Estudio del porcentaje de error en el pronóstico multitemporal de la irradiancia basado en RNA recurrente tipo LSTM

Daniel Sebastián Rosero Usamá
Andrés Felipe Zambrano Benavides

Programa de Ingeniería Electrónica
Facultad de Ingeniería
Universidad CESMAG
2024

Estudio del porcentaje de error en el pronóstico multitemporal de la irradiancia basado en RNA recurrente tipo LSTM

Daniel Sebastián Rosero Usamá
Andrés Felipe Zambrano Benavides

Proyecto de trabajo de grado en la modalidad de Estancia en Línea presentado al Comité Curricular del Programa de Ingeniería Electrónica

Asesor
John Evert Barco Jiménez

Programa de Ingeniería Electrónica
Facultad de Ingeniería
Universidad CESMAG
2024

Contenido

Lista de Tablas

Lista de Figuras

Introducción

Las prácticas sostenibles y conscientes del medio ambiente están en constante desarrollo, la Universidad CESMAG a la vanguardia de la innovación se hace parte del crecimiento con un proyecto para mejorar su calidad energética. Esta investigación busca establecer una base sólida con el desarrollo de un algoritmo basado en red neuronal de Memoria de corto largo plazo denominada LSTM por sus siglas en inglés, que llevará a cabo un pronóstico de irradiancia en la Universidad CESMAG, no solo para optimizar recursos energéticos, sino respondiendo a un llamado para adoptar tecnología apoyada con Deep Learning encaminada en contribuir a una sociedad más sostenible.

El objetivo primordial de este proyecto es contribuir a la Facultad de Ingeniería de la Universidad CESMAG por medio de la implementación de un sistema de pronóstico de irradiancia. Lo anterior será posible mediante el empleo de una red neuronal LSTM Multitemporal, debido a que le posibilita capturar patrones y tendencias, para ello, se diseñó un modelo basado en esta técnica durante la primera iteración. Técnica que se aplicará a la irradiancia registrada por la estación Davis Vantage Pro-2.0, con el propósito de proporcionar una estimación precisa y anticipada.

Este proyecto es relevante puesto que su enfoque se dirige de manera integral hacia la generación y uso de energías limpias, lo cual concuerda con los valores y objetivos de la comunidad CESMAG. La implementación de la red neuronal LSTM en la estimación de la irradiancia es complemento importante para la toma de decisiones sustentables en cuanto al uso de recursos energéticos, también busca sentar la base para una gestión eficiente y sostenible de la energía en la universidad. Esto no solo resultará en proyectos futuros afines en ahorros económicos significativos a largo plazo, sino que también contribuirá a la disminución de la huella ambiental de la institución.

La estructura del presente proyecto de grado se compone de diversas etapas fundamentales. En primer lugar, la revisión bibliográfica de las redes neuronales LSTM y su aplicación en la estimación de irradiancia. Posteriormente, recopilación y preparación de datos de irradiancia provenientes de la estación Davis Vantage Pro-2.0. La etapa central consistirá en el diseño, entrenamiento e implementación de la red neuronal LSTM usando Python. Finalmente se hará la respectiva evaluación del desempeño del algoritmo propuesto usando las métricas como error absoluto medio (MAE) y coeficiente de determinación R^2.

1. Problema de Investigación

1.1 Objeto o tema de Investigación

Error en el Pronóstico Multitemporal de la Irradiancia Basado en RNA Recurrente Tipo LSTM.

1.2 Línea de Investigación

Sistemas de Automatización y Control. “La línea de sistemas de automatización y control de la Universidad CESMAG desarrolla procesos investigativos orientados al modelamiento, simulación, diseño, desarrollo y evaluación de algoritmos de control, sistemas de control, sistemas inteligentes, control de procesos industriales, sistemas embebidos, acondicionamiento y procesamiento de señales, robótica, domótica e inteligencia artificial” [1].

1.3 Sublínea de Investigación

Inteligencia Artificial. “La inteligencia artificial es una respuesta al deseo de aproximar el comportamiento humano y el pensamiento racional a diversos sistemas para la solución de determinadas problemáticas a través de diferentes técnicas para la solución de problemas, entre las que se destacan la lógica difusa, las redes neurales, los sistemas neuro-difusos y los algoritmos genéticos. De esta forma los resultados que se desprenden de los procesos investigativos desarrollados en esta línea se orientan a la generación de algoritmos y metodologías que presentan un comportamiento autónomo, dinámico que se adecua a la evolución del entorno.”

1.4 Planteamiento del Problema

Colombia tiene un alto potencial en recursos energéticos a partir de fuentes renovables, la mayor parte del territorio nacional cuenta con un recurso de brillo solar de alrededor de 4 a 8 horas de sol al día, estos valores son altos en comparación a otros países, a partir de esto, el país es prometedor en cuanto a que se usen celdas fotovoltaicas como un método de generación de energía para desarrollar conjuntos de sistemas automáticos [2].

Los dispositivos denominados celdas fotovoltaicas convierten directamente la luz solar en electricidad mediante un proceso llamado efecto fotovoltaico, estas celdas se componen de

semiconductores como el silicio que tienen la capacidad de absorber energía proveniente de los fotones de luz [3]. Se recurre a tecnologías innovadoras como los sistemas fotovoltaicos que se constituyen como una alternativa de manera que se caracteriza por aprovechar la energía proveniente de los fotones de luz para ser renovables y sostenibles que da acceso al progreso de la eficiencia energética y la energía renovable, ya que el sol es una fuente de energía inagotable [4].

El desafío más difícil de los sistemas eléctricos es la capacidad de equilibrar tanto la generación como la demanda en todo momento ya que está sujeta a la integración de paneles solares para la generación eléctrica y a su vez se encuentra limitada por la incertidumbre e intermitencia de los parámetros meteorológicos, por lo tanto las predicciones enfocadas a la irradiancia carecen de poder determinar un periodo específico, por esto es necesario predecir los cambios en estos parámetros con ayuda de redes neuronales capaz de predecir en una serie multitemporal [5].

Una red neuronal es un enfoque de la inteligencia artificial, busca enseñarle a la computadora a procesar datos como la mente de un ser humano, existen diferentes arquitecturas de red neuronal siendo la LSTM una de las más completas ya que aborda secuencias y predicciones a largo plazo, tiene la capacidad de regular el flujo de información para realizar predicciones precisas en la red [6].

En este tipo de arquitectura se desconoce el porcentaje de error de irradiancia en una red neuronal basado en un algoritmo de tipo LSTM, además se acotan a establecer un tiempo fijo sin cambios en su predicción de manera que se limitan a no tener opciones multitemporales de las que se pueda variar, esto es debido a que no se ha realizado un estudio correspondiente donde se espere estimar el valor de la irradiancia en un periodo futuro, por lo que se percibe la ausencia de un algoritmo de predicción de la irradiancia que estime de manera precisa y con exactitud el coeficiente de determinación proveniente de variables meteorológicas. De continuar con esta situación será difícil estimar la calidad de la energía proveniente de las plantas solares puesto que el no tener una predicción de irradiancia, para una red de tamaño pequeño no es tan perceptible el cambio de la generación de kW/h, en cambio para una red de gran demanda estos costos pueden ser considerablemente significativos, demandando innecesariamente la producción de energía y sin considerar el impacto ambiental.

1.5 Objetivos

1.5.1 Objetivo General

Determinar el porcentaje de error en el pronóstico de la irradiancia en una red neuronal basado en un algoritmo de tipo LSTM multitemporal.

1.5.2 Objetivos Específicos

1. Gestionar la información de la estación Davis Vantage para preprocesamiento sobre datos adquiridos.
2. Implementar el modelo de red neuronal LSTM para el pronóstico de la irradiancia.
3. Validar el modelo desarrollado por medio del error entre datos reales y estimados.

1.6 Justificación

En la actualidad, la energía solar fotovoltaica es una de las fuentes más importantes de generación de electricidad limpia y renovable. Colombia, situada en la zona ecuatorial, experimenta una radiación solar media elevada. Esto ha presentado importantes oportunidades para aprovechar la energía solar en la región. Por lo tanto, es crucial comprender y estudiar el comportamiento de la radiación solar para lograr una obtención eficiente de esta energía.

El objetivo de este estudio es establecer el porcentaje de error en la predicción de la irradiancia mediante redes neuronales LSTM multitemporal. Este tipo de redes es adecuado para capturar dependencias a largo plazo en secuencias de datos, lo que resulta beneficioso para problemas como la predicción de irradiancia, ya que las dependencias pasadas pueden influir en gran medida en los valores futuros por parte de la predicción del modelo. Esto se debe a que, en comparación con otro tipo de redes, las LSTM cuenta con una estructura de memoria capaz de retener información relevante y mitigar problemas de *vanishing gradient* que ocurren en secuencias largas. Por todo lo anteriormente mencionado hace que este tipo de redes sea la mejor opción para trabajar con series temporales complejas y alta variabilidad, como las que se presentan en datos de irradiancia.

El proyecto es de tipo Estancia en Línea y contribuye al desarrollo del proyecto titulado “Análisis de rendimiento de algoritmos de predicción de irradiancia solar implementados en hardware y evaluados en tiempo real” del profesor Miller Manuel Ruales Luna del programa de ingeniería de la Universidad CESMAG.

El estudio requerirá el manejo de un conjunto de datos obtenidos durante los últimos 10 años a través de la estación Davis Vantage Pro de la Universidad CESMAG. Además, se implementará el modelo de red y posteriormente se verificarán los resultados de la predicción.

El desarrollo del algoritmo favorece a la Universidad CESMAG en un estudio sobre el tipo de red LSTM con característica multitemporal, trabajando de manera cooperativa con el proyecto investigativo [7] cuyo cronograma comprende en una sus actividades el desarrollo de una red neuronal de tipo LSTM como la del presente trabajo de grado, beneficiando a un desarrollo y profundización futura sobre el análisis y desempeño de este tipo de redes neuronales que complementan el servicio eléctrico ya que la energía solar es una fuente de energía descentralizada que puede generarse y consumirse en el mismo lugar. Esto reduce las pérdidas de transmisión y distribución, lo que a su vez aumenta la eficiencia del sistema eléctrico [8].

Este proyecto pretende estudiar de manera detallada el comportamiento de una red neuronal de tipo LSTM multitemporal a través del estudio de la predicción de la irradiancia. Con esto, se determinará en un periodo establecido una predicción de irradiación solar, por lo cual se podrá utilizar esta información para darle un mejor uso y aportando a tener un mayor rendimiento de estos generadores. La realización de este proyecto se ve enfocado al mejoramiento del estudio relacionado a energía fotovoltaica [9].

1.7 Delimitación

En este proyecto se determinará el error estimado en la predicción de irradiancia por la red neuronal LSTM multitemporal, utilizando los datos de irradiancia obtenidos en la ciudad de Pasto desde el año 2013 hasta el año 2024 por la estación Davis Vantage Pro-2.0 del grupo de investigación RAMPA de la Universidad CESMAG.

El proyecto es de tipo Estancia en Línea y contribuye al desarrollo del proyecto titulado “Análisis de rendimiento de algoritmos de predicción de irradiancia solar implementados en hardware y evaluados en tiempo real” del profesor Miller Manuel Ruales Luna del programa de ingeniería de la Universidad CESMAG.

2. Tópicos del Marco Teórico

2.1 Antecedentes

Las redes neuronales tienen diversas aplicaciones al momento de predecir la irradiancia, de este modo, se han estudiado diversos tipos de RNNA con el fin de dar un estimado al valor de la irradiancia futuro, la red de tipo LSTM tiene un comportamiento ideal para este tipo de aplicaciones, sin embargo, sobre todo lo que se ha trabajado carecen de poder estimar a criterio personal el periodo donde esta realice su predicción, esta característica Multitemporal comúnmente no es propia de este tipo de redes neuronales.

2.1.1 Predicción de series temporales en streaming mediante Deep Learning

El trabajo realizado por Pedro Benítez [10], desarrolló un estudio para mejorar la minería de datos en streaming donde estos se generan secuencialmente a gran velocidad por ende no permite el uso de técnicas de Deep Learning. Se presenta una solución mediante separar por medio de un Framework que él lo denomina (ADLStream) las fases de entrenamiento y predicción para reducir el coste computacional. Sumado a esto estudia la predicción en series temporales con una comparación de diferentes arquitecturas de redes neuronales y en combinación de las dos temáticas realiza una comprobación en la predicción de demanda eléctrica. Una de las conclusiones del estudio es que las redes LSTM Y CNN son las más adecuadas por su precisión para abordar temas de predicción. Del trabajo realizado basándose en sus pruebas y conclusiones se tomará las redes LSTM basándose en su precisión de predicción, en las pruebas realizadas se tomaron datos los cuales tenían una diferencia de tiempo de 10 minutos. En este proyecto los datos tomados para la predicción se toman con tiempo de 5 minutos.

2.1.2 Ten-minute prediction of solar irradiance based on cloud detection and a long short-term memory (LSTM) model.

El artículo realiza un estudio de la predicción de la irradiancia solar enfocado a una red neuronal de tipo LSTM de la cual hace énfasis en aquellas variables meteorológicas influyentes en el valor de la irradiancia, esto claramente afecta a la toma de datos y a la precisión del

algoritmo. Como lo describe Hui-Min Zo [11] si el algoritmo es de arquitectura LSTM este contiene entradas y salidas propias del modelo que aprende directamente en series temporales como las que ofrece la estación Davis Vantage Pro que registra datos de la Universidad CESMAG, además, almacena y elimina de forma automática la información. Una de las conclusiones importantes del artículo es la determinación de cómo actúa el algoritmo en un ambiente predictivo meteorológico, como lo es la irradiancia, ya que cada paso está en función del tiempo, la red neuronal LSTM procede a decidir qué valores son útiles, posteriormente toma un espacio temporal en una de sus células de memoria, procede a tomar decisiones mediante matrices generando una actualización, por último, la red neuronal decide cuál es la salida en función del estado de dicha célula. Dentro de los modelos verificados en este artículo, la autora determina que el algoritmo utilizado de arquitectura LSTM fue desarrollado y validado como el más importante para la predicción de la irradiancia puesto que el algoritmo propuesto utilizó variables meteorológicas, como la humedad, concentración de gases atmosféricos y el índice de cielo despejado, una vez comparado con los demás modelos este solventa la predicción de irradiancia de manera más precisa.

2.1.3 Time series forecasting on multivariate solar radiation data using deep learning (LSTM)

En el estudio realizado por Murat y Ozlem [12], acerca de la predicción de radiación solar haciendo uso de series temporales el objetivo es descubrir el efecto del uso de datos multivariantes en la predicción de la radiación solar, lo cual para este estudio se usó diferentes variables meteorológicas estas son la temperatura, presión, humedad, nebulosidad, velocidad del viento, dirección del viento, insolación y lluvia. estas variables se implementaron formando diferentes grupos para la implementación del set de entrenamientos de la RNN LSTM y poder observar cómo cada una de estas variables afecta a la predicción de la radiación solar, los datos recopilados abarcaron 87600 instancias para el periodo de 10 años entre enero de 1998 hasta diciembre del 2007, en este estudio para observar la métrica de rendimiento se usa el error cuadrático medio normalizado (NRMSE) comparándolo entre diferentes modelos. En Cada experimento de la red LSTM se utiliza una sola capa, como resultado entre esos modelos aplicados en el estudio la red LSTM (multivariante) otorga el NRMSE más bajo de 0.159 lo cual

concluye que las redes LSTM pueden ser muy competitivas para el desarrollo de predicciones temporales.

2.1.4 Solar Photovoltaic Forecasting of Power Output Using LSTM Networks

En el estudio realizado por María Konstantinou, Stefani Peratikou y Alexandros G. Charalambides [13], desarrollaron un modelo de pronóstico de energía solar basado en redes neuronales LSTM. La evaluación en datos de prueba arrojó un RMSE relativo de 0.11368, lo que indica una precisión significativa en las predicciones. Para garantizar la robustez del modelo, se empleó una validación donde se dividió el conjunto de entrenamiento en diez grupos y se obtuvo un RMSE promedio de 0.09394 con una desviación estándar de 0.01616. Esta validación respalda la estabilidad y el buen rendimiento del modelo en diferentes conjuntos de datos. En comparación con otros estudios relacionados con la predicción de la producción de energía solar, este enfoque se distingue por su uso exclusivo de datos de producción solar, sin depender de variables meteorológicas adicionales, además, el modelo utiliza 192 pasos de tiempo con intervalos de 15 minutos como entradas y cuenta con 4 capas de 50 celdas LSTM en su arquitectura. Esta elección permite capturar de manera efectiva las tendencias a largo plazo en los datos de producción de energía solar, lo que contribuye a su precisión y desempeño en el pronóstico.

2.1.5 Solar Radiation Prediction Based on Convolution Neural Network and Long Short-Term Memory

El estudio realizado por Jaihuni, Mustafa [14] desarrolló cinco variantes de redes neuronales recurrentes (RNN) para pronósticos a corto plazo de irradiación solar de 5 minutos. Se optimizaron las arquitecturas de las RNN con Hiperparámetros adecuados y se les dio consideración a la profundidad y amplitud de los modelos. Tras exhaustivas pruebas, se encontró que las RNN bidireccionales, es decir, Bi-LSTM y Bi-GRU, superaron a las versiones unidireccionales. El mejor desempeño se obtuvo con el modelo Bi-GRU, que logró un RMSE de 46.1 y un valor de R2 de 0.958. Además, el modelo Bi-GRU presentó el menor costo computacional, con 5.25*10^-5 s por peso por época, en comparación con los otros modelos. El estudio estableció que estos dos modelos requieren una gran cantidad de datos y configuraciones más profundas para lograr predicciones más precisas. Además, se mencionó la posibilidad de

aplicar estos modelos a pronósticos de irradiación solar para intervalos más largos, como 30 minutos o 1 hora. Se planteó la idea de estudiar modelos de conjunto, como el autorregresivo integrado de media móvil (ARIMA) y RNN. Se reconoció que los datos climáticos utilizados en el estudio se limitaban a una región, lo que podría restringir la aplicabilidad de las redes neuronales desarrolladas en otras regiones.

Aportes

A lo largo del proyecto, se han obtenido aportes significativos que complementan y contrastan con los antecedentes relacionados con la predicción de irradiancia utilizando redes neuronales LSTM. Además de analizar el comportamiento de estas redes con los datos de la estación meteorológica de la Universidad CESMAG, el proyecto logró desarrollar algoritmos con capacidad de predicción multitemporal de hasta 8 días, manteniendo un buen desempeño en sus métricas de evaluación, esto representa un avance diferencial en comparación con estudios previos los cuales suelen limitarse a comprobar su validación en el conjunto de test.

Se logró establecer una relación entre la precisión de las predicciones y el volumen de datos utilizados para el entrenamiento, validación y prueba, proporcionando así un enfoque más robusto para el manejo de series temporales en aplicaciones de energía solar. Así mismo, las predicciones para horizontes de corto y mediano plazo ofrecen un valor práctico en la gestión de energía solar, permitiendo decisiones informadas y mejorando la planificación de recursos.

Estos logros contribuyen al campo de la predicción de irradiancia al ofrecer una solución que mejora la capacidad de anticipación en la generación solar, optimizando el uso de recursos energéticos en contextos locales y con datos específicos. Este aporte, al contrastarse con los antecedentes, refleja un avance en la aplicación de redes LSTM multitemporales para sistemas de energía renovable, destacando el valor de los resultados obtenidos en condiciones reales y con conjuntos de datos extensos.

2.2 Supuestos Teóricos de la Investigación

2.2.1 Predicción

La predicción es un componente esencial en la inteligencia artificial, donde las redes neuronales, en particular las redes neuronales de tipo LSTM, desempeñan un papel destacado. Las redes LSTM son una variante de las redes neuronales recurrentes (RNN) que se utilizan para

hacer predicciones en secuencias de datos, como series temporales, lenguaje natural y otros datos secuenciales. Las redes LSTM son especialmente efectivas en la predicción porque pueden capturar dependencias a largo plazo en los datos, lo que las hace ideales para predecir eventos futuros en función de patrones complejos en secuencias temporales. Estas redes son capaces de aprender y recordar información relevante de entradas anteriores a medida que procesan nuevas entradas, lo que las convierte en un poderoso instrumento para la predicción. En aplicaciones de inteligencia artificial, las redes LSTM se utilizan comúnmente para tareas de predicción, como prever el comportamiento del mercado financiero, el tráfico de una ciudad, el pronóstico del tiempo y la detección de anomalías en datos de sensores. Además, se aplican en procesamiento del lenguaje natural para predecir palabras en una secuencia de texto o incluso para generar texto coherente en un contexto específico [15].

