



UNIVERSIDAD TÉCNICA DE COTOPAXI
FACULTAD DE CIENCIAS DE LA INGENIERÍA
Y APLICADAS
CARRERA DE ELECTRICIDAD

PROYECTO DE INVESTIGACIÓN

PRONÓSTICO DE RADIACIÓN SOLAR MEDIANTE
APRENDIZAJE AUTOMÁTICO UTILIZANDO SOFTWARE
DE CÓDIGO ABIERTO

Proyecto de Investigación presentado previo a la obtención del Título de
Ingeniero Eléctrico

AUTOR:

Muguicha Hinojoza Cristhian Ismael

TUTOR:

Ing. M.Sc. Edgar Roberto Salazar Achig

LATACUNGA – ECUADOR

FEBRERO – 2024



DECLARACIÓN DE AUTORÍA

Yo **Muguicha Hinojoza Cristhian Ismael**, con cedula de ciudadanía No. 020251174-7 declaro ser el autor del presente proyecto de investigación: **“PRONÓSTICO DE RADIACIÓN SOLAR MEDIANTE APRENDIZAJE AUTOMÁTICO UTILIZANDO SOFTWARE DE CÓDIGO ABIERTO”**, siendo el **Ing. M.Sc. Edgar Roberto Salazar Achig**, Tutor del presente trabajo; y, eximo expresamente a la Universidad Técnica de Cotopaxi y a sus representantes legales de posibles reclamos o acciones legales.

Además, certifico que las ideas, conceptos, procedimientos y resultados vertidos en el presente trabajo investigativo, son de mi exclusiva responsabilidad.

Latacunga, febrero 26 del 2024

.....
Muguicha Hinojoza Cristhian Ismael

C.C: 020251174-7



AVAL DEL TUTOR DEL PROYECTO DE INVESTIGACIÓN

En calidad de Tutor del proyecto de investigación sobre el título:

“PRONÓSTICO DE RADIACIÓN SOLAR MEDIANTE APRENDIZAJE AUTOMÁTICO UTILIZANDO SOFTWARE DE CÓDIGO ABIERTO”, de Muguicha Hinojoza Cristhian Ismael, de la carrera de Electricidad, considero que dicho Informe Investigativo es merecedor del aval de aprobación al cumplir las normas técnicas, traducción y formatos previstos, así como también ha incorporado las observaciones y recomendaciones propuestas en la pre-defensa.

Latacunga, 26 de febrero del 2024

.....
Ing. M.Sc. Salazar Achig Roberto Edgar

C.C: 050284761-9

TUTOR



AVAL DE APROBACIÓN DEL TRIBUNAL DE TITULACIÓN

En calidad de Tribunal de Lectores, aprueban el presente Informe de Investigación de acuerdo a las disposiciones reglamentarias emitidas por la Universidad Técnica de Cotopaxi, y, por la Facultad de Ciencias de la Ingeniería y Aplicadas; por cuanto, el postulante: Muguicha Hinojoza Cristhian Ismael, con el título del proyecto de investigación: **“PRONÓSTICO DE RADIACIÓN SOLAR MEDIANTE APRENDIZAJE AUTOMÁTICO UTILIZANDO SOFTWARE DE CÓDIGO ABIERTO”**, ha considerado las recomendaciones emitidas oportunamente y reúne los méritos suficientes para ser sometido al acto de sustentación del trabajo de titulación.

Por lo antes expuesto, se autoriza grabar los archivos correspondientes en un CD, según la normativa institucional.

Latacunga, 26 de febrero del 2024

Para constancia firman:

.....
Ing. M.Sc. Rueda Flores Walter Paul

C.C: 050368989-5

LECTOR 1(PRESIDENTE)

.....
Ing. M.Sc. Mullo Pallo Mauricio Eduardo

C.C: 050375513-8

LECTOR 2

.....
Ing. M.Sc. Proaño Maldonado Xavier Alfonso

C.C: 050265642-4

LECTOR 3

AGRADECIMIENTO

Agradezco a Dios, quien ha sido mi soporte inquebrantable, nunca abandonándome y brindándome la sabiduría necesaria para enfrentar este desafío que significó mi carrera como Ingeniero Eléctrico.

A mis padres, Oswaldo Gonzalo Muguicha y Laura María Hinojoza, les dedico un agradecimiento especial por ser mi fortaleza constante. A través de sus enseñanzas, apoyo incondicional y su presencia constante en los momentos difíciles. Gracias por demostrarme que, con esfuerzo, dedicación y la decisión de nunca rendirse, se pueden superar grandes obstáculos y alcanzar metas significativas.

A mis queridos hermanos y hermanas. Gracias a ustedes, he podido alcanzar la meta más importante de mi vida. Su aliento y respaldo han sido un impulso invaluable en este camino

A los docentes que, con su dedicación durante mi formación académica, me han proporcionado los conocimientos necesarios para desenvolverme en este tema y llegar a este punto. En especial al MSc. Edgar Roberto Salazar Achig como tutor de mi proyecto de investigación quien me ha guiado con su paciencia, y sabiduría como docente.

Cristhian I. Muguicha H.

DEDICATORIA

Con profundo agradecimiento y amor, dedico este trabajo de investigación a aquel que me da vida, salud y fortaleza día a día: Dios. Su guía y protección han sido fundamentales para alcanzar la culminación de esta etapa de mi vida.

A mis queridos padres, Laura María Hinojoza y Oswaldo Gonzalo Muguicha, quienes han sido mi mayor fuente de formación, mi motor de vida y el apoyo incondicional que ha impulsado cada paso que he dado. Su ejemplo de dedicación y sacrificio ha sido la luz que ha iluminado mi camino, y por ellos, hoy celebro este logro especial.

A mis hermanos, cuyos consejos sabios y palabras de ánimo han sido mi soporte en los momentos desafiantes. Cada uno de ustedes ha contribuido de manera significativa a mi camino, y por eso les dedico este logro compartido. Su inspiración y apoyo han sido esenciales para llegar a esta meta.

Cristhian I. Muguicha H.

UNIVERSIDAD TÉCNICA DE COTOPAXI

FACULTAD DE CIENCIAS DE LA INGENIERÍA Y APLICADAS

TÍTULO: “PRONÓSTICO DE RADIACIÓN SOLAR MEDIANTE APRENDIZAJE AUTOMÁTICO UTILIZANDO SOFTWARE DE CÓDIGO ABIERTO”

Autor:

Muguicha Hinojoza Cristhian Ismael

RESUMEN

El presente trabajo de investigación se enfoca en predecir la radiación solar. La importancia de realizar esta predicción radica en sus diversas aplicaciones como: la generación de energía renovable, la fotosíntesis para el crecimiento de las plantas, para sistemas térmicos para calentar agua, etc. La metodología utilizada implica el diseño, implementación y comparación de varios modelos de predicción, estos modelos se basan en técnicas de aprendizaje automático y son implementados mediante software de código abierto (Python), entre las técnicas empleadas se encuentran la regresión lineal simple, unidades recurrentes cerradas (GRU) y redes neuronales. Para evaluar la efectividad de estos modelos, se utilizan métricas como el Error Cuadrático Medio (MSE), el Error Absoluto Medio (MAE) y el coeficiente de determinación (R^2), estas métricas permiten medir la precisión de las predicciones en comparación con los datos reales. Para el análisis de los resultados obtenidos de cada modelo de predicción, se realiza en tres escenarios: un día (2023/02/06), una semana (2023/02/13 al 2023/02/19) y un mes (marzo del 2023), en el cual, el que mejor destaca en los tres escenarios de análisis, es el modelo de regresión lineal simple debido a su consistente desempeño superior en diversas métricas. Con un bajo Error Absoluto Medio ($MAE = 19,379$), alto Coeficiente de Determinación ($R^2 = 0,9937$) cercano a 1, y el Error Cuadrático Medio ($MSE = 525,963$) más bajo en todas las instancias, esta técnica demuestra ofrecer predicciones más precisas y un ajuste robusto.

Palabras claves: aprendizaje automático, pronóstico, métricas, Python, modelos, presión.

COTOPAXI TECHNICAL UNIVERSITY

ENGINEERING SCIENCES AND APPLIED FACULTY

THEME: “MACHINE LEARNING SOLAR RADIATION FORECASTING USING OPEN SOURCE SOFTWARE”

Author:

Muguicha Hinojoza Cristhian Ismael

ABSTRACT

The present research work focuses on predicting solar radiation. The importance of making this prediction lies in its various applications such as: renewable energy generation, photosynthesis for plant growth, for thermal systems for water heating, etc. The methodology used involves the design, implementation and comparison of several prediction models, these models are based on machine learning techniques and are implemented using open source software (Python), among the techniques used are simple linear regression, closed recurrent units (GRU) and neural networks. To evaluate the effectiveness of these models, metrics such as Mean Squared Error (MSE), Mean Absolute Error (MAE) and the coefficient of determination (R²) are used to measure the accuracy of the predictions compared to the real data. For the analysis of the results obtained from each prediction model, it is carried out in three scenarios: one day (2023/02/06), one week (2023/02/13 to 2023/02/19) and one month (March 2023), in which, the one that stands out best in the three analysis scenarios is the simple linear regression model due to its consistent superior performance in several metrics. With a low Mean Absolute Error (MAE = 19.379), high Coefficient of Determination (R² = 0.9937) close to 1, and the lowest Mean Squared Error (MSE = 525.963) in all instances, this technique proves to offer more accurate predictions and a robust fit.


Keywords: machine learning, forecasting, metrics, Python, models, pressure.

AVAL DE TRADUCCIÓN – PROFESIONAL EXTERNO

Yo Evelin Aracelly Santamaria Santana con cédula de identidad número: 175402236-4 Licenciada en: Pedagogía del idioma inglés con número de registro de la SENESCYT: 1010-2023-2634380; **CERTIFICO** haber revisado y aprobado la traducción al idioma Inglés del resumen del proyecto de investigación con el título: **“PRONÓSTICO DE RADIACIÓN SOLAR MEDIANTE APRENDIZAJE AUTOMÁTICO UTILIZANDO SOFTWARE DE CÓDIGO ABIERTO”** de: **Muguicha Hinojoza Cristhian Ismael**, egresado de la carrera de **Ingeniería en Electricidad** perteneciente a la Facultad de Ciencias de la Ingeniería y Aplicadas.

En virtud de lo expuesto y para constancia de lo mismo se registra la firma respectiva.

Latacunga, marzo 2024



Lic. Evelin Aracelly Santamaria Santana
CI: 175402236-4

ÍNDICE DE CONTENIDO

1. INFORMACIÓN GENERAL	1
2. INTRODUCCIÓN.....	2
2.1. PROBLEMA.....	2
2.1.1. Situación Problemática.....	2
2.1.2. Formulación del problema.....	2
2.2. OBJETO Y CAMPO DE ACCIÓN	2
2.3. JUSTIFICACIÓN	3
2.4. BENEFICIARIOS.....	3
2.4.1. Beneficiarios Directos	3
2.4.2. Beneficiarios Indirectos.....	3
2.5. HIPÒTESIS.....	4
2.6. OBJETIVOS	4
2.6.1. General.....	4
2.6.2. Específicos.....	4
2.7. SISTEMA DE TAREAS.....	5
3. FUNDAMENTACIÓN TEÓRICA	6
3.1. ANTECEDENTES	6
3.2. ENERGÍA	7
3.2.1. Tipos de energía.....	7
3.3. ENERGÍA SOLAR	10
3.4. RADIACIÓN SOLAR	10
3.4.1. Tipos de radiación solar.....	10
3.5. HORAS SOLAR PICO.....	12
3.6. PRONÓSTICO SOLAR	12
3.7. APRENDIZAJE AUTOMÁTICO	13

3.7.1.	Funcionamiento del aprendizaje automático	14
3.8.	MODELOS DE APRENDIZAJE AUTOMÁTICO	14
3.8.1.	Modelo de regresión lineal simple.....	15
3.8.2.	Modelo Gated Recurrent unit (GRU)	16
3.8.3.	Redes Neuronales	18
3.9.	MÉTRICAS PARA LA EVALUACIÓN DE LOS MODELOS	22
3.9.1.	Error absoluto medio (MAE).....	23
3.9.2.	Error cuadrático medio (MSE)	23
3.9.3.	R cuadrado (R^2)	24
3.9.4.	R^2 cuadrado ajustado	25
3.10.	ANÁLISIS DE CORRELACIÓN DE VARIABLES PARA CONSTRUIR MODELOS EXPLICATIVOS Y PREDICTIVOS	25
3.10.1.	Correlación de variables	25
3.10.2.	Interpretación de la correlación de variables	26
3.10.3.	Medición de la correlación de variables	27
3.11.	SOTFWARE DE CODIGO ABIERTO	28
3.11.1.	Lenguaje de programación	28
3.11.2.	Lenguaje de código abierto Python	29
3.11.3.	Principales usos de Python	29
3.11.4.	Tipos de librerías de Python	31
3.11.5.	Librerías de Python.....	31
4.	METODOLOGÍA.....	34
4.1.	TIPOS DE INVESTIGACIÓN	34
4.1.1.	Investigación Descriptiva	34
4.1.2.	Investigación Cuantitativo	34
4.1.3.	Investigación Experimental	34
4.2.	UBICACIÓN DEL LUGAR DE ESTUDIO	35

