se emplea para realizar predicciones futuras basadas en datos recopilados, lo que contribuye a optimizar la eficiencia y el rendimiento de estos sistemas. Mediante el desarrollo de modelos predictivos precisos, los responsables de los sistemas fotovoltaicos pueden anticipar escenarios y tomar decisiones estratégicas para maximizar la producción de energía solar.

En el campo del machine learning, existen principalmente cinco tipos de aprendizaje: supervisado, no supervisado, semi-supervisado, por refuerzo y profundo. El aprendizaje supervisado utiliza datos etiquetados para entrenar un modelo y predecir salidas basadas en ejemplos de entrada y sus etiquetas correspondientes. Por otro lado, el aprendizaje no supervisado se enfoca en descubrir patrones y estructuras ocultas en los datos sin la necesidad de etiquetas, lo que permite explorar y segmentar información sin conocimiento previo. El aprendizaje semi-supervisado combina elementos del aprendizaje supervisado y no supervisado, utilizando una combinación de datos etiquetados y no etiquetados para entrenar modelos más precisos y eficientes. El aprendizaje por refuerzo se centra en la toma de decisiones a través de la interacción con un entorno, donde un agente aprende a maximizar una recompensa acumulativa. Por último, el aprendizaje profundo se basa en redes neuronales, que pueden aprender y extraer características complejas y de alto nivel a partir de datos sin procesar, lo que permite un procesamiento más sofisticado y representaciones más abstractas (New Advances in Machine Learning, 2010).

El proceso del machine learning implica una serie de pasos clave para el desarrollo exitoso de un proyecto (Ilustración 2). El primer paso es la recopilación de datos, que consiste en recoger y preparar los datos relevantes para el problema en cuestión. A continuación, se realiza el preprocesamiento de datos, que implica limpiar y transformar los datos para eliminar ruido, manejar valores faltantes y normalizar características, entre otros procesos. Luego viene la selección del modelo, donde se elige la arquitectura y el algoritmo más adecuados para abordar el problema en cuestión. A continuación, se lleva a cabo el entrenamiento del modelo, donde se utilizan los datos recopilados y preprocesados para ajustar los parámetros del modelo y mejorar su capacidad para hacer predicciones precisas. Una vez entrenado, se procede a evaluar el modelo utilizando datos de prueba o validación. Esto implica medir su rendimiento en términos de métricas como la precisión, la exactitud o

el error. Si el modelo no cumple con los criterios deseados, se procede a mejorar el modelo, ajustando hiperparámetros, explorando diferentes técnicas de regularización o utilizando técnicas de optimización. Finalmente, se llega a la fase de implementación del modelo, donde se utiliza para realizar predicciones en datos no vistos en tiempo real. Es importante tener en cuenta que este proceso es iterativo, ya que se pueden realizar ajustes adicionales en cualquier etapa para mejorar aún más el rendimiento del modelo y garantizar su eficacia en la toma de decisiones o la solución de problemas (Bagnato, 2019).



*Ilustración 2 Proceso del Machine Learning*

Para la predicción de datos con series de tiempo, uno de los enfoques más utilizados es el aprendizaje profundo, también conocido como deep learning. Este tipo de machine learning se basa en el uso de redes neuronales artificiales de múltiples capas para aprender y extraer características complejas de los datos. Algunos de los tipos más comunes de redes neuronales utilizadas en el aprendizaje profundo incluyen las redes neuronales convolucionales (CNN), que se destacan en el procesamiento de datos en forma de cuadrícula, como imágenes o señales; las redes neuronales recurrentes (RNN), diseñadas para capturar la información secuencial en los datos, como las series de tiempo; y las redes neuronales de alimentación directa (FNN), que son redes estándar que transmiten la información en una sola dirección, sin retroalimentación. Cada tipo de red neuronal tiene sus propias características y aplicaciones específicas, pero todas comparten el objetivo común de aprender y representar de manera eficiente la información contenida en los datos de series de tiempo para realizar predicciones precisas y útiles (Ghoshal, 2023).

Dentro del análisis de series de tiempo, las redes neuronales recurrentes (RNN) desempeñan un papel fundamental. Algunos de los algoritmos más conocidos y utilizados en este contexto son los siguientes: memoria a largo plazo a corto plazo (LSTM), redes neuronales bidireccionales (BRNN) y unidades recurrentes cerradas (GRU). Las LSTM son un tipo de red neuronal recurrente diseñada para capturar relaciones a largo plazo en los datos secuenciales, gracias a su estructura de celdas de memoria. Por otro lado, las BRNN procesan la información en dos direcciones, capturando tanto la información pasada como la futura en la secuencia temporal. Las GRU, por su parte, son una variante más simple de las LSTM que ha demostrado ser efectiva en el modelado de dependencias temporales en series de tiempo. Cada uno de estos algoritmos tiene sus propias características y ventajas, y su elección depende del problema específico y los datos disponibles. En general, las LSTM, las BRNN y las GRU han demostrado ser herramientas poderosas para el análisis de series de tiempo,

permitiendo capturar patrones complejos y realizar predicciones precisas en una amplia gama de aplicaciones (¿Qué son las redes neuronales recurrentes? | IBM, s. f.).

El desarrollo de algoritmos basados en machine learning para la predicción de datos en sistemas fotovoltaicos es una herramienta clave para realizar análisis exhaustivos y tomar decisiones estratégicas efectivas. Esta aplicación del machine learning impulsa activamente un futuro sostenible y libre de emisiones al maximizar la eficiencia y optimizar el rendimiento de los sistemas fotovoltaicos. Por tanto, el objetivo principal de este proyecto es crear un modelo preciso y adecuado para la predicción y análisis de datos en sistemas fotovoltaicos. Utilizando técnicas de machine learning y datos históricos, se busca desarrollar un modelo capaz de identificar patrones y tendencias en la producción de energía solar. Esto facilitará la toma de decisiones estratégicas para mejorar el rendimiento y la eficiencia de los sistemas, así como optimizar la distribución y el consumo de energía solar.

## **2. OBJETIVOS**

### **2.1.Objetivo General**

- Desarrollar un modelo de predicción basado en redes neuronales LSTM para pronosticar la radiación generada por una planta fotovoltaica, con el fin de mejorar la eficiencia y la gestión de la planta.

### **2.2.Objetivos específicos**

- Recopilar y preprocesar los datos necesarios para el entrenamiento del modelo, incluyendo la temperatura ambiente, temperatura de los módulos, humedad relativa y fecha.
- Diseñar y conilustraciórnr una arquitectura de redes neuronales LSTM adecuada para capturar las relaciones temporales en los datos y realizar la predicción de la radiación.
- Evaluar y validar el modelo de predicción utilizando conjuntos de datos de prueba independientes, analizando métricas de evaluación como el error cuadrático medio (MSE), para determinar su precisión y capacidad de generalización.

### **3. MARCO TEORICO**

#### **3.1.Energías Renovables**

Las energías renovables son un tipo de energía obtenida de fuentes naturales que tienen la capacidad de regenerarse rápidamente. Un ejemplo de estas fuentes incluye la luz solar y el viento, las cuales se renuevan de manera constante. Estas fuentes de energía son ampliamente disponibles en diferentes entornos. En contraste, los combustibles fósiles, como el carbón, el petróleo y el gas, son fuentes de energía no renovables que requieren cientos de millones de años para formarse. La combustión de combustibles fósiles produce emisiones perjudiciales, como los gases de efecto invernadero, entre ellos el dióxido de carbono.

La generación de energías renovables emite considerablemente menos emisiones en comparación con la quema de combustibles fósiles. La transición de los combustibles fósiles, que actualmente son responsables de la mayoría de las emisiones, hacia las energías renovables es esencial para abordar la crisis del cambio climático (United Nations, s. f.).

##### **3.1.1. Energía solar**

La energía solar es una forma de energía renovable que se obtiene a partir de la radiación del sol. Esta radiación solar puede ser capturada y utilizada de diversas maneras para satisfacer nuestras necesidades energéticas. Existen varios tipos de tecnologías solares que permiten aprovechar esta radiación de manera eficiente. Una de las tecnologías más comunes es la energía solar fotovoltaica, que utiliza paneles solares para convertir la luz solar directamente en electricidad. Otra forma de aprovechar la energía solar es a través de la energía solar térmica. Esta tecnología utiliza paneles solares térmicos para calentar agua o fluidos. Además, existe la energía solar pasiva, que se basa en el diseño arquitectónico y la orientación de los edificios para maximizar la entrada de luz solar y aprovechar su calor natural (¿Qué es la energía solar? Tipos y principales ventajas | Repsol, 2023).

##### **3.1.2. Radiación solar**

La radiación solar es la energía emitida por el Sol, la cual se propaga en todas las direcciones a través del espacio mediante ondas electromagnéticas. La energía solar se origina en el núcleo del Sol, donde reacciones nucleares de fusión del hidrógeno generan radiación electromagnética. El Sol emite energía en forma de radiación de onda corta. Sin embargo, antes de llegar a la superficie terrestre, esta radiación solar atraviesa la atmósfera, donde experimenta procesos de debilitamiento como la difusión, la reflexión en nubes y la absorción por moléculas de gases como el ozono y el vapor de agua, así como partículas en suspensión (RADIACIÓN SOLAR - IDEAM, s. f.).

### **3.1.3. Sistemas fotovoltaicos**

La energía solar fotovoltaica es una forma de energía renovable y limpia que utiliza la radiación solar como fuente principal para producir electricidad. Se basa en un fenómeno conocido como efecto fotoeléctrico, en el cual ciertos materiales tienen la capacidad de absorber partículas de luz llamadas fotones y liberar electrones, generando así una corriente eléctrica. Para aprovechar este proceso, se utiliza un dispositivo semiconductor llamado célula fotovoltaica. Las cuales pueden estar compuestas de diferentes materiales, como silicio monocristalino, silicio policristalino, silicio amorfo u otros materiales semiconductores de capa delgada (Iberdrola, s. f.). En los sistemas fotovoltaicos existen algunos factores externos importantes que deben ser medidos para evaluar la eficiencia en la producción energética como por ejemplo la humedad relativa y la temperatura ambiente.

### **3.1.4. Paneles fotovoltaicos**

Los paneles fotovoltaicos son estructuras compuestas por grupos de celdas fotovoltaicas dispuestas entre capas de silicio. Estas celdas tienen la capacidad de capturar la radiación solar y llevar a cabo un proceso de conversión mediante el cual transforman la luz solar, en forma de fotones, en energía eléctrica, liberando electrones en el proceso. Estos se construyen conectando en serie o en paralelo múltiples celdas fotovoltaicas para aumentar la capacidad de generación de electricidad. Estas celdas están encapsuladas en un material transparente que las protege de factores ambientales y les permite recibir la mayor cantidad de luz solar posible (Iberdrola, s. f.).

### **3.1.5. Humedad relativa**

La humedad relativa es una medida que indica la cantidad de vapor de agua presente en una mezcla de aire y agua en comparación con la cantidad máxima posible. Esta es una relación de humedad de una mezcla específica de aire y agua, en comparación con la relación de humedad de saturación a una temperatura determinada. Es importante tener en cuenta que la relevancia de la humedad relativa en una aplicación particular depende de conocer tanto la temperatura como la humedad relativa (Yahia, 2019).

### **3.1.6. Temperatura ambiente**

La temperatura ambiente se refiere a la temperatura promedio del entorno en el que nos encontramos. Es la temperatura que experimentamos en nuestro entorno cotidiano, sin la influencia directa de fuentes de calor adicionales o sistemas de enfriamiento. La temperatura ambiente puede variar según la ubicación geográfica, la temporada y las condiciones climáticas.