2.2.2 Sistemas Fotovoltaicos

Los sistemas fotovoltaicos son sistemas de generación de energía que aprovechan la radiación solar para producir electricidad, estos sistemas se componen principalmente de paneles solares, que contienen células fotovoltaicas capaces de convertir la luz solar en energía eléctrica. Cuando la luz solar incide sobre los paneles, los electrones en las células fotovoltaicas se excitan, generando una corriente eléctrica continua. Un inversor convierte esta corriente continua en corriente alterna, que es la forma de electricidad utilizada en la mayoría de los dispositivos eléctricos y en las redes eléctricas convencionales. Los sistemas fotovoltaicos son una fuente de energía sostenible y limpia que contribuye a la reducción de las emisiones de carbono y la independencia energética, y son utilizados tanto en aplicaciones residenciales como comerciales y a gran escala en plantas de energía solar [16].

2.2.3 Machine Learning

Machine Learning (ML), es el estudio científico y diciplina del campo de la inteligencia artificial, que aplica algoritmos y modelos estadísticos a sistemas informáticos para realizar una tarea especifica sin estar programados explícitamente. El propósito del machine learning es aprender de una serie de datos de entrada. Dado que este tipo de aprendizaje se basa en diferentes algoritmos los científicos señalan que no existe un único tipo de algoritmo que pueda resolver todo tipo de problema, cada algoritmo o modelo para la implementación de ML dependen de

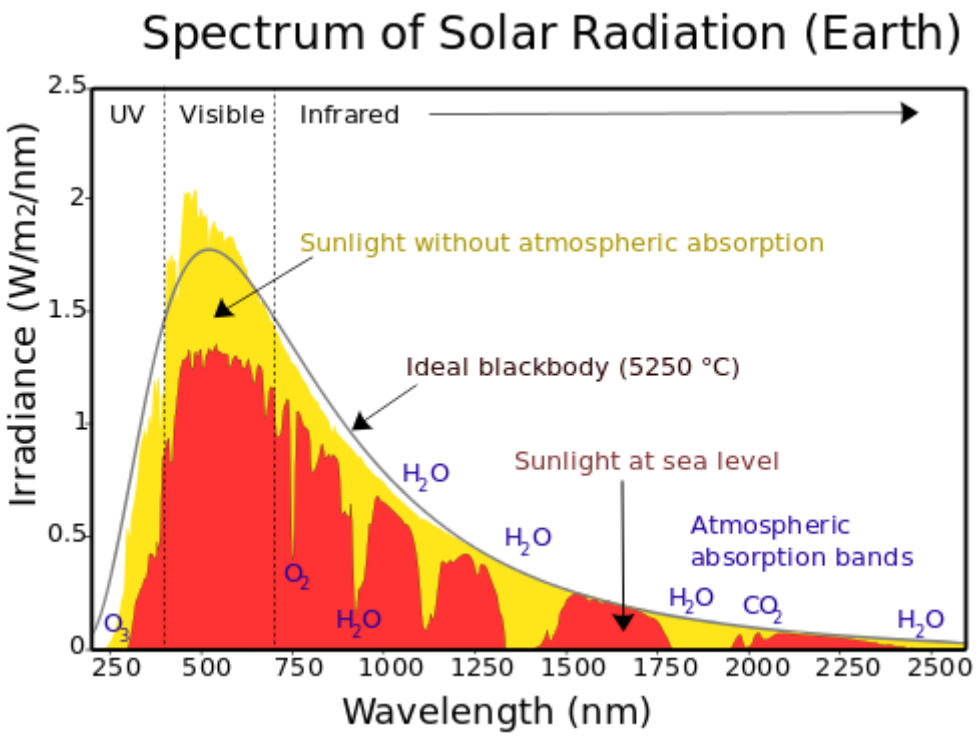
ciertas características como el número de variables de entrada, es por eso que existen distintos modelos para la implementación de este aprendizaje entre ellos se encuentran: modelo de aprendizaje supervisado, aprendizaje no supervisado, aprendizaje por refuerzo, aprendizaje conjunto, modelos generativos , redes neuronales etc. [17].

2.2.4 Irradiancia Solar

La tasa a la cual la radiación es recibida por una superficie por unidad de área se denomina irradiancia, la misma que se expresa en unidades de potencia por unidad de área, W/m2. La cantidad de radiación recibida por una superficie por unidad de área durante un determinado período se denomina irradiación y se expresa en unidades de energía por unidad de área, Wh/m2 [18].

La irradiancia conocida también como la radiación solar global en la superficie de la tierra se divide en varias ramas como es la radiación difusa, que es la que se recibe del Sol, después de ser desviada por dispersión atmosférica. Es radiación difusa la que se recibe a través de las nubes, así como la que proviene del cielo azul. La radiación que proviene de objetos terrestres es la radiación terrestre. Se conoce como radiación total, la suma de las radiaciones directa, difusa y terrestre que se reciben sobre una superficie. Un caso particular, pero de mucho interés práctico en el estudio de la energía solar, es el medir la radiación total sobre una superficie horizontal "viendo" hacia arriba como se puede observar en la Figura 1. En este caso puede considerarse que no existe radiación terrestre y se conoce también como radiación global es la suma de la directa más la difusa [19].

Figura 1 Spectrum of Solar Radiation (Earth)



Nota. Imagen tomada de [20].

2.2.5 Redes Neuronales

Una red neuronal puede ser caracterizada por el modelo de la neurona, el esquema de conexión que presentan sus neuronas, o sea su tipología, y el algoritmo de aprendizaje empleado para adaptar su función de cómputo a las necesidades del problema particular.

Existe una amplia variedad de modelos de neuronas, cada uno se corresponde con un tipo determinado de funciones de activación y de salida de la neurona.

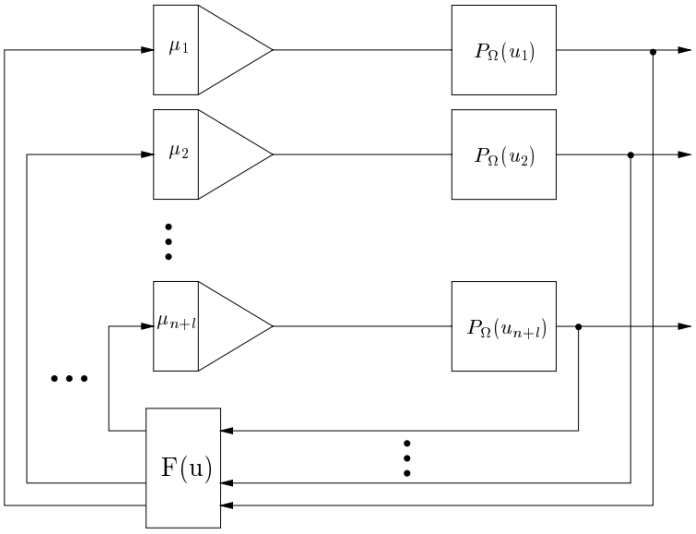
La topología es la forma específica de conexión (arquitectura) y la cantidad de neuronas conectadas (el número de parámetros libres) que describen una red. En los últimos años se han producido una amplia variedad de topologías de redes neuronales, sin embargo, la mayoría de ellas se encuentran ubicadas en dos grandes grupos: la redes multicapa de alimentación hacia adelante (feed-forward) y las redes recurrentes (RNN).

Por otro lado, las redes recurrentes son sistemas dinámicos. El cálculo de una entrada, en un paso, depende del paso anterior y en algunos casos del paso futuro.

Las RNN son capaces de realizar una amplia variedad de tareas computacionales incluyendo el tratamiento de secuencias, la continuación de una trayectoria, la predicción no lineal y la modelación de sistemas dinámicos.

Estas también se conocen como redes espaciotemporales o dinámicas, son un intento de establecer una correspondencia entre secuencias de entrada y de salida que no son más que patrones temporales de los cuales su estructura se observa en la Figura 2.

Figura 2 Arquitectura de una Red Neuronal



Nota. Imagen tomada de [21]

Existen tres tipos de tareas esenciales que se pueden realizar con este tipo de redes:

- Reconocimiento de secuencias.
- Reproducción de secuencias.
- Asociación temporal.

Una RNN se puede clasificar en parcial y/o totalmente recurrente. Las totalmente recurrentes son aquellas que cada neurona puede estar conectada a cualquier otra y sus conexiones recurrentes son variables. Las redes parcialmente recurrentes son aquellas que sus conexiones recurrentes son fijas. Estas últimas son la forma usual para reconocer o reproducir secuencias. Generalmente tienen la mayoría de las conexiones hacia adelante, pero incluyen un conjunto de conexiones retroalimentadas. Existen varios tipos de redes definidas con su tipo de topología y sus algoritmos de aprendizaje [22].

2.2.6 Modelos Bert y Convolucional (CNN)

Bert, cuyas siglas significan Bidireccional Encoder for Transformers. El objetivo de este modelo consiste en interpretar el lenguaje de una manera mucho más natural mediante programación neuro lingüística, Bert utiliza un modelo de análisis bidireccional, quiere decir que según la palabra que se esté analizando se revisa de derecha a izquierda para conocer los patrones de la oración, se usa una estructura multicapa donde en la entrada de cada una de las células es primordial la utilización de las componentes de las etapas anteriores de la red [23]. Normalmente este tipo de modelos se usan para mejorar la comunicación o interacción entre los usuarios y computadoras con el uso la interpretación del lenguaje natural.

Por otro lado, los modelos convolucionales son redes de neuronas que presentan múltiples capas para calcular la salida dado un conjunto de datos, este tipo de red se generaran pixeles con el uso de una matriz numérica que se denomina filtro, el resultado varía dependiendo de la configuración de este filtro por lo cual este modelo CNN es uno de los avances más importantes en el aprendizaje profundo utilizado en el reconocimiento de imágenes [24].

2.2.7 Redes Neuronales LSTM

Las redes neuronales de memoria de corto largo plazo (LSTM) son un tipo particular de RNN que solucionan el problema de las RNN clásicas asociado a la memoria de corto plazo: el desvanecimiento o decaimiento del gradiente o su “explosión”. Esta dificultad se supera mediante la implementación de tangentes hiperbólicas que hacen que su resultado se mantenga siempre acotado en la nueva celda del tipo LSTM. En la figura 1 todo lo que hay en el recuadro punteado corresponde a una sola unidad de la red. En primer lugar, hay que tener en cuenta que se ha mostrado una sola componente de una LSTM, por lo que a la izquierda se tiene la información procedente del procesamiento del dato asociado al período, utilizando para ello dos estructuras de datos en este caso del tipo tensor, que son arreglos de datos de más de dos dimensiones. En la parte inferior tiene la entrada con información de la unidad anterior. A la derecha tiene dos tensores que salen para informar a la siguiente unidad de tiempo y, como en la

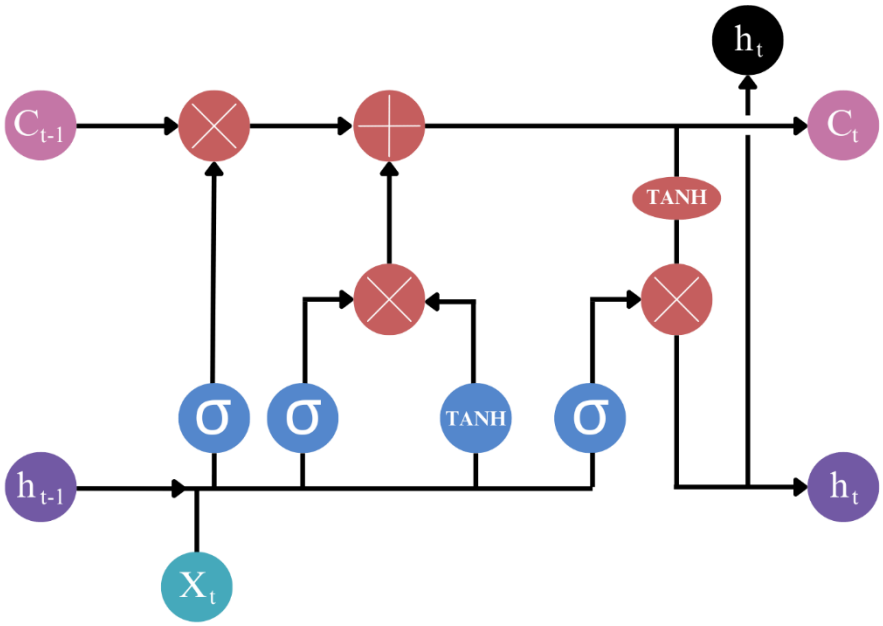
RNN simples, se tendría esta información “hacia arriba” en el diagrama para predecir la siguiente palabra y la pérdida (parte superior, lado derecho).

Nota. Imagen tomada de [25]

El objetivo es mejorar la memoria de la RNN de los eventos pasados entrenándola para que recuerde lo importante y olvide el resto. Para ello, las LSTM procesan dos versiones del pasado. La memoria selectiva “social” está en la parte superior y una versión más local en la parte inferior. La línea de tiempo de la memoria superior se llama el estado de la célula y se abrevia c. La línea inferior se llama h.

La Figura 3 introduce varias conectivas y funciones de activación nuevas. En primer lugar, se observa que la línea de memoria se modifica en dos lugares antes de pasar a la siguiente unidad de tiempo. Están etiquetados como tiempos X, y +. La idea es que las memorias se

Figura 3 Arquitectura de una Red Neuronal LSTM



eliminan en la unidad de tiempo X y se añaden en la unidad +. Se hace esto para que la incrustación de la información actual que viene en la parte inferior izquierda pase por una capa de unidades lineales seguida de activación sigmoidea, como indica la anotación W, b, S. Donde W son los pesos y b son las fallas. W y b forman la combinación lineal y S es la función sigmoidea de una neurona clásica. En notación matemática la operación queda de la siguiente manera:

$$\boldsymbol{h' = h_t \cdot e} \qquad (1)$$

$$\boldsymbol{f = S(h'W_f + b_f)} \quad (2)$$

Se utiliza un punto central para indicar la concatenación de vectores. Para repetir, en la parte inferior izquierda se concatena la línea h anterior h_t y el dato actual e para obtener h_0, que a su vez se introduce en la unidad lineal de "olvido" (seguida de una sigmoidea) para producir f, la señal de olvido que se desplaza hacia arriba en el lado izquierdo de la figura. La salida de la sigmoidea se multiplica elemento a elemento de la memoria que viene de la parte superior izquierda. La ecuación de esta operación se representa así:

$$\boldsymbol{c'_t = c_t \odot f} \quad (3)$$

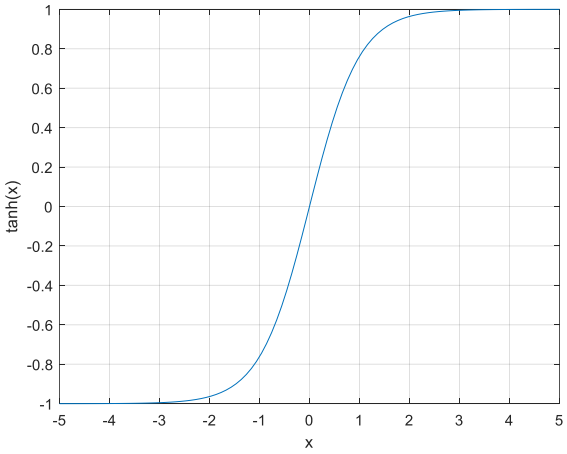
Dado que los sigmoides están limitados por cero y uno, el resultado de la multiplicación debe ser una reducción en el valor absoluto en cada punto de la memoria principal. Esto corresponde al "olvido". En general, esta conjugación, sigmoidal que alimenta una multiplicación es un patrón común cuando se quiere una compuerta que me lleve a obtener un cero o uno.

Contrasta esto con lo que sucede en la unidad aditiva con que se encuentra la memoria. De nuevo, el siguiente dato, que ha llegado desde abajo izquierda, pasa ahora por separado a través de dos capas lineales, una con una con activación sigmoidea y otra con la función de activación Tanh, como se muestra en la Figura 4.

$$\boldsymbol{a_1 = S(h'W_{a_1} + b_{a_1})} \quad (4)$$

$$\boldsymbol{a_2 = tanh((h_t \cdot e)W_{a_2} + b_{a_2})} \quad (5)$$

Figura 4 Tangente Hiperbólica



Nota. Imagen de Elaboración Propia

Es importante que, a diferencia de la función sigmoidea, Tanh puede tener como salidas tanto valores positivos o negativos, por lo que puede arrojar valores nuevos en lugar de sólo escalar el dato de entrada. El resultado de esto se añade al estado de la celda en la celda etiquetada "+":

$$\boldsymbol{c_{t+1}} = \boldsymbol{c'_t} \oplus (\boldsymbol{a_1} \odot \boldsymbol{a_2}) \quad (6)$$

Después de esto, la línea de memoria se divide. Una copia sale por la derecha, y una copia pasa por un Tanh y luego se combina con una transformación lineal de la historia y el valor actual, para convertirse en la nueva línea *h* de la parte inferior:

$$\boldsymbol{h''} = \boldsymbol{h'W_h} + \boldsymbol{b_h} \quad (7)$$

$$\boldsymbol{h_{t+1}} = \boldsymbol{h''} \odot \boldsymbol{a_2} \quad (8)$$

Esta se concatenará con la siguiente entrada, y el proceso se repetirá. El punto por destacar aquí es que la línea de memoria de celdas nunca pasa directamente por las unidades lineales. La memoria se irá desvirtuando (se "olvidan") en la unidad "X" y se añadirá nueva información del dato actual en "/+", pero no habrá operaciones matemáticas ni de combinación lineal ni de escalamiento/transformación no lineal [26].

2.3 Definición de Conceptos

2.3.1 Definición Nominal o Teórica de las Variables de Estudio

- Coeficiente de determinación

El coeficiente de determinación R^2 es una medida estadística utilizada para evaluar el desempeño de un modelo de regresión. Representa la proporción de la varianza en los datos observados que es explicada por el modelo. En términos simples, indica qué tan bien las predicciones del modelo se ajustan a los datos reales. Un valor de R^2 cercano a 1 sugiere que el modelo explica casi toda la variabilidad en los datos, mientras que un valor cercano a 0 indica que el modelo tiene poca capacidad predictiva y no captura adecuadamente la relación entre las variables.

- MAE

El MAE (Mean Absolute Error) es una métrica utilizada para medir la precisión de un modelo de predicción. Representa el promedio de las diferencias absolutas entre los valores predichos por el modelo y los valores reales observados, lo que significa que muestra el error promedio sin considerar la dirección del error (si es positivo o negativo). Al trabajar con diferencias absolutas, el MAE brinda una medida directa y fácilmente interpretable del error en las predicciones, expresada en las mismas unidades que los datos.

- RMSE

El RMSE (Root Mean Square Error) es una métrica que mide el error promedio de un modelo de predicción, penalizando los errores grandes de manera más severa que los errores pequeños. Se calcula tomando la raíz cuadrada del promedio de los errores al cuadrado entre los valores predichos y los valores reales. Al elevar los errores al cuadrado, el RMSE amplifica las diferencias grandes, lo que lo convierte en una métrica útil cuando se quiere poner mayor énfasis en minimizar errores significativos en las predicciones.

- Irradiancia

La irradiancia es una medida que se utiliza en el contexto de la radiometría y la óptica para cuantificar la cantidad de energía radiante que llega a una superficie por unidad de área y unidad de tiempo. Se representa típicamente con la letra "E" y se expresa en unidades de vatios por metro cuadrado (W/m^2). La irradiancia es fundamental para comprender cómo la energía electromagnética, como la luz visible o la radiación solar, se distribuye y se absorbe en diferentes superficies.

En términos simples, la irradiancia nos dice cuánta energía luminosa o radiante incide sobre una superficie específica en un período dado. Por ejemplo, en el caso de la radiación solar, la irradiancia solar se refiere a la cantidad de energía solar que llega a la tierra por unidad de área en un momento determinado y es esencial para comprender los fenómenos relacionados con el clima, la generación de energía solar y otros campos de estudio.

La fórmula básica para calcular la irradiancia (E) es:

$$E = P / A$$

Donde:

- E es la irradiancia en vatios por metro cuadrado (W/m^2).

- P es la potencia radiante total incidente sobre la superficie en vatios (W).
- A es el área de la superficie sobre la cual incide la radiación en metros cuadrados (m^2).

Esta fórmula te permite calcular la irradiancia cuando conoces la potencia radiante total y el área de la superficie en cuestión. La irradiancia es especialmente relevante en aplicaciones como la energía solar, donde se mide la cantidad de energía solar que llega a un panel solar o una superficie receptora para determinar su eficiencia y capacidad de generación de electricidad.

2.3.2 Definición Operativa o Empírica de las Variables de Estudio

El R^2 indica la cantidad de variabilidad de los datos que puede ser explicada por las variables independientes del modelo. El valor de R^2 varía entre 0 y 1. Un valor de $R^2 = 1$ indica que el modelo predice perfectamente los datos, mientras que $R^2 = 0$ significa que el modelo no explica ninguna de las variaciones en los datos. En términos operativos, se utiliza para determinar qué tan bien un modelo se ajusta a los datos observados, donde valores más cercanos a 1 indican un mejor ajuste.

$$R^2 = 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \bar{y})^2}$$

En esta fórmula:

- y_i son los valores observados.
- $\hat{y}_i$ son los valores predichos por el modelo.
- $\bar{y}$ es la media de los valores observados
- $\sum(y_i - \hat{y}_i)^2$ representa la suma de errores al cuadrado o residuos, que indica el error del modelo.
- $\sum(y_i - \bar{y})^2$ es suma total de la variabilidad en los datos observados.