4.3.	TÉCNICAS E INSTRUMENTOS.....	35
4.3.1.	Programación.....	35
4.3.2.	Simulación.....	36
4.4.	METODOLOGÍA GENERAL.....	36
4.5.	DATOS HISTORICOS.....	38
4.5.1.	Obtención de la base de datos.....	38
4.5.2.	Almacenamiento de datos.....	39
4.5.3.	Preprocesamiento de datos.....	40
4.6.	DETERMINACIÓN DE VARIABLES.....	44
4.7.	COMPORTAMIENTO DE LA RADIACIÓN SOLAR EN LA ZONA DE ESTUDIO	44
4.8.	PROGRAMACIÓN EN PYTHON.....	45
4.8.1.	Uso de las librerías.....	45
4.8.2.	Lectura de datos.....	46
4.8.3.	Datos de entrenamiento.....	47
4.8.4.	Predicción.....	47
4.9.	METODOLOGÍA DE LOS MODELOS DE PREDICCIÓN.....	47
4.9.1.	Regresión lineal múltiple.....	47
4.9.2.	Unidades Recurrentes Cerradas (GRU).....	49
4.9.3.	Redes Neuronales.....	52
4.10.	COMPARACIÓN DE LOS MODELOS DE APRENDIZAJE AUTOMATICO.....	56
5.	ANÁLISIS DE RESULTADOS.....	57
5.1.	CASO DE ESTUDIO.....	57
5.1.1.	Análisis de predicción para 1 día.....	57
5.1.2.	Análisis de predicción para 1 semana.....	59
5.1.3.	Análisis de predicción para 1 mes.....	61
6.	CONCLUSIONES Y RECOMENDACIONES.....	64

6.1. CONCLUSIONES	64
6.2. RECOMENDACIONES.....	65
7. REFERENCIAS	66

ÍNDICE DE FIGURAS

Figura 3.1. Energía	7
Figura 3.2. Tipos de energías renovables.	8
Figura 3.3. Tipos de energía no renovables.	8
Figura 3.4. Capacidad de producción de energías renovables en América Latina y el Caribe en 2021.	9
Figura 3.5. Radiación solar directa.	11
Figura 3.6. Radiación solar difusa.	11
Figura 3.7. Radiación solar reflejada.	11
Figura 3.8. Regresión lineal simple	16
Figura 3.9. Unidades recurrentes cerradas	17
Figura 3.10. Arquitectura de una red neuronal	18
Figura 3.11. Red neuronal monocapa	19
Figura 3.12. Red neuronal multicapa	20
Figura 3.13. Red neuronal recurrente	20
Figura 3.14. Error de una predicción	22
Figura 3.15. Interpretación de correlación mediante gráficos de dispersión	26
Figura 3.16. Representación gráfica de la relación lineal y no lineal (monótona).	27
Figura 3.17. Lenguajes de programación más populares.....	29
Figura 4.1. Universidad Nacional de Piura	35
Figura 4.2. Diagrama de flujo de la metodología utilizada para la obtención de modelos de predicción.	37
Figura 4.3. Ubicación de la estación meteorológica de la Universidad Nacional de Piura [49].	38
Figura 4.4. Base de datos históricos de la radiación solar.	39
Figura 4.5. Verificación de valores nulos.	39
Figura 4.6. Metodología de la simulación de Montecarlo.	40
Figura 4.7. Diagrama de flujo de la simulación de Montecarlo con una distribución de probabilidad normal.....	41
Figura 4.8. Obtención de las series aleatorias a partir de la simulación de Montecarlo.	42
Figura 4.9. Comportamiento de la radiación solar obtenidos a partir de la simulación de Montecarlo.....	42
Figura 4.10. Preprocesamiento de la base de datos.	43

Figura 4.11. Verificación de valores nulos.	43
Figura 4.12. Comportamiento de la radiación solar del día 25/04/2022.....	44
Figura 4.13. Comportamiento de la radiación solar durante la primera semana de mayo del 2022.	45
Figura 4.14. Librerías usadas en la programación de los modelos de aprendizaje automático.	46
Figura 4.15. Lectura de la base de datos de estudio.	46
Figura 4.16. Porcentaje de datos de entrenamiento.	47
Figura 4.17. Proceso general de los modelos de predicción.....	47
Figura 4.18. Metodología de la regresión lineal simple.	48
Figura 4.19. Metodología del modelo unidades recurrentes cerradas (GRU).	49
Figura 4.20. Arquitectura del modelo GRU.....	51
Figura 4.21. Selección del optimizador para el modelo GRU.....	51
Figura 4.22. Diagrama de bloques de la metodología de una red neuronal.....	52
Figura 4.23. Comportamiento de la radiación solar con los optimizadores.....	54
Figura 4.24. Comportamiento de la radiación solar en relación al número de época.....	55
Figura 5.1. Resultados obtenidos de los modelos de predicción durante el 2023/02/06.	57
Figura 5.2. Técnicas de predicción de la radiación solar del día 2023/02/06: a) Regresión lineal simple, b) Unidades recurrentes cerradas (GRU), c) Redes neuronales.....	58
Figura 5.3. Resultados obtenidos de los modelos de predicción durante el 2023/02/13 al 2023/02/19.....	59
Figura 5.4. Técnicas de predicción de la radiación solar de una semana (2023/02/13 al 2023/02/19): a) Regresión lineal simple, b) Unidades recurrentes cerradas (GRU), c) Redes neuronales.....	60
Figura 5.5. Resultados obtenidos de los modelos de predicción durante todo el mes de marzo del 2023.	61
Figura 5.6. Técnicas de predicción de la radiación solar del mes de marzo del 2023: a) Regresión lineal simple, b) Unidades recurrentes cerradas (GRU), c) Redes neuronales.....	62

ÍNDICE DE TABLAS

Tabla 3.1. Tipos de radiación solar.....	11
Tabla 4.1. Determinación de variables dependientes e independientes.....	44
Tabla 4.2. Parámetros de la regresión lineal simple.	48
Tabla 4.3. Parámetros de la arquitectura de las unidades recurrentes cerradas (GRU).	50
Tabla 4.4. Comportamiento del número de épocas.....	52
Tabla 4.5. Parámetros de las redes neuronales.	53
Tabla 4.6. Optimizadores para la red neuronal.	54
Tabla 4.7. Selección del número de neuronas y épocas para la red neuronal.....	55
Tabla 4.8. Comparación de los modelos de aprendizaje automático.....	56
Tabla 5.1. Métricas de evaluación del modelo Regresión Lineal Simple.....	59
Tabla 5.2. Métricas de evaluación para el modelo GRU.	61
Tabla 5.3. Métricas de evaluación para el modelo de Redes Neuronales.....	63

1. INFORMACIÓN GENERAL

Título: Pronóstico de radiación solar mediante aprendizaje automático utilizando software de código abierto.

Fecha de inicio: octubre del 2023

Fecha de finalización:

Lugar de ejecución: Latacunga

Facultad que auspicia: Facultad de Ciencias de la Ingeniería y Aplicadas.

Carrera que auspicia: Carrera de Ingeniería en Electricidad.

Proyecto Marco asociado: Desarrollo de sistemas eléctricos eficientes para el abastecimiento y uso de energía eléctrica a nivel local, regional o nacional.

Equipo de Trabajo: Muguicha Hinojoza Cristhian Ismael

Tutor de Titulación: Ing. Ms.C Edgar Roberto Salazar Achig

Grupo de investigación: Energía y tecnología apropiadas

Área de conocimiento: 07 Ingeniería, Industria y Construcción / 071 Ingeniería y Profesiones Afines / 0713 Electricidad y Energía.

Línea de investigación: Energías Alternativas y Renovables, Eficiencia Energética y Protección Ambiental.

Sublíneas de investigación de la Carrera: Conversión y uso racional de la energía eléctrica.

Grupos Temáticos:

Modelación y simulación de sistemas de energía eléctrica.

2. INTRODUCCIÓN

2.1. PROBLEMA

2.1.1. Situación Problémica

Los métodos de pronóstico tradicionales basados en modelos estadísticos tienen limitaciones en cuanto a precisión y efectividad, especialmente cuando se trata de relaciones complejas no lineales entre variables meteorológicas. Las técnicas de aprendizaje automático son prometedoras para mejorar la precisión de los pronósticos meteorológicos en este caso radiación solar, por ende, también es necesario comparar y evaluar diferentes algoritmos para determinar los métodos efectivos. Además, el uso de software propietario para la predicción meteorológica puede resultar costoso y limitar el acceso a los últimos avances en este campo.

La predicción precisa de las variables meteorológicas es crucial para la gestión eficiente de fuentes de energía renovables como la eólica y la solar. La variabilidad de variables meteorológicas en este caso la radiación solar puede tener un impacto significativo en el rendimiento de los sistemas de energía renovable, y una predicción precisa puede ayudar a optimizar su funcionamiento y reducir costes.

Además, las condiciones meteorológicas pueden tener un impacto significativo en los resultados. Sin embargo, los métodos tradicionales de predicción basados en modelos estadísticos tienen limitaciones en cuanto a precisión y eficacia. Por lo tanto, se necesitan métodos de predicción más avanzados y precisos, como los basados en técnicas de aprendizaje automático, para mejorar la exactitud y la eficiencia de las predicciones meteorológicas.

2.1.2. Formulación del problema

Determinación del modelo adecuado de aprendizaje automático para el pronóstico de la radiación solar, utilizando software de código abierto.

2.2. OBJETO Y CAMPO DE ACCIÓN

El campo de acción para el proyecto: 330000 ciencias Tecnológicas / 3306 Ingeniería y Tecnología Eléctricas / 3306.99 Transmisión y Distribución.

2.3. JUSTIFICACIÓN

La radiación solar es una fuente de energía renovable y su disponibilidad varía según las condiciones meteorológicas. La predicción precisa de la radiación solar permite que los operadores de sistemas de energía renovable puedan planificar y gestionar mejor la producción de energía, lo que puede mejorar la eficiencia y a su vez reducir los costos.

Las técnicas de aprendizaje automático han demostrado ser prometedoras para mejorar la precisión de las predicciones meteorológicas, pero es necesario comparar y evaluar diferentes algoritmos para determinar el enfoque adecuado. Además, el uso de software propietario para la predicción meteorológica puede resultar caro y limitar el acceso a los últimos avances en este campo. Por lo tanto, este estudio pretende explorar el potencial del software de código abierto como alternativa rentable y flexible para desarrollar modelos de predicción de aprendizaje automático para la radiación solar, para comparar y a su vez evaluar las diferentes técnicas de aprendizaje automático determinando la técnica de predicción adecuada [1].

Este estudio tiene como objetivo principal comparar y evaluar distintas técnicas de aprendizaje automático como: la regresión lineal simple, unidades recurrentes cerradas (GRU) y redes neuronales, con el fin de determinar el enfoque eficaz para predecir la radiación solar. Además, desarrollar un sistema de predicción en software de código abierto que pueda reproducirse para otros tipos de variables.

2.4. BENEFICIARIOS

2.4.1. Beneficiarios Directos

Empresas e instituciones dedicadas a la generación fotovoltaica.

2.4.2. Beneficiarios Indirectos

Los beneficiarios indirectos son estudiantes de ingeniería y energía, docentes, el Ministerio de Electricidad y Energía Renovable del Ecuador, entidades que utilizan datos para la planificación del sector energético e instituciones de investigación.

2.5. HIPÒTESIS

¿Cuál de los tres modelos de aprendizaje automático comparados resultará ser el más óptimo en términos de precisión para el pronóstico de la radiación solar?

2.6. OBJETIVOS

2.6.1. General

Comparar modelos de pronóstico para la variable de radiación solar utilizando diversas técnicas de aprendizaje automático y software de código abierto, para determinar la precisión de las predicciones.

2.6.2. Específicos

- Revisar el estado del arte referente a las técnicas de aprendizaje automático, modelos de pronósticos y uso de software de código abierto.
- Construir los modelos de pronóstico utilizando los parámetros de la radiación solar.
- Comparar los resultados de los modelos de pronóstico con los datos reales para determinar la precisión de las predicciones.

2.7. SISTEMA DE TAREAS

Objetivos específicos	Actividad (tareas)	Resultados Esperados	Técnicas, Medios e Instrumentos
<p>Revisar el estado del arte referente a las técnicas de aprendizaje automático, modelos de pronósticos y uso de software de código abierto.</p>	<ul style="list-style-type: none"> • Revisión el estado del arte referente a modelos de pronóstico. • Revisión el estado de arte referente a técnicas de aprendizaje automático • Redacción 	<ul style="list-style-type: none"> • Fuentes de información sobre modelos de pronóstico. • Tipos de técnicas de aprendizaje automático. • Elaboración del informe con los recursos recopilados 	<ul style="list-style-type: none"> • Artículos científicos, tesis, libros • Artículos científicos, tesis, libros • Formatos otorgados
<p>Construir los modelos de pronóstico utilizando los parámetros de la radiación solar.</p>	<ul style="list-style-type: none"> • Análisis de la base de datos. • Preprocesamiento de datos. • Construcción de los modelos de pronóstico. 	<ul style="list-style-type: none"> • Determinación de la fiabilidad de los datos disponibles. • Limpieza de datos incompletos • Archivo ejecutable del software Python. 	<ul style="list-style-type: none"> • Data de la Universidad Nacional de Piura. • Librerías de Python. • Software Python.
<p>Comparar los resultados de los modelos de pronóstico con los datos reales para determinar la precisión de las predicciones.</p>	<ul style="list-style-type: none"> • Análisis de resultados • Comparación de los resultados obtenidos. 	<ul style="list-style-type: none"> • Análisis de las métricas de evaluación para comprender la precisión de las predicciones • Informe de resultados. 	<ul style="list-style-type: none"> • Intercambio de datos. • Porcentaje de error.

3. FUNDAMENTACIÓN TEÓRICA

3.1. ANTECEDENTES

Castañeda G, Fabara F [2], en su investigación cuyo objetivo principal es la predicción de la radiación solar para el dimensionamiento de plantas fotovoltaicas en la provincia de Pichincha, en cual su principal enfoque, está basado en redes neuronales artificiales utilizando datos reales de tres años de radiación solar. La metodología abarca enfoques descriptivos y experimentales, con herramientas de programación y simulación. La implementación del modelo de predicción se realiza en Python.