### **3.1.7. Parámetros medidos**

En los sistemas fotovoltaicos existen diferentes parámetros que deben ser medidos para obtener un riguroso análisis del sistema, esta medición se rige por la norma IEC 61724-1 Photovoltaic system performance – Part 1: Monitoring. En la cual se describen que parámetros deben medirse, como medirlos y con que dispositivos se debe hacer. Entre los requerimientos se encuentran que parámetros se deben medir con sensores en el lugar de la planta y cuales pueden ser medidos meteorológicamente, la ubicación y alineación de los sensores. Algunos de los parámetros a medir son: Irradiancia, Factores ambientales (temperatura, humedad, velocidad del viento), sistema de seguimiento y mediciones eléctricas.

## **3.2. Machine Learning**

El aprendizaje automático, también conocido como machine learning, se centra en el desarrollo de algoritmos y modelos estadísticos que permiten a los sistemas informáticos llevar a cabo tareas sin necesidad de instrucciones explícitas. En lugar de depender de comandos específicos, estos sistemas se basan en la detección de patrones e inferencias. Para lograrlo, utilizan algoritmos de machine learning para procesar grandes volúmenes de datos históricos e identificar patrones ocultos. Esta capacidad les permite generar resultados con mayor precisión a partir de un conjunto de datos de entrada. (¿Qué es el machine learning? - Explicación sobre el machine learning empresarial - AWS, s. f.). Existen varios tipos de aprendizaje automático entre los que se encuentran: supervisado, no supervisado, semi-supervisado, por refuerzo y profundo.

### **3.2.1. Deep Learning**

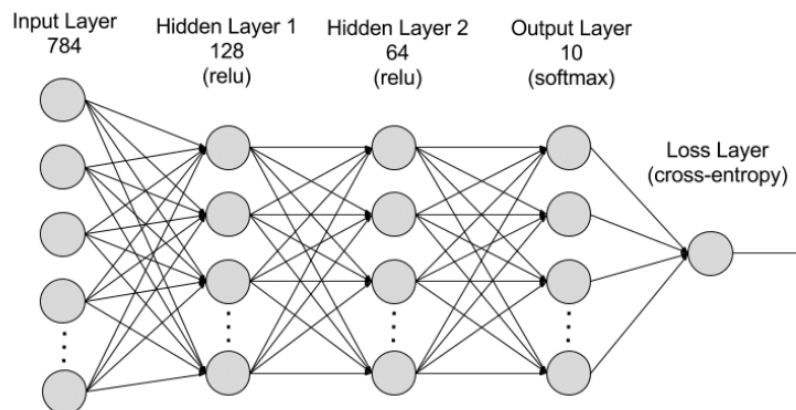
El deep learning o aprendizaje profundo es una rama del machine learning que se enfoca en el uso de redes neuronales con tres o más capas. Estas redes neuronales buscan imitar el funcionamiento del cerebro humano, lo que les permite "aprender" a partir de grandes volúmenes de datos. Mientras que una red neuronal con una sola capa puede realizar predicciones aproximadas, las capas ocultas adicionales contribuyen a optimizar y mejorar la precisión. En resumen, el deep learning utiliza redes neuronales profundas para realizar un aprendizaje más avanzado y refinado a partir de la información proporcionada (¿Qué es Deep Learning? | IBM, s. f.).

### **3.2.2. Redes neuronales**

Una red neuronal es un enfoque de la inteligencia artificial que busca enseñar a las computadoras a procesar datos de manera similar al cerebro humano. Se trata de una técnica dentro del campo del machine learning conocida como aprendizaje profundo, que utiliza una estructura de capas interconectadas, similar a las neuronas del cerebro, para procesar la información. Este enfoque crea un sistema adaptable en el que las computadoras aprenden de sus propios errores y mejoran de forma continua.

Las redes neuronales artificiales se aplican para resolver problemas complejos, como el resumen de documentos o el reconocimiento facial, con mayor precisión. Estas redes se entrenan utilizando grandes cantidades de datos y ajustando los pesos y conexiones entre las neuronas para lograr resultados más precisos en tareas específicas. De esta manera, se busca emular el funcionamiento del cerebro humano y utilizarlo como inspiración para el procesamiento de información en las computadoras (¿Qué es una red neuronal? - Explicación de las redes neuronales artificiales - AWS, s. f.) Existen varios tipos de redes neuronales entre los cuales se destacan los siguientes: redes neuronales convolucionales (CNN), las redes neuronales recurrentes (RNN) y las redes neuronales de alimentación directa (FNN).

Las redes neuronales consisten en múltiples capas ocultas que contienen millones de neuronas artificiales interconectadas. Estas conexiones entre nodos se representan mediante pesos, que pueden ser positivos si un nodo estimula a otro o negativos si un nodo inhibe a otro. Los nodos con pesos más altos ejercen una mayor influencia sobre los demás nodos. En la ilustración 3 se puede observar un ejemplo de la arquitectura de una red neuronal.

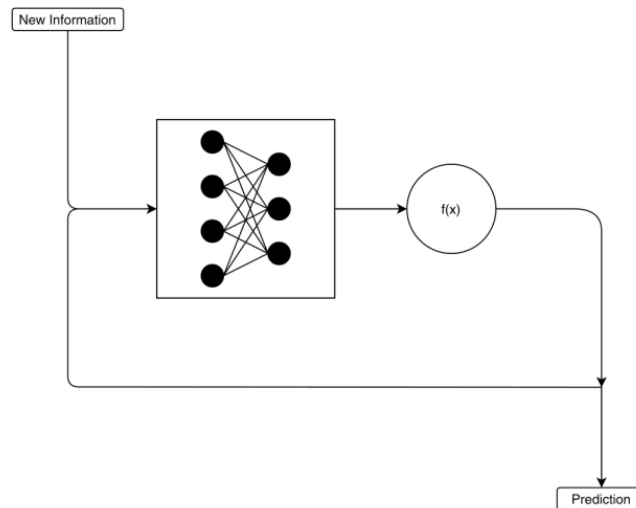


*Ilustración 3 Arquitectura de una red neuronal*

### 3.2.3. Redes neuronales Recurrentes (RNN)

A diferencia de otras redes neuronales, las RNN tienen la capacidad de reconocer y procesar características secuenciales de los datos, utilizando patrones para predecir el próximo escenario probable. Las RNN son especialmente efectivas en casos donde el contexto es crucial para realizar predicciones precisas. A través de bucles de retroalimentación (en la ilustración 4 se puede observar un ejemplo de la retroalimentación en una arquitectura de una RNN), las RNN procesan secuencias de datos, permitiendo que la información persista a lo largo del tiempo. Este efecto a menudo se describe como memoria, ya que las RNN pueden retener información previa y utilizarla para influir en los resultados futuros.

Las redes neuronales recurrentes son una herramienta poderosa en el aprendizaje profundo y han sido utilizadas para simular la actividad neuronal en el cerebro humano. Su capacidad para capturar dependencias a largo plazo en los datos secuenciales las convierte en una opción eficaz para problemas que requieren un entendimiento profundo del contexto y la evolución temporal de los datos (Redes neuronales recurrentes – KW Foundation, 2021).



*Ilustración 4. Diagrama de grafo cíclico de una RNN (Charniak, 2019)*

### 3.2.4. Long-Short Term Memory (LSTM)

Dentro de las RNN, se encuentra una arquitectura conocida como Long-Short Term Memory (LSTM), la cual aborda el desafío de las dependencias a largo plazo en el procesamiento de secuencias. En algunas situaciones, si la influencia del estado anterior en la predicción actual no proviene del pasado cercano, es posible que las RNN convencionales no puedan predecir con precisión el estado actual. Para solucionar este problema, las LSTM incorporan "celdas" en las capas ocultas de la red neuronal, las cuales están equipadas con tres puertas fundamentales: una puerta de entrada, una puerta de salida y una puerta de olvido. Estas puertas tienen la función de regular el flujo de información que se necesita para realizar predicciones precisas en la red (¿Qué son las redes neuronales recurrentes? | IBM, s. f.).

La puerta de entrada determina qué nueva información se debe agregar a la celda, mientras que la puerta de salida controla qué parte de la información de la celda se debe transmitir a la siguiente capa o a la salida final. Por otro lado, la puerta de olvido regula la cantidad de información antigua que se debe descartar de la celda. Mediante el uso de estas puertas, las LSTM pueden capturar dependencias a largo plazo en los datos secuenciales y mantener información relevante a lo largo del tiempo, lo que mejora significativamente la capacidad de predicción de la red neuronal.

Para añadir o remover datos de la memoria de la neurona LSTM usa 3 compuertas, compuerta del olvido, compuerta de actualización y la compuerta de salida, en la ilustración 5 se puede observar un ejemplo de esta arquitectura (Codificando Bits, 2019). La compuerta del olvido decide que información se descarta con la ecuación 1

$$f_t = \text{sigmoidal}(W_f[a_{t-1}, x_t] + b_f) \quad (1)$$

Donde  $a_t$  es el estado oculto anterior,  $x_t$  es la entrada actual,  $W_f$  y  $b_f$  son coeficientes que se aprenden durante el entrenamiento y como salida genera el vector  $f_t$ . Debido a la función sigmoidal si el valor del vector es  $f_t$  es 0 o cercano a 0 la celda LSTM elimina esa información y por el contrario si es 1 o cercano a 1 la información se mantendrá.

La compuerta de actualización permite actualizar la información con lo aprendido con la ecuación 2

$$u_t = \text{sigmoidal}(W_i[a_{t-1}, x_t] + b_i) \quad (2)$$

Donde  $a_t$  es el estado oculto anterior,  $x_t$  es la entrada actual,  $W_i$  y  $b_i$  son coeficientes que se aprenden durante el entrenamiento y como salida genera el vector  $u_t$ . En este caso si el vector  $u_t$  es 1 o cercano a 1 actualiza la información de lo contrario no es actualizada. A continuación, ya teniendo los datos de los vectores  $f_t$  y  $u_t$  se actualiza la celda de estado. Primero se elimina la información irrelevante como se muestra a continuación.

$$f_t \times c_{t-1} \quad (3)$$

Donde,  $c_{t-1}$  es el paso anterior de la celda de estado. A continuación, se crea un vector de valores candidatos ( $\tilde{c}_t$ ) con la ecuación 4

$$\tilde{c}_t = \text{tanh}(W_c[a_{t-1}, x_t] + b_c) \quad (4)$$

Donde,  $w_c$  y  $b_c$  son aprendidos durante el entrenamiento y estos valores se multiplican al vector  $u_t$  como se observa en la ecuación 5

$$u_t \times \tilde{c}_t \quad (5)$$

Y los resultados de las ecuaciones 3 y 5 son sumados obteniendo el nuevo estado de la celda de memoria

$$c_t = f_t \times c_{t-1} + u_t \times \tilde{c}_t \quad (6)$$

Finalmente, con la compuerta de salida se calcula el nuevo estado de la celda de memoria haciendo uso de ecuación 7 y ecuación 8

$$o_t = \sigma(W_o[a_{t-1}, x_t] + b_o) \quad (7)$$

$$\tanh(c_t) \quad (8)$$

Donde  $o_t$  es el vector de la compuerta de salida y  $w_o$  y  $b_o$  son nuevamente coeficientes que se aprenden durante el entrenamiento. La ecuación 8 se usa para verificar que el  $c_t$  se mantenga en el rango entre -1 y 1 y se filtran los valores de la compuerta de salida dando el resultado final ( $a_t$ ).