El MAE varía entre 0 y valores positivos, donde un valor de MAE = 0 indica que el modelo predice perfectamente los datos. Cuanto mayor sea el valor del MAE, mayor será la desviación promedio entre las predicciones del modelo y los valores reales.

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i|$$

En esta fórmula:

- n es el número total de observaciones.

- y_i son los valores observados.
- $\hat{y}_i$ son los valores predichos por el modelo.
- $|y_i - \hat{y}_i|$ es la diferencia absoluta entre el valor real y el valor predicho.

El error RMSE (Root Mean Squared Error) es una métrica utilizada comúnmente para evaluar la precisión de un modelo en relación con los valores observados o reales. Se emplea principalmente en problemas de regresión para medir la diferencia entre los valores pronosticados por el modelo y los valores reales.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

En esta fórmula:

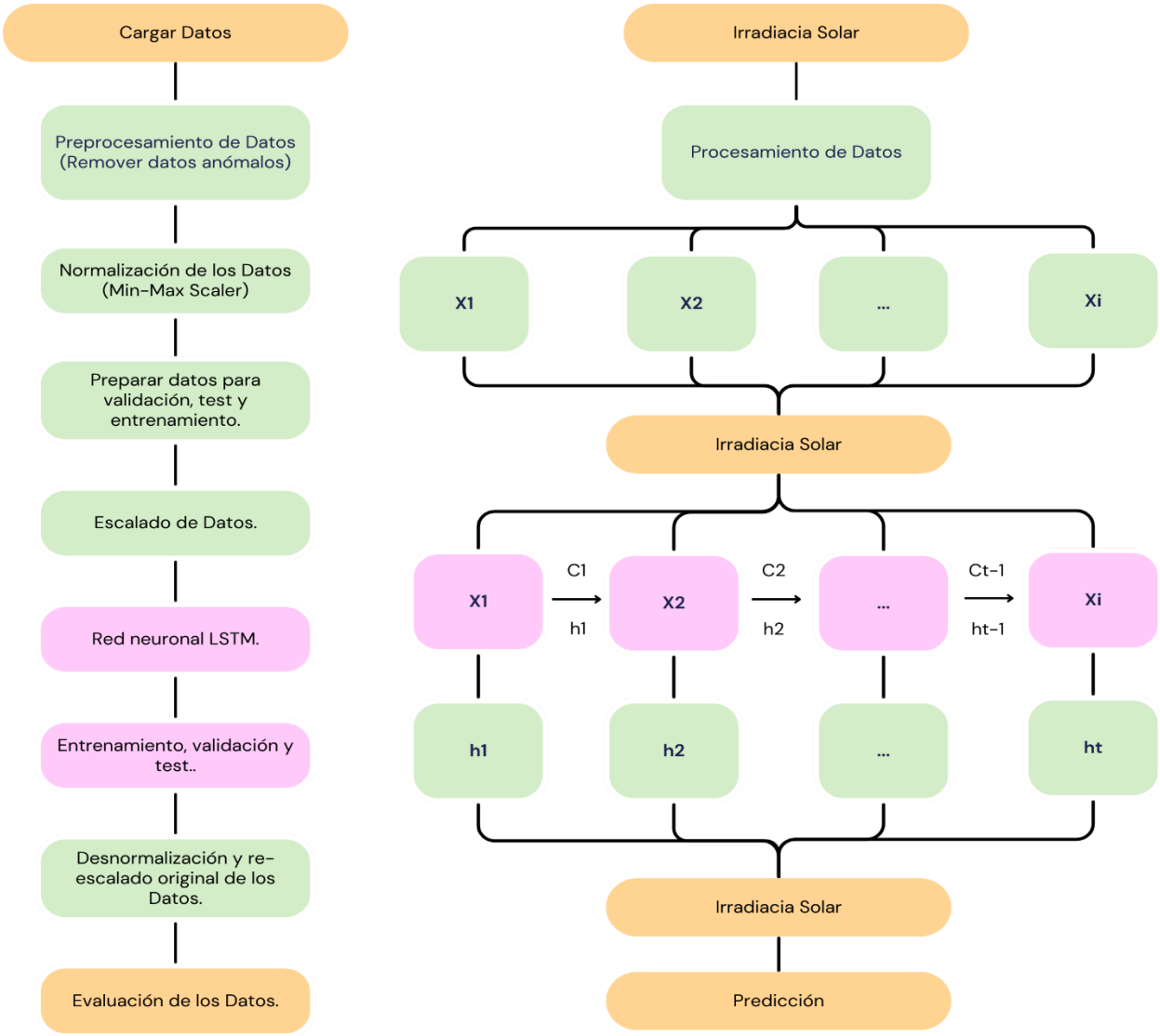
- n es el número total de observaciones.
- y_i son los valores reales u observados.
- $\hat{y}_i$ son los valores predichos por el modelo.

3. Metodología

3.1 Secuencia Metodológica

El enfoque de esta investigación es cuantitativo, ya que utiliza la recolección y el análisis de datos para dar solución a la pregunta de investigación, además las variables asociadas al objeto de investigación se miden en métodos estadísticos con porcentajes y mediante los datos obtenidos permitirán determinar el error estimado en la predicción de la irradiancia usando la red neuronal recurrente LSTM.

Figura 5 Secuencia Metodológica del Proyecto



Nota. Imagen de Elaboración Propia

En la Figura 5 los componentes son los siguientes:

X: Representa las entradas de la red, en este caso, los datos de irradiancia solar en diferentes instantes de tiempo. Cada X_t corresponde al valor de entrada en el instante t.

C: Denota el estado de la celda de la red LSTM, que se usa para almacenar la memoria a largo plazo de la red. Cada C_t representa el estado de la celda en el instante t.

h: Indica el estado oculto de la red, que es la salida de la celda LSTM en cada paso de tiempo. Cada h_t representa el estado oculto en el instante t y captura la información de la secuencia procesada hasta ese momento.

3.2 Técnicas de recolección de información

Para recolectar la información de la investigación, se utilizará la base de datos obtenida mediante la estación meteorológica Davis Vantage Pro-2.0 del grupo de investigación RAMPA, la cual proporcionará diferentes datos de irradiancia a lo largo del tiempo a través del sensor de radiación solar asociado a la estación.

3.2.1 Validez de la técnica

La técnica es válida, ya que la estación meteorológica proporciona datos fiables obtenidos a través de los sensores de radiación solar. Estos datos se almacenan en el software Weatherlink y se envían a la nube, específicamente a Google Drive, lo que facilita su acceso y manejo. Además, la precisión y velocidad con la que se monitorizan y registran los datos es eficaz, con mediciones de radiación solar cada cinco minutos, y una resolución que va desde 1 W/m² hasta 1800 W/m², con una precisión nominal del 5% en escala total [28]. Esto garantiza una cantidad suficiente de datos, que serán almacenados en Excel para su posterior depuración e implementación en los entrenamientos de la red neuronal LSTM.

3.2.2 Confiabilidad técnica

A pesar de las limitaciones potenciales dentro de la fiabilidad del sensor y frecuencia de muestreo, la estación meteorológica en combinación con el preprocesamiento de datos y las herramientas de almacenamiento asegura un flujo de trabajo técnico confiable. Esto contribuye a la confianza en la calidad de los resultados que se obtendrán de los entrenamientos.

3.3 Instrumentos de recolección de información

La información de irradiancia será obtenida de la estación meteorológica. La información recolectada será almacenada en una hoja de cálculo de Excel (ver Tabla 1) e importada al software Python, donde será depurada utilizando técnicas de limpieza de datos, como la eliminación de valores atípicos, para descartar aquellos que puedan afectar la correcta predicción de la irradiancia. Posteriormente, se identificarán los valores faltantes, se normalizarán los datos, se corregirán las inconsistencias, se documentarán los cambios y se verificará la coherencia temporal, ya que el tratamiento de los datos es la fase inicial del proceso de desarrollo de la red.

Tabla 1 Recolección de datos en hoja de Excel

Hora	Fecha	Dia	2014	2015	2016	2017	2018	2019	2020	2021	2022	2023
6:00 a. m.	1/1/2015	1		0						0	0	
6:05 a. m.	1/1/2015	1		0						0	0	
6:10 a. m.	1/1/2015	1		0						0	0	
6:15 a. m.	1/1/2015	1		0						0	5	
6:20 a. m.	1/1/2015	1		0						0	7	
6:25 a. m.	1/1/2015	1		0						3	10	
6:30 a. m.	1/1/2015	1		0						7	12	
6:35 a. m.	1/1/2015	1		7						10	13	
6:40 a. m.	1/1/2015	1		15						12	31	
6:45 a. m.	1/1/2015	1		31						15	44	
6:50 a. m.	1/1/2015	1		46						21	60	
6:55 a. m.	1/1/2015	1		46						27	77	
7:00 a. m.	1/1/2015	1		46						31	96	
7:05 a. m.	1/1/2015	1		52						34	114	
7:10 a. m.	1/1/2015	1		68						40	133	
7:15 a. m.	1/1/2015	1		81						51	154	
7:20 a. m.	1/1/2015	1		96						70	176	
7:25 a. m.	1/1/2015	1		142						83	199	
7:30 a. m.	1/1/2015	1		183						90	221	
7:35 a. m.	1/1/2015	1		175						97	242	
7:40 a. m.	1/1/2015	1		187						98	259	
7:45 a. m.	1/1/2015	1		175						102	276	
7:50 a. m.	1/1/2015	1		166						114	296	
7:55 a. m.	1/1/2015	1		153						112	317	
8:00 a. m.	1/1/2015	1		150						110	338	
8:05 a. m.	1/1/2015	1		137						150	358	
8:10 a. m.	1/1/2015	1		149						154	375	
8:15 a. m.	1/1/2015	1		154						386	394	
8:20 a. m.	1/1/2015	1		150						442	412	

4. Resultados

4.1 Preprocesamiento sobre datos adquiridos de la estación Davis Vantage.

Para llevar a cabo un estudio detallado de la irradiancia solar y su predicción a través de redes neuronales, es necesario iniciar con la recolección de datos e importación de librerías (4.1.1), seguido de una limpieza de datos (4.1.2) para suprimir elementos indeseables. Se llevará a cabo la sustitución de cadenas de texto por NaN (4.1.3) y la identificación de valores ausentes (4.1.4), para posteriormente efectuar la imputación de datos (4.1.5). Luego, se realizará la última visualización del DataFrame (4.1.6) y, finalmente, se verificará el relleno de NaN (4.1.7), garantizando la calidad de los datos que respaldarán el modelo predictivo.

4.1.1 Extracción de datos e importación de librerías.

En primer lugar, se extrajeron los datos de la estación Davis Vantage Pro-2.0, obteniendo un archivo plano con problemas de *outliers*, desfases temporales y campos nulos, causados por la desconexión de la estación, caídas en la energía eléctrica y una mala sincronización. Para abordar este problema, se desarrolló un algoritmo en Python. En este caso, se importa el archivo Excel que contiene los datos de cualquier mes dentro del rango de junio de 2013 a diciembre de 2023 (ver Anexo 1).

Figura 6 Importación del Set de Datos

```python
from google.colab import drive
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

# Montar Google Drive
drive.mount('/gdrive')

# Leer set de datos
ruta = '/gdrive/MyDrive/Colab_Notebooks/'
df = pd.read_excel(ruta+'Radiacion_2013_2023_NaN.xlsx',
sheet_name = 'Enero')
```

Nota. Imagen de Elaboración Propia

Se importan las librerías necesarias para el manejo de datos (*pandas*), el cálculo y manejo de *arrays* (*numpy*), y la visualización de datos (*matplotlib*). Además, se importa la librería para montar Google Drive (*google.colab.drive*). Posteriormente, se monta Google Drive en el entorno de Google Colab para acceder a los archivos almacenados en él. Por último, se especifica la ruta del archivo de datos en Google Drive y se carga el archivo Excel en un DataFrame de *pandas*. Tras esto, se obtiene el siguiente resultado:

Figura 7 Set de datos con NaN's

	Hora	Fecha	Dia	2014	2015	2016	2017	2018	2019	2020	2021	2022	2023
0	06:00:00	2015-01-01	1	NaN	0	NaN	NaN	NaN	NaN	NaN	0.0	0.0	NaN
1	06:05:00	2015-01-01	1	NaN	0	NaN	NaN	NaN	NaN	NaN	0.0	0.0	NaN
2	06:10:00	2015-01-01	1	NaN	0	NaN	NaN	NaN	NaN	NaN	0.0	0.0	NaN
3	06:15:00	2015-01-01	1	NaN	0	NaN	NaN	NaN	NaN	NaN	0.0	5.0	NaN
4	06:20:00	2015-01-01	1	NaN	0	NaN	NaN	NaN	NaN	NaN	0.0	7.0	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...	...
4862	18:40:00	2015-01-31	31	0.0	0	0.0	0.0	NaN	NaN	0.0	0.0	0.0	0.0
4863	18:45:00	2015-01-31	31	0.0	0	0.0	0.0	NaN	NaN	0.0	0.0	0.0	0.0
4864	18:50:00	2015-01-31	31	0.0	0	0.0	0.0	NaN	NaN	0.0	0.0	0.0	0.0
4865	18:55:00	2015-01-31	31	0.0	0	0.0	0.0	NaN	NaN	0.0	0.0	0.0	0.0
4866	19:00:00	2015-01-31	31	0.0	0	0.0	0.0	NaN	NaN	0.0	0.0	0.0	0.0

4867 rows × 13 columns

Nota. Imagen de Elaboración Propia

4.1.2 Limpieza de datos.

Se descartan las columnas 'Hora', 'Fecha' y 'Num' del DataFrame, puesto que estas columnas no son necesarias de momento para realizar el promedio. Solo son necesarias las columnas que contengan valores de irradiancia.

Figura 8 Drop de datos innecesarios

```
df.drop(['Hora','Fecha', 'Dia'], axis=1, inplace=True)
```

Nota. Imagen de Elaboración Propia

Por lo que se obtiene un set de datos más dinámico para la sección de código que hará el promedio.

Figura 9 DataSet listo para promediar

	2014	2015	2016	2017	2018	2019	2020	2021	2022	2023
0	NaN	0	NaN	NaN	NaN	NaN	NaN	0.0	0.0	NaN
1	NaN	0	NaN	NaN	NaN	NaN	NaN	0.0	0.0	NaN
2	NaN	0	NaN	NaN	NaN	NaN	NaN	0.0	0.0	NaN
3	NaN	0	NaN	NaN	NaN	NaN	NaN	0.0	5.0	NaN
4	NaN	0	NaN	NaN	NaN	NaN	NaN	0.0	7.0	NaN
...	...	...	...	...	...	...	...	...	...	...
4862	0.0	0	0.0	0.0	NaN	NaN	0.0	0.0	0.0	0.0
4863	0.0	0	0.0	0.0	NaN	NaN	0.0	0.0	0.0	0.0
4864	0.0	0	0.0	0.0	NaN	NaN	0.0	0.0	0.0	0.0
4865	0.0	0	0.0	0.0	NaN	NaN	0.0	0.0	0.0	0.0
4866	0.0	0	0.0	0.0	NaN	NaN	0.0	0.0	0.0	0.0

4867 rows × 10 columns

Nota. Imagen de Elaboración Propia

4.1.3 Reemplazo de cadenas de texto por NaN.

Para la sección donde se realiza el promedio se necesita que todos los datos del DataFrame sean de tipo numérico, debido a la posibilidad de que en las columnas existan celdas conteniendo un valor de tipo caracter, por lo que habrá de reemplazarlos por celdas nulas (NaN).

Figura 10 DataSet listo para promediar

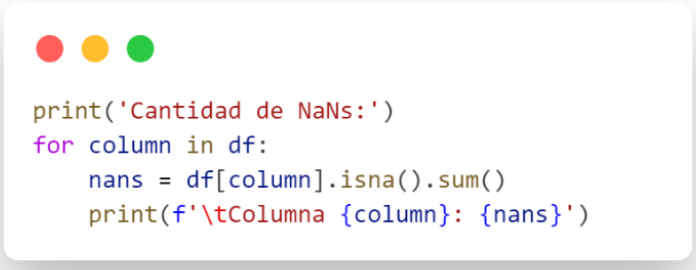
```
# Reemplazar cadenas de texto por NaNs
for index, row in df.iterrows():
    for col_name, cell_value in row.items():
        if isinstance(cell_value, str):
# Verifica si el valor es una cadena de texto
            df.at[index, col_name] = np.nan
# Reemplaza la cadena de texto con NaN
```

Nota. Imagen de Elaboración Propia

4.1.4 Determinar valores faltantes.

Con el fin de dimensionar cifras exactas la cantidad de valores faltantes que se tiene en el mes, se realiza un conteo e imprimen aquellos valores NaN (nulos) en cada columna del Dataframe.

Figura 11 Imprimir cantidad de NaN's



```
print('Cantidad de NaNs:')
for column in df:
    nans = df[column].isna().sum()
    print(f'\tColumna {column}: {nans}')
```

Nota. Imagen de Elaboración Propia

Figura 12 Cantidad exacta de NaN's

```
Cantidad de NaNs:
        Columna 2014: 2485
        Columna 2015: 0
        Columna 2016: 2724
        Columna 2017: 4840
        Columna 2018: 4867
        Columna 2019: 4867
        Columna 2020: 2048
        Columna 2021: 2448
        Columna 2022: 117
        Columna 2023: 446
```

Nota. Imagen de Elaboración Propia

4.1.5 Imputación de Datos.

A partir de esto, se recorren todas las filas del DataFrame en busca de valores NaN que serán eliminados, una vez realizado este proceso, continua la lectura de las celdas con valores mayores o iguales a cero, con esto aseguramos una correcta integración de todos los valores numéricos existentes.

Para cada fila, se calculan los promedios teniendo en cuenta las filas aledañas existentes en el mismo instante de tiempo que tengan un valor diferente a NaN, de manera

que se tendrá el relleno promediado en las celdas en cada uno de los espacios vacíos, esto se hace sin afectar los valores ya existentes que posee el Dataframe por defecto.

Figura 13 Promedio recorriendo fila a fila

```
# Recorrer fila a fila y calcular el promedio
for index, row in df.iterrows():
    valores_validos = row.dropna()  # Eliminar valores NaN
    if len(valores_validos) >= 0:  # Verificar si hay valores válidos
        promedio = valores_validos.mean()  # Calcular el promedio
        for col in row.index:
            if pd.isnull(row[col]):  # Rellenar solo las celdas vacías
                df.loc[index, col] = promedio  # Rellenar la celda vacía con el promedio calculado
```

Nota. Imagen de Elaboración Propia

4.1.6 Visualización final del Dataframe.

Se obtiene un nuevo Dataframe que tendrá todas sus celdas con un valor numérico, dicho valor provendría por defecto o será resultado del relleno que se realiza en base al promedio. Una visualización final del Dataframe se verá así:

Figura 14 Resultado de DataSet promediado

	2014	2015	2016	2017	2018	2019	2020	2021	2022	2023
0	0.000000	0	0.000000	0.000000	0.000000	0.000000	0.000000	0.0	0.0	0.000000
1	0.000000	0	0.000000	0.000000	0.000000	0.000000	0.000000	0.0	0.0	0.000000
2	0.000000	0	0.000000	0.000000	0.000000	0.000000	0.000000	0.0	0.0	0.000000
3	1.666667	0	1.666667	1.666667	1.666667	1.666667	1.666667	0.0	5.0	1.666667
4	2.333333	0	2.333333	2.333333	2.333333	2.333333	2.333333	0.0	7.0	2.333333
...	...	...	...	...	...	...	...	...	...	...
4862	0.000000	0	0.000000	0.000000	0.000000	0.000000	0.000000	0.0	0.0	0.000000
4863	0.000000	0	0.000000	0.000000	0.000000	0.000000	0.000000	0.0	0.0	0.000000
4864	0.000000	0	0.000000	0.000000	0.000000	0.000000	0.000000	0.0	0.0	0.000000
4865	0.000000	0	0.000000	0.000000	0.000000	0.000000	0.000000	0.0	0.0	0.000000
4866	0.000000	0	0.000000	0.000000	0.000000	0.000000	0.000000	0.0	0.0	0.000000

4867 rows × 10 columns

Nota. Imagen de Elaboración Propia

4.1.7 Validación del relleno de NaN.

Guardamos el nuevo Dataframe para su posterior uso en el entrenamiento de la red neuronal (ver Anexo 2).