Segovia J, Toaquiza J, Llanos J, Rivas D, describe en su documento [1], las técnicas de predicción de variables meteorológicas son muy estudiadas ya que su conocimiento previo permite la gestión eficiente de las energías renovables, y también para otras aplicaciones de la ciencia como la agricultura, la salud, la ingeniería, la energía, etc. En esta investigación se ha estudiado el diseño, implementación y la comparación de modelos de predicción de variables meteorológicas se ha realizado mediante diferentes técnicas de Machine Learning como parte del software de código abierto Python. Para identificar la mejor técnica, se utilizan el error cuadrático medio (RMSE), el error porcentual absoluto medio (MAPE), el error absoluto medio (MAE) y el coeficiente de determinación (R^2) se utilizan como métricas de evaluación.

Aguirre G [3], en su investigación “Series temporales en variaciones aleatorias de radiación solar”, propone un algoritmo confiable para analizar amplitudes impredecibles en series temporales observadas en la naturaleza. El enfoque algorítmico implica la correlación de componentes típicos, como efectos de tendencia temporal, componentes periódicos y lecturas secuenciales, así como la identificación de los efectos de interacción de estos factores. Se realiza un estudio de simulación para verificar la confiabilidad del algoritmo propuesto, el cual luego se aplica a un conjunto de datos de radiación solar observada en el área metropolitana de Guadalajara. Los datos provienen de estaciones automatizadas en zonas urbanas y forestales que abarcan once años y se leen cada diez minutos. Los resultados destacan signos de atenuación a largo plazo en las zonas urbanas, mientras que ocurre lo contrario en las zonas semiurbanas. Además, encontramos que los cambios a corto plazo muestran los efectos de factores como efectos cíclicos o tendencias a largo plazo, en conclusión, el modelo de Weibull se ajusta mejor a los datos observados en ambas áreas de investigación.

Mendaña I, en su trabajo final de grado [4], “Predicción de radiación solar a partir de modelos físicos y aprendizaje automático”, cuyo objetivo de la investigación es comprobar si

los algoritmos de inteligencia artificial pueden mejorar los resultados obtenidos con los modelos físicos tradicionales utilizados para predecir la radiación solar. Los conjuntos de datos se utilizan para aplicar transformaciones para construir modelos básicos y experimentales. Luego se utilizan técnicas de aprendizaje automático con diferentes algoritmos para evaluar la efectividad de sus predicciones y determinar si representan una mejora con respecto al modelo de referencia. Los resultados del análisis muestran mejoras significativas en comparación con los modelos de referencia genéricos, lo que indica que la incorporación de algoritmos de inteligencia artificial puede mejorar con éxito las predicciones de radiación solar global.

3.2. ENERGÍA

La energía es aquello capaz de producir un cambio en la materia. De acuerdo con el principio de conservación, esta no se crea ni se destruye, solo se transforma y puede transferirse entre objetos y cuerpos como se muestra en la Figura 3.1. Según la forma o el sistema físico en que se manifiesta, se consideran diferentes formas de energía: térmica, mecánica, eléctrica, química, electromagnética, nuclear, luminosa [5].

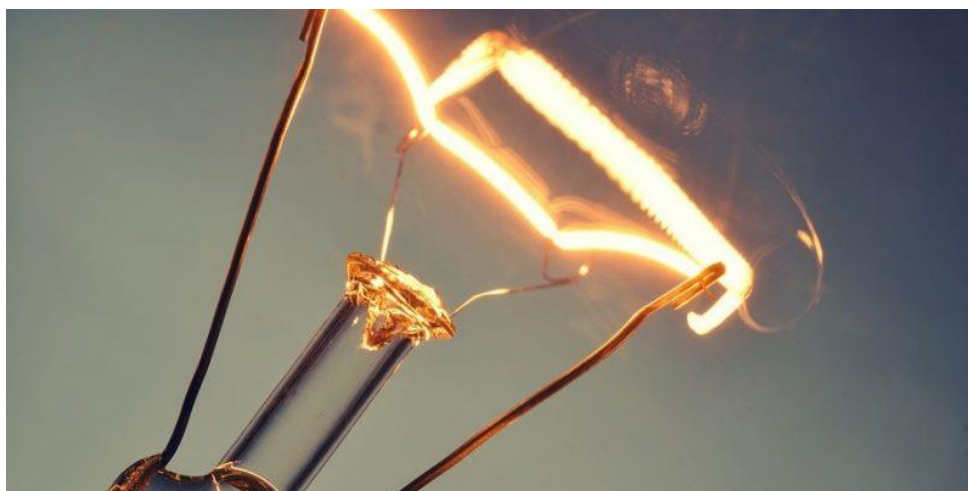


Figura 3.1. Energía [6].

3.2.1. Tipos de energía

Existen dos tipos de energías:

- Energías renovables: Estos son recursos puros y casi infinitos que nos brinda la naturaleza. Por su carácter autóctono, ayudan a nuestro país a reducir la dependencia del abastecimiento exterior [7]. En la Figura 3.2, se describe los diferentes tipos de energía renovable.

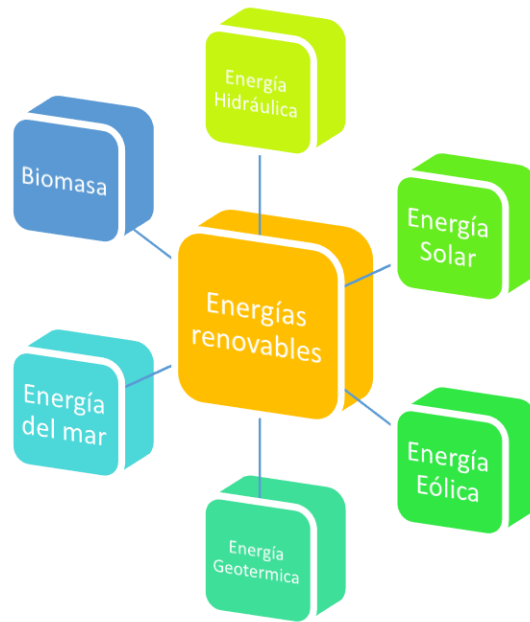


Figura 3.2. Tipos de energías renovables.

- Energías no renovables: hacen referencia a las formas de energía convencionales, aquellas cuyos recursos de los que se obtienen son limitados, es decir, que no se auto regeneran como las energías renovables [8]. A continuación, en la Figura 3.3, se detalla los tipos de energía no renovables.

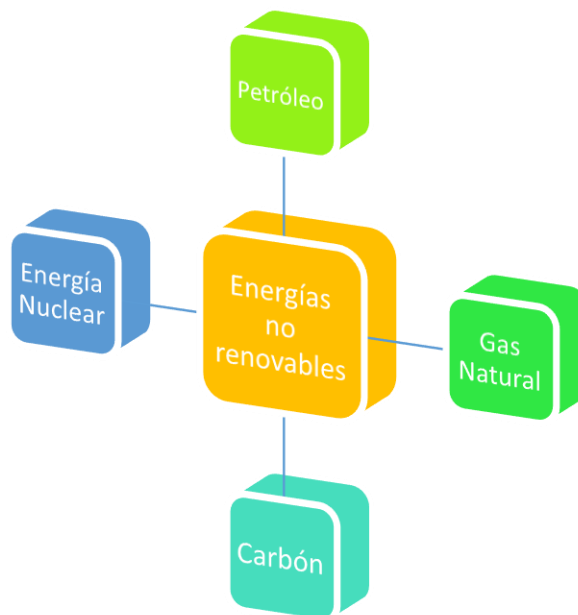


Figura 3.3. Tipos de energía no renovables.

En 2021, Brasil fue el país con la mayor capacidad de producción de energías renovables en América Latina y el Caribe como se puede apreciar en la Figura 3.4, con una capacidad de aproximadamente 159.000 megavatios. México ocupó el segundo lugar con 14,89 megavatios. La capacidad instalada total de energía renovable en la región superó los 291,7 gigavatios en 2021 [9].

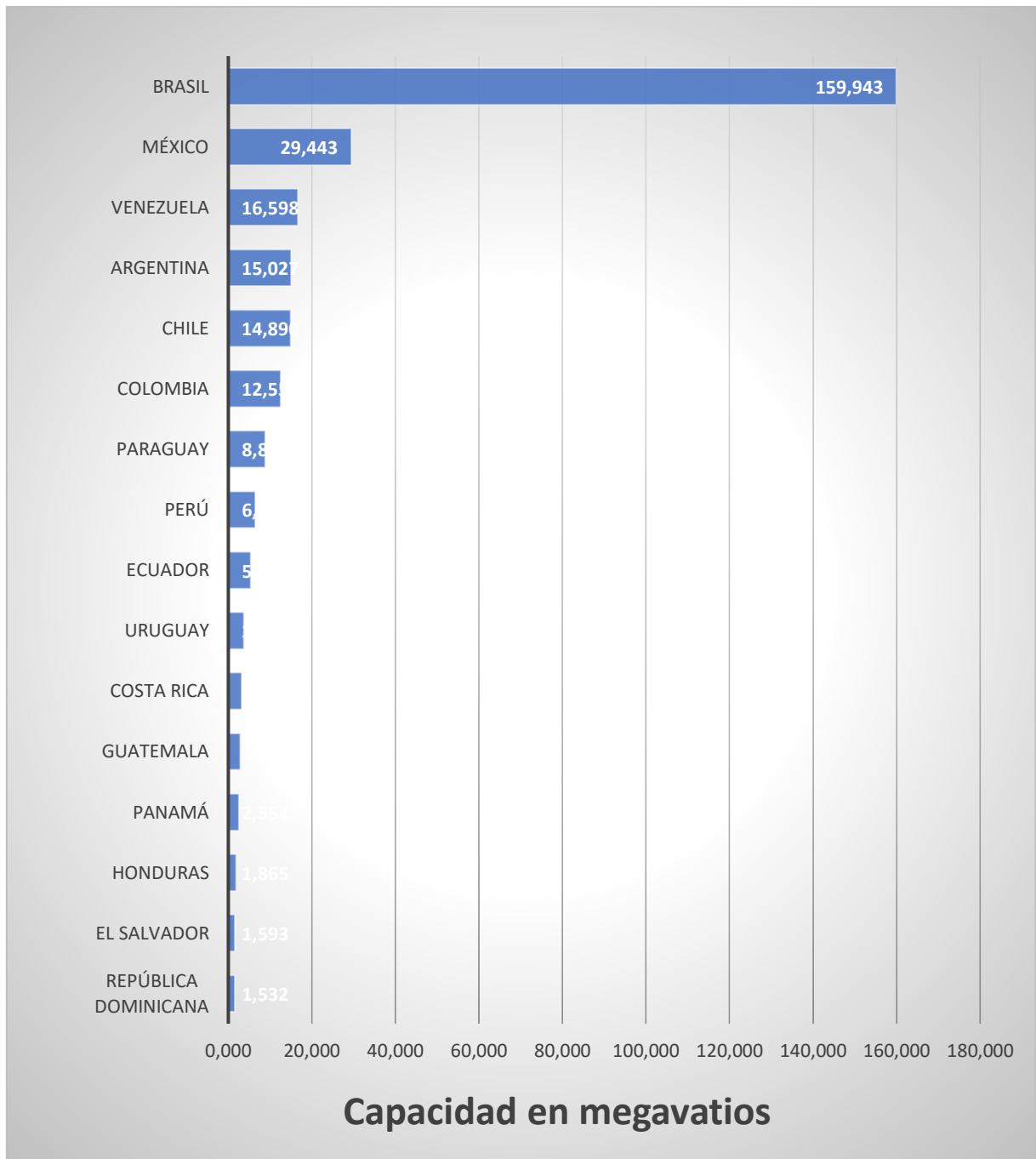


Figura 3.4. Capacidad de producción de energías renovables en América Latina y el Caribe en 2021.

3.3. ENERGÍA SOLAR

Este tipo de energía proporciona el sol en forma de radiación electromagnética (luz, calor y rayos ultravioleta principalmente).

El uso de la energía del sol se puede derivar en energía solar térmica (usada para producir agua caliente de baja temperatura para uso sanitario y calefacción) solar fotovoltaica (a través de placas de semiconductores que se alteran con la radiación solar), etc. La energía solar es una de las fuentes de energía inagotable, su aplicación es variada, por ende, debe ser explotada al máximo con el fin de mejorar la matriz energética como se hace referencia en [10].

3.4. RADIACIÓN SOLAR

El aprovechamiento de la energía solar, como fuente de energía útil tiene un doble objetivo: por un lado, ahorrar en energías no renovables, sobre todo energías fósiles y energías nucleares, y por otro amortiguar el impacto ambiental generado por ellas.


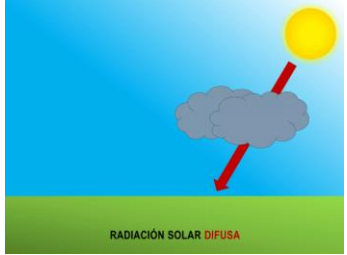
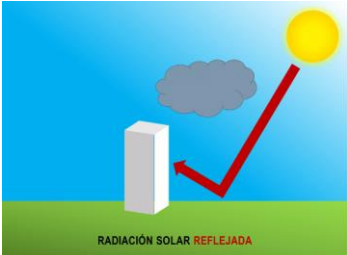
Mendaña [4], considera que la radiación solar es el conjunto de ondas electromagnéticas emitidas por el sol. Esta radiación puede ser medida en el espacio o en la superficie terrestre. Si es medida en la superficie terrestre hay factores que afectan a su valor, como en qué posición se encuentra el sol en ese momento, o las condiciones climáticas.

Si se aprovecha esta energía, de la mejor manera posible, se debe conocer bien algunas de sus características más importantes, tales como: horas del sol para situación geográfica determinada, trayectoria aparente del sol respecto a un punto situado sobre la superficie de la Tierra, cantidad de energía recibida por metro cuadrado y unidad de tiempo, energía recibida a lo largo de un día, mes o año [11].