$$a_t = o_t \times \tanh(c_t) \quad (9)$$

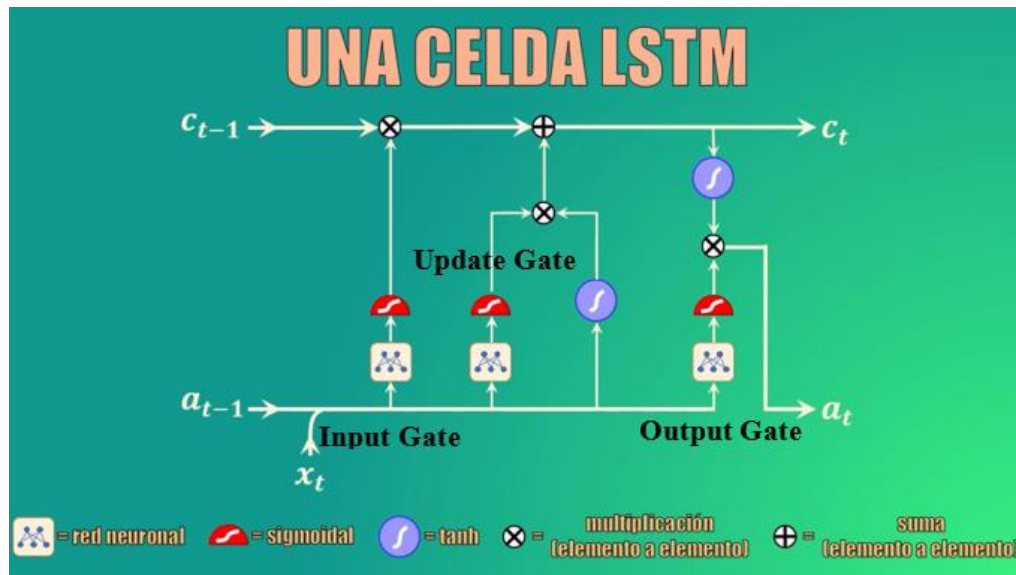
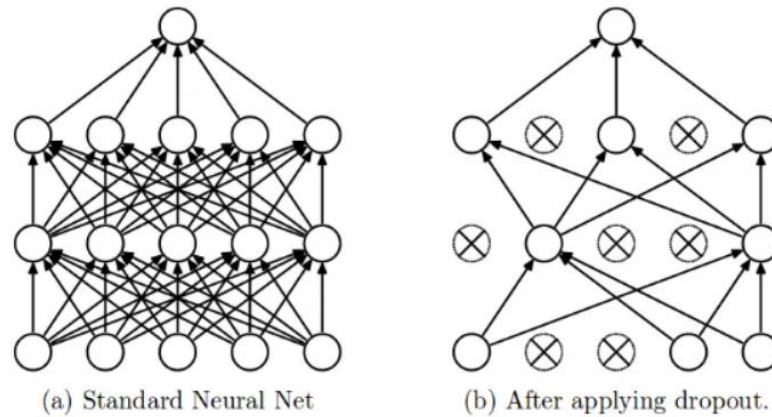


Ilustración 5 Estructura celda LSTM (Codificando Bits, 2019)

### 3.2.5. Dropout

El término "dropout" se refiere a la eliminación de nodos (tanto de la capa de entrada como de la capa oculta) en una red neuronal como se observa en la ilustración 6. Todas las conexiones hacia adelante y hacia atrás con un nodo eliminado se eliminan temporalmente, creando así una nueva arquitectura de red a partir de la red original. Los nodos se eliminan mediante una probabilidad de dropout de que varía usualmente entre 0 y 0.5 (Yadav, 2023).



*Ilustración 6 Dropout aplicado a una red neuronal (Yadav, 2023).*

### 3.2.6. Hiperparámetros

Los hiperparámetros son variables de configuración externa utilizadas para gestionar el entrenamiento de modelos de machine learning. A diferencia de los parámetros, que se derivan automáticamente durante el proceso de aprendizaje, los hiperparámetros se configuran manualmente antes de entrenar un modelo. Algunos ejemplos de hiperparámetros incluyen el número de nodos y capas en una red neuronal, así como el número de ramificaciones en un árbol de decisiones. Estos hiperparámetros son fundamentales para determinar la arquitectura del modelo, la tasa de aprendizaje y la complejidad del modelo. Estos se ajustan para optimizar el rendimiento y la generalización del modelo durante el entrenamiento. Esto implica experimentar con diferentes combinaciones de valores de hiperparámetros y evaluar cómo afectan el desempeño del modelo en datos de validación o prueba (¿Qué es el ajuste de hiperparámetros? - Explicación de los métodos de ajuste de hiperparámetros - AWS, s. f.).

#### 3.2.6.1. Número y tamaño de capas

La arquitectura de una red neuronal, determinada por el número de capas y neuronas, influye en la complejidad del modelo y su capacidad de aprendizaje. Cuanto mayor sea el número de neuronas y capas, más complejas serán las relaciones que puede aprender el modelo. Sin embargo, esto también aumenta el número de parámetros a aprender y el tiempo de entrenamiento. La elección de la arquitectura debe equilibrar la capacidad de aprendizaje y el costo computacional.

#### 3.2.6.2. Función de activación

La función de activación es esencial en una red neuronal, ya que determina la salida de una neurona en función de su entrada o conjunto de entradas. Cada capa de la red neuronal utiliza una función de activación para reconstruir o predecir resultados. Es importante destacar que se utiliza una función no lineal en la red

neuronal, ya que permite al modelo adaptarse a una amplia variedad de datos y patrones, proporcionando flexibilidad y capacidad de aprendizaje. Existen diferentes funciones como: Lineal, umbral, sigmoide, tangente hiperbólica, ReLu, SoftMax, entre otras (Freire & Silva, 2019).

### **3.2.6.3. Learning rate**

El learning rate, o tasa de aprendizaje, es un hiperparámetro crítico que determina la velocidad de ajuste de los parámetros de un modelo durante el proceso de optimización. Establecer el learning rate adecuado puede ser un desafío, ya que su elección depende de los datos y su interacción con otros hiperparámetros. Si el learning rate es demasiado grande, el proceso de optimización puede saltar de una región a otra sin lograr un aprendizaje efectivo. Por otro lado, si el learning rate es demasiado pequeño, el entrenamiento puede volverse muy lento y no alcanzar la convergencia deseada (Amat, 2021).

### **3.2.6.4. Optimizador**

En el aprendizaje profundo, los optimizadores se utilizan para ajustar los parámetros de un modelo. El propósito de un optimizador es modificar los pesos del modelo para maximizar una función de pérdida. La función de pérdida se utiliza como una forma de medir qué tan bien está desempeñándose el modelo. Es necesario utilizar un optimizador al entrenar un modelo de red neuronal (Vadapalli, 2022).

### **3.2.6.5. Epoch**

Hace referencia a una ronda completa de entrenamiento sobre todo el conjunto de datos. Pocos epoch no permiten un correcto entrenamiento del modelo y muchos epoch generan un sobre entrenamiento del modelo que ocasiona retardos sin justificación en el modelo de aprendizaje.

### **3.2.6.6. Batch Size**

Es el tamaño de los lotes en los cuales se dividen los conjuntos de datos para ser entrenados. Es decir, si tenemos un batch size de 100 significa que se entrenaran los primeros 100 datos y se entrena la red neuronal, luego toma los siguientes 100 y reentrena la red y así sucesivamente hasta terminar con el conjunto de datos.

### **3.2.6.7. Funciones de coste**

La función de coste (también conocida como función de pérdida) tiene como objetivo medir la discrepancia entre el valor real y el valor predicho por la red neuronal. En otras palabras, cuantifica cuánto se equivoca la red al hacer predicciones. Por lo general, la función de costo produce valores positivos, y cuanto más cercano a cero sea el valor de costo, mejores serán las predicciones de la red (es decir, menor error). Un valor de costo igual a cero indica que las predicciones coinciden exactamente con los valores reales. El optimizador se

encarga de optimizar esta función, es decir, encontrar el mejor valor para la función de coste (Amat, 2021).

### **3.2.7. Optimización Bayesiana**

En el ajuste de modelos de aprendizaje automático, es necesario establecer hiperparámetros que no se pueden aprender directamente de los datos. Dos estrategias comunes son la búsqueda en cuadrícula y la búsqueda aleatoria. Sin embargo, estas estrategias no aprovechan los resultados previos y pueden ser ineficientes en espacios de búsqueda amplios o con evaluaciones lentas. La optimización bayesiana de hiperparámetros ofrece una alternativa, utilizando un modelo probabilístico para dirigir la búsqueda hacia las regiones más prometedoras y reducir el número de combinaciones evaluadas. Esto es especialmente beneficioso en casos donde el espacio de búsqueda es amplio o la evaluación del modelo es lenta (Amat, 2020).

## **4. METODOLOGIA**

Con el fin de obtener un modelo preciso para la predicción de datos se siguieron los 7 pasos que se observan en la ilustración 2. Selección de datos, preprocesamiento de los datos, selección del modelo, entrenamiento del modelo, evaluación del modelo, mejoramiento del modelo, implementación del modelo. Este es un proceso iterativo lo cual implica que en cualquier paso se pueden realizar correcciones del proceso con el fin de mejorar los resultados finales.

Durante el desarrollo del proyecto, se realizaron múltiples iteraciones del proceso completo con el fin de mejorar los resultados obtenidos en la etapa de evaluación. Las métricas establecidas como criterios de éxito no se estaban cumpliendo inicialmente, lo que implicaba la necesidad de ajustar y afinar los diferentes parámetros involucrados en el proyecto. En cada iteración, se llevó a cabo un análisis exhaustivo de los resultados obtenidos, identificando las áreas en las que se requerían mejoras y determinando qué parámetros debían ajustarse para lograr un rendimiento óptimo. A través de esta retroalimentación constante, se adquirió un mayor conocimiento sobre cómo optimizar los distintos componentes del proyecto y cómo hacerlos trabajar en conjunto de manera más efectiva.

El proceso de ajuste de parámetros fue un ejercicio de aprendizaje continuo, en el cual se exploraron diferentes enfoques y estrategias con el objetivo de encontrar la combinación adecuada que condujera a un resultado satisfactorio. Se realizaron experimentos, se realizaron cambios graduales y se evaluaron los efectos en los resultados obtenidos. Cada iteración permitió obtener información valiosa sobre qué funcionaba y qué no, lo que condujo a una mejora gradual del proyecto en cada ciclo. El objetivo final de todas estas iteraciones y ajustes era lograr una solución óptima y satisfactoria para el proyecto. A medida que se avanzaba en el proceso, se buscaba optimizar tanto el rendimiento general del sistema como el cumplimiento de las métricas predefinidas. A través de la perseverancia y el enfoque en la



mejora continua, se buscaba alcanzar un resultado final que cumpliera con los estándares y expectativas establecidos.

Antes de comenzar el desarrollo de la metodología del proyecto, se llevó a cabo una exhaustiva revisión bibliográfica. El objetivo principal fue adquirir un profundo conocimiento de los fenómenos físicos detrás de las plantas fotovoltaicas y los fundamentos matemáticos del aprendizaje profundo. Esta revisión fue fundamental para comprender y abordar de manera efectiva el proyecto. Durante la revisión bibliográfica, se examinaron diversas fuentes como publicaciones científicas, investigaciones y estudios relevantes. Estos recursos proporcionaron información detallada sobre los principios físicos involucrados en la generación de energía fotovoltaica, así como los conceptos matemáticos y algoritmos utilizados en el aprendizaje profundo.