Figura 15 Guardado del nuevo DataFrame

```
# Guardar el nuevo DataFrame en un archivo Excel
df.to_excel(ruta + 'enero.xlsx', index=False)
```

Nota. Imagen de Elaboración Propia

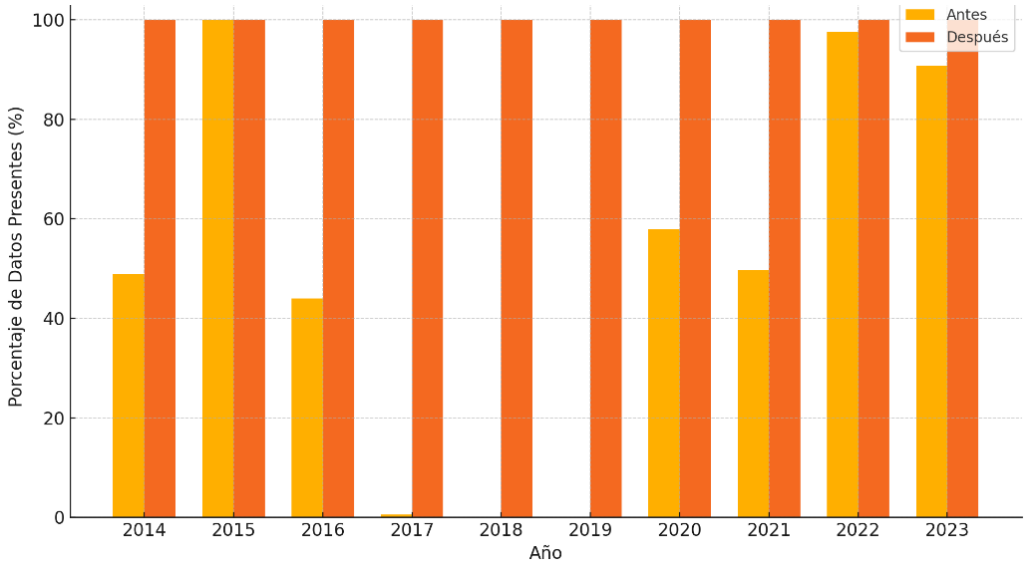
Se imprime nuevamente la cantidad de NaNs para determinar el resultado del relleno por promedio.

Figura 16 Resultado del preprocesamiento

```
Cantidad de NaNs:
        Columna 2014: 0
        Columna 2015: 0
        Columna 2016: 0
        Columna 2017: 0
        Columna 2018: 0
        Columna 2019: 0
        Columna 2020: 0
        Columna 2021: 0
        Columna 2022: 0
        Columna 2023: 0
```

Nota. Imagen de Elaboración Propia

Figura 17 Comparación del porcentaje de datos presentes Antes y Después



Nota. Imagen de Elaboración Propia

Se importaron las librerías necesarias para la importación del set de datos promediado (ver Anexo 3), a través de Google Drive se realiza la lectura de dicho set de datos y se define un nuevo *df* (Dataframe), así mismo se establece el *Datetime* como índice para tener una serie temporal que permita realizar un mejor desarrollo de la división de datos.

Figura 18 Lectura del set de datos

```
from google.colab import drive
import pandas as pd
import numpy as np
import plotly.graph_objs as go
from plotly.subplots import make_subplots

# Montar Google Drive
drive.mount('/gdrive')

# Leer set de datos
ruta = '/gdrive/MyDrive/Colab Notebooks/RNN_LSTM/'
df = pd.read_excel(ruta+'Dataset_Completo_Means_2013_2023_VarTemporal.xlsx', index_col="DateTime")
df.index = pd.to_datetime(df.index)
df
```

Nota. Imagen de Elaboración Propia

El set de datos se puede visualizar de la siguiente manera:

Figura 19 Set de datos

DateTime	Irradiancia
2013-08-01 06:00:00	0.000000
2013-08-01 06:05:00	0.000000
2013-08-01 06:10:00	0.000000
2013-08-01 06:15:00	0.857143
2013-08-01 06:20:00	4.000000
...	...
2023-12-31 18:40:00	0.000000
2023-12-31 18:45:00	0.000000
2023-12-31 18:50:00	0.000000
2023-12-31 18:55:00	0.000000
2023-12-31 19:00:00	0.000000

597385 rows × 1 columns

Nota. Imagen de Elaboración Propia

Se filtra el set de datos en el rango de enero de 2019 a diciembre de 2022, puesto que son los datos que menos intervención de preprocesamiento tuvieron, es decir, son más fieles a los datos que provienen directamente de la estación.

Figura 20 Filtrado del Dataframe

```
# Filtrar el DataFrame para obtener solo los datos del año 2021 hasta 2022
df_filtrado = df.loc['2019-01-01':'2022-12-31']
df_filtrado
```

Nota. Imagen de Elaboración Propia

Una vez se tiene el set de datos preparado se hará uso de la columna de *Irradiancia* ya que el modelo tendrá un solo feature, es decir, será Univariado por sólo se hace uso de una característica como lo es la irradiancia y a su vez multistep ya que se predicen futuros pasos hacia adelante. También se han definido el tamaño de los conjuntos con un 70% para entrenamiento, 15% para prueba y 15% para validación.

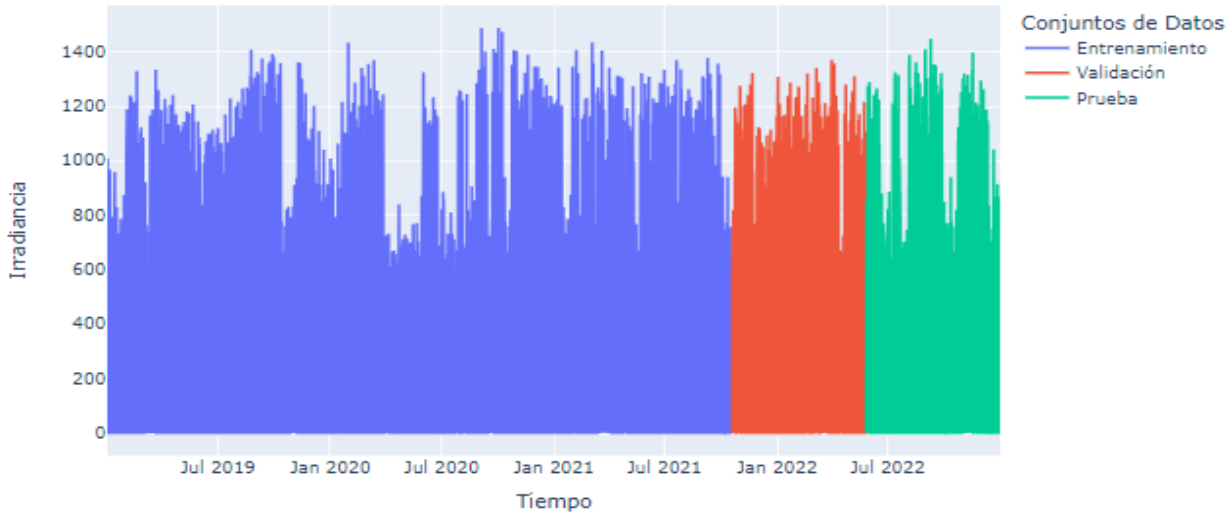
Figura 21 División del set de datos

```
DATOS = df_filtrado['Irradiancia'].values
DATETIME = df_filtrado.index

# Definir el tamaño de los conjuntos
TRAIN_SIZE = 0.70
TEST_SIZE = 0.15
VAL_SIZE = 0.15

# Indices para separar los conjuntos
idx_train = round(len(DATOS) * TRAIN_SIZE)
idx_test = round(len(DATOS) * (TRAIN_SIZE + TEST_SIZE))

datos_train = DATOS[0:idx_train]
datos_val = DATOS[idx_train:idx_test]
datos_test = DATOS[idx_test:]

time_train = DATETIME[0:idx_train]
time_val = DATETIME[idx_train:idx_test]
time_test = DATETIME[idx_test:]

print(f'Tamaño set de entrenamiento: {datos_train.shape}')
print(f'   Tamaño set de validación: {datos_val.shape}')
print(f'       Tamaño set de prueba: {datos_test.shape}')

# Crear la figura
fig = make_subplots()

# Agregar las series de datos
fig.add_trace(go.Scatter(x=time_train, y=datos_train, mode='lines', name='Entrenamiento'))
fig.add_trace(go.Scatter(x=time_val, y=datos_val, mode='lines', name='Validación'))
fig.add_trace(go.Scatter(x=time_test, y=datos_test, mode='lines', name='Prueba'))

# Agregar etiquetas y leyenda
fig.update_layout(
    title="Datos de Irradiancia",
    xaxis_title="Tiempo",
    yaxis_title="Irradiancia",
    legend_title="Conjuntos de Datos"
)

# Mostrar la figura
fig.show()
```

Nota. Imagen de elaboración propia

La división del set de datos se visualizaría de la siguiente manera:

Figura 22 División del set de datos



Nota. Imagen de Elaboración Propia

La división de datos para entrenamiento, validación y prueba se realizó respetando la coherencia temporal, evitando el uso de técnicas de aleatorización. Dado que las series temporales requieren de un orden cronológico, la aleatorización de los datos podría romper la secuencia temporal, introduciendo sesgos y afectando la capacidad del modelo para aprender patrones. Además, los datos ingresados al modelo son completos, sin datos faltantes, lo que asegura continuidad y una temporalidad secuencial para una predicción precisa y coherente.

Los datos para ser ingresados se normalizan en el modelo, su escala se encuentra entre -1 y 1 haciendo uso de la librería *MinMaxScaler*, este proceso se realiza con el fin de acelerar el proceso de convergencia y mejorar la precisión al tener una escala más pequeña y uniforme.

Figura 23 Normalización de los datos

```
# Escalemos los datos entre -1 y 1
from sklearn.preprocessing import MinMaxScaler

scaler = MinMaxScaler(feature_range=(-1,1))
datos_train_s = scaler.fit_transform(datos_train.reshape(-1,1))
datos_val_s = scaler.transform(datos_val.reshape(-1,1))
datos_test_s = scaler.transform(datos_test.reshape(-1,1))
```

Nota. Imagen de Elaboración Propia

Posteriormente, se realiza el ingreso de los datos (*INPUT*) definiendo el número de días pasados que entran al modelo y se calcula la longitud de la secuencia de entrada, es decir, la frecuencia de un día a partir de las 6 de la mañana a 7 de la noche cada cinco minutos es igual a 156 muestras, a su vez, estos se multiplican con el número de días a predecir (*OUTPUT*) para cada uno de los sets correspondientes a la división de datos.

Se inicializan seis listas vacías: *X_train, Y_train, X_val, Y_val, X_test, Y_test* que contendrán las secuencias de entrada (*X*) y las correspondientes salidas (*Y*). En cada *for*, se recorren los datos de entrenamiento, validación y prueba para crear subconjuntos (ventanas de tiempo) con una longitud de secuencia de entrada *LONG_SEC* y secuencia de salida *N_STEPS.*

A partir de esto, se convierten las listas a arrays para facilitar la operación de grandes volúmenes de datos de manera eficiente. Finalmente se imprime el tamaño final de cada uno de los conjuntos indicando las dimensiones tanto de las entradas como de las salidas.

Figura 24 Subconjunto de datos

```
Dias_pasados = 7 # N
LONG_SEC = 156 * Dias_pasados # toma N días = INPUT

# Definir el número de pasos a predecir
Dias_futuros = 1 # X
N_STEPS = 156 * Dias_futuros  # X días hacia adelante = OUTPUT

X_train, Y_train = [], []
for i in range(len(datos_train_s) - LONG_SEC - N_STEPS):
    X_train.append(datos_train_s[i:i+LONG_SEC])
    Y_train.append(datos_train_s[i+LONG_SEC:i+LONG_SEC+N_STEPS])

X_val, Y_val = [], []
for i in range(len(datos_val_s) - LONG_SEC - N_STEPS):
    X_val.append(datos_val_s[i:i+LONG_SEC])
    Y_val.append(datos_val_s[i+LONG_SEC:i+LONG_SEC+N_STEPS])

X_test, Y_test = [], []
for i in range(len(datos_test_s) - LONG_SEC - N_STEPS):
    X_test.append(datos_test_s[i:i+LONG_SEC])
    Y_test.append(datos_test_s[i+LONG_SEC:i+LONG_SEC+N_STEPS])

X_train, Y_train = np.array(X_train), np.array(Y_train)
X_train = np.reshape(X_train, (X_train.shape[0], X_train.shape[1], 1))

X_val, Y_val = np.array(X_val), np.array(Y_val)
X_val = np.reshape(X_val, (X_val.shape[0], X_val.shape[1], 1))

X_test, Y_test = np.array(X_test), np.array(Y_test)
X_test = np.reshape(X_test, (X_test.shape[0], X_test.shape[1], 1))

print(f'Tamaños de entrada (BATCH x LONG_SEC x FEATURES) y de salida (BATCH x N_STEPS x FEATURES):')
print(f'Tamaño set de entrenamiento X_train: {X_train.shape}, Y_train: {Y_train.shape}')
print(f'     Tamaño set de validación X_val: {X_val.shape},     Y_val: {Y_train.shape}')
print(f'        Tamaño set de prueba X_test: {X_test.shape},   Y_test: {Y_train.shape}')
```

Nota. Imagen de Elaboración Propia

Figura 25 Tamaño de entrada y salida

Nota. Imagen de Elaboración Propia

4.2 Implementación del modelo de red neuronal LSTM para el pronóstico de la irradiancia.

En la subsección 4.2.1, se describen las diferentes configuraciones utilizadas para entrenar el modelo LSTM, variando en el número de capas, unidades y horizontes de predicción, para evaluar su impacto en el rendimiento. En la subsección 4.2.2, se detalla la selección del modelo final, destacando la arquitectura LSTM con 20 unidades y activación tanh, el cual tiene la variable de irradiancia como entrada y salida del modelo, junto a esto se apropian técnicas como predicción multistep y evaluación de métricas como R^2 y MAE.

4.2.1 Configuraciones de modelos LSTM

En la Tabla 1 se resume el proceso experimental que se ha desarrollado para entrenar y evaluar distintos modelos LSTM con el fin de encontrar el más adecuado para el pronóstico de la irradiancia. Los modelos 1 hasta el 16 tienen como variable de entrada la irradiancia, los modelos 17 hasta 22 incluyen variables categóricas como el periodo del día, por últimos están los modelos 23 y 24 que comprende variables temporales cíclicas. (Ver Anexo 4)

Se evaluaron distintas configuraciones, cada uno contiene distintas capas con el fin de determinar cómo la complejidad de la red afecta el rendimiento del modelo. Los horizontes de predicción varían desde predicciones a corto plazo de 30 minutos hasta predicciones de largo plazo como 1 día y 2 días.

Para los modelos la función de activación es *tanh*, debido a que es la más adecuada para garantizar que las salidas de las neuronas estén en el rango de -1 y 1, esto permite un control sobre los valores entre capas. Los tamaños de lote utilizados fueron de 64 y 128, además de afectar en la convergencia del modelo, estos valores definen cuántas muestras se entrenan en cada iteración del entrenamiento. El número de épocas en cada entrenamiento es igual, sin embargo, el tiempo de entrenamiento se encuentra en función del *Callback Early Stop Patience,* y *Reduce LR* los cuales actúan cuando el modelo deja de mejorar en sus métricas.

Tabla 2 Configuración de modelos LSTM

Nombre del Modelo	Características de entrada	Capas LSTM	Unidades por Capa	Horizonte de Predicción	Activación	Tamaño del Lote (batch)	Épocas	Optimización	Early Stop patience	Reduce LR
LSTM 1	Irradiancia	[1]	5	1 día	tanh	64	100	Adam	10	patience [4] min = 1e^-10
LSTM 2			20							
LSTM 3			50							
LSTM 4			128							
LSTM 5		[2]	[20, 20]							
LSTM 6			[40, 70]							
LSTM 7		[3]	[50, 50,50]							
LSTM 8			[20, 20, 20]							
LSTM 9		[4]	[20, 20, 20, 20]							
LSTM 10		[1]	20	30 min						
LSTM 11			50							
LSTM 12			128							
LSTM 13		[2]	[40, 70]							
LSTM 14		[1]	20	2 días		128			20	
LSTM 15			70							
LSTM 16			128							
LSTM 17	Irradiancia Periodo del Dia: Mañana [0] Medio Dia [1] Tarde [2]	1	50	1 día		64			10	
LSTM 18			300							
LSTM 19		2	[40, 40]							
LSTM 20	Irradiancia Periodo del Dia OneHot: Mañana [1,0,0] Medio Dia [0,1,0] Tarde [0,0,1]	1	20						20	
LSTM 21			50							
LSTM 22		2	[40, 70]							
LSTM 23	Irradiancia Variable Temporal Cíclica	1	20						10	
LSTM 24		2	[50, 50]			128				

4.2.2 Selección del modelo

Para la implementación de un modelo de predicción de series temporales basado en redes neuronales recurrentes (*RNN*) de tipo LSTM tendrá la configuración descrita en la presente sección, partiendo de este, se harán distintas configuraciones en la búsqueda del estudio porcentual de sus métricas de desempeño, y encontrar su mejor configuración.

El modelo está construido utilizando Keras, una biblioteca para el desarrollo de redes neuronales que se integra con TensorFlow. Se ha empleado una sola capa LSTM con 20 unidades, captando los patrones a lo largo del tiempo en los datos de entrada en función de las secuencias de longitud variable (*LONG_SEC*).

La activación seleccionada es *tanh*, una función que se ajusta bien a este tipo de redes recurrentes, de esta manera puede permitirse la oscilación entre los valores de -1 y 1, lo que facilita estabilidad en el aprendizaje. La capa recibe como entrada un conjunto de secuencias de longitud *LONG_SEC*, donde cada paso de tiempo tiene una característica unidimensional.

La capa de salida es una capa densa que contiene un número de neuronas igual al horizonte de predicción *N_STEPS*. El propósito de esta capa es generar una secuencia de predicciones de longitud igual al horizonte, estos corresponden a la cantidad de pasos que se desean predecir en función de los datos pasados.

Figura 26 Modelo LSTM 1 Capa de 20 Neuronas

```
# Creación del modelo
from keras.models import Sequential
from keras.layers import LSTM, Dense
from keras.optimizers import Adam
from keras.callbacks import EarlyStopping, ModelCheckpoint, ReduceLROnPlateau
import tensorflow as tf
from time import time

# Semilla de los generadores aleatorios
SEED = 42
tf.random.set_seed(SEED)
np.random.seed(SEED)

# Definir el modelo 1 capa
N_UNITS = 20
modelo = Sequential()
modelo.add(LSTM(N_UNITS, activation='tanh', input_shape=(LONG_SEC,1)))
modelo.add(Dense(N_STEPS))
# Capa de salida con longitud igual al horizonte de predicción

# Resumen del modelo
modelo.summary()
```

Nota. Imagen de elaboración propia

Puesto que se trata de una regresión no se especifica ninguna función de activación en la capa densa, puesto que esto permite que las salidas sean valores continuos. Además, con el objetivo de asegurar la reproducibilidad de los resultados, se establecen semillas en los generadores aleatorios tanto de *Numpy* como de *TensorFlow*.

Las funciones *tf.random.set_seed(SEED)* y *np.random.seed(SEED)* garantizan que al ejecutar el código varias veces se obtengan los mismos resultados, eliminando la posibilidad de una variación no deseada en el cálculo de los pesos o en el orden de los datos. El modelo propuesto, en su forma más básica marca un punto de referencia, lo que permite evaluaciones comparativas posteriores con modelos más complejos en términos de series temporales.

El proceso de compilación y entrenamiento del modelo de predicción utiliza técnicas para optimizar el rendimiento y prevenir el sobreajuste. Se detallan los mecanismos empleados para mejorar la eficiencia del entrenamiento y garantizar la mejor versión del modelo en cada configuración.

El callback *EarlyStopping* monitorea la pérdida en el conjunto de validación (val_loss). Si no se consigue una mejora después de 10 épocas (especificado por *patience=10*), el entrenamiento se detendrá de manera anticipada, evitando el desgaste computacional y el exceso en tiempo de entrenamiento. Además, *restore_best_weights=True* asegura que, al finalizar el entrenamiento, se restauren los pesos del modelo donde se alcanza la mejor validación.

Para guardar el mejor modelo durante el proceso de entrenamiento se hace uso de *ModelCheckpoint*, esto se realiza según el valor mínimo de la pérdida de validación (val_loss). Dicho modelo se almacena en la ruta especificada como '*ruta+best_mode_lstm.kerasl*'. El parámetro *save_best_only=True* asegura que solo se guarde el modelo cuando se detecte una mejora en el monitoreo de *val_loss*.

Para reducir la tasa de aprendizaje cuando no se tiene una mejora en la pérdida de validación durante 4 épocas (*patience=4*), multiplicando la tasa de aprendizaje por un factor de 0,1. La tasa de aprendizaje establece un valor mínimo (*min_lr*) evitando que disminuya por completo. Esta técnica ayuda a ajustar dinámicamente el ritmo de aprendizaje del modelo.