3.4.1. Tipos de radiación solar

La radiación solar superficial se caracteriza por variaciones temporales, algunas aleatorias, como la nubosidad, y otras pre-visibles, como los cambios estacionales del día y la noche provocados por el movimiento de la Tierra. A continuación, en la Tabla 3.1, se detallará sobre los tipos de radiación solar.

Tabla 3.1. Tipos de radiación solar

Tipo	Característica	Grafico
Radiación directa	La forman los rayos recibidos directamente del Sol [12].	 <p>RADIACIÓN SOLAR DIRECTA</p> <p>Figura 3.5. Radiación solar directa.</p>
Radiación difusa	La radiación difusa procede de toda la bóveda de cielo, excluyendo el disco solar, la forma los rayos dispersados por la atmósfera en dirección al receptor (por ejemplo, en un día completamente nublado toda la radiación es difusa) [12].	 <p>RADIACIÓN SOLAR DIFUSA</p> <p>Figura 3.6. Radiación solar difusa.</p>
Radiación reflejada	Como su propio nombre lo dice es reflejada por la superficie terrestre hacia el receptor. Depende directamente por la naturaleza de las montañas, lagos, edificios, etc. Que rodean al receptor [12].	 <p>RADIACIÓN SOLAR REFLEJADA</p> <p>Figura 3.7. Radiación solar reflejada.</p>
Radiación global	La suma de todas las radiaciones descritas anteriormente se denomina radiación global, es decir, la radiación solar total que recibe la superficie, por lo que es necesario conocerla y cuantificarla [12].	

3.5. HORAS SOLAR PICO

Más comúnmente abreviada (HSP) (Wh/m²/día) es empleada para considerar cuanta energía diaria podría producirse en un panel solar en un metro cuadrado de superficie, está relacionado con la cantidad de insolación global total (kWh/m²/día) que incide en el día sobre los módulos solares, en teoría es un concepto que define la equivalencia entre la cantidad diaria en horas en las que el panel está expuesto (varía dependiendo la inclinación del arreglo fotovoltaico) con una radiación de 1000 W/m² y la radiación total diaria que recibe [13].

Las horas solar pico se puede expresar de la siguiente manera como se muestra en la ecuación (3.1):

$$HSP_{\beta} = \frac{R_{\beta}}{I_{\beta}} \quad (3.1)$$

Donde:

HSP_{β} son las horas solar pico según la inclinación β

R_{β} corresponde a la radiación media diaria expresada en Wh/m²/día según si inclinación β

I_{β} corresponde a la constante equivalente de 1 kW/m²

3.6. PRONÓSTICO SOLAR

Un pronóstico solar fotovoltaico tiene como principal objetivo estimar la irradiancia solar incidente y posteriormente la potencia eléctrica generada, en un intervalo de tiempo, denominado horizonte de pronóstico, en base a los datos observados durante el período previo al intervalo deseado. Existen pronósticos a largo plazo, comprendidos en horizontes de un día en adelante, pronósticos de corto - mediano plazo con horizontes que van desde 1-6 horas y pronósticos de corto plazo o también llamados inmediatos, comprendidos en intervalos menores a 1 hora, usualmente entre 15 a 30 minutos. Es importante mencionar que mientras más largo sea el horizonte de pronóstico, menor será la precisión del algoritmo [14].

Los pronósticos solares son requeridos para abordar las necesidades de planificación, operación y equilibrio de las redes de transmisión eléctrica. Los pronósticos solares a corto plazo intra-horarios son particularmente útiles para labores operacionales en el sector, como la toma de decisiones en centrales eléctricas en tiempo real, el equilibrio de la red, despacho de unidades móviles y activación de otras fuentes de energía ante posibles fluctuaciones en el recurso solar que provoquen incumplimiento en la demanda energética, además de influir en

otros aspectos tales como el comercio energético. Por otro lado, los pronósticos con horizontes de tiempo más largos son de interés para empresas de servicios públicos, además de mejorar el rendimiento de la unidad encargada del equilibrio de la red, pues obtienen una mirada hacia el futuro, que, si bien no es tan precisa como en un pronóstico a corto plazo, permite mantener un registro aproximado de la irradiancia o potencia eléctrica, tal que el balance no requiera una variación excesiva y por tanto que el tiempo de respuesta sea más corto [15].

En resumen, la predicción de irradiancia solar es una tecnología que permite la integración de un nivel cada vez mayor de energía solar en la matriz energética, pues mejora la calidad de la energía entregada a la red y reduce los costos secundarios asociados con la dependencia del clima. La combinación de estos dos factores ha sido la motivación impulsora para el desarrollo de un complejo campo de investigación que apunta a producir mejores capacidades de pronóstico solar.

3.7. APRENDIZAJE AUTOMÁTICO

El aprendizaje automático es una subcategoría de inteligencia artificial que se refiere al proceso por el cual los PC desarrollan el reconocimiento de patrones o la capacidad de aprender continuamente y realizar predicciones basadas en datos, tras lo cual realizan ajustes sin haber sido programados específicamente para ello [16].

El aprendizaje automático o machine learning (ML) es el campo de estudio encargado de desarrollar algoritmos que permiten a las computadoras extraer e identificar patrones de los datos. El aprendizaje automático posee una gran variedad de aplicaciones en la sociedad moderna, donde se encuentran principalmente los motores de búsqueda, análisis de mercado para los diferentes sectores de actividad, aplicaciones en diagnósticos médicos, detección de fraude en el uso de tarjetas de crédito, predicción y análisis de datos [6].

La gran mayoría de algoritmos de ML pueden ser clasificados en dos categorías, aprendizaje supervisado y aprendizaje no supervisado.

El objetivo del aprendizaje supervisado es generar una función a partir de los datos de entrenamiento, la cual se encarga de mapear desde los valores de los atributos que describen una instancia del valor, como puede ser los datos de entrada, hasta el valor de otro atributo, conocido como atributo de salida o de destino, es decir, los resultados. La función creada busca ser capaz de predecir el valor de cualquier atributo de entrada, a partir de un conjunto de datos de ejemplo, denominados datos de entrenamiento, para lo cual debe generalizar patrones usando

estos datos, y así, aplicarlos a los atributos de entrada que no han sido pre visualizados, también denominados datos de prueba o testeo.

El aprendizaje supervisado opera mediante la búsqueda en una variada gama de funciones diferentes, con la finalidad de encontrar la que mejor representa la variación entre atributos de entrada y de salida. No obstante, en cualquier conjunto de datos de carácter más complejo, la cantidad de posibles combinaciones entre atributos de entrada y sus correspondientes caminos hacia el atributo de salida escala a un número muy elevado, tal que el algoritmo no puede probar todas las funciones posibles, por ende, un algoritmo de aprendizaje automático se diseña para preferir ciertos tipos de funciones mientras lleva a cabo su búsqueda. Dichas preferencias se conocen como sesgo de aprendizaje.

Por otro lado, el aprendizaje no supervisado se diferencia de su contraparte, en que no hay un atributo de salida definido. Esto implica que se trabaje únicamente con los atributos de entrada, los cuales muchas veces se trabajan como variables aleatorias, construyendo funciones de densidad para el conjunto de datos.

Una consecuencia de esto es que los algoritmos de aprendizaje no supervisado pueden ser aplicados sin tener que invertir tiempo en etiquetar las instancias del conjunto de datos con un atributo de salida. No obstante, esto implica que el proceso de aprendizaje, es decir, la inferencia de patrones, sea más compleja [14].

3.7.1. Funcionamiento del aprendizaje automático

El aprendizaje automático es increíblemente complejo y el modo de funcionar varía según la tarea y el algoritmo utilizado para lograrla. Sin embargo, básicamente, un modelo de aprendizaje automático es un PC que analiza datos e identifica patrones, y luego usa esos conocimientos para completar mejor la tarea asignada. Cualquier tarea que se base en un grupo de puntos de datos o normas puede automatizarse con el uso del aprendizaje automático [16].

3.8. MODELOS DE APRENDIZAJE AUTOMÁTICO

En este punto de la investigación se desarrollará la explicación de los tres métodos de aprendizaje automático empleados para la generación de pronósticos de radiación de solar interés para el trabajo de titulación.

3.8.1. Modelo de regresión lineal simple

3.8.1.1. Regresión lineal

La regresión lineal es un modelo matemático y un procedimiento estadístico, utilizado para modelar la relación entre una variable dependiente (respuesta) y una o más variables independientes (predictores). En su forma más simple, la regresión lineal asume que la relación entre las variables puede ser aproximada por una línea recta. Este tipo de modelos es utilizado en campos científicos y en negocios, sin embargo, en las últimas décadas se ha implementado en el área de aprendizaje automático [17].

La regresión lineal en el ámbito de aprendizaje automático desempeña un papel importante que consiste en predecir un parámetro salida (Y) a partir de un parámetro de entrada (X).

3.8.1.2. Importancia de la regresión lineal

Los modelos de regresión lineal son muy populares en diversos campos de investigación debido a su velocidad y facilidad de interpretación. Gracias a su capacidad para transformar datos, se pueden utilizar para modelar muchas relaciones y a su forma más simple que las redes neuronales, sus parámetros estadísticos se pueden analizar y comparar fácilmente, lo que permite extraer información valiosa [18].

La regresión lineal no sólo se utiliza con fines de pronosticar: también se ha demostrado que es eficaz para describir sistemas. Si desea modelar los valores de una sola variable, tendrá una lista relativamente corta de variables independientes y, dado que espera que el modelo sea interpretable.

3.8.1.3. Regresión lineal simple

La regresión lineal implica establecer una relación entre una variable independiente y su correspondiente variable dependiente. Esta relación está representada por una línea recta. Es imposible trazar una línea recta que pase por todos los puntos de la gráfica si están colocados al azar. Por lo tanto, la ubicación óptima de esta línea está determinada únicamente por regresión lineal. Algunos puntos todavía estarán lejos de la línea, pero esta distancia será mínima. Calcular la distancia mínima desde la línea hasta cada punto se llama función de pérdida [19].

La regresión lineal simple es un caso especial de regresión lineal en el que hay una única variable independiente (x) que se utiliza para predecir una variable dependiente (y). A continuación, en el Figura 3.8, se representa un ejemplo del modelo de regresión lineal simple.

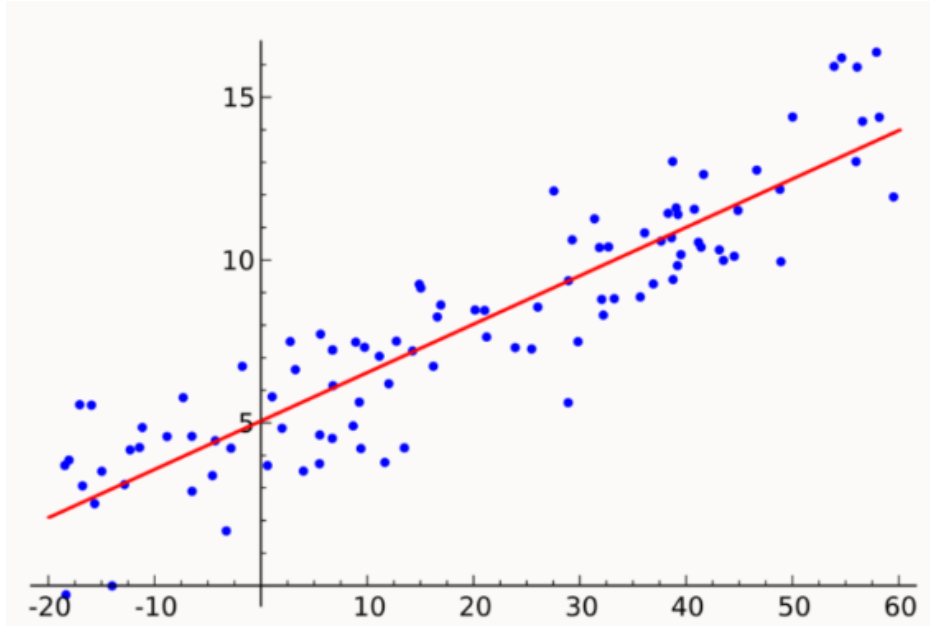


Figura 3.8. Regresión lineal simple

La fórmula general de la regresión lineal simple se representa en la ecuación (3.2).

$$Y = \beta_0 + \beta_1 X + \varepsilon \quad (3.2)$$

Donde:

Y es la variable dependiente.

X es la variable independiente.

β_0 es el intercepto (la intersección en el eje y).

β_1 es la pendiente de la línea (tasa de cambio de Y respecto a X).

ε es el error, que es la diferencia entre el valor observado Y y el valor predicho por el modelo.

3.8.2. Modelo Gated Recurrent unit (GRU)

Unidades recurrentes cerradas (GRU) también conocidas como “Gated Recurrent unit”, fue desarrollada para resolver problemas comunes de gradientes desvanecidos, además se considera a la GRU como una versión más avanzada de la LSTM, esto se debe a que sus diseños idénticos y sus resultados excelentes [20].

Este modelo permite profundizar el procesamiento debido a que son capaces de obtener relaciones de tendencias no lineales y patrones complejos de datos como se puede ver en la Figura 3.9, es recurrente en este tipo de proyectos. Los datos históricos que se desea manejar pueden llegar a tener vacíos en su secuencia con el modelo mencionado los errores son minimizados ya que puede predecir lagunas de datos.