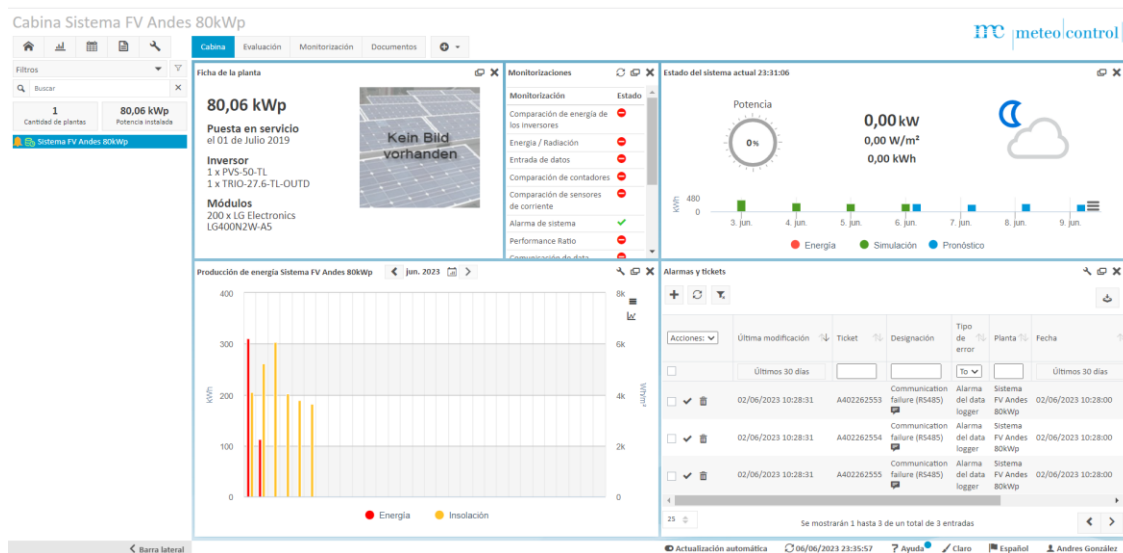
El objetivo principal fue interpretar y comprender cómo estos fenómenos y conceptos se aplican al proyecto en cuestión. La revisión bibliográfica permitió establecer una base sólida de conocimiento teórico, lo que fue esencial para diseñar una metodología efectiva. Al comprender los fenómenos físicos y los fundamentos matemáticos detrás del aprendizaje profundo, se pudo identificar las variables clave, los métodos de análisis adecuados y las mejores prácticas para abordar los desafíos del proyecto. Esta comprensión enriqueció el enfoque metodológico y proporcionó una base sólida para el desarrollo del proyecto.

#### **4.1. Selección de datos**

Para la primera etapa del diseño de la red neuronal, que es la selección de datos, es fundamental comprender a fondo el problema que deseamos abordar mediante el aprendizaje automático y revisar fuentes bibliográficas para obtener antecedentes e investigaciones previas sobre el tema. En este caso, el objetivo del proyecto es desarrollar un modelo de predicción basado en redes neuronales LSTM para pronosticar la radiación generada por una planta fotovoltaica, con el propósito de mejorar su eficiencia y gestión. El primer paso consistió en investigar y determinar qué datos se obtienen en la planta fotovoltaica que serán evaluados en nuestro modelo. Esto implica identificar las variables relevantes relacionadas con la radiación solar, se recopiló información sobre los diferentes sensores y dispositivos utilizados en la planta para medir y registrar estos datos.

Para garantizar la obtención de datos precisos y confiables, se tomaron las medidas necesarias para obtener la autorización requerida de la planta fotovoltaica del edificio Julio Mario Santo Domingo de la Universidad de los Andes, ubicada en Bogotá, Colombia. Esta autorización permitió acceder a los datos relevantes para nuestro proyecto. Una vez obtenida la autorización, se procedió a descargar los datos del software de recopilación utilizado en la planta, que en este caso es VCOM de meteocontrol. Este software es utilizado para el monitoreo y registro de datos en tiempo real de la planta fotovoltaica.

A través del software VCOM, se accedió a los datos recolectados por los sensores y dispositivos instalados en la planta, que registran variables como la radiación solar, la temperatura, la humedad y otros parámetros relevantes. Estos datos son fundamentales para el desarrollo de nuestro modelo de predicción basado en redes neuronales LSTM. Cabe destacar que se aplicaron los protocolos y procedimientos establecidos por la Universidad de los Andes y los responsables de la planta fotovoltaica para garantizar la confidencialidad y seguridad de los datos obtenidos.



*Ilustración 7 Software VCOM Universidad de los Andes.*

Además, se revisaron estudios científicos y técnicos que abordaban la recolección de datos en plantas fotovoltaicas similares, con el fin de comprender las mejores prácticas y metodologías empleadas en la industria. Esta investigación bibliográfica nos brindó una visión general de los tipos de datos disponibles y las técnicas utilizadas para su adquisición. También nos permitió identificar posibles desafíos y consideraciones importantes para el manejo y la calidad de los datos.

Durante la revisión bibliográfica exhaustiva realizada, se identificó un artículo científico relevante titulado "Very-Short-Term Power Prediction for PV Power Plants Using a Simple and Effective RCC-LSTM Model Based on Short Term Multivariate Historical Datasets". Este artículo fue fundamental para comprender los enfoques y metodologías utilizados en proyectos similares al nuestro. En dicho estudio, los investigadores desarrollaron un sistema de predicción de energía a muy corto plazo para plantas fotovoltaicas utilizando un modelo RCC-LSTM simple y efectivo. Para ello, utilizaron datos como la temperatura ( $^{\circ}\text{C}$ ), la humedad relativa (%), la radiación global horizontal ( $\omega/m^2 \times sr$ ) y la radiación horizontal difusa ( $\omega/m^2 \times sr$ ). Además, el estudio presentó resultados prometedores al demostrar la efectividad del modelo RCC-LSTM en la predicción de la generación de energía a corto plazo. Estos resultados respaldaron nuestra elección de utilizar una arquitectura LSTM en nuestro propio modelo de predicción basado en redes neuronales (Chen et al., 2020).

Después de analizar los datos proporcionados por el software VCOM y llevar a cabo la revisión bibliográfica correspondiente, se realizó una selección de los siguientes datos para el desarrollo del proyecto actual: Fecha (Timestamp), humedad relativa del aire, radiación meteorológica, temperatura de los módulos y temperatura ambiente. Estos datos se consideraron relevantes debido a su influencia en la generación de energía en una planta fotovoltaica. La fecha y hora permitirán tener una referencia temporal para el análisis y la predicción. La humedad relativa del aire puede afectar la eficiencia de los paneles solares y, por lo tanto, es un factor importante para tener en cuenta. La radiación meteorológica proporciona información sobre la cantidad de radiación solar disponible en un determinado momento. Las temperaturas de los módulos y del ambiente también son variables críticas, ya que afectan el rendimiento y la vida útil de los paneles solares. Utilizando estos datos seleccionados, el objetivo principal del proyecto será predecir la radiación global horizontal [ $W/m^2$ ] medida por los sensores. Esta predicción permitirá mejorar la eficiencia y la gestión de la planta fotovoltaica, al proporcionar información anticipada sobre la radiación solar y optimizar así la generación de energía.

Variable	Unidades
Fecha	dd/MM/aaaa HH:mm a.m./p.m.
Humedad relativa	%
Radiación meteorológica	$W/m^2$
Temperatura de los módulos	°C
Temperatura Ambiente	°C

*Tabla 1 Datos Seleccionados*

Los datos seleccionados se descargaron en formato CSV desde 1 de enero del 2020 a las 00:00 hasta el 12 de marzo de 2023 a las 17:30 con un intervalo de 15 minutos entre cada muestra tomada. Posteriormente se eliminaron datos aleatoriamente con el fin de tener una muestra más pequeña y evitar el sobre entrenamiento.

## 4.2. Preprocesamiento

El preprocesamiento de datos es importante porque garantiza la calidad y la adecuación de los datos para su uso en el entrenamiento de modelos de machine learning. Permite corregir problemas, normalizar los datos, codificar variables categóricas, seleccionar características relevantes y preparar los conjuntos de entrenamiento y prueba. Un preprocesamiento adecuado contribuye a mejorar la precisión y la eficacia del modelo predictivo basado en redes neuronales.

El preprocesamiento de los datos se llevó a cabo utilizando Excel, ya que los datos estaban en formato CSV, y esta herramienta facilita la manipulación de este tipo de archivos de manera sencilla. En la primera etapa del preprocesamiento, se realizó una limpieza de los datos, eliminando los valores nulos que se encontraron. Estos valores nulos pueden haber sido el resultado de posibles errores en la planta o de su desconexión,

y se tomaron medidas para evitar que estos valores afectaran el funcionamiento del programa en etapas posteriores.

Después de la limpieza de datos, se procedió a realizar una modificación en la columna que contenía el timestamp. Se separaron los valores en diferentes columnas correspondientes al día, mes, hora, minuto y AM/PM. En particular, se realizó una modificación en la columna del AM/PM para convertirla en una casilla numérica asignando el valor 1 para AM y 0 para PM. Esto fue necesario ya que las redes neuronales requieren valores numéricos como entrada. Además, la columna del año se eliminó, ya que no era relevante para el problema en cuestión.

Después de la modificación de los datos en Excel, se procedió a cargarlos en Python en forma de dataframe utilizando la biblioteca panda. Esto facilitó la manipulación y análisis de los datos. A continuación, se aplicó la normalización a los datos utilizando la función StandardScaler de la biblioteca sklearn. La normalización es una técnica común en el preprocesamiento de datos que permite escalar los valores a un rango específico, lo que ayuda a mejorar el desempeño de los modelos de machine learning. En este caso, se utilizó la función preprocessing.StandardScaler para normalizar los datos.

Una vez que los datos se cargaron como un dataframe, se procedió a dividirlos en dos conjuntos: un conjunto de entrenamiento y un conjunto de prueba. Esta división se realizó utilizando la biblioteca sklearn y la función train\_test\_split. El conjunto de entrenamiento se asignó al 80% de los datos, mientras que el conjunto de prueba se asignó al 20% restante. Es importante destacar que los datos originales se dividieron en dos conjuntos distintos: X, que contiene los datos utilizados para realizar las predicciones, y Y, que contiene los datos objetivo que se desea predecir. Después de la separación en entrenamiento y prueba, se obtuvieron cuatro conjuntos de datos denominados X\_train, X\_test, Y\_train y Y\_test. Estos conjuntos se utilizarán posteriormente para entrenar y evaluar el modelo de predicción.

```
15
16 dataset = pd.read_csv("datosf - copia.csv", delimiter = ";", encoding='ISO-8859-1')
17
18 X = dataset.iloc[:, 0:9].values
19 y = dataset.iloc[:, 9].values
20
21 from sklearn.model_selection import train_test_split
22 X_train, X_test, Y_train, Y_test = train_test_split(X,y, test_size = 0.2, random_state = 0)
23
24 from sklearn.preprocessing import StandardScaler
25 sX = StandardScaler()
26 X_train = sX.fit_transform(X_train)
27 X_test = sX.transform(X_test)
28
```

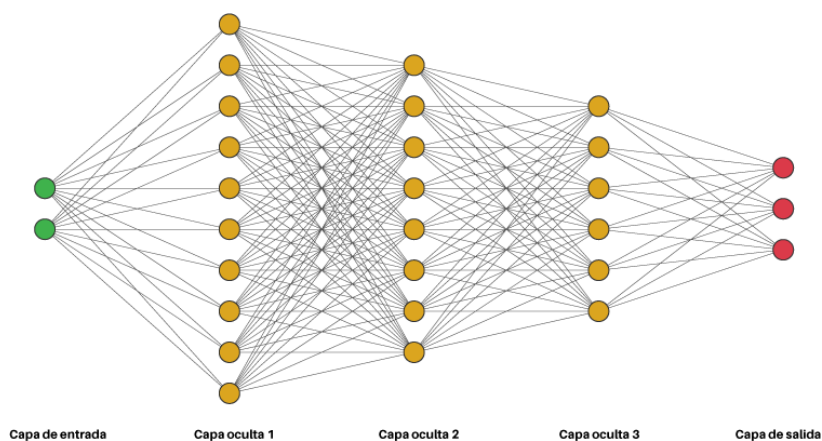
*Ilustración 8 Código preprocesamiento de datos.*

### 4.3. Selección del modelo

Una vez que los datos fueron seleccionados y preprocesados, se procedió a elegir el modelo ideal para el proyecto. Basándose en las diversas fuentes bibliográficas investigadas, se optó por utilizar una red neuronal de tipo LSTM. Una vez seleccionada esta arquitectura de red, se procedió al diseño de esta y a la conilustración de los hiperparámetros.