Figura 27 Callbacks del modelo LSTM

```
# Callbacks
early_stopping = EarlyStopping(monitor='val_loss', patience=10, restore_best_weights=True)
model_checkpoint = ModelCheckpoint(ruta+'300U1L64B.keras', save_best_only=True, monitor=
'val_loss', mode='min')
reduce_lr = ReduceLROnPlateau(monitor='val_loss', factor=0.1, patience=4, min_lr=0.00000000001)
```

Nota. Imagen de elaboración propia

Con el fin de visualizar la evolución de la pérdida en el proceso de entrenamiento, se toma en cuenta tanto el conjunto de entrenamiento como en el conjunto de validación. Haciendo uso de la biblioteca *Plotly*, que permite graficar el desempeño del modelo y detectar posibles problemas como el *underfitting* o el *overfitting.*

El objeto *historial*, generado durante el entrenamiento del modelo almacena los valores de la función de pérdida (error cuadrático medio, MSE) en cada época, tanto para el conjunto de entrenamiento como para el conjunto de validación. Los valores de pérdida se extraen los diccionarios *historial.history ['loss']* (para el conjunto de entrenamiento) e *historial.history ['val_loss']* (para el conjunto de validación). La pérdida en el conjunto de entrenamiento se añade con la función *go.Scatter(),* lo que genera una gráfica continua con base en las épocas transcurridas. Los datos del eje *x* corresponden a las épocas *list(range(len(los_train)))*, mientras que el eje *y* representa los valores de la pérdida en entrenamiento. De la misma manera, se añade una segunda traza para representar la pérdida en el conjunto de validación.

Figura 28 Pérdida en el conjunto de entrenamiento y validación

```python
import plotly.graph_objs as go

# Accediendo a los datos de pérdida y validación del entrenamiento
loss_train = historial.history['loss']
loss_val = historial.history['val_loss']

# Crear la gráfica de pérdida con Plotly
fig = go.Figure()

# Añadir la línea de pérdida en entrenamiento
fig.add_trace(go.Scatter(x=list(range(len(loss_train))), y=loss_train, mode='lines', name='Pérdida en entrenamiento'))

# Añadir la línea de pérdida en validación
fig.add_trace(go.Scatter(x=list(range(len(loss_val))), y=loss_val, mode='lines', name='Pérdida en validación'))

# Personalizar el layout
fig.update_layout(
    title='Pérdida durante el entrenamiento y la validación',
    xaxis_title='Épocas',
    yaxis_title='Pérdida (MSE)',
    legend_title='Datos'
)

fig.show()
```

Nota. Imagen de elaboración propia

La interpretación del gráfico de pérdidas es esencial para evaluar el rendimiento y la capacidad de generalización del modelo. El overfitting ocurre cuando el modelo se ajusta demasiado a los datos de entrenamiento, esto deteriora la capacidad de generar nuevos datos. Por otro lado, el underfitting ocurre cuando el modelo es demasiado simple o no ha aprendido suficiente de los datos.

El enfoque principal es predecir múltiples pasos hacia adelante (multistep forecasting) y calcular métricas de error para comparar los valores predichos con los valores reales. Con *y_pred=modelo.predict(X_test)* se utiliza el modelo entrenado para hacer predicciones sobre el conjunto de prueba *X_test*. Las predicciones *y_pred* corresponden al horizonte de predicción con múltiples pasos hacia adelante.

Los valores reales *Y_test* como las predicciones del modelo *y_pred* son matrices de múltiples dimensiones debido al formato en las secuencias utilizado para la predicción. Dichos datos se “aplanan”, se convierte en una sola columna para facilitar la transformación inversa de la normalización y se calculan las métricas.

Después de la predicción, es necesario invertir la transformación de los datos a su escala original. Utilizando el método *inverse_transform* del objeto *scaler* se aplica la trasformación inversa tanto a los valores reales (*Y_test_flat*) como a los valores predichos (*y_pred_flat*).

Los datos se reestructuran nuevamente en su forma original de múltiples pasos (multistep), es decir, una matriz donde cada fila representa una secuencia de predicción y cada columna representa un paso hacia adelante en el horizonte de predicción.

Para evaluar el rendimiento del modelo, se emplea una métrica clave: el coeficiente de determinación (R^2). Esta métrica mide qué tan bien se ajusta el modelo a los datos observados, donde un valor cercano a 1 indica un ajuste excelente. Sin embargo, en este caso, se espera que el valor de R^2 se sitúe alrededor del 40%. Esta expectativa se debe a que el modelo tiende a generalizar sus predicciones y la variabilidad diaria influye significativamente en el ajuste entre las predicciones y los valores reales.

Figura 29 Predicción y métricas de desempeño

```
from sklearn.metrics import
 mean_squared_error, mean_absolute_error, r2_score

# Aplanar los datos antes de aplicar la transformación inversa
Y_test_flat = Y_test.reshape(-1, 1)
y_pred_flat = y_pred.reshape(-1, 1)

# Invierte la escala de los datos predichos y de prueba
y_test_inv = scaler.inverse_transform(Y_test_flat)
y_pred_inv = scaler.inverse_transform(y_pred_flat)

# Reestructura los datos invertidos a su forma original
y_test_inv = y_test_inv.reshape(Y_test.shape)
y_pred_inv = y_pred_inv.reshape(y_pred.shape)

# Cálculo de Métricas (por ejemplo, para el primer paso de predicción)
mse = mean_squared_error(y_test_inv[:, 0], y_pred_inv[:, 0])
mae = mean_absolute_error(y_test_inv[:, 0], y_pred_inv[:, 0])
r2 = r2_score(y_test_inv[:, 0], y_pred_inv[:, 0])

print(f"R^2: {r2}")
```

Nota. Imagen de elaboración propia

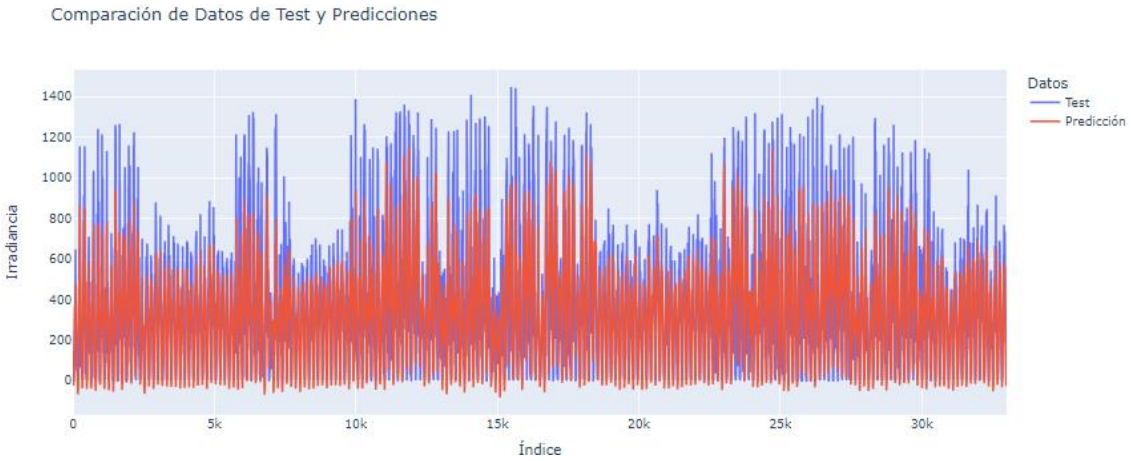
Para visualizar gráficamente la comparación entre los datos reales de prueba y las predicciones generadas por el modelo, se hace uso de la librería *Plotly* con el módulo *graph_objects*, creando un gráfico interactivo que permite evaluar visualmente el rendimiento del modelo en la predicción en el conjunto de prueba.

Si las líneas de predicciones y de datos de pruebas están cercanas entre sí, indica que el modelo está prediciendo correctamente los valores de irradiancia. Si existe una diferencia significativa entre las líneas (test y predicción), significa que el modelo no está capturando adecuadamente los patrones de los datos.

Figura 30 Figura Plotly

```
# Crea una figura de Plotly
import plotly.graph_objects as go

fig = go.Figure()

# Añade la línea de datos de prueba
fig.add_trace(go.Scatter(x=list(range(len(y_test_inv[:, 0]))), y=y_test_inv[:, 0].flatten(), mode=
'lines', name='Test'))

# Añade la línea de predicciones
fig.add_trace(go.Scatter(x=list(range(len(y_pred_inv[:, 0]))), y=y_pred_inv[:, 0].flatten(), mode=
'lines', name='Predicción'))

# Personaliza el layout
fig.update_layout(title='Comparación de Datos de Test y Predicciones',
                  xaxis_title='Índice',
                  yaxis_title='Irradiancia',
                  legend_title='Datos')

fig.show()
```

Figura 31 Gráfica test vs predicción



Nota. Imagen de elaboración propia

Nota. Imagen de elaboración propia

La predicción de la irradiancia solar para un periodo futuro se evalúa en calidad de las predicciones en relación con los datos reales. Se asegura que el dataframe *df* está en formato datetime puesto que se trata de trabajar con series temporales, *ultimo_X* toma los últimos datos del conjunto de prueba *X_test*, serán usados como punto de partida para las predicciones futuras.

Para la transformación de los últimos datos de prueba de vuelta a su escala original se hace uso de *scaler*. Además, *eje_x_datos* filtra los tiempos entre las 6:00 y las 19:00 horas, eliminando datos irrelevantes para la irradiancia solar puesto que se encuentran en un periodo de noche.

Figura 32 Toma de últimos datos de *df*

```python
import numpy as np
import pandas as pd
from sklearn.metrics import mean_absolute_error, r2_score
import plotly.graph_objs as go

# Asegúrate de que df tiene el índice en formato datetime
df.index = pd.to_datetime(df.index)

# Últimos puntos de X_test para comenzar la predicción
ultimo_X = X_test[-1].reshape(1, LONG_SEC, 1)

# Últimos puntos del X_test para graficar (solo la primera variable para ilustrar)
ultimos_datos_test = scaler.inverse_transform(X_test[-1][:, 0].reshape(-1, 1))

# Eje x para el gráfico, filtrando solo las horas de 06:00 a 19:00
eje_x_datos = time_test[-LONG_SEC:]
eje_x_datos = [x for x in eje_x_datos if 6 <= x.hour < 19]
# Ajustar el tamaño de los datos
ultimos_datos_test = ultimos_datos_test[-len(eje_x_datos):]
```

Nota. Imagen de elaboración propia

Se generan predicciones a futuro durante un número de días definidos por *Dias_futuros*, para cada iteración el modelo predice un conjunto de valores (*pred_nueva*) y luego actualiza el conjunto de entrada (*ultimo_X*) para incluir las predicciones que se acumulan en la lista *predicciones_futuras*.

Las *predicciones_futuras* se transforma de vuelta a su escala original usando *scaler*. Se crea un eje de tiempo (*eje_x_pred*) para las predicciones futuras, limitando solo dentro del rango de las 6:00 y las 19:00 horas.

Figura 33 Predicciones futuras

```
predicciones_futuras = []
dias_pred = Dias_futuros
num_predicciones = dias_pred * 156 # (Dias_futuros * 156 datos por día)

for i in range(int(num_predicciones // N_STEPS)):
    # Hacer la predicción usando el último punto disponible
    pred_nueva = modelo.predict(ultimo_X)
    predicciones_futuras.extend(pred_nueva[0])  # Agrega todos los pasos predichos

    # Actualizar el último_X para incluir la nueva predicción
    ultimo_X = np.roll(ultimo_X, -N_STEPS, axis=1)
    ultimo_X[0, -N_STEPS:, 0] = pred_nueva[0].flatten()
# Aplanar el array antes de asignarlo

# Transformar las predicciones de vuelta a la escala original
predicciones_futuras = np.array(predicciones_futuras).reshape(-1, 1)
predicciones_futuras = scaler.inverse_transform(predicciones_futuras)

# Crear fechas para las predicciones futuras, excluyendo las horas de 19:00 a 06:00
fecha_actual = pd.Timestamp('2023-01-01 06:00:00')
eje_x_pred = []
while len(eje_x_pred) < len(predicciones_futuras):
    if 6 <= fecha_actual.hour < 19:
        eje_x_pred.append(fecha_actual)
    fecha_actual += pd.Timedelta(minutes=5)
```

Nota. Imagen de elaboración propia

Se obtienen valores reales de irradiancia para las fechas predichas y proporcionar una evaluación cuantitativa se calculan dos métricas clave vistas anteriormente (*MAE* y R^2). Los valores reales de irradiancia para las fechas correspondientes a las predicciones (*eje_x_pred*) se extraen y se utilizan para comparar las predicciones generadas por el modelo.

El gráfico interactivo que se genera tiene tres componentes principales: los últimos puntos observados de prueba donde muestra los valores reales más recientes (*LONG_SEC*), las predicciones futuras la cual representa las predicciones generadas por el modelo para el futuro y, por último, los valores reales tomados de la estación para realizar la comparación.

Figura 34 Métricas y comparación

```
# Asegurar que los valores reales existen para las fechas predichas
valores_reales = df.loc[eje_x_pred, 'Irradiancia'].values

# Calcular las métricas de evaluación
mse = mean_squared_error(valores_reales, predicciones_futuras)
mae = mean_absolute_error(valores_reales, predicciones_futuras)
r2 = r2_score(valores_reales, predicciones_futuras)

print(f"R^2: {r2}")

# Graficar los datos con Plotly
fig = go.Figure()

# Añade la línea de últimos puntos de X_test
fig.add_trace(go.Scatter(x=eje_x_datos, y=ultimos_datos_test.flatten(), mode='lines', name=
'Últimos puntos de X_test'))

# Añade la línea de predicciones futuras
fig.add_trace(go.Scatter(x=eje_x_pred, y=predicciones_futuras.flatten(), mode='lines', name=
'Predicciones futuras', line=dict(color='red')))

# Añade la línea de valores reales
fig.add_trace(go.Scatter(x=eje_x_pred, y=valores_reales, mode='lines', name='Valores reales'
, line=dict(color='rosybrown')))

# Personaliza el layout
fig.update_layout(
    title='Comparación de Predicciones Futuras con Valores Reales',
    xaxis_title='Tiempo',
    yaxis_title='Valor',
    legend_title='Datos'
)

fig.show()
```

Nota. Imagen de elaboración propia

En resumen, los principales Hiperparámetros se muestran en la Tabla 3.

Tabla 3 Principales Hiperparámetros del modelo

Modelo LSTM 300 Unidades 1 Capa - Univariado												
Set de Datos %			**Tamaño Set**			**INPUT**			**OUTPUT**			**Horizonte**
Train.	Val.	Test	Train.	Val.	Test	X_train	X_val	X_test	Y_train	Y_val	Y_test	
70%	15%	15%	232732	49360	49360	231732, 1099, 1	48670, 1099, 1	48670, 1099, 1	231732, 157, 1	48670, 157, 1	48670, 157, 1	157

Nro de Capas / Nro de Neuronas		**Paciencia Callbacks**		**Optimizador Adam**	**Entrenamiento**	
#Nodos Layer1	Capa Densa	EarlyStopping	Reduce_lr	Learning rate	Epochs	Batch_Size
300	Si	10	4	0,0001	100	64

4.3 Validar el modelo desarrollado por medio del error entre datos reales y estimados.

En esta sección se presentan los resultados de predicción de irradiancia solar generados por el modelo. Se discutirán las subsecciones 4.3.1 y 4.3.2, en las que se comparan las predicciones de los modelos entrenados con datos segmentados, evaluando cómo esta segmentación impacta la capacidad del modelo para generalizar patrones de irradiancia. En la subsección 4.3.3, se explorará la predicción multitemporal, destacando cómo el modelo puede pronosticar la irradiancia hasta 14 días en el futuro y su desempeño sobre este horizonte de predicción. La sección 4.3.4 se realiza una comparativa con otro modelo basa en red neuronal recurrente GRU. Por último, en la subsección 4.3.5, se presenta el análisis de resultados del modelo en los pronósticos de irradiancia, destacando las ventajas que ofrece en la gestión de la energía solar y la sostenibilidad institucional, particularmente para la Universidad CESMAG.

4.3.1 Predicciones vs datos reales (Entrenamiento por meses).

El modelo ha tardado 39 épocas en aprender sobre los datos, con un tiempo de 1 hora y 46 minutos, teniendo como resultado el resumen del modelo (Figura 35) y las curvas de aprendizaje (Figura 36) de las cuales se puede determinar que el modelo ha tenido un aprendizaje óptimo sin tener unerfitting u overfitting.

Figura 35 Resumen del modelo

Layer (type)	Output Shape	Param #
lstm (LSTM)	(None, 300)	362,400
dense (Dense)	(None, 156)	46,956

Nota. Imagen de elaboración propia

El entrenamiento del modelo de predicción de irradiancia se realizó dividiendo los datos en meses, para la predicción del período comprendido entre octubre de 2023 y septiembre de 2024. Esta segmentación por meses permitió analizar de manera más precisa cómo el modelo se comporta bajo diferentes condiciones climáticas a lo largo del año. A pesar de que el modelo tiene la capacidad de predecir sin considerar de manera explícita el mes o el día, esta diferenciación mensual se empleó con el objetivo de comprender mejor los patrones estacionales

y evaluar la capacidad del modelo para generalizar entre períodos con diferentes características atmosféricas.

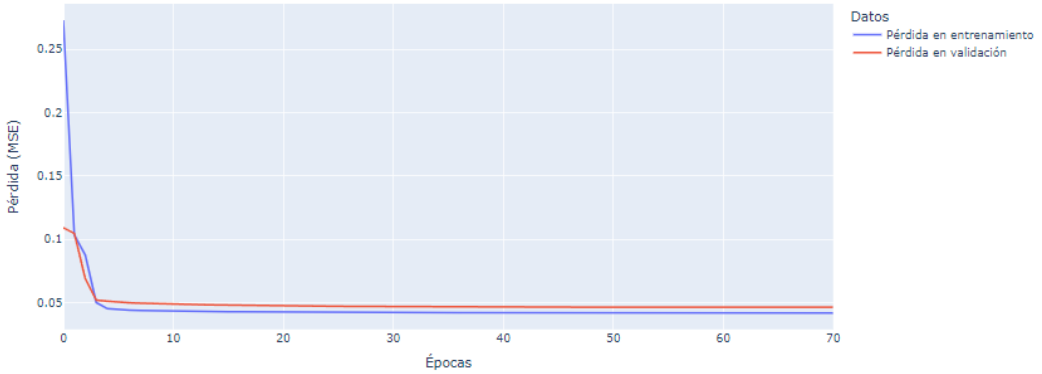
Uno de los enfoques distintivos de este trabajo, en comparación con los modelos encontrados en la literatura, es la capacidad de realizar predicciones a largo plazo. Mientras que la mayoría de los estudios se enfocan en predicciones de minutos, una hora, dos horas o hasta tres horas hacia adelante, nuestro modelo se diseñó para predecir hasta 8 días en el futuro. Esto representa una ventaja significativa en la planificación de la gestión de energía solar, ya que permite una anticipación mayor para optimizar la operación de sistemas fotovoltaicos y otros sistemas dependientes de la irradiancia. Sin embargo, se observó que más allá del octavo día, la precisión de las predicciones comienza a deteriorarse, lo que implica que los errores aumentan y el modelo no logra captar con exactitud las tendencias a largo plazo.

Para evaluar el rendimiento del modelo, se realizaron comparaciones detalladas entre las predicciones generadas y los valores reales observados. Estas comparaciones se realizaron utilizando tanto datos de test como datos de predicciones futuras, abarcando diferentes períodos del año. A continuación, se presentan tres gráficos clave que resumen el comportamiento del modelo durante el proceso de entrenamiento y en las pruebas de predicción:

La Figura 36 muestra la evolución de la pérdida del modelo tanto en el conjunto de entrenamiento como en el de validación. Se puede observar cómo el modelo va ajustándose a los datos con el paso de las épocas de entrenamiento, y cómo se estabiliza en un valor de pérdida bajo, indicando que ha aprendido adecuadamente los patrones de los datos de irradiancia. El comportamiento de la curva de validación sugiere un entrenamiento óptimo, ya que no se

evidencia ni overfitting ni underfitting, lo que significa que el modelo generaliza bien en los datos de validación.