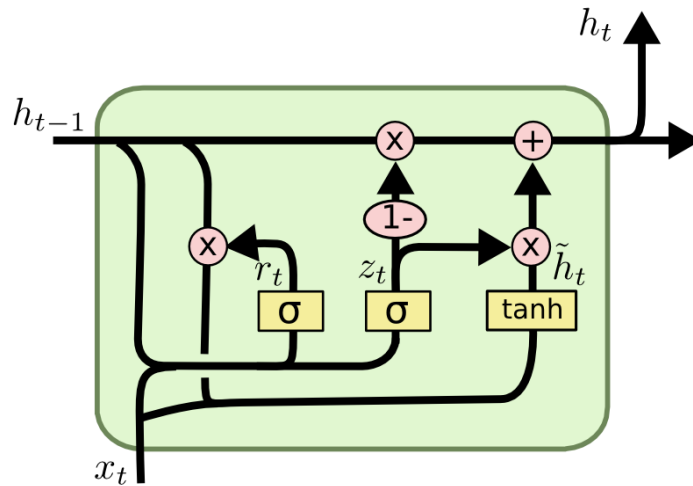


Figura 3.9. Unidades recurrentes cerradas [21]

Ecuación característica de las unidades recurrentes cerradas:

$$z_t = \sigma(W_z[h_{t-1}, x_t] + b_z) \quad (3.3)$$

$$r_t = \sigma(W_r[h_{t-1}, x_t] + b_r) \quad (3.4)$$

$$\tilde{h}_t = \tanh(W_h[r_t \odot h_{t-1}, x_t] + b_h) \quad (3.5)$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t \quad (3.6)$$

Donde:

h_t vectores de capa oculta

x_t vector de entrada

b_z, b_r, b_h vector de sesgo

W_z, W_r, W_h matrices de parámetros

σ, \tanh funciones de activación

3.8.3. Redes Neuronales

La red neuronal artificial es un método que puede resolver problemas individualmente o en combinación con otros métodos para clasificar, identificar y diagnosticar problemas donde el equilibrio entre datos y conocimientos. Se pueden expresar en muchos campos diferentes como: neurociencia, matemáticas, estadística, física, informática e ingeniería. Las redes neuronales se utilizan en campos tan diversos como el modelado, el análisis de series temporales, el reconocimiento de patrones, el procesamiento de señales y el control debido a una propiedad importante: la capacidad de aprender de los datos de entrada, ya sean supervisados o no [22].

Una red neuronal es un modelo computacional inspirado en la estructura y funcionamiento del cerebro humano. Es parte del campo de la inteligencia artificial y se utiliza para realizar tareas de aprendizaje y reconocimiento de patrones. Las redes neuronales están compuestas por unidades básicas llamadas "neuronas" que están organizadas en capas y conectadas entre sí mediante "pesos sinápticos", como se puede observar en la Figura 3.10. Estas conexiones permiten que la información fluya a través de la red y se procese de manera no lineal.

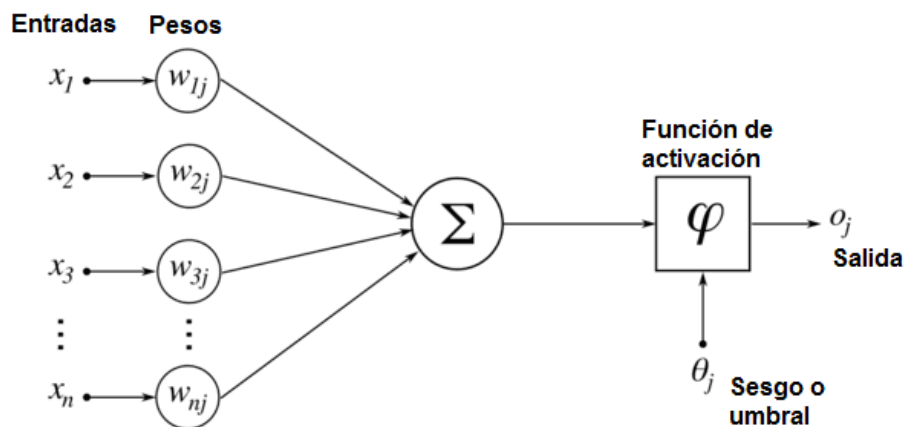


Figura 3.10. Arquitectura de una red neuronal [23].

3.8.3.1. Rol de una neurona

En las redes neuronales, las neuronas desempeñan un papel clave en el procesamiento de la información de entrada. Toma entradas ponderadas, calcula una suma ponderada de esas entradas, agrega un desplazamiento y pasa el resultado a través de una función de activación no lineal. Este proceso transforma los datos de entrada de forma no lineal, lo que permite a la red aprender patrones complejos, cada neurona contribuye a la capacidad de la red para modelar relaciones no lineales en los datos. Durante el entrenamiento, el peso y los sesgos de neurona

se ajustan para minimizar la diferencia entre la predicción y los valores reales, lo que ayuda a la red a adaptarse a este tipo de tareas [24].

3.8.3.2. Arquitectura de una red neuronal

Dependiendo de la tarea que la red deba realizar, puede haber diferentes arquitecturas. Aquí se describen algunas de las arquitecturas más comunes de redes neuronales:

- **Según el número de capas**

Redes neuronales monocapa: Cuando se menciona una "red neuronal monocapa" o "red neuronal de una capa" generalmente se refiere a una red de alimentación directa (feedforward) con una única capa de neuronas, la cual puede ser tanto la capa de entrada como la capa de salida, como se muestra en la Figura 3.11. Esta configuración simple se utiliza comúnmente en problemas de clasificación lineal, donde se trata de separar las clases con una línea recta en el espacio de entrada.

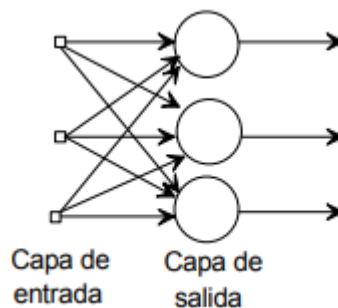


Figura 3.11. Red neuronal monocapa [25].

Redes neuronales multicapa: Una red neuronal multicapa (Multilayer Perceptron o MLP) es un tipo de red neuronal artificial organizada en capas, con al menos una capa de entrada, una o más capas ocultas y una capa de salida, como se muestra en la Figura 3.12. La presencia de capas ocultas le permite a la red aprender representaciones más complejas de los datos y abordar problemas no lineales [25].

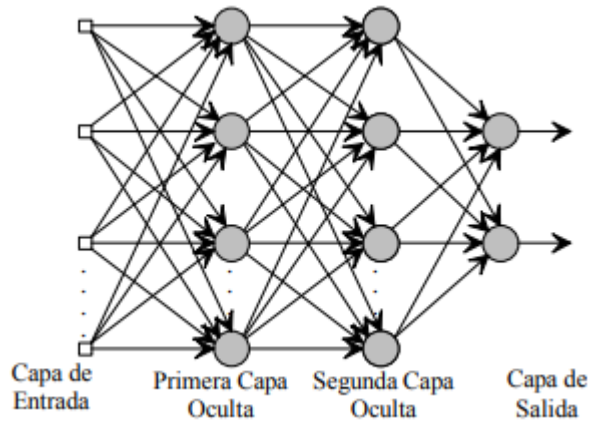


Figura 3.12. Red neuronal multicapa [25].

- **Según el tipo de conexión**

Redes neuronales no recurrentes: este tipo de red neuronal no tiene memoria, esta red no tiene posibilidades de retroalimentaciones ya que la propagación de las señales se produce solamente en un sentido.

Redes neuronales recurrentes: son un tipo de red neuronal que se ha diseñado específicamente para trabajar con datos secuenciales o temporales. A diferencia de las redes neuronales feedforward, las RNN tienen conexiones retroalimentadas que les permiten mantener información sobre estados anteriores y procesar secuencias de datos de longitud variable como se puede apreciar en la Figura 3.13. Esta estructura recurrente es adecuada para el estudio de sistemas no lineales [25].

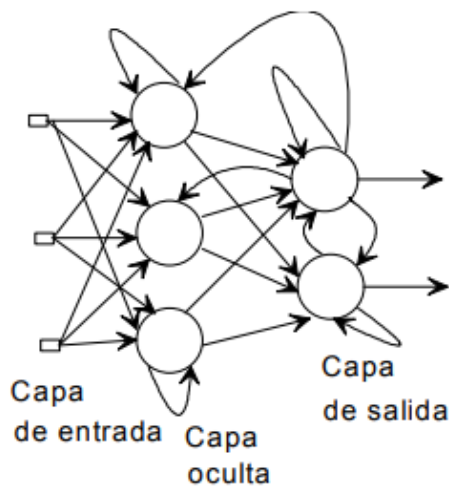


Figura 3.13. Red neuronal recurrente [25].

- **Según el grado de conexión**

Redes neuronales totalmente conectadas: este tipo de redes neuronales hace referencia a las neuronas de una capa el cual se encuentran conectadas a las capas siguientes (redes no recurrentes) o a su vez con las anteriores (redes recurrentes.)

Redes parcialmente conectadas: para simplificar en este caso no existe conexión total entre las neuronas de diferentes capas.

3.8.3.3. Modelamiento matemático de una red neuronal

El ingreso total hacia la neurona es calculado como la suma ponderada de todas las entradas ponderadas compuesto por ciertos valores. Para obtener la ecuación matemática de la red neuronal presentada en la Figura 3.10, se representa un grupo de variables de entradas $x_1 \dots x_2$, comúnmente conocida como las sinapsis de una neurona biológica, en el cual cada una de estas se encuentran multiplicadas por un peso al que está asociado $w_1 \dots w_n$ [26], como se muestra en la ecuación (3.7).

$$\sum_{i=1}^n w_1 x_1 + w_2 x_2 + \dots + w_n \quad (3.7)$$

Donde:

$x_1 \dots x_2$ representa las entradas de la red neuronal

$w_1 \dots w_2$ representa los pesos asociados a cada entrada

Al incorporar el sesgo (θ_j) a la ecuación (3.7), esta se reescribe en la ecuación (3.8), obteniendo una única entrada.

$$u = \sum_{i=1}^n w_{ij} x_i + \theta_j \quad (3.8)$$

Ahora al incorporar el tiempo en la red neuronal, especialmente si estás tratando con datos secuenciales o de series temporales, se modificar la expresión para que refleje la dependencia temporal. Una forma común de hacerlo es considerar $u(t)$ como la entrada neta en el tiempo t (observe la ecuación (3.9)), donde t representa algún punto en el tiempo [26].

$$u(t) = \sum_{i=1}^n w_{ij} x_i + \theta_j \quad (3.9)$$

La relación funcional entre la salida de una neurona en una red neuronal y su entrada neta en el cual la función de activación introduce no linealidades, permitiendo a la red aprender patrones más complejos en los datos. La entrada neta generalmente se calcula como la suma ponderada de las entradas más un sesgo. La ecuación (3.10), encapsula el proceso mediante el cual la red neuronal introduce no linealidades en la relación entre la entrada neta y la salida de una neurona.

$$y = f(u) \quad (3.10)$$

Donde:

y representa la salida de la neurona o unidad.

f es la función de activación aplicada a la entrada neta

(u) representa la entrada neta a la neurona, que a menudo es la suma ponderada de las entradas más un sesgo.

3.9. MÉTRICAS PARA LA EVALUACIÓN DE LOS MODELOS

El error es un criterio muy sencillo [27], básicamente es la diferencia que existe entre el resultado que nuestro modelo de predicción arroja y el valor real de la observación con la que estamos haciendo el test como indica la Figura 3.14.

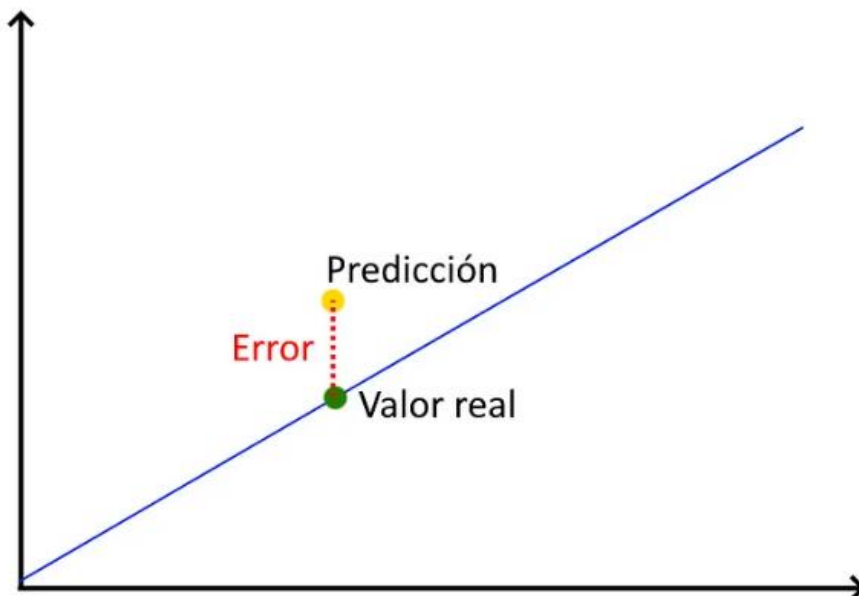


Figura 3.14. Error de una predicción [28].

A continuación, se detallará las diferentes métricas para determinar el error:

3.9.1. Error absoluto medio (MAE)

Esta métrica es una medida de la diferencia entre dos valores, lo que significa que nos permite saber en qué medida difiere el valor predicho del valor real u observado. Para evitar que los errores con valores positivos sean borrados por errores con valores negativos, utilizamos el valor absoluto de la diferencia. Como nos interesa preservar el error de todas las observaciones y no solo de una, obtenemos el promedio de los valores absolutos de la diferencia [29].

Fórmula para calcular el error absoluto medio (MAE) se denota en la ecuación (3.11).

$$MAE = \frac{\sum_{i=0}^n |y_i - x_i|}{n} \quad (3.11)$$

Donde:

y_i es el valor real.

x_i es el valor de la predicción.

n es el tamaño de muestra

Ventajas de la métrica de evaluación MAE:

- Fácil de calcular.
- Estos errores se ponderan en la misma balanza porque se toman valores absolutos.
- Estos son útiles si sus datos de entrenamiento contienen valores atípicos porque MAE no penaliza los errores grandes causados por valores atípicos.
- Proporciona una evaluación única del rendimiento del modelo.