Para realizar estas pruebas y conilustraciónes, se utilizó un entorno de trabajo en mi computadora personal con las siguientes especificaciones: procesador Intel Core i7-8750H y 16 GB de memoria RAM a una frecuencia de 2600 MHz. Se empleó el IDE Spyder de Anaconda como entorno de desarrollo, utilizando el lenguaje de programación Python. Además, se contó con un entorno personalizado que incluía las librerías necesarias para el desarrollo del proyecto. En este entorno, se llevaron a cabo múltiples pruebas y ajustes de la arquitectura y los hiperparámetros de la red neuronal LSTM para obtener el mejor rendimiento y precisión en la predicción de la radiación generada por la planta fotovoltaica.

Durante las iteraciones iniciales, se probaron diferentes conilustraciónes de capas ocultas en la red neuronal, todas ellas basadas en el tipo LSTM. Se experimentó con distintas cantidades de neuronas en cada capa y se evaluó el desempeño de la red en cada caso. Tras analizar los resultados, se determinó que la arquitectura óptima consistía en utilizar 3 capas ocultas LSTM, la cual sería similar a la arquitectura mostrada en la ilustración 9. Sin embargo, aún quedaba pendiente la revisión y ajuste de los hiperparámetros necesarios para el correcto funcionamiento de la red neuronal. Los hiperparámetros son valores que no se aprenden del conjunto de datos, sino que deben ser establecidos previamente. Estos parámetros incluyen el learning rate (tasa de aprendizaje), el número de épocas de entrenamiento, el batch size (tamaño del lote), entre otros. La revisión de los hiperparámetros es fundamental para obtener un modelo óptimo y mejorar el rendimiento de la red neuronal en la predicción de la radiación generada por la planta fotovoltaica.



*Ilustración 9 Arquitectura red neuronal (¿Qué es una Red Neuronal? | Codificando Bits, s.f.).*

Para la selección de los hiperparámetros, función de activación, función de coste y optimizador, se tomaron en cuenta las recomendaciones y buenas prácticas establecidas en la literatura especializada. Estas elecciones se basaron en la efectividad demostrada en casos similares al proyecto en cuestión.

La función de activación utilizada fue la función "ReLU" (Rectified Linear Unit). Esta función aplica una transformación lineal a los valores de entrada, descartando los valores negativos y manteniendo los valores positivos sin cambios. Es una función no lineal que se ha vuelto muy popular en redes neuronales debido a su eficiencia computacional y su capacidad para manejar gradientes de manera más estable.

La función de coste seleccionada fue el Mean Square Error (MSE), que calcula la diferencia cuadrática promedio entre los valores predichos por el modelo y los valores reales. Esta función mide el nivel de discrepancia entre las predicciones y los objetivos, siendo más sensible a los errores más grandes. Es ampliamente utilizada en problemas de regresión.

Para el optimizador se eligió el algoritmo Adam, que combina las ventajas de los algoritmos de descenso de gradiente estocástico (SGD) y el método de adaptación del learning rate. Adam ajusta los pesos de la red neuronal de manera eficiente y adaptativa, permitiendo un aprendizaje más rápido y una convergencia efectiva hacia los mínimos locales.

Estas selecciones se basaron en la experiencia y el conocimiento acumulado en la comunidad de aprendizaje automático y se espera que proporcionen un buen rendimiento en el modelo de predicción de radiación fotovoltaica.

Para determinar los valores óptimos de los hiperparámetros restantes, como la cantidad de neuronas LSTM en cada capa, el valor de Dropout y el learning rate, se empleó el método de búsqueda bayesiana utilizando la librería Keras Tuner, en particular la función BayesianOptimization.

En el proceso de búsqueda bayesiana, se realizó una iteración exhaustiva de combinaciones de hiperparámetros. Para la cantidad de neuronas en cada una de las tres capas LSTM, se exploraron valores entre 1 y 1000 con un paso de 100. Para el Dropout, se consideraron valores entre 0.0 y 0.5 con un paso de 0.1. El learning rate se varió entre 0.0001 y 0.01. Además, se definió un valor fijo de 20 epochs para el entrenamiento de cada combinación de hiperparámetros. Esto indica la cantidad de veces que el modelo fue entrenado con los datos de entrenamiento. Asimismo, se estableció un límite máximo de 15 pruebas para la búsqueda bayesiana, lo que determina el número máximo de combinaciones evaluadas.

El enfoque de búsqueda bayesiana permitió explorar sistemáticamente diferentes combinaciones de hiperparámetros y encontrar los valores óptimos que maximizan el rendimiento del modelo de predicción de radiación fotovoltaica.

Es importante destacar que la combinación de hiperparámetros descrita anteriormente no fue la única probada durante el proceso de selección. Se realizaron pruebas adicionales con diferentes valores de iteraciones y máximos intentos. Se observó que utilizar un número muy bajo de iteraciones o intentos no permitía un entrenamiento adecuado del modelo, mientras que utilizar un número excesivamente alto conducía a un sobreentrenamiento. Por lo tanto, se encontró que la combinación de 20 epochs y un límite máximo de 15 pruebas proporcionó los mejores resultados en términos de rendimiento y capacidad de generalización del modelo. Esta combinación permitió un equilibrio adecuado entre el ajuste del modelo a los datos de entrenamiento y su capacidad para realizar predicciones precisas en nuevos datos.

Es fundamental realizar múltiples pruebas y ajustes de hiperparámetros para encontrar la configuración óptima que se adapte al problema específico y los datos disponibles. El proceso de selección de hiperparámetros es iterativo y requiere una evaluación cuidadosa del desempeño del modelo en cada combinación probada. Además, el tiempo de ejecución de las pruebas con diferentes combinaciones de hiperparámetros varió significativamente. Dependiendo de la complejidad de la arquitectura de la red neuronal, la cantidad de datos y el poder computacional disponible, el tiempo de ejecución puede variar desde aproximadamente una hora hasta varias semanas.

El ajuste de hiperparámetros es un proceso que requiere paciencia y dedicación, ya que cada combinación debe ser entrenada y evaluada para determinar su desempeño. Es posible que algunas configuraciones requieran un mayor número de iteraciones y, por lo tanto, más tiempo de ejecución para alcanzar la convergencia y obtener resultados óptimos.

Después de realizar la búsqueda bayesiana, se obtuvo una arquitectura óptima de red neuronal con las siguientes características:

- La primera capa LSTM tiene 301 neuronas.
- La segunda capa LSTM tiene 301 neuronas.
- La tercera capa LSTM tiene 201 neuronas.
- Se utilizó un valor de Dropout de 0.0, lo que significa que no se aplicó ninguna regularización de dropout en las capas LSTM.
- El learning rate utilizado fue 0.007772896641209602, que determina la tasa de ajuste de los pesos durante el proceso de optimización.

Estos valores fueron los seleccionados como resultado de la búsqueda bayesiana, con el objetivo de maximizar el rendimiento y la precisión del modelo de predicción. Cabe destacar que la elección de estos hiperparámetros puede depender de la naturaleza de los datos, la complejidad del problema y otros factores específicos del proyecto.

#### **4.4. Entrenamiento del modelo**

Durante el entrenamiento del modelo, se utilizaron las características que se encontraron en la etapa de selección del modelo. Estas características incluyen la función de activación ReLu, la función de coste MSE (Mean Square Error) y el optimizador Adam. La arquitectura de la red neuronal consta de tres capas LSTM, con la primera capa LSTM compuesta por 301 neuronas, la segunda capa LSTM por 301 neuronas y la tercera capa LSTM por 201 neuronas. No se aplicó ninguna regularización de dropout, ya que se utilizó un valor de Dropout de 0.0. El learning rate utilizado fue 0.007772896641209602, lo que determina la tasa de ajuste de los pesos durante el proceso de optimización. Para el entrenamiento definitivo del modelo, se realizaron 40 epochs, lo que significa que el conjunto de datos se pasó por el modelo de entrenamiento 40 veces.

Estas conilustraciónes fueron determinadas como óptimas durante el proceso de selección del modelo y se utilizaron para entrenar el modelo final con el objetivo de obtener resultados precisos y confiables en la predicción de la radiación generada por la planta fotovoltaica.

Finalmente, el modelo entrenado fue guardado como un archivo con la extensión .h5. Esto permite acceder al modelo en cualquier momento para realizar predicciones sin tener que repetir todo el proceso de entrenamiento desde cero. Al guardar el modelo, se guardan todos los pesos, conilustraciónes y parámetros necesarios para su funcionamiento. Esta capacidad de cargar el modelo previamente entrenado y realizar reentrenamiento con nuevos datos es extremadamente útil, ya que reduce significativamente el tiempo de procesamiento. Esto es especialmente beneficioso cuando se trabaja con grandes conjuntos de datos o cuando se requiere actualizaciones frecuentes del modelo. Al guardar el modelo como un archivo .h5, se asegura su portabilidad y la posibilidad de utilizarlo en diferentes entornos o plataformas. Esto facilita su implementación en aplicaciones prácticas, donde se pueden realizar predicciones en tiempo real utilizando el modelo previamente entrenado.

#### **4.5. Evaluación del modelo**

En el proceso de evaluación del modelo, se utilizó el conjunto de datos de prueba ( $Y_{test}$ ) que representaba el 20% de los datos totales. Utilizando el modelo entrenado previamente, se realizaron predicciones utilizando este conjunto de datos y se obtuvo una lista llamada  $y_{pred}$  que contiene las predicciones generadas por el modelo. Luego, se llevó a cabo la verificación de la viabilidad del modelo mediante la comparación entre las predicciones obtenidas ( $y_{pred}$ ) y los valores reales correspondientes del conjunto de prueba ( $Y_{test}$ ). Esto se hizo utilizando métricas de evaluación como el error cuadrático medio (MSE), graficas de comparación y correlación.

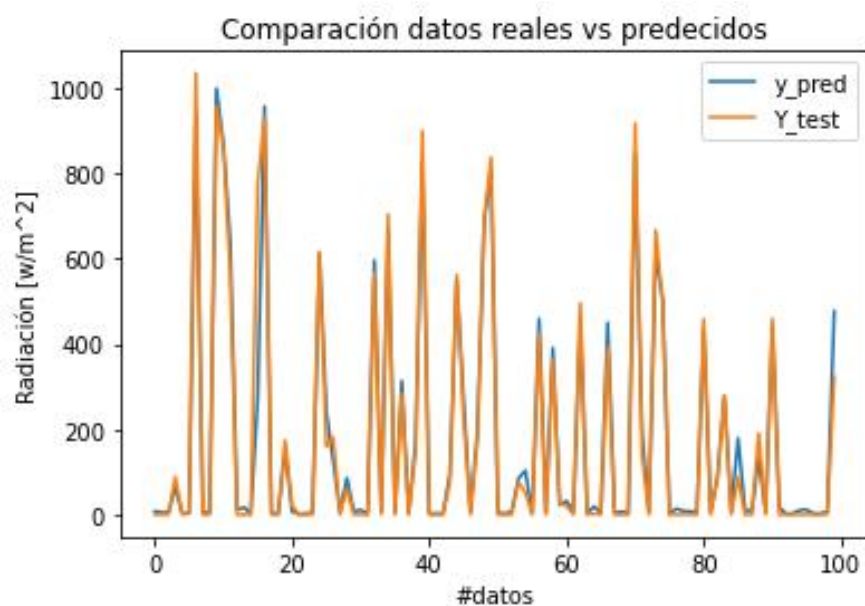
Estas métricas permiten medir qué tan bien se ajusta el modelo a los datos de prueba y proporcionan una medida objetiva de su rendimiento. Si las predicciones del modelo son consistentes con los valores reales y las métricas de evaluación son satisfactorias, esto indica que el modelo es capaz de realizar buenas predicciones y es viable para su implementación.