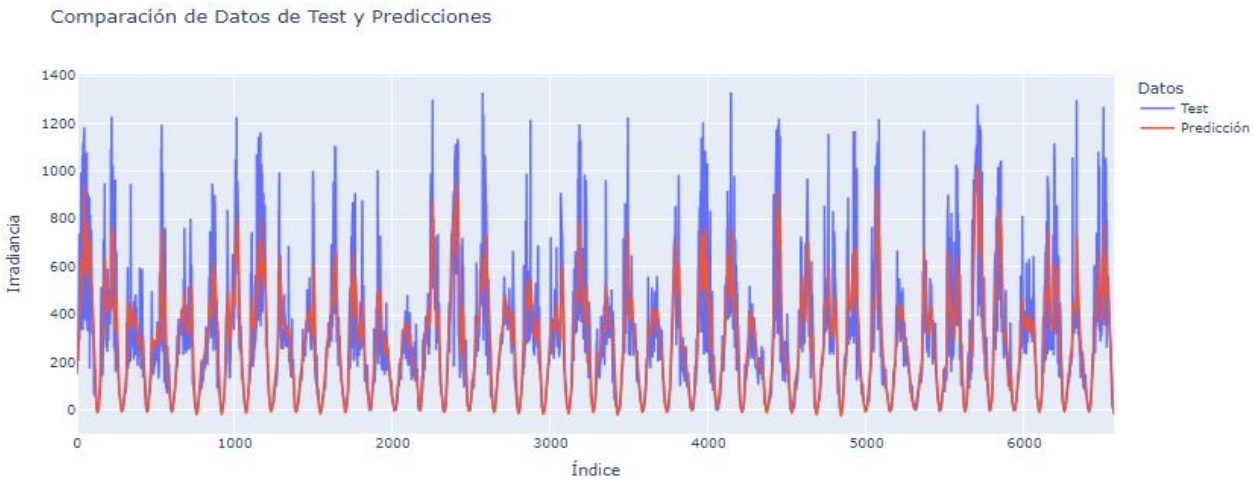
Figura 36 Curvas de aprendizaje



Nota. Imagen de elaboración propia

La Figura 37 ilustra la comparación directa entre los datos reales de irradiancia del conjunto de test y las predicciones generadas por el modelo. Los datos de test corresponden a un conjunto de datos que no fue utilizado durante el entrenamiento, lo que permite medir el desempeño del modelo en condiciones desconocidas. El coeficiente de determinación (R^2) se utilizó como métrica principal para evaluar qué tan bien el modelo logra ajustarse a los datos reales. Un valor de R^2 cercano a 1 indica un buen ajuste, mientras que valores más bajos reflejan una menor capacidad del modelo para capturar las variaciones en los datos. Como se discutirá más adelante, el valor del R^2 varía según el mes y las condiciones climáticas asociadas a dicho mes, siendo más bajo en los meses con mayor variabilidad atmosférica, como noviembre y diciembre.

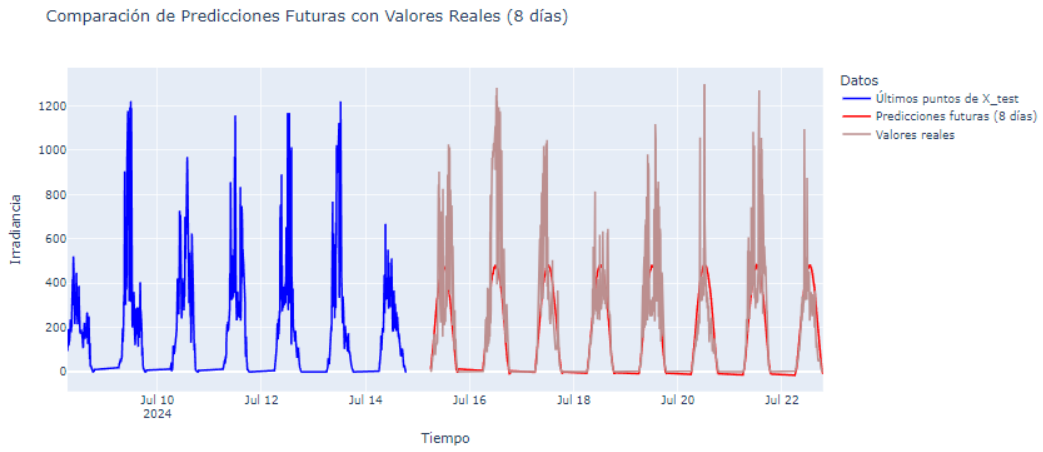
Figura 37 Test vs Predicción



Nota. Imagen de elaboración propia

En la Figura 38 se observa la comparación de las predicciones futuras con los valores reales, hasta 8 días adelante. Este gráfico representa la capacidad del modelo para predecir la irradiancia futura, cubriendo un rango de hasta 8 días en el futuro. Las predicciones futuras son esenciales para aplicaciones prácticas en sistemas de energía solar, donde se requiere prever con suficiente antelación los posibles cambios en la irradiancia para optimizar la captación de energía. Si bien el modelo muestra un desempeño adecuado para las primeras predicciones dentro de este rango, se nota una disminución en la precisión a medida que las predicciones se alejan en el tiempo, especialmente después del octavo día, donde el error de predicción aumenta y la capacidad del modelo para captar patrones a largo plazo disminuye considerablemente.

Figura 38 Test vs Predicción



Nota. Imagen de elaboración propia

El comportamiento del coeficiente de determinación R^2, que evalúa la calidad de las predicciones realizadas por el modelo, se detalla en la Tabla 2. La tabla muestra los valores de R^2 desglosados por semanas para cada mes desde octubre de 2023 hasta septiembre de 2024. Estos resultados permiten analizar el rendimiento del modelo de manera más precisa durante el período de estudio. Como se puede observar, el valor de R^2 fluctúa a lo largo de las semanas y meses, lo que refleja las variaciones estacionales y meteorológicas que impactan el comportamiento de la irradiancia.

Por ejemplo, en enero de 2024, el coeficiente alcanza su valor más alto en la Semana 2 con 0.57, mientras que, en otros meses como julio de 2024, los valores de R^2 presentan una mayor estabilidad entre las semanas, lo que puede estar relacionado con condiciones climáticas más uniformes durante ese período. Este comportamiento sugiere que el modelo es más preciso en ciertos momentos del año, pero en otros meses, especialmente en condiciones de alta variabilidad climática, la precisión disminuye, tal como se observa en marzo de 2024, donde el R^2 de la Semana 2 alcanza su valor más bajo, 0.25.

Es importante destacar que el valor relativamente bajo del coeficiente de determinación en ciertos periodos no debe interpretarse como una falla del modelo, sino como una característica propia de la naturaleza estadística de la arquitectura LSTM. Al ser un modelo de predicción estadístico, su desempeño está basado en el promedio de los datos históricos y, por lo tanto, está diseñado para generalizar sus predicciones sin ajustarse específicamente a las particularidades de

cada día. Un modelo que ajuste demasiado bien a los datos de entrenamiento podría caer en sobreajuste, lo que llevaría a predicciones inexactas cuando se enfrenta a nuevos datos. En contraste, el modelo ha aprendido a capturar patrones generales en los datos sin sobre ajustarse, lo cual es deseable para poder predecir adecuadamente bajo diferentes condiciones.

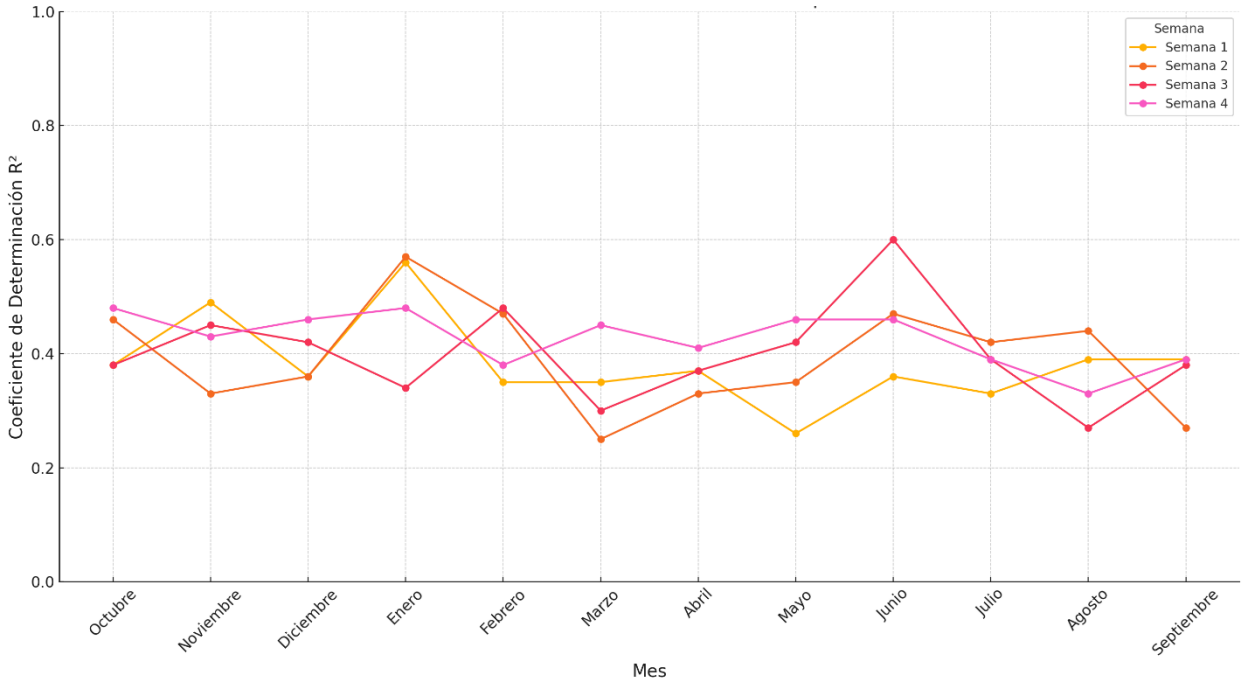
El comportamiento del modelo sugiere que no se ha sobre ajustado a los datos de entrenamiento, lo que indica que su entrenamiento ha sido adecuado. Los resultados presentados en la Tabla 2 reflejan cierta capacidad del modelo para adaptarse a las variaciones estacionales y meteorológicas de la región. Sin embargo, aunque el coeficiente de determinación no alcanza valores cercanos a 1, el modelo logra un equilibrio razonable entre precisión y generalización, lo que puede ofrecer predicciones útiles de irradiancia a largo plazo, aunque con limitaciones.

Tabla 4 Coeficiente de Determinación 2023-2024

Espacio Temporal		Coeficiente de determinación R^2			
Mes	Año	Semana 1	Semana 2	Semana 3	Semana 4
Octubre	2023	0,38	0,46	0,38	0,48
Noviembre	2023	0,49	0,33	0,45	0,43
Diciembre	2023	0,36	0,36	0,42	0,46
Enero	2024	0,56	0,57	0,34	0,48
Febrero	2024	0,35	0,47	0,48	0,38
Marzo	2024	0,35	0,25	0,3	0,45
Abril	2024	0,37	0,33	0,37	0,41
Mayo	2024	0,26	0,35	0,42	0,46
Junio	2024	0,36	0,47	0,6	0,46
Julio	2024	0,33	0,42	0,39	0,39
Agosto	2024	0,39	0,44	0,27	0,33
Septiembre	2024	0,39	0,27	0,38	0,39

La Figura 39 muestra la evolución del Coeficiente de Determinación R^2 a lo largo de los meses entre octubre de 2023 y septiembre de 2024, segmentada por semanas dentro de cada mes. En la gráfica, cada línea de color representa una semana específica (de la Semana 1 a la Semana 4), permitiendo observar cómo fluctúa la precisión del modelo en función del mes y la semana. Esto ofrece una visión detallada de las variaciones estacionales y semanales en el desempeño del modelo, lo que puede ayudar a identificar patrones o tendencias relevantes en el periodo analizado.

Figura 39 Variación del R^2



Nota. Imagen de elaboración propia

4.3.2 Predicciones vs datos reales (Entrenamiento por año)

Los resultados obtenidos al entrenar el modelo de predicción de irradiancia utilizando datos agrupados por año en lugar de dividirlos mensualmente, como se hizo en el ítem anterior. Este cambio en la estrategia de entrenamiento permite evaluar la capacidad del modelo para captar patrones a largo plazo y generalizar su rendimiento de forma continua a lo largo del año, sin la influencia directa de segmentaciones estacionales.

El entrenamiento del modelo LSTM se llevó a cabo utilizando 229.377 datos con distintos diseños de arquitectura, variando el número de capas y la cantidad de neuronas en cada una. Se entrenaron 14 configuraciones distintas del modelo, cuyos resultados se resumen en la Tabla 3. Cada modelo fue evaluado tanto en los datos de test como en las predicciones futuras, considerando el coeficiente de determinación (R^2) y el error absoluto medio (MAE) como principales métricas de evaluación.

Tabla 5 Arquitecturas LSTM

Modelo	Univariado												
			Nro de Capas / Nro de Neuronas				Entrenamiento			En Test		En Predicción	
	Días Futuros	Cant. Datos	#Nodos Layer1	#Nodos Layer2	#Nodos Layer3	#Nodos Layer4	Epochs	Batch_Size	Tiempo (min)	MAE	R^2 %	MAE	R^2 %
LSTM	1	229377	5	-	-	-	100	64	74.65	78	77%	80	72%
	2		5	-	-	-	100	64	79.42	83	77%	85	68%
	1		20	20	20	-	100	64	110.36	68	80%	79	73%
	2		20	20	20	-	100	64	81.71	90	72%	85	67%
	1		**20**	**-**	**-**	**-**	**100**	**64**	**64,39**	**59.36**	**82%**	**76.6**	**75%**
	1		**50**	**-**	**-**	**-**	**100**	**64**	**80.46**	**57.47**	**82%**	**77.8**	**74%**
	1		128	-	-	-	100	64	245.05	73.7	77%	80.3	73%
	1		20	20	-	-	100	64	317,49	101.6	64%	81.6	71%
	1		40	70	-	-	100	64	424.61	83.3	72%	78.5	73%
	1		50	50	50	-	100	64	293.40	102.4	65%	80.7	71%
	1		20	20	20	20	100	64	323.22	109.9	60%	85.82	71%
	1		300	-	-	-	100	64	106.38	82.33	78%	75.81	76%
	1		40	40	-	-	100	64	74.72	105.36	67%	77.49	74%

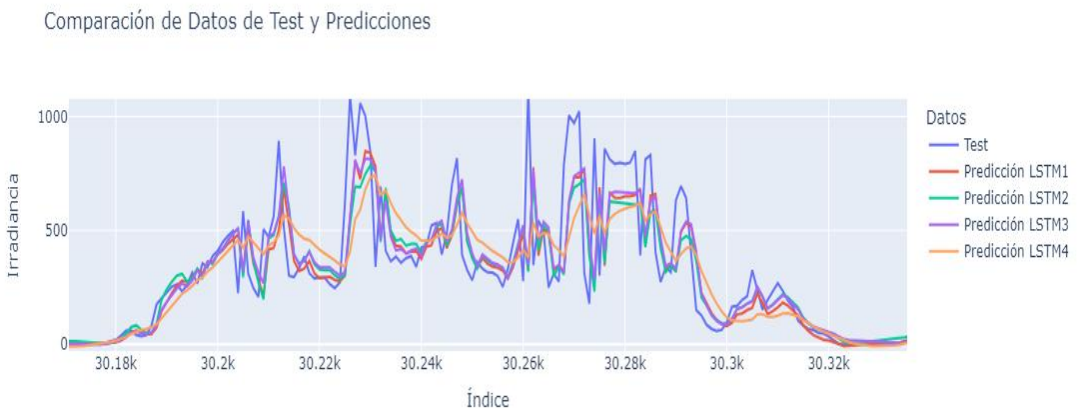
A diferencia de la segmentación por meses presentada en la sección anterior, en este caso el modelo entrenado anualmente mostró una mejora significativa en el R^2 en casi todas las configuraciones, superando el rendimiento obtenido por meses. En particular, algunos modelos alcanzaron valores de R^2 en las pruebas superiores al 80%, lo cual indica que el modelo es capaz de capturar los patrones generales de irradiancia de manera más efectiva cuando se consideran los datos anuales en lugar de mensuales. Esto sugiere que, al agrupar los datos de manera continua, el modelo tiene más contexto para aprender los patrones subyacentes, mejorando así su capacidad de predicción.

Al comparar los resultados obtenidos por meses (ítem 4.3.1) con los del entrenamiento por año, se puede observar que el R^2 en los modelos anuales es claramente superior en varias configuraciones. En el entrenamiento mensual, el R^2 fluctuaba notablemente en función de la variabilidad estacional y climática. Por ejemplo, en meses con alta variabilidad atmosférica, como noviembre y diciembre, los valores de R^2 descendían considerablemente. En contraste, en los modelos entrenados anualmente, se consiguió una mayor estabilidad en el R^2 a lo largo del año, con un mejor rendimiento global, alcanzando valores máximos de R^2 en predicción del 85.6% (Modelo con 50 nodos en una capa y 20 nodos en la segunda capa).

Los modelos LSTM fueron seleccionados al azar para evaluar su rendimiento predictivo en esta fase del estudio. Es evidente que cada modelo sigue en mayor o menor grado las variaciones de los datos reales de irradiancia. Sin embargo, se notan ligeras diferencias entre las predicciones de cada modelo, lo cual está relacionado con la variabilidad de la arquitectura.

Al observar la coincidencia general en las tendencias entre los datos de prueba y las predicciones (Figura 40), se puede afirmar que los modelos LSTM son capaces de capturar patrones clave en los datos de irradiancia, aunque con variaciones en la precisión, especialmente en los picos de irradiancia, donde algunos modelos logran un ajuste más cercano a los datos reales, mientras que otros presentan una menor coincidencia.

Figura 40 Test vs predicción modelos LSTM



Nota. Imagen de elaboración propia

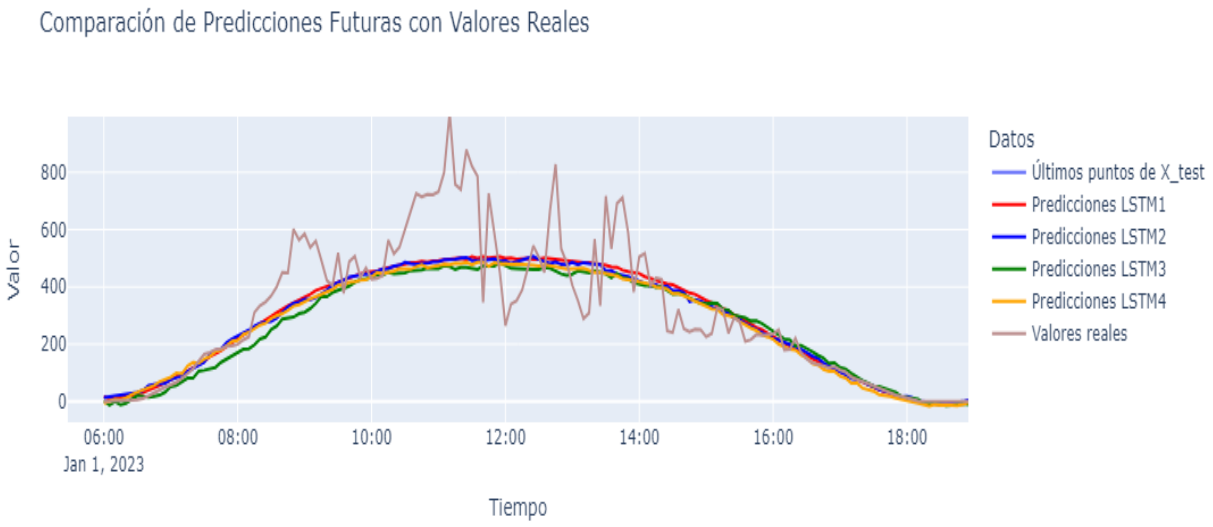
La Figura 41 compara las predicciones de irradiancia de un día completo, generadas por cuatro modelos LSTM, con los valores reales observados. Las predicciones siguen de manera precisa la tendencia general de los datos reales, mostrando que los modelos capturan el comportamiento esperado de irradiancia durante el día, con un aumento matutino, un pico alrededor del mediodía y una caída progresiva en la tarde.

Un aspecto positivo es que, aunque los modelos no replican exactamente los valores reales en cada momento, esto no implica un mal rendimiento, sino una buena capacidad de generalización. Los modelos no están sobre ajustados a los datos reales, lo que significa que son capaces de generar predicciones coherentes sin depender exclusivamente de las fluctuaciones específicas del conjunto de datos de entrenamiento. Este es un comportamiento esperado y

deseable en modelos LSTM para forecasting, donde el objetivo es capturar las tendencias y patrones generales en lugar de replicar los valores exactos.