3.9.2. Error cuadrático medio (MSE)

MSE (Mean Squared Error o Error Cuadrático Medio) es una métrica utilizada comúnmente en problemas de regresión para evaluar la calidad de un modelo predictivo. Mide el promedio de los cuadrados de las diferencias entre las predicciones del modelo y los valores reales del conjunto de datos [30].

A continuación, en la ecuación (3.12), se describe la formula del error cuadrático medio (MSE):

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \tilde{y}_i)^2 \quad (3.12)$$

Donde:

n es el número total de observaciones en el conjunto de datos.

y_i es el valor real de la observación i .

\tilde{y}_i es la predicción correspondiente para la observación i .

Ventajas del error cuadrático medio:

- El MSE castiga de manera significativa los errores grandes debido a la naturaleza cuadrática de la fórmula [31].
- Es una métrica diferenciable, lo cual es crucial en la optimización de modelos mediante algoritmos de descenso de gradiente.
- El MSE proporciona una medida cuantitativa de la dispersión de las predicciones en comparación con los valores reales
- El uso del cuadrado en la fórmula tiene propiedades matemáticas favorables, como la convexidad, que facilitan la optimización y el análisis matemático [31].

3.9.3. R cuadrado (R^2)

R^2 es el coeficiente de determinación e indica cuánta variación en la variable dependiente se puede predecir en función de la variable independiente. En otras palabras, qué tan bien coincide el modelo con las observaciones reales que tenemos. Cuando usamos R^2 , todas las variables independientes que tenemos en el modelo contribuyen a su valor [32].

El mejor valor posible de R^2 es 1 y el peor valor posible es 0. La desventaja de este enfoque es que supone que cada variable ayuda a explicar la variación en el pronóstico, lo que no siempre es así. Si se agrega otra variable, el valor de R^2 aumentará o permanecerá igual pero nunca disminuirá.

Fórmula para calcular el R cuadrado (R^2), se representa en la ecuación (3.13):

$$R^2 = 1 - \frac{\sum (y_i - x_i)^2}{\sum (y_i - \mu_y)^2} \quad (3.13)$$

Donde:

R^2 coeficiente de determinación.

y_i valor de la variable dependiente.

x_i valor aproximado que el modelo de regresión arroja.

μ_y medida de la variable dependiente.

3.9.4. R^2 cuadrado ajustado

R^2 Ajustado compensa la desventaja de R^2 con la integración de variables, esto se logra al penalizar la integración de variables independientes al modelo.

Fórmula para calcular R^2 ajustado, se representa en la ecuación (3.14):

$$R_{ajustado}^2 = 1 - \frac{(1 - R^2)(N - 1)}{N - M - 1} \quad (3.14)$$

Donde:

N es el número de filas.

M el número de columnas o variables.

3.10. ANÁLISIS DE CORRELACIÓN DE VARIABLES PARA CONSTRUIR MODELOS EXPLICATIVOS Y PREDICTIVOS

A menudo se utiliza el análisis de correlación para observar y medir la relación entre dos variables numéricas. Es uno de los métodos más comunes de análisis de datos y es un primer paso necesario antes de crear un modelo explicativo o predictivo.

3.10.1. Correlación de variables

La correlación es la relación entre dos variables numéricas, ya que evalúa específicamente la tendencia (mayor o menor) de los datos. Dos variables están relacionadas si una variable da información sobre la otra. En cambio, en ausencia de correlación, un aumento o disminución en una variable no indica nada sobre el comportamiento de la otra.

Dos variables están correlacionadas si tienden a aumentar o disminuir.

3.10.1.1. Matriz de correlación

La matriz de correlación es una matriz que contiene coeficientes de correlación en las posiciones i, j entre las variables i y j como se puede apreciar en la ecuación (3.15), por lo que la matriz de correlación es una matriz cuadrada cuyo diagonal principal asume el valor de uno,

incluidos los elementos de la fila i y la columna j , el cual consiste el valor del coeficiente de correlación entre la variable i y la variable j [33].

$$R = \begin{pmatrix} 1 & r_{12} & r_{13} & \cdots & r_{1n} \\ r_{21} & 1 & r_{23} & \cdots & r_{2n} \\ r_{31} & r_{32} & 1 & \cdots & r_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ r_{n1} & r_{n2} & r_{n3} & \cdots & 1 \end{pmatrix} \quad (3.15)$$

Donde:

r_{ij} es el coeficiente de correlación entre las variables i y j .

3.10.2. Interpretación de la correlación de variables

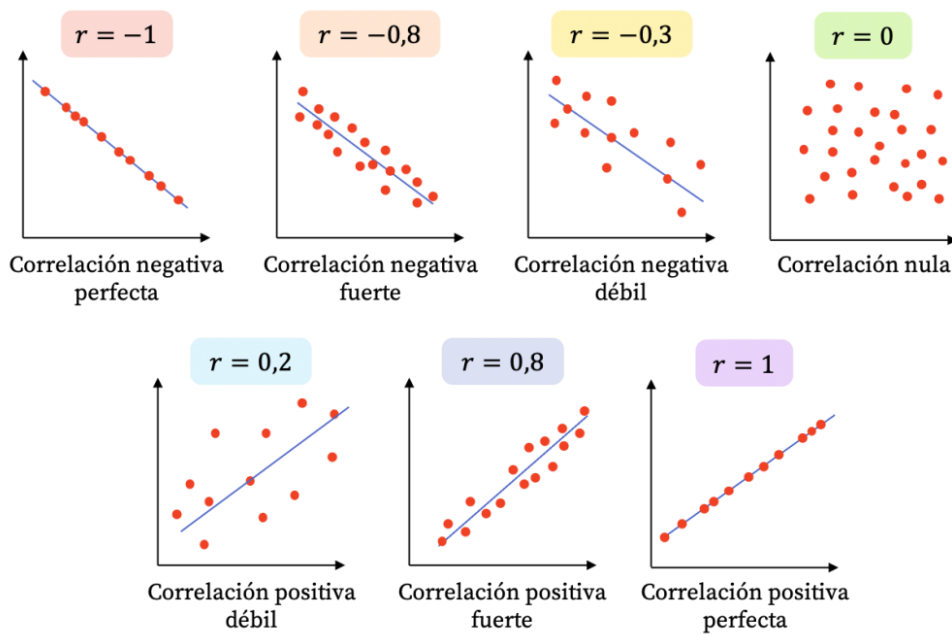


Figura 3.15. Interpretación de correlación mediante gráficos de dispersión [34].

En la Figura 3.15, se observan diferentes tipos de resultados del coeficiente de correlación y sus diagramas de dispersión correspondientes, en cual indica que la correlación permite medir el signo y magnitud de la tendencia entre dos variables.

La dirección de la relación nos indica el signo de un valor, como se observó en el diagrama de dispersión (ver la Figura 3.15).

- El valor positivo indica relación directa o positiva,
- El valor negativo indica una relación indirecta, inversa o negativa,

- En cambio, el valor nulo indica que no existe una tendencia entre ambas variables esto quiere decir que, puede ocurrir que no exista relación o que la relación sea más compleja que una tendencia.

El valor representa la fuerza de la relación y varía de -1 a 1. Cuanto más cerca esté el valor de los valores extremos del rango (1 o -1), en este caso más fuerte será la tendencia de las variables o menor existirá varianza en puntos en torno a una tendencia particular. Cuanto más cercano a 0 esté el coeficiente de correlación, más débil será la tendencia, es decir, mayor será la diferencia en la nube de puntos [35].

- Si la correlación vale 1 o -1 la correlación es “perfecta”,
- Si la correlación vale 0 las variables no están correlacionadas.

3.10.3. Medición de la correlación de variables

Los coeficientes de correlación utilizados para medir la correlación son; El coeficiente de correlación lineal de Pearson se utiliza para cuantificar tendencias lineales, y el coeficiente de correlación de Spearman se utiliza para aumentar o reducir tendencias que no son necesariamente lineales sino monótonas, es decir, las variables tienden a moverse en la misma dirección relativa pero no siempre a un ritmo constante.

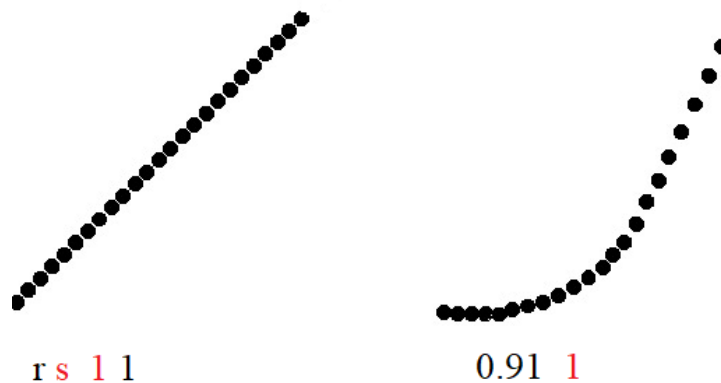


Figura 3.16. Representación gráfica de la relación lineal y no lineal (monótona).

En la Figura 3.16, se representa la “r” en color negro el coeficiente Pearson y la “s” de color rojo el coeficiente Spearman. Cuando los dos coeficientes tienen un valor de 1 la relación es lineal, cuando no es lineal la relación el coeficiente de correlación Spearman indica mejor la relación entre dos variables [35].

- Para medir la tendencia lineal entre dos variables se utiliza el coeficiente de correlación Pearson.

- En cambio, para medir la tendencia monótona (aumentando o disminuyendo) entre dos variables se utiliza el coeficiente de correlación Spearman.

3.11. SOTFWARE DE CODIGO ABIERTO

El software de código abierto ha evolucionado significativamente desde que Richard Stallman fundó el Proyecto GNU en los años 1970 y 1980. Con el lanzamiento del kernel de Linux en 1991, el software libre se hizo popular y dominó industrias clave como servidores y aplicaciones empresariales en la década de 2000. En la era móvil, Android basado en Linux se ha convertido en el sistema operativo dominante para dispositivos móviles. En la década de 2020, el software libre se ha convertido en una fuerza importante en inteligencia artificial, penetrando en todos los ámbitos, desde la investigación científica hasta la atención sanitaria. Además, se centra en aspectos éticos, privacidad y seguridad, lo que ha llevado a su adopción e integración generalizada en la infraestructura tecnológica global [36].

El software de código abierto es importante porque fomenta la innovación, la transparencia y el acceso igualitario a la tecnología, permitiendo a individuos y organizaciones utilizar herramientas poderosas sin las limitaciones de costos o restricciones de licencia. En los últimos años el software de código abierto que más fuerza a tomado es Python el cual utilizaremos para la construcción de los modelos de predicción.

3.11.1. Lenguaje de programación

El mundo IT (empresa dedicada al desarrollo de software y de soluciones tecnológicas para su comercio o industria) está en constante evolución, un ejemplo típico es la aparición de lenguajes de programación. Python es uno de los lenguajes más utilizados en la actualidad y esta tendencia crece constantemente [37].

Cada vez más personas en todo el mundo lo eligen como su primer lenguaje de programación, e incluso los programadores de mayor edad recurren a él por una razón principal: su sintaxis permite a los programadores expresar y lograr su problema con menos líneas de código. Python se está convirtiendo en uno de los lenguajes de programación más utilizados, desde aplicaciones web hasta ciencia de datos. Sus aplicaciones son muy diversas.

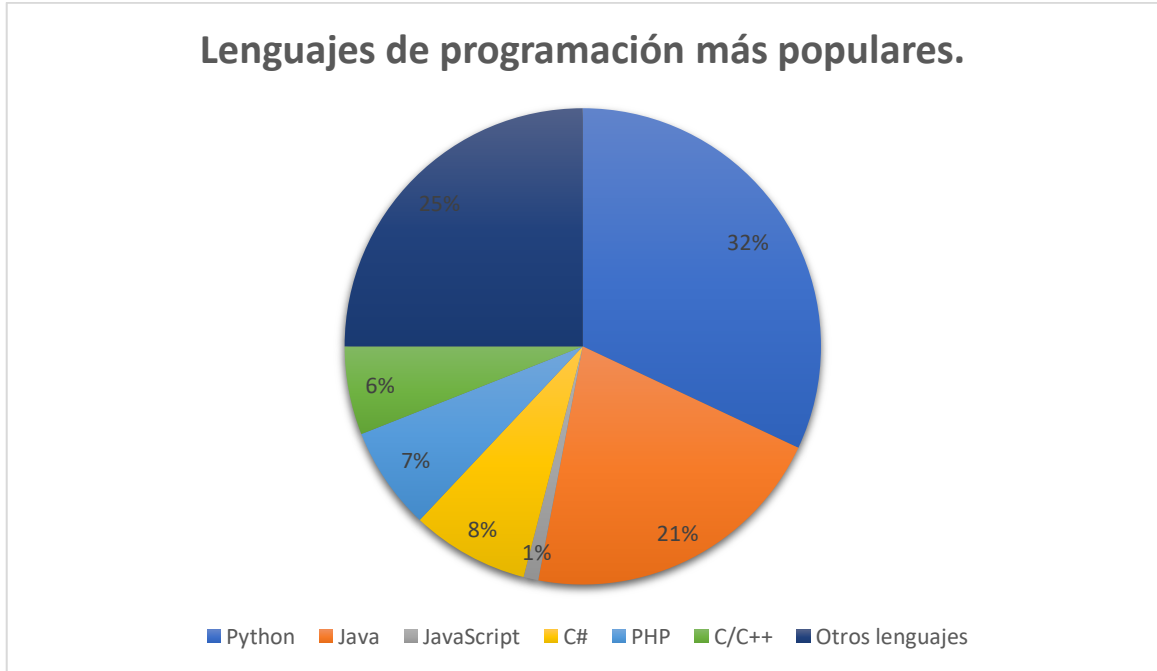


Figura 3.17. Lenguajes de programación más populares.