La evaluación del modelo es crucial para determinar su calidad y su capacidad para generalizar a nuevos datos. Si los resultados de la evaluación son satisfactorios, se puede considerar que el modelo es exitoso y puede ser utilizado en aplicaciones prácticas. En caso contrario, se pueden realizar ajustes en la arquitectura del modelo, los hiperparámetros o el preprocesamiento de datos para mejorar su desempeño.

En la ilustración 10, se presenta una gráfica que muestra una comparación entre los datos reales de prueba y los datos predichos por el modelo. Para esta visualización, se seleccionaron únicamente los primeros 100 datos con el propósito de mejorar la legibilidad de la gráfica y evitar la saturación visual. Esta representación gráfica permite evaluar visualmente qué tan cercanas están las predicciones del modelo en comparación con los valores reales. Al limitar la cantidad de datos a mostrar, se facilita la interpretación de la tendencia y los patrones de los resultados.

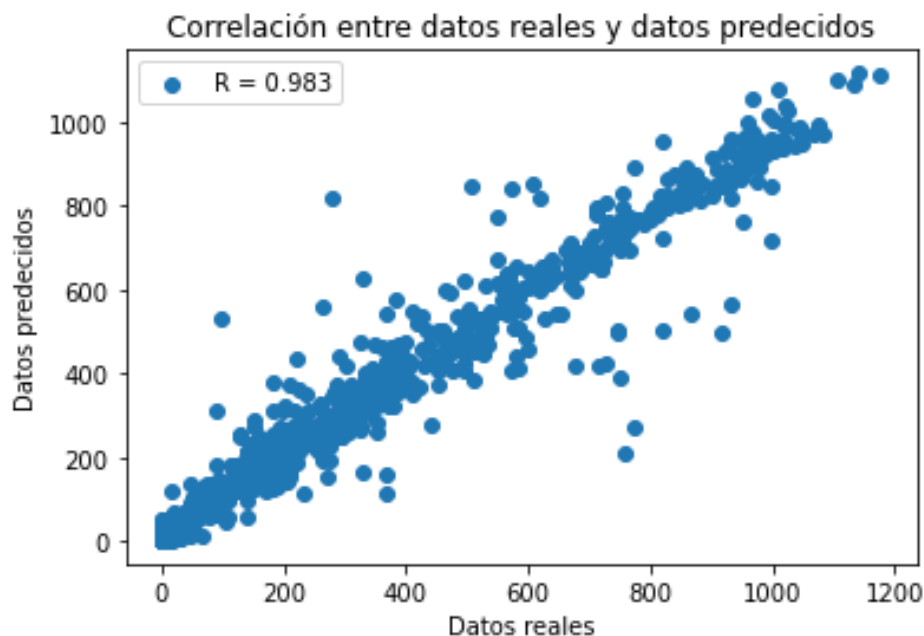
Al observar la gráfica, es posible identificar cómo se ajustan las predicciones del modelo a los valores reales. Si las líneas que representan los datos predichos se asemejan a las líneas correspondientes a los datos reales, esto indica que el modelo ha logrado capturar de manera precisa el comportamiento de los datos y realizar predicciones adecuadas. Esta visualización proporciona una perspectiva inicial sobre la calidad del modelo y su capacidad para capturar la variabilidad de los datos. Sin embargo, es importante tener en cuenta que esta ilustración se basa en una muestra reducida de datos y no refleja necesariamente el rendimiento general del modelo en todo el conjunto de prueba.



*Ilustración 10 Gráfico de comparación de datos reales vs predichos.*

En la ilustración 11, se muestra la gráfica de correlación que representa la relación entre los datos reales y los datos predichos. En este caso, se utilizó el conjunto completo de datos para realizar un análisis completo del sistema. El valor de correlación  $R$  obtenido fue de 0.983.

Un coeficiente de correlación cercano a 1, como en este caso, indica una fuerte correlación positiva entre los datos reales y los datos predichos. Esto significa que el modelo ha sido capaz de capturar y predecir de manera efectiva los patrones y tendencias presentes en los datos reales. En otras palabras, existe una alta concordancia entre los valores predichos por el modelo y los valores reales observados. Un valor de R de 0.983 es considerado muy bueno, ya que indica una correlación muy cercana entre los datos reales y los datos predichos. Esto sugiere que el modelo de predicción basado en redes neuronales LSTM ha logrado capturar de manera precisa la relación subyacente entre las variables y ha obtenido resultados prometedores en términos de precisión y rendimiento.



*Ilustración 11 Gráfico de correlación entre datos reales vs predichos.*

Si bien es cierto que en la gráfica de correlación se pueden observar algunos puntos que se alejan de la línea de tendencia, es importante tener en cuenta el contexto del problema. En el caso específico de la radiación durante la noche, es esperado que los valores predichos no sean exactamente cero, ya que el modelo puede capturar ciertos patrones o influencias que ocurren en esas horas, como la reflexión de la luz artificial o condiciones atmosféricas particulares. Sin embargo, a pesar de estas discrepancias, el resultado general del modelo sigue siendo bueno. La correlación alta obtenida ( $R = 0.983$ ) indica que, en la mayoría de los casos, los datos predichos se acercan de manera precisa a los datos reales. Esto sugiere que el modelo ha aprendido correctamente los patrones y tendencias subyacentes en los datos, y es capaz de realizar predicciones confiables.

Es importante considerar también que en cualquier modelo de predicción siempre existirá cierto grado de error, y es normal encontrar puntos dispersos en la gráfica de correlación. Lo crucial es evaluar el rendimiento general del modelo y su capacidad para realizar predicciones precisas en la mayoría de los casos.

## 4.6. Mejoramiento del modelo

Durante el desarrollo del proyecto, se realizaron múltiples iteraciones en cada etapa con el objetivo de mejorar el modelo en cada proceso. Estas iteraciones permitieron ajustar y optimizar diferentes aspectos del modelo, desde la selección de datos y preprocesamiento hasta la arquitectura de la red neuronal y los hiperparámetros. Aunque se obtuvo un resultado final satisfactorio, es importante destacar que siempre existe espacio para seguir mejorando el modelo en futuros procesos. A continuación, se presentan algunas posibles áreas de mejora:

- **Recopilación de datos:** En este proyecto, se utilizaron datos específicos de una planta fotovoltaica. En futuros procesos, se podría considerar la inclusión de más variables o fuentes de datos adicionales que puedan influir en la generación de radiación, como la posición geográfica, la orientación de los paneles solares, la sombra, entre otros.
- **Selección de características:** Durante el preprocesamiento de los datos, se realizó una selección de las características más relevantes para el modelo. En futuros procesos, se podrían explorar diferentes técnicas de selección o extracción de características para mejorar la representación de los datos y aumentar la precisión del modelo.
- **Optimización de la arquitectura:** Aunque se realizó un proceso exhaustivo de selección de la arquitectura de la red neuronal, se podrían explorar diferentes conilustraciónes de capas, neuronas y conexiones para evaluar si se obtiene una mejora en el rendimiento del modelo.
- **Aumento de datos:** En algunos casos, el rendimiento del modelo puede mejorar al aumentar la cantidad de datos disponibles para el entrenamiento. En futuros procesos, se podrían considerar técnicas de generación o recopilación de datos adicionales para fortalecer el modelo y mejorar su capacidad de generalización.
- **Regularización y optimización de hiperparámetros:** Aunque se realizaron ajustes de hiperparámetros mediante la búsqueda bayesiana, siempre se puede explorar más a fondo el espacio de hiperparámetros para encontrar combinaciones óptimas. Además, se podrían aplicar técnicas de regularización, como la inclusión de más capas de dropout o la implementación de técnicas de regularización L1 o L2, para mejorar la capacidad de generalización del modelo.

## 4.7. Implementación del modelo

En la implementación del modelo, se logró obtener un resultado final bastante satisfactorio, con buenas aproximaciones en la predicción de datos. Sin embargo, el objetivo final es poder realizar predicciones a futuro basadas en el modelo desarrollado. Para lograr esto, se continuó con la siguiente etapa del proyecto, que consiste en la predicción futura de la radiación generada por la planta fotovoltaica.

En la ilustración 12 se presenta el código correspondiente a la predicción de datos futuros. En este código, se tienen en cuenta dos parámetros importantes: "look\_back" y la cantidad de valores a predecir. El parámetro "look\_back" representa la cantidad de pasos anteriores que el programa utilizará para predecir el siguiente paso. En el caso de este proyecto, considerando que los datos de entrenamiento se tomaron cada 15 minutos, un día de predicciones sería equivalente a 96 pasos.

Con estos parámetros definidos, el código utiliza el modelo entrenado previamente para realizar las predicciones de datos futuros. Esto permite obtener una estimación de la radiación generada por la planta fotovoltaica en los próximos pasos de tiempo. Es importante destacar que esta etapa de predicción a futuro es de gran relevancia, ya que proporciona información valiosa para la toma de decisiones en la gestión y optimización de la planta fotovoltaica. La capacidad de prever la radiación futura permite ajustar y planificar de manera más eficiente el funcionamiento de la planta, maximizando así su rendimiento y producción energética.

```
146 look_back = 10
147 last_sequence = X_test[-look_back]
148 num_steps = 15
149 predicted_values = []
150 for _ in range(num_steps):
151     next_value = clf.predict(np.array([last_sequence]))
152     predicted_values.append(next_value)
153     last_sequence = np.concatenate([last_sequence[1:], next_value.flatten()])
154
155 print(predicted_values)
156
```

*Ilustración 12 Implementación de la predicción a futuro de la radiación*

## 5. RESULTADOS

Durante el desarrollo de este proyecto, se llevaron a cabo diversas etapas para lograr la predicción de la radiación solar generada por una planta fotovoltaica. El objetivo final era obtener un modelo preciso que permitiera estimar la radiación futura y, así, optimizar la gestión de la planta.

En primer lugar, se realizó el preprocesamiento de los datos. Estos se encontraban en formato CSV y se utilizó la herramienta Excel para su manipulación. Se eliminaron las columnas con valores faltantes, ya que podrían afectar el funcionamiento del modelo posteriormente. Además, se separaron los datos del timestamp en día, mes, hora, minuto y am/pm, y se convirtió la columna de am/pm en una variable numérica para su uso en la red neuronal.

Posteriormente, se procedió a seleccionar el modelo más adecuado para el proyecto. Después de investigar diversas fuentes bibliográficas, se optó por utilizar una red neuronal de tipo LSTM (Long Short-Term Memory). Esta arquitectura es especialmente eficaz para modelar secuencias temporales, lo que la hace adecuada para la predicción de la radiación solar.

Se realizaron iteraciones para ajustar los hiperparámetros del modelo. Se probaron diferentes tamaños de capas ocultas y cantidad de neuronas por capa. Se utilizó la librería Keras Tuner y la función de búsqueda bayesiana para encontrar la combinación óptima de hiperparámetros. Al final, se determinó que la mejor arquitectura consistía en tres capas LSTM con 301, 301 y 201 neuronas respectivamente.

Una vez seleccionado el modelo, se procedió al entrenamiento del mismo. Se utilizaron los datos preprocesados y se conilustraciónron los hiperparámetros seleccionados previamente. La función de activación utilizada fue ReLU (Rectified Linear Unit), la función de coste fue el Mean Square Error (MSE) y se empleó el optimizador Adam. Se entrenó el modelo durante 40 epochs para obtener resultados óptimos.