Nota. Imagen de elaboración propia

Figura 41 Predicción 1 día de los modelos LSTM



4.3.3 Predicción multitemporal

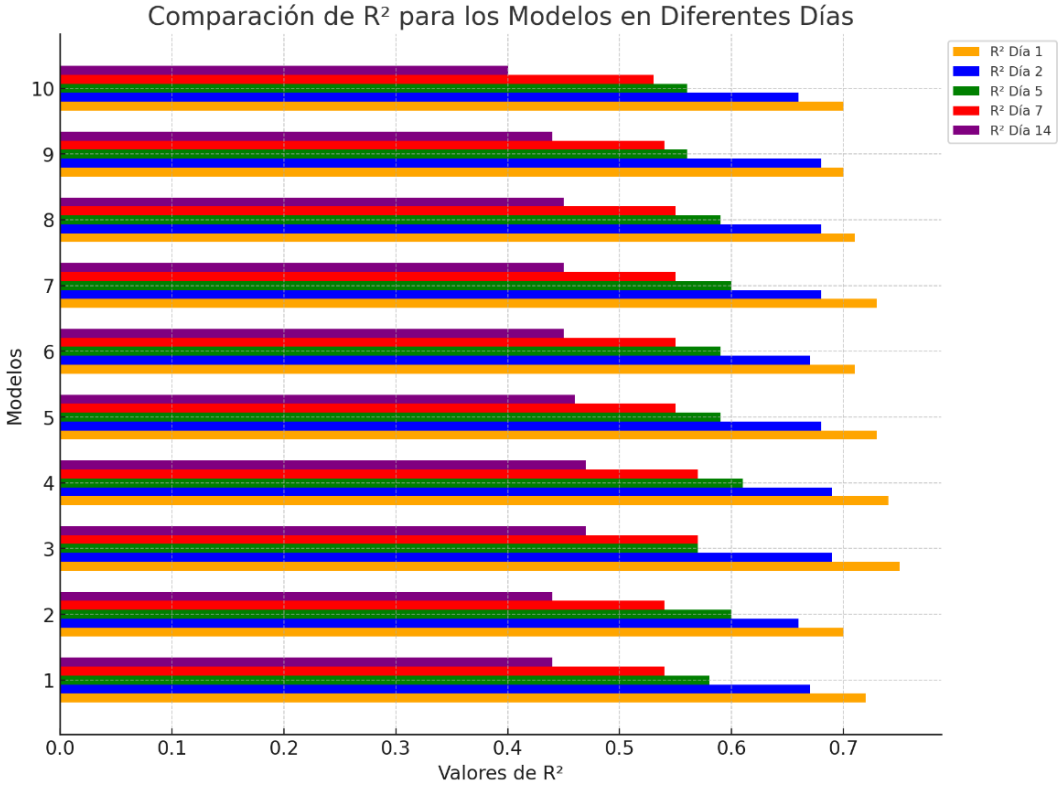
En esta tabla 4, se presentan diversas configuraciones de redes LSTM para predicciones multitemporales de irradiancia solar, donde se evalúa el rendimiento del modelo para prever diferentes días en el futuro (1, 2, 5, 7 y 14 días) utilizando un enfoque multitemporal. Esto significa que, aunque los datos de entrenamiento son por muestras de un día (156 muestras por día), el modelo se emplea para predecir múltiples días hacia adelante. Se están probando modelos con diferentes arquitecturas de redes LSTM para predecir la irradiancia hasta 14 días hacia adelante, ajustando el número de neuronas y capas.

Para los primeros días (del Día 1 al Día 7), las predicciones mantienen un coeficiente de determinación R^2 relativamente alto. En los primeros dos días, los mejores modelos alcanzan un R^2 entre 65% y 76%, lo que indica una alta confiabilidad en la predicción de estos días. A medida que el horizonte de predicción se extiende hasta el Día 7, los valores de R^2 se mantienen dentro de un rango razonable (entre 50% y 60%). Esto sugiere que el modelo todavía puede capturar de manera adecuada las tendencias generales de la irradiancia solar, manteniendo una buena capacidad predictiva.

Tabla 6 Modelos LSTM en función del tiempo

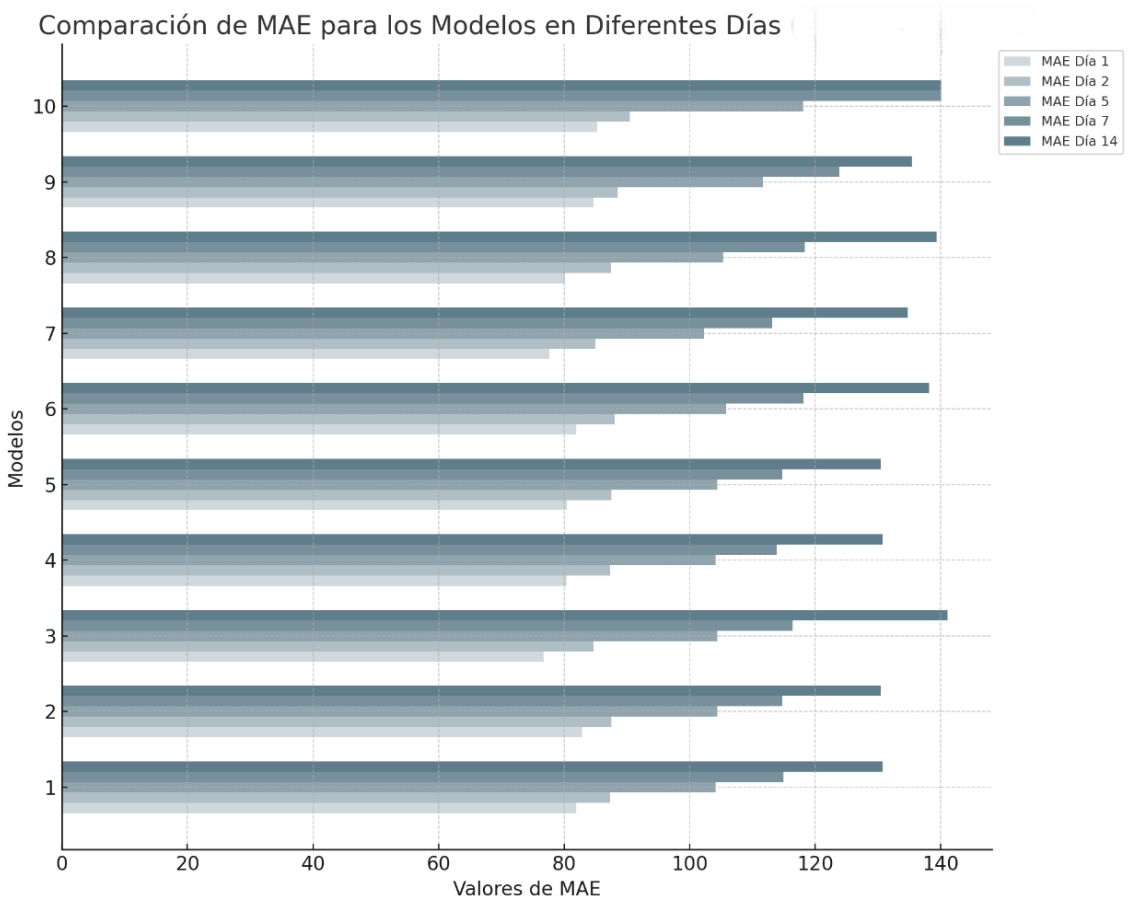
Modelo	Univariado									
		Nro de Capas / Nro de Neuronas				Coeficiente de determinación R^2				
	Días Futuros	#Nodos Layer1	#Nodos Layer2	#Nodos Layer3	#Nodos Layer4	Día 1	Día 2	Día 5	Día 7	Día 14
LSTM	1	5	-	-	-	72%	67%	58%	54%	44%
	2	5	-	-	-	-	68%	61%	56%	47%
	1	20	20	20	-	70%	66%	57%	54%	44%
	2	20	20	20	-	-	67%	69%	55%	46%
	1	20	-	-	-	75%	69%	60%	56%	41%
	1	50	-	-	-	74%	69%	61%	57%	47%
	1	128	-	-	-	73%	68%	60%	56%	45%
	1	20	20	-	-	71%	67%	59%	55%	43%
	1	40	70	-	-	73%	68%	60%	56%	44%
	1	50	50	50	-	71%	68%	59%	55%	44%
	1	20	20	20	20	71%	65%	57%	54%	41%
	1	300	-	-	-	76%	67%	58%	54%	40%
	1	40	40	-	-	74%	67%	58%	54%	40%

Figura 42 Comparación Coeficiente de Correlación



Nota. Imagen de elaboración propia

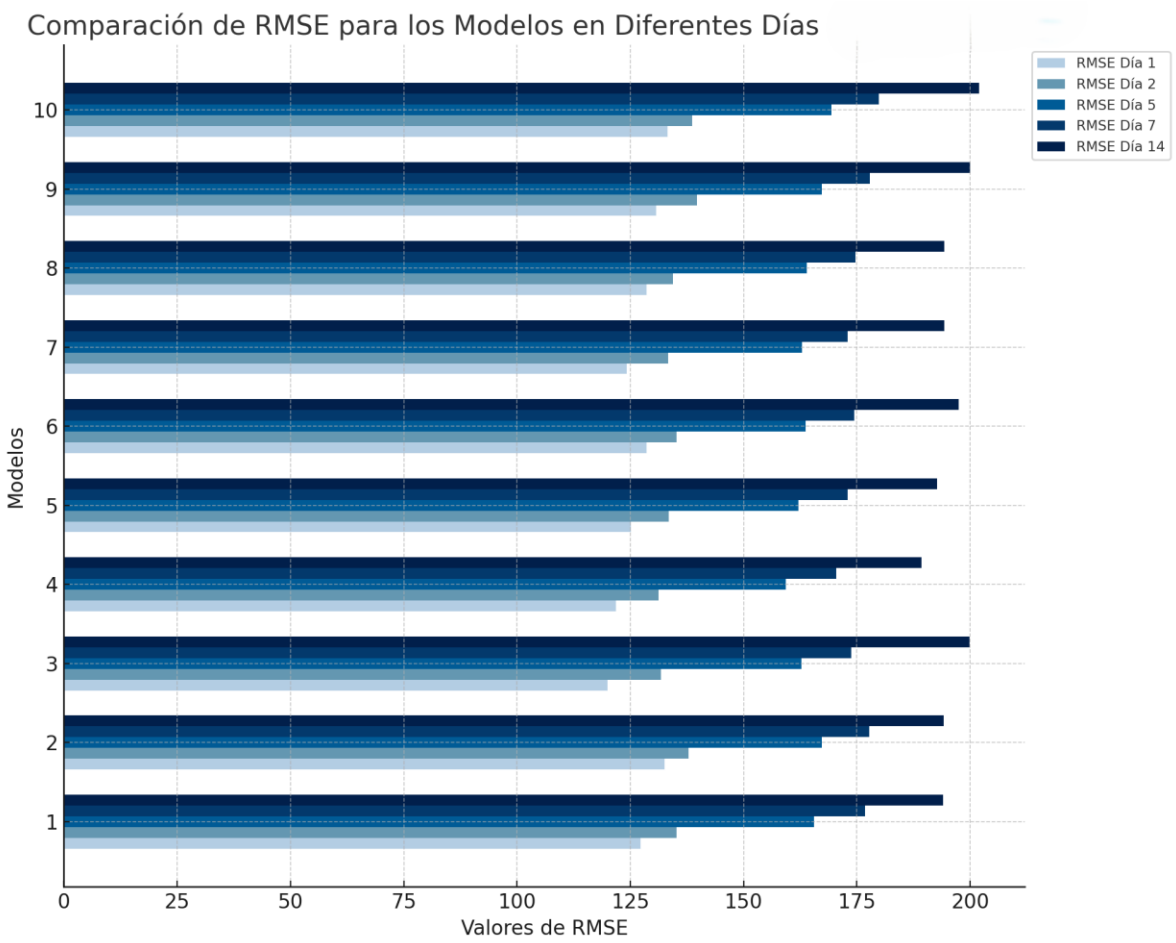
Figura 43 Comparación MAE En Diferentes Horizontes de Predicción



Nota. Imagen de elaboración propia

Para predicciones más allá del Día 7, especialmente para el Día 14, se observa una caída considerable en las métricas de desempeño donde el coeficiente de determinación, R^2 que cae por debajo del 50%, llegando incluso al 40% en algunos casos. Esto indica que la confiabilidad del modelo se reduce drásticamente después de una semana. Esta disminución en la precisión se debe a la complejidad y la naturaleza variable de los datos de irradiancia solar, que son más difíciles de predecir en horizontes largos.

Figura 44 Comparación RMSE En Diferentes Horizontes de Predicción



Nota. Imagen de elaboración propia

A pesar de la disminución en la precisión más allá del séptimo día, es importante resaltar que el hecho de que este modelo pueda hacer predicciones confiables hasta 7 días hacia adelante es una gran ventaja. En comparación con otros modelos con arquitecturas equiparables los cuales predicen unos minutos o unas pocas horas hacia adelante, este modelo LSTM multitemporal tiene como característica principal su capacidad de extender las predicciones hasta una semana sin perder de manera significativa su capacidad predictiva.

Para una explicación más detallada de lo mencionado anteriormente, consulte el Anexo 4 ubicado al final del presente documento.

4.3.4 Comparación modelo LSTM vs GRU

Tabla 7 LSTM vs GRU en función del tiempo

Modelo	Univariado								
	Nro de Capas / Nro de Neuronas				Coeficiente de determinación R^2				
	#Nodos Layer1	#Nodos Layer2	#Nodos Layer3	#Nodos Layer4	Día 1	Día 2	Día 5	Día 7	Día 14
LSTM	5	-	-	-	72%	67%	58%	54%	44%
	20	20	20	-	70%	66%	60%	54%	44%
	20	-	-	-	75%	69%	57%	56%	41%
	50	-	-	-	74%	69%	61%	57%	47%
	128	-	-	-	73%	68%	60%	56%	45%
	20	20	-	-	71%	67%	59%	55%	43%
	40	70	-	-	73%	68%	60%	56%	44%
	50	50	50	-	71%	68%	59%	55%	44%
	20	20	20	20	71%	65%	57%	54%	41%
	300	-	-	-	70%	66%	56%	53%	40%
GRU	5	-	-	-	70%	66%	61%	56%	47%
	20	20	20	-	72%	68%	60%	56%	43%
	20	-	-	-	72%	68%	59%	55%	46%
	50	-	-	-	71%	67%	58%	55%	45%
	128	-	-	-	71%	67%	57%	53%	39%
	20	20	-	-	72%	68%	58%	55%	44%
	40	70	-	-	72%	68%	57%	53%	39%
	50	50	50	-	72%	67%	59%	55%	42%
	20	20	20	20	71%	67%	57%	53%	37%
	300	-	-	-	71%	67%	58%	54%	40%

La Tabla 5 lleva a cabo una comparación en la que el modelo LSTM exhibe un desempeño superior durante los primeros días (Día 1 al Día 7) con relación al coeficiente de determinación R^2, alcanzando en ciertas situaciones el 75% el día 1 y manteniendo valores relativamente elevados hasta el día 7 (cerca del 61%). No obstante, tras el día 7, la precisión experimentó una notable reducción, en particular el día 14, donde el valor más alto registrado fue del 45% y, en ciertas situaciones, bajó al 40%.

En contraposición, el modelo GRU, a pesar de su comportamiento similar en los primeros días, con un cambio del 72% al 71% el día 1, parece tener una disminución más marcada que la de LSTM. El día 14, la mayoría R^2 observa que los valores disminuyen entre un 39% y un

46%, lo que señala que GRU posee una habilidad restringida para conservar la exactitud en horizontes de predicción más lejanos.

El modelo LSTM se distingue por su capacidad para almacenar datos en secuencias extensas, debido a su estructura fundamentada en puertas de entrada, salida y olvido. Esto se manifiesta en que, pese a que la precisión se reduce con el tiempo, LSTM conserva un desempeño más constante que GRU en periodos de tiempo más extensos. Como variante, GRU puede ofrecer beneficios en cuanto a eficiencia computacional, pero a expensas de una capacidad de memoria reducida, lo que podría justificar su disminución en el rendimiento en la predicción de largo alcance. Lo previamente expuesto se muestra en la Tabla 6.

Tabla 8 LSTM vs GRU en Test y Predicción

Modelo	Univariado										
		Nro de Capas / Nro de Neuronas				Entrenamiento			En Test		En Predicción	
	Cant. Datos	#Nodos Layer1	#Nodos Layer2	#Nodos Layer3	#Nodos Layer4	Epochs	Batch_Size	Tiempo (min)	MAE	R^2 %	MAE	R^2 %
LSTM	229377	5	-	-	-	100	64	74.65	78	77%	80	72%
		20	20	20	-	100	64	110.36	68	80%	79	73%
		20	-	-	-	100	64	64,39	59.36	82%	76.6	75%
		50	-	-	-	100	64	80.46	57.47	82%	77.8	74%
		128	-	-	-	100	64	245.05	73.7	77%	80.3	73%
		20	20	-	-	100	64	317,49	101.6	64%	81.6	71%
		40	70	-	-	100	64	424.61	83.3	72%	78.5	73%
		50	50	50	-	100	64	293.40	102.4	65%	80.7	71%
		20	20	20	20	100	64	323.22	109.9	60%	85.82	71%
		300	-	-	-	100	64	106.38	63.44	81%	81.92	70%
GRU	229377	5	-	-	-	100	64	190.59	79.43	78%	87.79	70%
		20	20	20	-	100	64	57.94	77.33	78%	79.67	72%
		20	-	-	-	100	64	85.02	64.61	81%	80.21	72%
		50	-	-	-	100	64	82.43	62.06	82%	80.95	71%
		128	-	-	-	100	64	76.96	60.37	82%	80.96	71%
		20	20	-	-	100	64	107.89	69.93	80%	80.8	72%
		40	70	-	-	100	64	110.17	62.6	82%	80.03	72%
		50	50	50	-	100	64	111.46	63.68	82%	81.32	72%
		20	20	20	20	100	64	176.02	73.89	80%	81.6	71%
		300	-	-	-	100	64	128.54	64.98	81%	80.58	71%

Por lo general, el modelo LSTM demanda considerablemente más tiempo de entrenamiento que el modelo GRU, particularmente cuando se incrementa el número de nodos. Por ejemplo, un modelo LSTM de 128 nodos requiere 245.05 minutos, en cambio, un modelo equivalente de GRU solo requiere 76.96 minutos. En realidad, el tiempo de entrenamiento para configuraciones con más capas (como 2 capas de 40 y 70 nodos) llega a 424.61 minutos en LSTM, en contraste con 110.17 minutos en GRU. Esto evidencia que, a pesar de que LSTM puede conservar más datos a través de secuencias temporales, este beneficio se asocia a un costo computacional notablemente superior.

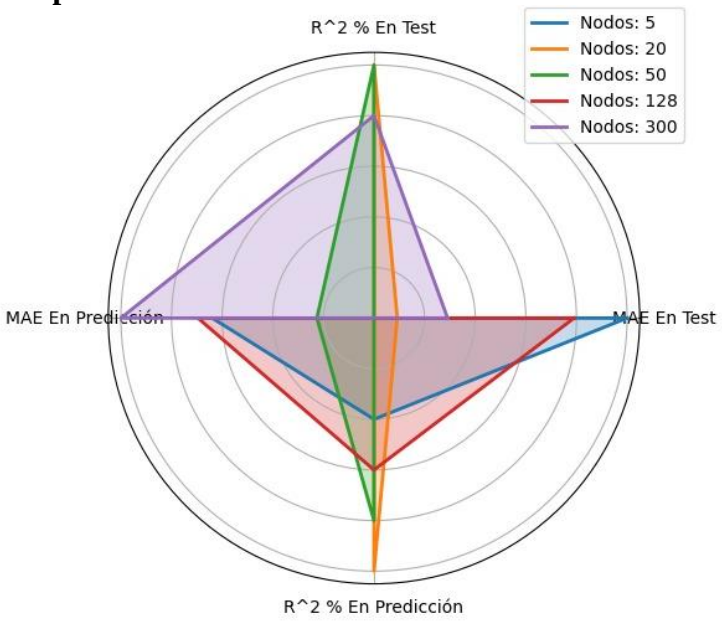
En relación con R^2 en el contexto de prueba, LSTM exhibe un desempeño sólido, con valores que varían entre el 77% y el 82% en la mayoría de las configuraciones. Es notable que la arquitectura de 20 nodos y 50 nodos alcanzan un R^2 del 82%, lo que indica que dicha configuración es extremadamente eficaz para el conjunto de test. Respecto al Error Absoluto Medio (MAE), los valores inferiores, como 57.47 en la misma configuración, fortalecen la exactitud del modelo en el examen. El rendimiento en el escenario de prueba para GRU varía un poco respecto al de LSTM. Los valores de R^2 oscilan entre el 78% y el 81%, siendo los valores de MAE usualmente superiores a los de LSTM.

En el escenario de predicción, el desempeño de LSTM se mantiene competitivo, con valores de R^2que oscilan entre 70% y 75%. Nuevamente, las arquitecturas de 20 y 50 nodos son de las más destacadas, alcanzando un R^2 de 75% y 74% respectivamente, con un MAE de 76.6 y 77.8, lo que indica que esta configuración logra un buen equilibrio entre precisión y eficiencia en la predicción. Aunque GRU muestra un rendimiento similar, el desempeño en predicción tiende a ser ligeramente inferior. Los valores de R^2varían entre 70% y 72%, con un MAE en algunos casos superior a LSTM.

4.3.5 Análisis de resultados en el pronóstico de irradiancia.

La Figura 45 muestra un gráfico de tipo radar el cual compara cinco modelos LSTM con diferentes cantidades de nodos (5, 20, 50, 128, y 300), evaluando su rendimiento mediante el Error Absoluto Medio (MAE) y el coeficiente de determinación (R^2) en las fases de test y predicción. El objetivo es analizar cómo el número de nodos influye en la precisión del modelo.

Figura 45 Comparación de modelos LSTM



Nota. Imagen de elaboración propia.

En las pruebas llevadas a cabo, los **modelos de 20 y 50 nodos** han probado ser los más eficaces, sobresaliendo sobre otros modelos en cuanto a exactitud y fiabilidad en la predicción. Estos modelos consiguen un balance ideal entre la complejidad del sistema y la precisión de las proyecciones, lo que los hace las mejores alternativas para la predicción de la irradiancia solar a largo plazo. Debido a su habilidad para adaptarse con exactitud a patrones climáticos, sobresalen como instrumentos esenciales para una administración energética eficaz en entornos cambiantes y dinámicos.