En la Figura 3.17, se puede observar que los datos de uso de lenguajes de programación, Python es el más popular con un 32%, seguido de Java con un 21%. Sorprendentemente, a pesar de su importancia, JavaScript sólo representa el 1% del desarrollo web. C# y PHP comparten el 8% y el 7% respectivamente, mientras que C/C tiene el 6%. La categoría "Otros idiomas" representa el 25%, lo que indica una considerable diversidad de preferencias de programación. En general, Python ocupa el primer lugar, pero JavaScript es más lento y la gran presencia de "otros lenguajes" muestra que la selección de herramientas de programación en este conjunto de datos es diversa.

3.11.2. Lenguaje de código abierto Python

Python es un lenguaje de programación abierto, orientado a objetos, de alto nivel, simple y fácil de aprender. Aunque fue creado a principios de la década de 1990 en los Países Bajos y en comparación con otros lenguajes de programación, se considera un proyecto relativamente joven [38]. Tiene una gran biblioteca de herramientas, lo que lo convierte en un lenguaje de programación único.

3.11.3. Principales usos de Python

Es muy versátil y con él se puede desarrollar software de todo tipo, principalmente en:

3.11.3.1. Python en el aprendizaje automático

Python admite tecnologías de inteligencia artificial porque:

- Tiene bibliotecas como Numpy, Scipy y Pybrain que se utilizan para informática científica, informática avanzada y aprendizaje automático.
- Permite expresar ideas en tan solo unas pocas líneas de código, lo que no es posible en otros lenguajes.
- Tiene bibliotecas como Keras y TensorFlow que contienen información sobre funciones de aprendizaje automático.
- Todas estas razones hacen de este lenguaje una mejor opción para aplicaciones de IA en comparación con otros lenguajes.

3.11.3.2. Python en Big Data

Python es la mejor opción en el campo de Big Data gracias a su flexibilidad y abundancia de bibliotecas y marcos especializados. Con herramientas como Pandas para la manipulación eficiente de datos, PySpark para computación distribuida y Scikit-learn para aprendizaje automático, Python facilita el procesamiento y análisis de grandes conjuntos de datos. Además, las bibliotecas de visualización como Matplotlib y Plotly le permiten crear representaciones visuales impresionantes. La flexibilidad de Python se extiende a entornos de nube como GCP y AWS, al igual que su integración con tecnologías como Apache Kafka para la recopilación de datos en tiempo real [38].

3.11.3.3. Python en Data Science

Es uno de los mejores lenguajes utilizados por los científicos de datos en diversos proyectos y aplicaciones de análisis de datos. Python proporciona una gran funcionalidad para trabajar con funciones matemáticas, estadísticas y científicas. Proporciona excelentes bibliotecas para trabajar con aplicaciones de ciencia de datos.

3.11.3.4. Python en Frameworks de Pruebas

La comprobación es otra operación que ha sido modificada en este idioma. Es ideal para probar ideas o productos porque tiene muchos marcos integrados que ayudan a depurar el código y garantizan un flujo de trabajo y una ejecución rápidos [38].

Las herramientas de prueba como Unittest, Pytest y Nose test facilitan las pruebas para los desarrolladores y admiten pruebas multiplataforma y entre navegadores utilizando varios marcos como PyTest y Robot.

3.11.4. Tipos de librerías de Python

Existen diferentes tipos de bibliotecas de Python según el propósito de la biblioteca de Python.

A continuación, se muestran algunos tipos de bibliotecas de Python:

- Deep learning: se centra en la predicción de datos; a través de grandes datos.
- Machine learning: Estas bibliotecas son muy útiles en el aprendizaje automático porque mejoran el procesamiento de la información y resuelven problemas de clasificación de datos y análisis de regresión.
- Cálculos numéricos: Preparar datos y sugerir propiedades importantes para sus cálculos.
- Visualización: se utilizan para comprender y dar sentido a los datos en una forma más legible.
- IA explicable: Buscan resultados óptimos en IA utilizando diversos métodos tecnológicos.
- Procesamiento del lenguaje natural: basándose en cálculos de frecuencia normalizados, los modelos se construyen utilizando datos de texto.

3.11.5. Librerías de Python

3.11.5.1. Matplotlib

De hecho, ya hemos hecho referencia a Matplotlib antes. Esta es la biblioteca estándar y más famosa de Python, también generan una variedad de gráficos de alta calidad para publicar online o en papel sin muchas líneas de código, gráficos de barras, histogramas, series temporales, espectros de potencia... Sólo algunos ejemplos. de las capacidades de esta biblioteca en Python [39].

Propósito: Visualización de datos.

Matplotlib: Ofrece flexibilidad para crear una amplia gama de gráficos y visualizaciones.

3.11.5.2. TensorFlow

Por su parte, TensorFlow fue desarrollado por Google y juega un papel importante en la computación numérica. Su estructura permite la creación de una red neuronal que proporciona un nuevo enfoque para conceptualizar las matemáticas. [40].

Propósito: Desarrollo de modelos de aprendizaje profundo.

TensorFlow: Desarrollado por Google, es ampliamente utilizado y respaldado por una gran comunidad.

3.11.5.3. PyTorch

En este caso, Facebook es el encargado de desarrollar esta biblioteca de Python. También se utiliza en cálculo numérico y su tarjeta gráfica es fundamental para la ejecución rápida de código.

Propósito: Desarrollo de modelos de aprendizaje profundo.

PyTorch: Con énfasis en la flexibilidad y la expresividad, es preferido por su interfaz más intuitiva.

3.11.5.4. Scikit-learn

Estas bibliotecas de Python de muestra están destinadas a crear modelos de análisis de datos y aprendizaje automático. Puede realizar operaciones de regresión, clasificación y agrupación de datos. Su interfaz es muy sencilla ya que sólo requiere una línea de código [41].

Propósito: Aprendizaje automático y minería de datos.

Funcionalidades: Contiene implementaciones de algoritmos de clasificación, regresión, clustering y técnicas de reducción de dimensionalidad.

3.11.5.5. Pandas

Pandas se utiliza ampliamente en la ciencia de datos. Es fácil trabajar con datos porque su estructura consta de una serie de datos y marcos de datos para dos dimensiones. Las bibliotecas de este tipo se utilizan en estadística, economía, ingeniería y ciencias sociales [42].

Propósito: Diseñado para análisis y manipulación de datos.

Estructura: Introduce los objetos `DataFrame` y `Series` que permiten organizar datos tabulares y realizar operaciones sofisticadas como filtrado, agrupación y fusión.

3.11.5.6. Seaborn

Si bien la manipulación de datos es el núcleo de Pandas, Seaborn se basa en la visualización de datos estadísticos para comprenderlos. Gracias a esto, su interfaz es de alto nivel e interactiva [43].

Propósito: Visualización de datos.

Seaborn: Basado en Matplotlib, simplifica la creación de gráficos estadísticos atractivos.

3.11.5.7. Bokeh

Bokeh es también una de las bibliotecas de Python destacadas para la interactividad. Sus gráficos son flexibles y funcionan en tiempo real.

3.11.5.8. NumPy

Por su parte, NumPy permite crear estructuras de datos universales, que ayudan a un mejor análisis de datos cuando se intercambian datos entre diferentes algoritmos. Sus vectores son de alta dimensión y sus matrices son grandes [44].

Propósito: NumPy se utiliza para realizar operaciones matriciales y numéricas eficientes.

Características: Introduce el objeto `array` para representar y manipular datos de manera eficiente. Ofrece funciones para álgebra lineal, transformadas y operaciones estadísticas.

4. METODOLOGÍA

4.1. TIPOS DE INVESTIGACIÓN

4.1.1. Investigación Descriptiva

La investigación descriptiva se centrará en analizar y detallar las características de la radiación solar en la Universidad Nacional de Piura. A través de la recopilación de datos históricos de radiación comprendido entre abril del 2022 – abril del 2023, dando un año de toma de datos, en el cual se busca identificar patrones y tendencias en diferentes horas del día. Este análisis permitirá comprender las condiciones actuales y futuras de la radiación solar.

Posteriormente, los resultados obtenidos se comparan entre los modelos de aprendizaje automático para evaluar la precisión predictiva del modelo. Este estudio descriptivo combina el análisis detallado con la aplicación de técnicas de aprendizaje automático para proporcionar una visión integral de la radiación solar en la zona de estudio [45].

4.1.2. Investigación Cuantitativo

Esta investigación tiene un papel importante al abordar la naturaleza numérica de la radiación solar. La recopilación de datos históricos implica la obtención de información cuantitativa sobre los niveles de radiación a lo largo del tiempo, lo que proporciona una base numérica para analizar patrones y tendencias. La implementación de modelos de aprendizaje automático (regresión lineal simple, redes neuronales, GRU) también se basa en métodos cuantitativos que utilizan algoritmos y métodos numéricos para procesar datos y hacer predicciones. Las comparaciones entre los resultados de los modelos y los datos históricos se realizan mediante mediciones cuantitativas y evalúan la efectividad del modelo en términos de precisión [46].

4.1.3. Investigación Experimental

La investigación experimental tiene como objetivo principal el diseño y ejecución de experimentos el cual implica la manipulación de variables clave (independientes, dependientes), relevantes para las predicciones de la radiación solar. La implementación de modelos de aprendizaje automático (regresión lineal simple, unidades recurrentes cerradas y redes neuronales) es una parte esencial del uso de software de código abierto (Python), para entrenar y probar estos modelos. La recopilación de datos se realiza en condiciones controladas

y la eficacia del modelo se puede evaluar comparando las predicciones obtenidas con datos reales de radiación solar [47].

4.2. UBICACIÓN DEL LUGAR DE ESTUDIO

En este estudio se utilizará la base de datos de radiación solar de la Universidad Nacional de Piura (ver la Figura 4.1), ubicada en la Av. Ramón Mendoza s/n, Urb. Santa Ana, Piura, Perú. La base de datos a utilizar esta comprendida entre abril del 2022 hasta abril del 2023 dando un año en la toma de datos.



Figura 4.1. Universidad Nacional de Piura [48].

4.3. TÉCNICAS E INSTRUMENTOS

4.3.1. Programación

En este punto de la investigación se centra en el diseño de algoritmos de aprendizaje automático, en el cual los algoritmos utilizarán datos depurados que se encuentran en una base de datos general. En este proceso, se introducirá el historial de la radiación solar cuyo objetivo principal es comparar los resultados obtenidos de los métodos de aprendizaje automático de acuerdo a las métricas de evaluación, con el propósito de verificar la precisión y validez de los datos que se están prediciendo.

4.3.2. Simulación

Mediante el uso del lenguaje de programación Python se realizará la simulación de los métodos de aprendizaje automático, considerando los datos históricos comprendidos entre abril del 2022 – abril del 2023, dando un año en toma de datos, con el fin de comparar cuál de los tres métodos de aprendizaje automático es más preciso, de acuerdo a las métricas de evolución,

4.4. METODOLOGÍA GENERAL

Las técnicas a ser diseñadas para su evaluación y comparación son las siguientes: regresión lineal simple, unidades recurrentes cerradas y redes neuronales. Para obtener el pronóstico de la radiación solar se implementa la siguiente metodología que se muestra en la Figura 4.2.

Como parte fundamental de la metodología es la revisión exhaustiva de la literatura para fundamentar teóricamente la investigación. Posteriormente, inicia con la recopilación de datos históricos de radiación solar en la Universidad Nacional de Piura, los cuales son sometidos a un proceso de preprocesamiento y selección de características para garantizar su calidad y relevancia. La implementación de modelos de aprendizaje automático ya mencionados anteriormente se lleva a cabo utilizando software de código abierto (Python), ajustando parámetros para optimizar el rendimiento de los modelos.

En el siguiente paso para verificar que los modelos funcionan correctamente, la base de datos disponible se divide en tres grupos: conjunto de entrenamiento, conjunto de prueba y conjunto de validación. Como su nombre lo indica, el primero se utilizará para entrenar los modelos de pronóstico, el segundo se utilizará para evaluar el conjunto de pruebas y el tercero para validar cada uno de los modelos implementados.

Finalmente, para el proceso de comparación y evaluación de diversos modelos de pronóstico, se emplean diferentes métricas para medir su eficiencia. Entre estas métricas se encuentran el Error Cuadrático Medio (MSE), el Error Absoluto Medio (MAE), Error Porcentual Absoluto Medio (MAPE) entre otras métricas, que evalúan la precisión en términos de error de pronóstico a lo largo del rango de validación.