Posteriormente, se evaluó el modelo utilizando el conjunto de datos de prueba. Se compararon los datos reales con las predicciones del modelo y se calculó el coeficiente de correlación (R), que resultó ser de 0.983. Este valor indica una fuerte correlación entre los datos reales y los datos predichos, lo que demuestra la eficacia del modelo en la estimación de la radiación solar.

Además de la evaluación, se realizó la predicción de datos futuros utilizando el modelo entrenado. Se conilustraciónron los parámetros "look\_back", que determina la cantidad de pasos anteriores utilizados para predecir el siguiente paso, y la cantidad de valores a predecir. Esto permitió obtener estimaciones de la radiación solar generada por la planta fotovoltaica en periodos futuros.

En resumen, el proyecto logró desarrollar un modelo de red neuronal LSTM que proporciona buenas aproximaciones en la predicción de la radiación solar. Sin embargo, siempre existe margen para la mejora continua. Se pueden explorar diferentes arquitecturas de red neuronal, ajustar hiperparámetros adicionales y considerar la incorporación de otras variables relevantes para aumentar la precisión de las predicciones.

## **6. DISCUSION**

La predicción de la radiación solar generada por una planta fotovoltaica es de vital importancia para optimizar su rendimiento y maximizar la producción de energía limpia. En este proyecto, se desarrolló un modelo de red neuronal LSTM que demostró ser eficaz en la estimación de la radiación solar futura.

Tras el entrenamiento y evaluación del modelo, se obtuvo un coeficiente de correlación (R) de 0.983 entre los datos reales y las predicciones del modelo. Este valor indica una fuerte correlación, lo que significa que el modelo es capaz de capturar las tendencias y patrones subyacentes en los datos de radiación solar.

La ilustración 11 muestra la gráfica de correlación, donde se puede observar la cercanía de los datos predichos a la línea de regresión. Sin embargo, se identificaron algunas desviaciones en los datos predichos durante las horas nocturnas. Estas discrepancias pueden explicarse por el hecho de que, en el modelo, la radiación no se reduce a cero durante la noche, a diferencia de la realidad donde la radiación solar es mínima o nula. Esto podría ser un área de mejora para futuras investigaciones, donde se podría incorporar una consideración más precisa del comportamiento nocturno de la radiación solar.

A pesar de esta pequeña discrepancia, el modelo demostró un buen desempeño en la predicción de la radiación solar, lo que brinda oportunidades para optimizar la gestión y el rendimiento de las plantas fotovoltaicas. Con las estimaciones de radiación futura proporcionadas por el modelo, los operadores de las plantas pueden planificar de manera más efectiva la producción de energía, ajustar el seguimiento solar y realizar mantenimientos preventivos.

Es importante destacar que este proyecto representa una base sólida para futuras investigaciones y mejoras. Se podrían explorar diferentes arquitecturas de red neuronal, ajustar los hiperparámetros de manera más exhaustiva e incorporar variables adicionales, como datos meteorológicos o información geoespacial, para aumentar aún más la precisión de las predicciones.

## 7. CONCLUSIONES

En este proyecto, se llevó a cabo el desarrollo e implementación de un modelo de red neuronal LSTM para la predicción de la radiación solar en una planta fotovoltaica. A partir del análisis y procesamiento de los datos históricos de radiación solar, se logró obtener un modelo capaz de realizar estimaciones precisas de la radiación solar futura. Las principales conclusiones obtenidas son las siguientes:

- El modelo de red neuronal LSTM demostró ser efectivo en la predicción de la radiación solar. El coeficiente de correlación ( $R$ ) obtenido entre los datos reales y las predicciones del modelo fue de 0.983, lo cual indica una fuerte correlación y un buen ajuste del modelo a los datos.
- La arquitectura del modelo, con tres capas ocultas LSTM y una combinación adecuada de hiperparámetros, resultó en un rendimiento satisfactorio. Sin embargo, existen posibilidades de mejorar aún más el modelo mediante la exploración de diferentes arquitecturas y ajustes de hiperparámetros.
- Se identificó una ligera discrepancia entre los datos predichos y los valores reales durante las horas nocturnas. Esto podría atribuirse a la falta de una consideración precisa del comportamiento nocturno de la radiación solar en el modelo.
- El modelo desarrollado proporciona una herramienta valiosa para la gestión y optimización de plantas fotovoltaicas. Las estimaciones de radiación solar futura

permiten a los operadores planificar y ajustar de manera más efectiva la producción de energía, maximizando así el rendimiento de la planta.

- El proyecto sienta las bases para futuras investigaciones y mejoras. Se podrían explorar enfoques más avanzados, como la incorporación de variables adicionales o el uso de técnicas de aprendizaje profundo más complejas, para mejorar aún más la precisión de las predicciones.

## 8. RECOMENDACIONES

Basado en los resultados y conclusiones obtenidas en este proyecto de predicción de radiación solar en plantas fotovoltaicas, se pueden hacer las siguientes recomendaciones:

- Explorar nuevas combinaciones de hiperparámetros: Aunque se realizó una búsqueda exhaustiva de hiperparámetros utilizando el método de búsqueda bayesiana, aún existe la posibilidad de encontrar combinaciones alternativas que mejoren el rendimiento del modelo. Se recomienda continuar experimentando con diferentes configuraciones de capas LSTM, número de neuronas, valores de dropout y learning rate para determinar si se puede lograr una mayor precisión en las predicciones.
- Estudiar los datos nocturnos y aplicar técnicas de estandarización: Dado que se observó una discrepancia entre las predicciones y los valores reales durante las horas nocturnas, se sugiere realizar un estudio más detallado de estos datos y explorar métodos de estandarización alternativos. Se podría considerar la aplicación de un offset que permita ajustar las predicciones en los momentos en los que la radiación solar es igual a cero, de modo que se eviten problemas de sobreestimación en estas situaciones.
- Realizar una búsqueda óptima del valor de look\_back: En la implementación actual, se utilizó un valor de look\_back que representa la cantidad de pasos anteriores utilizados para predecir el siguiente paso. Sería recomendable realizar una búsqueda exhaustiva para determinar el valor óptimo de look\_back que produzca las mejores predicciones a futuro. Esto implica probar diferentes valores y evaluar su impacto en la precisión de las predicciones, teniendo en cuenta la frecuencia de muestreo de los datos originales.

Estas recomendaciones ayudarán a mejorar aún más la precisión y robustez del modelo de predicción de radiación solar en plantas fotovoltaicas. Al realizar ajustes en los hiperparámetros, estudiar y abordar los desafíos asociados con los datos nocturnos y buscar el valor óptimo de look\_back, se podrán obtener predicciones más precisas y confiables, lo que contribuirá a una mejor gestión y optimización de las plantas fotovoltaicas.

Además de las recomendaciones mencionadas anteriormente, una posible continuación del proyecto sería utilizar los resultados obtenidos en la predicción de radiación solar para llevar

a cabo la predicción de la producción energética de la planta fotovoltaica. La radiación solar es un factor determinante en la generación de energía en una planta fotovoltaica, pero no es el único. Otros factores como la eficiencia de los paneles solares, la temperatura ambiente y la orientación de los paneles también influyen en la producción de energía.

Para realizar la predicción de la producción energética, se podrían combinar los datos de radiación solar predichos con información adicional sobre los factores mencionados anteriormente. Esto permitiría obtener una estimación más precisa de la cantidad de energía que la planta será capaz de generar en un período de tiempo determinado. Sería necesario recopilar datos históricos de producción energética de la planta y combinarlos con los datos predichos de radiación solar para entrenar un nuevo modelo de predicción. Este modelo podría utilizar técnicas de aprendizaje supervisado, como regresión lineal o redes neuronales, para estimar la producción energética en función de los diferentes factores ambientales.

La predicción de la producción energética de la planta fotovoltaica sería de gran utilidad para la planificación y la toma de decisiones en la gestión de la planta. Permite optimizar la operación de la planta, programar el mantenimiento de los paneles solares y anticiparse a posibles fluctuaciones en la generación de energía.

## REFERENCIAS

Amat, J. (2020). *Optimización bayesiana de hiperparámetros*. [https://www.cienciadedatos.net/documentos/62\\_optimizacion\\_bayesiana\\_hiperparametros](https://www.cienciadedatos.net/documentos/62_optimizacion_bayesiana_hiperparametros)

Amat, J. (2021). *Redes neuronales con Python*. <https://www.cienciadedatos.net/documentos/py35-redes-neuronales-python.html>

Bagnato. (2019). 7 Pasos Aprendizaje Automático. *Aprende Machine Learning*. <https://www.aprendemachinelearning.com/7-pasos-machine-learning-construir-maquina/>

Charniak, E. (2019). *Introduction to Deep Learning*. MIT Press.

Chen, B., Lin, P., Lai, Y., Cheng, S., Chen, Z., & Wu, L. (2020). Very-Short-Term Power Prediction for PV Power Plants Using a Simple and Effective RCC-LSTM Model Based on Short Term Multivariate Historical Datasets. *Electronics*, 9(2), 289. <https://doi.org/10.3390/electronics9020289>

Codificando Bits. (2019, 20 julio). *¿Qué es una red LSTM?* [Vídeo]. YouTube. <https://www.youtube.com/watch?v=1BubAvTVBYs>

Freire, E., & Silva, S. (2019). Redes neuronales - Bootcamp AI - Medium. *Medium*. <https://bootcampai.medium.com/redes-neuronales-13349dd1a5bb>

Ghoshal, A. (2023). Types of Neural Networks. *EDUCBA*. <https://www.educba.com/types-of-neural-networks/>

Iberdrola. (s. f.). *¿Cómo funcionan las plantas fotovoltaicas?* <https://www.iberdrola.com/sostenibilidad/que-es-energia-fotovoltaica>



IEC. (2017). IEC - International Electrotechnical Commission. <https://webstore.iec.ch/publication/65561>

La Marca, E., Arriojas, M., & Costa, F. (2017). REPRESAS HIDROELÉCTRICAS EN LOS ANDES VENEZOLANOS: PROBLEMÁTICA AMBIENTAL, CRISIS ENERGÉTICA Y ENERGÍAS ALTERNATIVAS. *Universidad de Oriente*. [https://www.researchgate.net/profile/Enrique-Marca/publication/330281753\\_REPRESAS\\_HIDROELECTRICAS\\_EN\\_LOS\\_ANDES\\_VENEZOLANOS\\_PROBLEMATICA\\_AMBIENTAL\\_CRISIS\\_ENERGETICA\\_Y\\_ENERGIAS\\_ALTERNATIVAS/links/5c372ad6458515a4c71b1e39/REPRESAS-HIDROELECTRICAS-EN-LOS-ANDES-VENEZOLANOS-PROBLEMATICA-AMBIENTAL-CRISIS-ENERGETICA-Y-ENERGIAS-ALTERNATIVAS.pdf](https://www.researchgate.net/profile/Enrique-Marca/publication/330281753_REPRESAS_HIDROELECTRICAS_EN_LOS_ANDES_VENEZOLANOS_PROBLEMATICA_AMBIENTAL_CRISIS_ENERGETICA_Y_ENERGIAS_ALTERNATIVAS/links/5c372ad6458515a4c71b1e39/REPRESAS-HIDROELECTRICAS-EN-LOS-ANDES-VENEZOLANOS-PROBLEMATICA-AMBIENTAL-CRISIS-ENERGETICA-Y-ENERGIAS-ALTERNATIVAS.pdf)

*New Advances in Machine Learning*. (2010). Google Books. <https://books.google.es/books?hl=es&lr=&id=XAqhDwAAQBAJ&oi=fnd&pg=PA19&dq=machine+learning+types+of+classifiers&ots=r2NndXxiMk&sig=tBaCn99kF4PwvZv91Qb45pSQUec#v=onepage&q=machine%20learning%20types%20of%20classifiers&f=false>