Los modelos desarrollados tienen un impacto significativo en la predicción de irradiancia solar, ofreciendo herramientas valiosas para la gestión de energía en sistemas fotovoltaicos. Al extender la capacidad de predicción hasta 8 días en el futuro, los modelos representan una mejora considerable respecto a los enfoques tradicionales, que generalmente se limitan a predicciones a corto plazo (minutos o pocas horas). Esta capacidad de anticipación a más largo plazo permite una mejor planificación en la operación y mantenimiento de sistemas solares, facilitando la optimización de la producción energética y la reducción de pérdidas.

Una de las principales ventajas de contar con estos modelos que predice a largo plazo es que permite a los operadores y gestores de energía prepararse para fluctuaciones en la producción de energía solar debido a cambios en las condiciones climáticas. Por ejemplo, en días nublados o durante condiciones meteorológicas adversas, los sistemas pueden ajustar su

operación anticipadamente, maximizando así su eficiencia y minimizando costos. Este enfoque proactivo es crucial en un contexto donde la energía solar está ganando terreno como fuente principal de energía renovable.

Además, la Universidad CESMAG se beneficiará directamente de este modelo de predicción. La institución, que está comprometida con la investigación y el desarrollo en el ámbito de las energías renovables, podrá utilizar este modelo para mejorar sus proyectos relacionados con la energía solar. La capacidad de predecir la irradiancia solar con hasta 8 días de antelación permitirá a la universidad optimizar la generación y el uso de energía en sus instalaciones, reduciendo costos operativos y mejorando la eficiencia de sus sistemas solares. Esto no solo contribuirá a la sostenibilidad de la universidad, sino que también servirá como un valioso recurso educativo para estudiantes e investigadores interesados en el área de energías renovables.

Sin embargo, es importante destacar que, a pesar de la capacidad del modelo para realizar predicciones extendidas, la precisión tiende a disminuir después de 8 días. Esta disminución en la exactitud puede atribuirse a varios factores, incluidos cambios impredecibles en las condiciones atmosféricas, como la nubosidad y otros fenómenos climáticos. En meses donde la variabilidad climática es más pronunciada, se observan errores significativos en las predicciones, lo que resalta la necesidad de continuar mejorando la robustez del modelo.

Los resultados obtenidos en las comparaciones de las predicciones y los datos reales indican que, aunque el modelo es efectivo en el corto y mediano plazo, su desempeño a largo plazo podría beneficiarse de la inclusión de variables adicionales que capturen mejor la complejidad de las condiciones meteorológicas. A medida que la tecnología avanza, se espera que la incorporación de más datos geográficos y meteorológicos, así como el uso de técnicas de aprendizaje profundo más sofisticadas, puedan mejorar aún más la precisión de las predicciones.

5. Conclusiones

En el presente trabajo se implementó y evaluó un modelo de redes neuronales de tipo LSTM para el pronóstico de irradiancia en la Universidad CESMAG, utilizando datos recopilados durante varios años. A lo largo del proceso de investigación se logró determinar la eficiencia de la red LSTM multitemporal, logrando predicciones precisas en horizontes temporales que se extienden por varios días, lo cual representa una mejora significativa en comparación con los modelos revisados en la literatura, cuya validación se limita al subconjunto de test. Este enfoque aporta una nueva perspectiva al campo de la predicción de irradiancia, al demostrar que las redes temporales pueden utilizarse para ofrecer un pronóstico con mayor anticipación y precisión.

Los principales resultados de este trabajo muestran que el modelo LSTM es capaz de capturar patrones relevantes en los datos de irradiancia y ofrece predicciones con un coeficiente de determinación óptimo, especialmente en los primeros días de predicción. Aunque la precisión decrece a medida que se incrementa el horizonte temporal, los niveles de error siguen siendo manejables y permiten la toma de decisiones informadas para futuros proyectos en la gestión de energía solar de la Universidad CESMAG.

El modelo de red LSTM desarrollado en este trabajo logró avances significativos respecto a los antecedentes en el campo de la predicción de irradiancia. Mientras que los estudios previos generalmente se limitaron a predicciones de corto alcance, restringidas a periodos de horas, nuestra implementación permitió extender con éxito las predicciones hasta varios días. Esto representa una mejora en términos de capacidad de pronóstico, al ofrecer una mayor anticipación para la toma de decisiones relacionadas con sistemas de energía solar. La capacidad de la red LSTM Multitemporal para capturar patrones en los datos de irradiancia muestran que este enfoque proporciona resultados más robustos y una mejor capacidad de generalización comparada con los modelos revisados en la literatura.

Este trabajo de grado abre nuevas líneas de investigación en el campo del pronóstico de irradiancia con redes neuronales LSTM. Futuras investigaciones podrían explorar la integración de otras variables meteorológicas y el uso de arquitecturas híbridas que combinen LSTM con otros modelos de predicción para mejorar la precisión a largo plazo. De igual manera, se recomienda estudiar la posibilidad de aplicar este enfoque a diferentes regiones geográficas y climas para validar la robustez del modelo en escenarios más diversos.

6. Recomendaciones

Aunque la imputación de datos faltantes permitió mantener la consistencia en los entrenamientos, es recomendable mejorar los mecanismos de adquisición de datos desde la estación meteorológica. Esto reduciría la necesidad de preprocesamiento y aumentaría la precisión del modelo, ya que siempre es preferible trabajar con datos reales. Se sugiere implementar soluciones para asegurar la continuidad en la captura de datos, como la automatización de los sistemas de respaldo de energía o la implementación de alertas tempranas ante fallos en la transmisión de información.

En el presente trabajo se abordó la imputación de datos con un algoritmo propio desarrollado en Python el cual promedia los datos existentes en el mismo instante de tiempo para poder rellenar las celdas con valores nulos. Esto podría adaptarse a otras formas de imputación, como el uso de métodos de interpolación o algoritmos basados en aprendizaje automático para grandes lotes de datos faltantes.

Para mejorar la precisión del modelo a largo plazo, se recomienda la inclusión de más variables meteorológicas, como nubosidad, temperatura, velocidad del viento, entre otras. Esto podría ayudar a capturar mejor la complejidad del clima y su impacto en la irradiancia solar. Sin embargo, debido al aumento en el coste computacional que implicaría un modelo multivariable, es aconsejable utilizar tecnologías de mayor capacidad, como TPU’s (Tensor Processing Units), o software especializado para entrenamientos con grandes volúmenes de datos.

El modelo propuesto ha mostrado un rendimiento adecuado para la predicción de irradiancia en la región de Pasto. Sin embargo, para validar su generalización, sería útil replicar el estudio en otras zonas geográficas con características climáticas diferentes. Esto permitiría evaluar la robustez del modelo y su capacidad para adaptarse a diversas condiciones atmosféricas.

Si bien el modelo LSTM multitemporal ofrece buenos resultados para predicciones de hasta 8 días, su precisión disminuye a medida que el horizonte temporal se extiende. Una posible solución sería explorar arquitecturas híbridas que combinen LSTM con otros modelos de predicción, como redes convolucionales o modelos autorregresivos, para mejorar la capacidad del modelo de capturar patrones a largo plazo sin comprometer la eficiencia.

7. Referencias

[1] Programa de Ingeniería Electrónica, “Proyecto Eductivo del Programa.” Universidad Cesmag, Pasto, p. 67, 2015.

[2] María Vázquez Fernández, “Energía solar en Colombia,” *Prim. pasos Energía Sol. en Colomb.*, pp. 1–8, 2022, [Online]. Available: http://www.laguiasolar.com/energia-solar-en-colombia/

[3] O. Luis, L. V. Daniel, B. G. Víctor, and O. Hugo, “Solar Radiation Prediction on Photovoltaic Systems Using Machine Learning Techniques,” Universidad Técnológica y Pedagógica de Colombia, 2020.

[4] M. Moran, “Energía Desarrollo Sostenible,” *Organización de las Naciones Unidas*, 2020.

[5] M. V. José Alejandro, “Predicción de la generación eléctrica de la central solar Rubí utilizando redes neuronales LSTM,” Universidad de Piura, 2023. [Online]. Available: https://pirhua.udep.edu.pe/bitstream/handle/11042/5950/IME_2304.pdf?sequence=1&isAllowed=y

[6] A. H. Jiménez, “Análisis Y Predicción De Radiación En Sistemas Fotovoltaicos Haciendo Uso De Machine Learning,” Universidad De Los Andes, 2023.

[7] M. Ruales y J. Eraso, Análisis de rendimiento de algoritmos de predicción de irradiancia solar implementados en hardware y evaluados en tiempo real, Universidad CESMAG, 2023.

[8] U. Quirama Estrada, J. Sepúlveda Aguirre, M. Morelo Machado, C. Mosquera Romaña, and L. C. Valle Beleño, “Beneficios económicos de la energía renovable en Colombia,” *Adm. Desarro.*, vol. 52, no. 2, pp. 152–164, 2022, doi: 10.22431/25005227.vol52n2.9.

[9] M. O. Moreira, P. P. Balestrassi, A. P. Paiva, P. F. Ribeiro, and B. D. Bonatto, “Design of experiments using artificial neural network ensemble for photovoltaic generation forecasting,” *Renew. Sustain. Energy Rev.*, vol. 135, no. September 2020, p. 110450, 2021, doi: 10.1016/j.rser.2020.110450.

[10] P. Lara Benítez, “Predicción de series temporales en streaming mediante Deep Learning,” p. 1, 2022, [Online]. Available: https://dialnet.unirioja.es/servlet/tesis?codigo=307565&info=resumen&idioma=ENG

[11] H. M. Zuo, J. Qiu, Y. H. Jia, Q. Wang, and F. F. Li, “Ten-minute prediction of solar irradiance based on cloud detection and a long short-term memory (LSTM) model,”

Energy Reports, vol. 8, pp. 5146–5157, 2022, doi: 10.1016/j.egyr.2022.03.182.

[12] M. C. Sorkun, Ö. Durmaz Incel, and C. Paoli, "Time series forecasting on multivariate solar radiation data using deep learning (LSTM)," *Turkish J. Electr. Eng. Comput. Sci.*, vol. 28, no. 1, pp. 211–223, 2020, doi: 10.3906/elk-1907-218.

[13] M. Konstantinou, S. Peratikou, and A. G. Charalambides, "Solar photovoltaic forecasting of power output using lstm networks," *Atmosphere (Basel).*, vol. 12, no. 1, pp. 1–17, 2021, doi: 10.3390/atmos12010124.

[14] M. Jaihuni *et al.*, "A novel recurrent neural network approach in forecasting short term solar irradiance," *ISA Trans.*, vol. 121, pp. 63–74, 2022, doi: 10.1016/j.isatra.2021.03.043.

[15] S. Gbémou, J. Eynard, S. Thil, E. Guillot, and S. Grieu, "A comparative study of machine learning-based methods for global horizontal irradiance forecasting," *Energies*, vol. 14, no. 11, pp. 1–23, 2021, doi: 10.3390/en14113192.

[16] Y. Li, F. Ye, Z. Liu, Z. Wang, and Y. Mao, "A Short-Term Photovoltaic Power Generation Forecast Method Based on LSTM," *Math. Probl. Eng.*, vol. 2021, 2021, doi: 10.1155/2021/6613123.

[17] Mahesh Batta, "Machine Learning Algorithms - A Review," *Int. J. Sci. Res.*, no. October, 2020, doi: 10.21275/ART20203995.

[18] O. Álvarez Hernández, T. Montaño Peralta, and J. Maldonado Correa, "La radiación solar global en la provincia de Loja, evaluación preliminar utilizando el método de Hottel," *Ingenius*, no. 11, p. 25, 2014, doi: 10.17163/ings.n11.2014.03.

[19] N. Falcón, F. Peña, H. Mavo, and R. Muñoz, "Irradiación Solar Global En La Ciudad De Valencia. (Global Solar Irradiación in the City of Valencia)," 2001.

[20] E. B. Babatunde, *Solar Radiation*. Rijeka: IntechOpen (March 21, 2012), 2012. doi: 10.5772/38994.

[21] Y. Xia and J. Wang, "Recurrent Neural Networks for Optimization: the State of the Art," *Book, Chapter_Reurrent neural networks Des. Appl.*, 2001.

[22] I. Bonet Cruz, S. Salazar Martínez, A. Rodríguez Abed, R. Grau Ábalo, and M. M. García Lorenzo, "Redes neuronales recurrentes para el analisis de secuencias," *Rev. Cuba. Ciencias Informáticas*, vol. 1, no. 4, pp. 48–57, 2007, [Online]. Available: https://www.redalyc.org/pdf/3783/378343634004.pdf

[23] M. Mascarell, "Clasificación de Textos Basado en los Modelos Pre-entrenados BERT,"

Univ. Politec. Val., pp. 1–53, 2021.

[24] C. Bonilla Carrión, “Redes Convolucionales,” 2020.

[25] J. De Lucio, “Estimación adelantada del crecimiento regional mediante redes neuronales LSTM,” *Investig. Reg. - J. Reg. Res.*, vol. 49, pp. 45–64, 2021, doi: 10.38191/iirr-jorr.21.007.

[26] C. Arana, “Modelos de aprendizaje automático mediante árboles de decisión,” UCEMA, 2021.

[27] Armando Jose Quijano Vodniza, *Guía de Investigación Cuantitativa*. San Juan de Pasto - Nariño, 2009.

[28] Davis Instruments Corp, “Manual de la consola Vantage Pro2.” p. 55, 2006.

8. Anexos

Anexo 1:

https://docs.google.com/spreadsheets/d/1YbzJ4gCfAHT1eUmX_7e-roJ0yYLTB35E/edit?usp=sharing&ouid=109095232544394642416&rtpof=true&sd=true

Anexo 2:

https://docs.google.com/spreadsheets/d/17S0D3F1UDcmrRZmlgWCvIVt2g-92Dltt/edit?usp=sharing&ouid=109095232544394642416&rtpof=true&sd=true

Anexo 3:

https://docs.google.com/spreadsheets/d/1NSMHhMPEPnk8vtOSB6Ile2K9sQZhcP6Z/edit?usp=sharing&ouid=109095232544394642416&rtpof=true&sd=true

Anexo 4:

https://drive.google.com/file/d/1K-c27axhJnmVR6b3aWPgNB6aakV_SmI6/view?usp=sharing

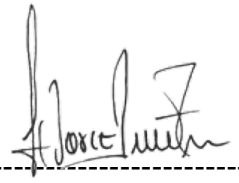
San Juan de Pasto, 26 de Noviembre 2024

Biblioteca
REMIGIO FIORE FORTEZZA OFM. CAP.
Universidad CESMAG
Pasto

Saludo de paz y bien.

Por medio de la presente se hace entrega del Trabajo de Grado / Trabajo de Aplicación denominado Estudio del porcentaje de error en el pronóstico multitemporal de la irradiancia basado en RNA recurrente tipo LSTM, presentado por el (los) autor(es) Andres Felipe Zambrano Benavides, y Daniel Sebastián Rosero Usamá, del Programa Académico de Ingeniería Electrónica al correo electrónico biblioteca.trabajosdegrado@unicesmag.edu.co. Manifiesto como asesor(a), que su contenido, resumen, anexos y formato PDF cumple con las especificaciones de calidad, guía de presentación de Trabajos de Grado o de Aplicación, establecidos por la Universidad CESMAG, por lo tanto, se solicita el paz y salvo respectivo.

Atentamente,



John Evert Barco Jiménez
87067512
ingeniería Electrónica
3158222096
jebarco@unicesmag.edu.co

INFORMACIÓN DEL (LOS) AUTOR(ES)	
Nombres y apellidos del autor: Andres Felipe Zambrano Benavides	**Documento de identidad:** 1004213994
Correo electrónico: afzambrano.3994@unicesmag.edu.co	**Número de contacto:** 3002487899
Nombres y apellidos del autor: Daniel Sebastián Rosero Usamá	**Documento de identidad:** 1193029981
Correo electrónico: dsrosero.9981@unicesmag.edu.co	**Número de contacto:** 3166420844
Nombres y apellidos del asesor: John Evert Barco Jiménez	**Documento de identidad:** 87067512
Correo electrónico: jebarco@unicesmag.edu.co	**Número de contacto:** 3158222096
Título del trabajo de grado: Estudio del porcentaje de error en el pronóstico multitemporal de la irradiancia basado en RNA recurrente tipo LSTM	
Facultad y Programa Académico: Facultad de Ingeniería Programa de Ingeniería Electrónica	

En mi (nuestra) calidad de autor(es) y/o titular (es) del derecho de autor del Trabajo de Grado o de Aplicación señalado en el encabezado, confiero (conferimos) a la Universidad CESMAG una licencia no exclusiva, limitada y gratuita, para la inclusión del trabajo de grado en el repositorio institucional. Por consiguiente, el alcance de la licencia que se otorga a través del presente documento, abarca las siguientes características:

a) La autorización se otorga desde la fecha de suscripción del presente documento y durante todo el término en el que el (los) firmante(s) del presente documento conserve (mos) la titularidad de los derechos patrimoniales de autor. En el evento en el que deje (mos) de tener la titularidad de los derechos patrimoniales sobre el Trabajo de Grado o de Aplicación, me (nos) comprometo (comprometemos) a informar de manera inmediata sobre dicha situación a la Universidad CESMAG. Por consiguiente, hasta que no exista comunicación escrita de mi(nuestra) parte informando sobre dicha situación, la Universidad CESMAG se encontrará debidamente habilitada para continuar con la publicación del Trabajo de Grado o de Aplicación dentro del repositorio institucional. Conozco(conocemos) que esta autorización podrá revocarse en cualquier momento, siempre y cuando se eleve la solicitud por escrito para dicho fin ante la Universidad CESMAG. En estos eventos, la Universidad CESMAG cuenta con el plazo de un mes después de recibida la petición, para desmarcar la visualización del Trabajo de Grado o de Aplicación del repositorio institucional.
b) Se autoriza a la Universidad CESMAG para publicar el Trabajo de Grado o de Aplicación en formato digital y teniendo en cuenta que uno de los medios de publicación del repositorio institucional es el internet, acepto(amos) que el Trabajo de Grado o de Aplicación circulará con un alcance mundial.
c) Acepto (aceptamos) que la autorización que se otorga a través del presente documento se realiza a título gratuito, por lo tanto, renuncio(amos) a recibir emolumento alguno por la publicación, distribución, comunicación pública y/o cualquier otro uso que se haga en los términos de la presente autorización y de la licencia o programa a través del cual sea publicado el Trabajo de grado o de Aplicación.

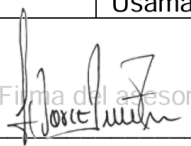
d) Manifiesto (manifestamos) que el Trabajo de Grado o de Aplicación es original realizado sin violar o usurpar derechos de autor de terceros y que ostento(amos) los derechos patrimoniales de autor sobre la misma. Por consiguiente, asumo(asumimos) toda la responsabilidad sobre su contenido ante la Universidad CESMAG y frente a terceros, manteniéndose indemne de cualquier reclamación que surja en virtud de la misma. En todo caso, la Universidad CESMAG se compromete a indicar siempre la autoría del escrito incluyendo nombre de(los) autor(es) y la fecha de publicación.
e) Autorizo(autorizamos) a la Universidad CESMAG para incluir el Trabajo de Grado o de Aplicación en los índices y buscadores que se estimen necesarios para promover su difusión. Así mismo autorizo (autorizamos) a la Universidad CESMAG para que pueda convertir el documento a cualquier medio o formato para propósitos de preservación digital.

NOTA: En los eventos en los que el trabajo de grado o de aplicación haya sido trabajado con el apoyo o patrocinio de una agencia, organización o cualquier otra entidad diferente a la Universidad CESMAG. Como autor(es) garantizo(amos) que he(hemos) cumplido con los derechos y obligaciones asumidos con dicha entidad y como consecuencia de ello dejo(dejamos) constancia que la autorización que se concede a través del presente escrito no interfiere ni transgrede derechos de terceros.

Como consecuencia de lo anterior, autorizo(autorizamos) la publicación, difusión, consulta y uso del Trabajo de Grado o de Aplicación por parte de la Universidad CESMAG y sus usuarios así:

- Permito(permitimos) que mi(nuestro) Trabajo de Grado o de Aplicación haga parte del catálogo de colección del repositorio digital de la Universidad CESMAG por lo tanto, su contenido será de acceso abierto donde podrá ser consultado, descargado y compartido con otras personas, siempre que se reconozca su autoría o reconocimiento con fines no comerciales.

En señal de conformidad, se suscribe este documento en San Juan de Pasto a los 26 días del mes de Noviembre del año 2024

 Firma del autor	 Firma del autor
Nombre del autor: Andres Felipe Zambrano Benavides.	Nombre del autor: Daniel Sebastián Rosero Usamá
 Firma del asesor _____ Nombre del asesor: John Evert Barco Jiménez	