*¿Qué es Deep Learning?* | IBM. (s. f.). <https://www.ibm.com/es-es/topics/deep-learning>

*¿Qué es el ajuste de hiperparámetros?* - Explicación de los métodos de ajuste de hiperparámetros - AWS. (s. f.). Amazon Web Services, Inc. <https://aws.amazon.com/es/what-is/hyperparameter-tuning/>

*¿Qué es el machine learning?* - Explicación sobre el machine learning empresarial - AWS. (s. f.). Amazon Web Services, Inc. <https://aws.amazon.com/es/what-is/machine-learning/>

*¿Qué es la energía solar? Tipos y principales ventajas* | Repsol. (2023, 26 abril). REPSOL. <https://www.repsol.com/es/energia-futuro/transicion-energetica/energia-solar/index.cshtml>

*¿Qué es una Red Neuronal?* | Codificando Bits. (s. f.). Codificando Bits. <https://www.codificandobits.com/blog/que-es-una-red-neuronal/>

*¿Qué es una red neuronal?* - Explicación de las redes neuronales artificiales - AWS. (s. f.-a). Amazon Web Services, Inc. <https://aws.amazon.com/es/what-is/neural-network/>

*¿Qué es una red neuronal?* - Explicación de las redes neuronales artificiales - AWS. (s. f.-b). Amazon Web Services, Inc. <https://aws.amazon.com/es/what-is/neural-network/>

*¿Qué son las redes neuronales recurrentes?* | IBM. (s. f.). [https://www.ibm.com/mx-es/topics/recurrent-neural-networks#:~:text=Una%20red%20neuronal%20recurrente%20\(RNN,voz%20y%20subt%C3%ADtulos%20de%20im%C3%A1genes.](https://www.ibm.com/mx-es/topics/recurrent-neural-networks#:~:text=Una%20red%20neuronal%20recurrente%20(RNN,voz%20y%20subt%C3%ADtulos%20de%20im%C3%A1genes.)

*RADIACIÓN SOLAR* - IDEAM. (s. f.). <http://www.ideam.gov.co/web/tiempo-y-clima/radiacion-solar-ultravioleta>

*Redes neuronales recurrentes* – KW Foundation. (2021, 13 julio). <https://kwfoundation.org/blog/2021/07/13/redes-neuronales-recurrentes/>

Ruiz Molano, K. V., & García Martínez, A. F. (s. f.). MAQUINA DE VAPOR - PDFCOFFEE.COM. *pdfcoffee.com*. <https://pdfcoffee.com/maquina-de-vapor-5-pdf-free.html>

United Nations. (s. f.). *¿Qué son las energías renovables? | Naciones Unidas*. <https://www.un.org/es/climatechange/what-is-renewable-energy>

*Urban Energy Transition - 1st Edition*. (2008, 3 marzo). <https://www.elsevier.com/books/urban-energy-transition/droege/978-0-08-045341-5>

Vadapalli, P. (2022). Types of Optimizers in Deep Learning Every AI Engineer Should Know. *upGrad blog*. <https://www.upgrad.com/blog/types-of-optimizers-in-deep-learning/#:~:text=In%20deep%20learning%2C%20optimizers%20are,training%20a%20neural%20network%20model>.

Yadav, H. (2023, 31 mayo). Dropout in Neural Networks - Towards Data Science. *Medium*. <https://towardsdatascience.com/dropout-in-neural-networks-47a162d621d9>

Yahia, E. M. (2019). *Postharvest Technology of Perishable Horticultural Commodities*. <https://www.sciencedirect.com/book/9780128132760/postharvest-technology-of-perishable-horticultural-commodities>

## ANEXOS

A continuación, se encuentra el código fuente utilizado, adicionalmente se encuentra un repositorio de GitHub (<https://github.com/alejo2612/An-lisis-Y-Prediccion-De-Radiacion-En-Sistemas-Fotovoltaicos-Haciendo-Uso-De-Machine-Learning.git>) donde se encuentra el código fuente, el archivo CSV y archivo.h5 ya entrenado.

```
1  """
2  @author: Alejandro Herrera Jiménez
3  Proyecto: Análisis Y Predicción De Radiación En Sistemas Fotovoltaicos Haciendo Uso De Machine Learning
4  """
5
6  #Librerías necesarias para importar
7  import tensorflow as tf
8  from keras_tuner.tuners import BayesianOptimization
9  import pandas as pd
10 from keras.models import Sequential
11 from keras.layers import Dense, LSTM
12 from keras.layers import Dropout
13 import matplotlib.pyplot as plt
14 import numpy as np
15 import scipy.stats as stats
16 from sklearn.model_selection import train_test_split
17 from sklearn.preprocessing import StandardScaler
18
19 #Lectura del dataset en dataframe
20
21 dataset = pd.read_csv("dataset.csv", delimiter = ";", encoding='ISO-8859-1')
22
23 #Datos usados para predecir (timestamp, humedad relativa, radiación metereologica, temperatura de los modulos, temperatur
24 X = dataset.iloc[:, 0:9].values
25 #Datos que se predicen (Radiación global horizontal)
26 y = dataset.iloc[:, 9].values
27
28 #Separación entre datos de entrenamiento y datos de preuba
29 X_train, X_test, Y_train, Y_test = train_test_split(X,y, test_size = 0.2, random_state = 0)
30
31 #Preprocesamiento de los datos
32 sX = StandardScaler()
33 X_train = sX.fit_transform(X_train)
34 X_test = sX.transform(X_test)
35
36 #Definición de la función para la búsqueda Bayesiana
37 def model_builder(hp):
38     clf = Sequential()
```

```

39
40 #Definición de las iteraciones de la cantidad de neuronas por capa oculta
41 hp_LSTM_1 = hp.Int('LSTM_1', min_value=1, max_value=1000, step=100)
42
43 hp_LSTM_2 = hp.Int('LSTM_2', min_value=1, max_value=1000, step=100)
44
45 hp_LSTM_3 = hp.Int('LSTM_3', min_value=1, max_value=1000, step=100)
46
47 #Iteración del Dropout
48 hp_Dropout = hp.Float('Dropout_1', min_value=0.0, max_value=0.5, step=0.1)
49
50 #Iteración de learning rate
51 hp_learning_rate = hp.Float('learning_rate', min_value=1e-4, max_value=1e-2)
52
53 #Preentrenamiento de prueba para la búsqueda Bayesiana
54
55 clf.add(LSTM(units=hp_LSTM_1, activation='relu', input_shape = (X_train.shape[1], 1), return_sequences=True))
56
57 clf.add(LSTM(units=hp_LSTM_2, activation='relu', return_sequences=True))
58
59 clf.add(LSTM(units=hp_LSTM_3, activation='relu'))
60 clf.add(Dropout(hp_Dropout))
61 clf.add(Dense(units = 1))
62
63 # Compilación del modelo con Adam y MSE
64 clf.compile(optimizer=tf.keras.optimizers.Adam(learning_rate=hp_learning_rate),
65             loss=tf.keras.losses.MeanSquaredError(),
66             metrics=['MSE'])
67
68 return clf
69
70 #Se establecen los parametros de la búsqueda Bayesiana
71 tuner = BayesianOptimization(model_builder,
72                             objective='val_loss',
73                             max_trials=15)
74
75 tuner.search(X_train, Y_train, epochs=20, validation_split=0.2)
76

```

```

76
77 best_hps = tuner.get_best_hyperparameters(num_trials=1)[0]
78
79 #Una vez seleccionados los mejores Hiperparámetros se realiza el entrenamiento final del modelo
80 clf = tuner.hypermodel.build(best_hps)
81 #En este caso tiene epoch de 40 pero se puede variar
82 history = clf.fit(X_train, Y_train, epochs=40)
83
84 #Se guarda el modelo como un archivo .h5
85 clf.save('modelo_entrenado')
86
87 #Se predicen los datos con el modelo entrenado
88 y_pred = clf.predict(X_test)
89
90 #Se observan los resultados
91 # En esta sección se observan e imprimen los resultados de la Búsqueda Bayesiana
92 learning_rate = best_hps.get('learning_rate')
93 LSTM1 = best_hps.get('LSTM_1')
94 LSTM2 = best_hps.get('LSTM_2')
95 LSTM3 = best_hps.get('LSTM_3')
96 Dropout_1 = best_hps.get('Dropout_1')
97 print(learning_rate)
98 print(LSTM1)
99 print(LSTM2)
100 print(LSTM3)
101 print(Dropout_1)
102
103

```

```

104 #Graficación de resultados
105
106 #Grafica de error relativo (no es muy diciente para este caso)
107 error = []
108 for i in range(len(y_pred)):
109     val1= (y_pred[i])
110     val2= (Y_test[i])
111     if Y_test[i] != 0:
112         a = ((val1-val2)/val2)*100
113         error.append(a)
114
115
116 plt.plot(error)
117 plt.xlabel('#datos')
118 plt.ylabel('% error')
119 plt.title('Error relativo')
120 plt.show()
121
122 #Grafica de compración entre los datos predecidos y los datos reales
123 primeros_p = []
124 primeros_t = []
125 e = 0
126 #Se recomienda usar solo 100 datos para poder observar de mejor manera la grafica
127 while e<100:
128     primeros_p.append(y_pred[e])
129     primeros_t.append(Y_test[e])
130     e +=1
131
132 plt.plot(primeros_p, label="y_pred")
133 plt.plot(primeros_t, label="Y_test")
134 plt.xlabel('#datos')
135 plt.ylabel('Radiación [w/m^2]')
136 plt.title('Comparación datos reales vs predecidos')
137 plt.legend()
138 plt.show()
139

```

```

140
141 #Grafica de correlación (es la mejor manera de observar que tan bien quedó el modelo)
142 plt.scatter(Y_test, y_pred)
143 plt.xlabel('Datos reales')
144 plt.ylabel('Datos predecidos')
145 plt.title('Correlación entre datos reales y datos predecidos')
146 r, _ = stats.pearsonr(Y_test, y_pred)
147 plt.legend(["R = 0.983"])
148 plt.show()
149
150
151 # En esta parte se hace la predicción de los datos futuros
152 #Tener en cuenta que cada paso temporal son 15 minutos
153 look_back = 10 # Cantidad de pasos anteriores para predecir el siguiente
154 last_sequence = X_test[-look_back]
155 num_steps = 15 #Cantidad de pasos temporales siguientes
156 predicted_values = []
157 for _ in range(num_steps):
158     next_value = clf.predict(np.array([last_sequence]))
159     predicted_values.append(next_value)
160     last_sequence = np.concatenate([last_sequence[1:], next_value.flatten()])
161
162 print(predicted_values)
163

```

```

140
141 #Grafica de correlación (es la mejor manera de observar que tan bien quedó el modelo)
142 plt.scatter(Y_test, y_pred)
143 plt.xlabel('Datos reales')
144 plt.ylabel('Datos predecidos')
145 plt.title('Correlación entre datos reales y datos predecidos')
146 r, _ = stats.pearsonr(Y_test, y_pred)
147 plt.legend(["R = 0.983"])
148 plt.show()
149
150
151 # En esta parte se hace la predicción de los datos futuros
152 #Tener en cuenta que cada paso temporal son 15 minutos
153 look_back = 10 # Cantidad de pasos anteriores para predecir el siguiente
154 last_sequence = X_test[-look_back]
155 num_steps = 15 #Cantidad de pasos temporales siguientes
156 predicted_values = []
157 for _ in range(num_steps):
158     next_value = clf.predict(np.array([last_sequence]))
159     predicted_values.append(next_value)
160     last_sequence = np.concatenate([last_sequence[1:], next_value.flatten()])
161
162 print(predicted_values)
163

```