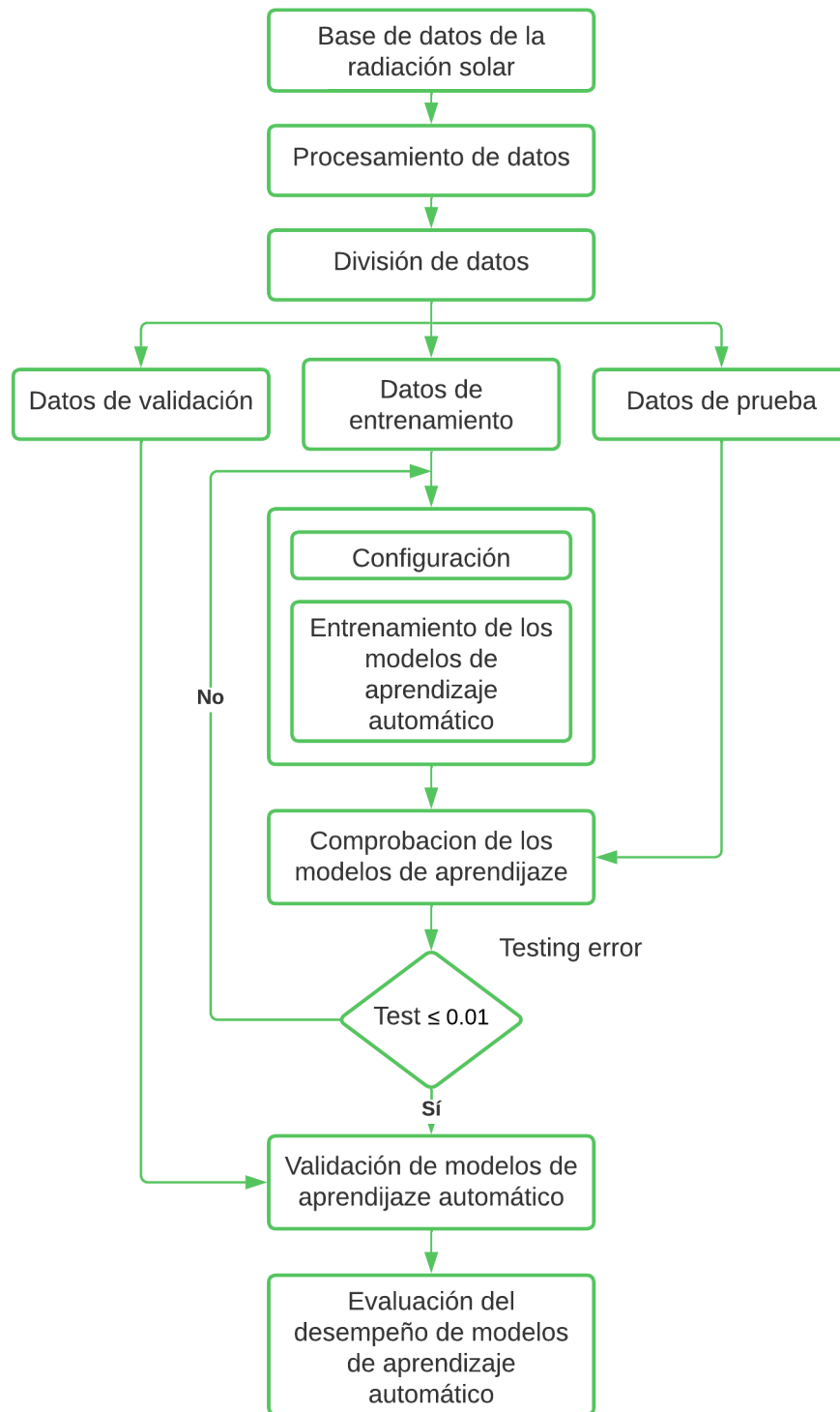


Figura 4.2. Diagrama de flujo de la metodología utilizada para la obtención de modelos de predicción.

4.5. DATOS HISTORICOS

4.5.1. Obtención de la base de datos

Para la implementación de los modelos de aprendizaje automático se obtuvo información de la Universidad Nacional de Piura, donde se encuentra la Estación Meteorológica Convencional Miraflores (ver la Figura 4.3), que cuentan con parámetros tales como: precipitación (mm), temperatura (°C), humedad relativa (%), velocidad del viento (EM) y la radiación solar (). Para el diseño de los modelos de aprendizaje automático solo se tomará en cuenta los valores de la radiación solar ya que la investigación se centra en ese parámetro.



Figura 4.3. Ubicación de la estación meteorológica de la Universidad Nacional de Piura [49].

4.5.2. Almacenamiento de datos

Se presenta una base de datos históricos de la radiación solar comprendida ente abril del 2022 hasta abril del 2023, dando un año en la toma de datos. Los parámetros a considera son; el mes, el día y las horas como se puede apreciar en la Figura 4.4. Para que la predicción sea más precisa se toma en cuenta las 24 horas del día en intervalos de una hora.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1		25/4/2022	26/4/2022	27/4/2022	28/4/2022	29/4/2022	30/4/2022	1/5/2022	2/5/2022	3/5/2022	4/5/2022	5/5/2022	6/5/2022	7/5/2022
2	12:00 a. m.		0	0	0	0	0	0	0	0	0	0	0	0
3	1:00 a. m.		0	0	0	0	0	0	0	0	0	0	0	0
4	2:00 a. m.		0	0	0	0	0	0	0	0	0	0	0	0
5	3:00 a. m.		0	0	0	0	0	0	0	0	0	0	0	0
6	4:00 a. m.		0	0	0	0	0	0	0	0	0	0	0	0
7	5:00 a. m.		0	0	0	0	0	0	0	0	0	0	0	0
8	6:00 a. m.		0	3	1	26	17	1	28	25	16	11	16	18
9	7:00 a. m.		9	52	64	190	97	42	191	175	148	87	93	137
10	8:00 a. m.		153	150	184	395	334	140	394	391	390	219	356	380
11	9:00 a. m.		379	444	377	580	413	264	583	573	576	461	577	572
12	10:00 a. m.		577	674	647	729	797	594	730	722	725	724	722	717
13	11:00 a. m.		634	795	784	821	823	819	818	810	818	650	809	803
14	12:00 p. m.	893	572	852	813	842	850	836	839	829	818	471	826	825
15	1:00 p. m.	893	867	835	674	776	778	753	784	775	631	802	773	772
16	2:00 p. m.	818	832	747	492	688	676	661	661	655	345	385	655	649
17	3:00 p. m.	712	674	590	446	497	499	477	487	479	389	164	490	476
18	4:00 p. m.	603	517	388	292	290	283	290	284	277	189	73	212	271
19	5:00 p. m.	234	322	170	95	85	85	79	77	77	59	27	27	70
20	6:00 p. m.	74	101	20	2	2	1	1	1	1	1	0	1	1
21	7:00 p. m.	4	4	0	0	0	0	0	0	0	0	0	0	0
22	8:00 p. m.	0	0	0	0	0	0	0	0	0	0	0	0	0
23	9:00 p. m.	0	0	0	0	0	0	0	0	0	0	0	0	0
24	10:00 p. m.	0	0	0	0	0	0	0	0	0	0	0	0	0
25	11:00 p. m.	0	0	0	0	0	0	0	0	0	0	0	0	0

Figura 4.4. Base de datos históricos de la radiación solar.

La base de datos que se presenta consta con 8592 datos que equivale al 100 % que será analizado, pero toda base de datos tiene datos faltantes, en este caso contamos con 8339 datos reales que equivale al 97.06 % y 253 datos faltantes que equivalen al 2.94 % como se muestra en la Figura 4.5.

```
# Verificar valores nulos en el DataFrame
valores_nulos = datos.isnull().sum()

print("Valores nulos:")
print(valores_nulos)

Valores nulos:
AÑO          0
MES          253
DIA          253
HORA         253
RADIACION    253
dtype: int64
```

Figura 4.5. Verificación de valores nulos.

4.5.3. Preprocesamiento de datos

La base de datos obtenida que consta con la información de 1 año (del 25 de abril de 2022 al 17 de abril de 2023), la cual fue preprocesada para tomar datos cada 1 hora de la radiación solar. Para realizar un pronóstico es importante verificar que no falten datos en las mediciones o implementar algún método de llenado de datos (observe la Figura 4.10); en este caso se implementó un algoritmo “Series sintéticas con la simulación de Montecarlo”, el cual consiste en generar números aleatorios a partir de una distribución de probabilidad normal.

4.5.3.1. Simulación de Montecarlo

En una base de datos es habitual encontrar valores nulos, es decir datos faltantes, esto se puede dar ya sea por error del mismo ser humano o a su vez por la propia tecnología (computadoras, instrumentos, etc.), para este problema se emplea la simulación de Montecarlo en combinación con una distribución de probabilidad normal ya que es simétrica y se caracteriza por su media y desviación estándar, el cual nos ayuda a generar series aleatorias para completar los datos faltantes, para llegar a este punto se debe seguir el siguiente método que se refleja en la Figura 4.6.

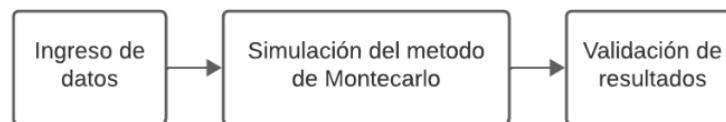


Figura 4.6. Metodología de la simulación de Montecarlo.

En la Figura 4.7, se muestra un diagrama de flujo en cual se indica el proceso de simulación de Montecarlo basado en datos deterministas de entrada correspondientes a la radiación solar. Primero se ingresa una matriz de perfiles vectoriales correspondientes a 24 mediciones por día y se especifica el número n de series a obtener (en este caso 50 series). Luego se calculan los parámetros μ y σ de la matriz del vector de datos, lo que facilita la generación de valores aleatorios. El proceso se realiza según un límite de error relativo porcentual, donde si el error relativo porcentual es menor que un límite predeterminado, el valor generado se retiene; de lo contrario, se genera un nuevo valor hasta que se cumplan las condiciones predeterminadas. Es importante que los límites se ajusten a la variabilidad del material. A continuación, se utiliza el error cuadrático medio normalizado (NMES) para determinar la precisión de generar secuencias sintéticas utilizando métodos de Monte Carlo.

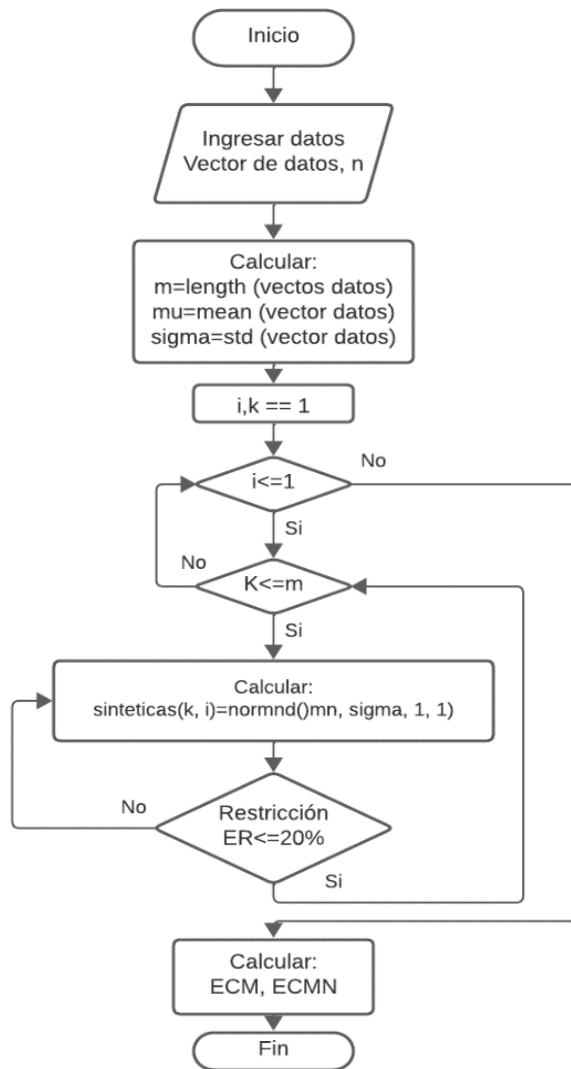


Figura 4.7. Diagrama de flujo de la simulación de Montecarlo con una distribución de probabilidad normal.

En la Figura 4.8, se puede apreciar las 50 series aleatorias que nos arrojó la simulación de Montecarlo, posteriormente estos datos obtenidos nos ayudara a completar la base de datos de la radiación solar de forma manual para poder analizar muestra base de datos para la implementación de los modelos de aprendizaje automático, también en la Figura 4.9, se muestra el comportamiento de la radiación solar en base a las series aleatorias obtenidas a partir de la simulación de Montecarlo con una distribución de probabilidad normal, en el cual se puede apreciar las 11 series de datos aleatorios, que se utilizaron para llenar de forma manual la base de datos final.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	13	9	10	11	11	13	12	13	11	12	10	13	12	12	10	12	11	13	11	10	14	11	12	11	12	14
8	97	94	92	89	93	74	70	88	94	94	91	89	73	79	95	94	76	78	87	84	79	90	75	96	95	68
9	239	236	217	237	246	178	202	240	188	217	196	261	266	183	221	218	227	208	184	187	236	223	196	225	225	250
10	370	369	395	337	411	343	356	350	413	366	384	425	403	347	323	408	324	412	406	425	338	339	436	343	368	350
11	575	642	521	522	547	513	467	513	547	490	688	630	667	539	474	673	663	564	543	465	467	562	630	551	489	607
12	561	794	734	791	754	572	804	740	642	636	661	567	677	781	765	787	630	709	773	632	613	590	618	613	812	678
13	708	792	759	727	847	667	903	727	648	668	850	770	886	668	645	681	753	684	732	676	659	683	827	712	661	760
14	659	666	720	626	633	670	761	843	682	824	666	753	696	886	703	752	682	740	781	657	725	783	919	811	689	814
15	722	549	610	572	775	584	758	722	698	709	789	547	548	571	678	695	564	637	727	789	724	539	680	554	620	720
16	544	613	446	511	577	599	521	630	516	476	530	468	469	532	445	548	534	497	435	442	438	612	612	615	445	616
17	423	424	309	299	294	384	304	305	415	284	332	316	333	405	310	380	298	406	323	332	350	385	305	310	302	312
18	155	140	143	152	159	195	137	153	143	154	177	137	166	176	151	199	194	180	182	164	171	163	167	161	150	185
19	33	34	45	36	44	43	40	40	48	34	48	46	42	43	42	42	32	34	42	42	36	38	36	35	38	35
20	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	1	1	1	1	1	1	1	2	2	1
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figura 4.8. Obtención de las series aleatorias a partir de la simulación de Montecarlo.

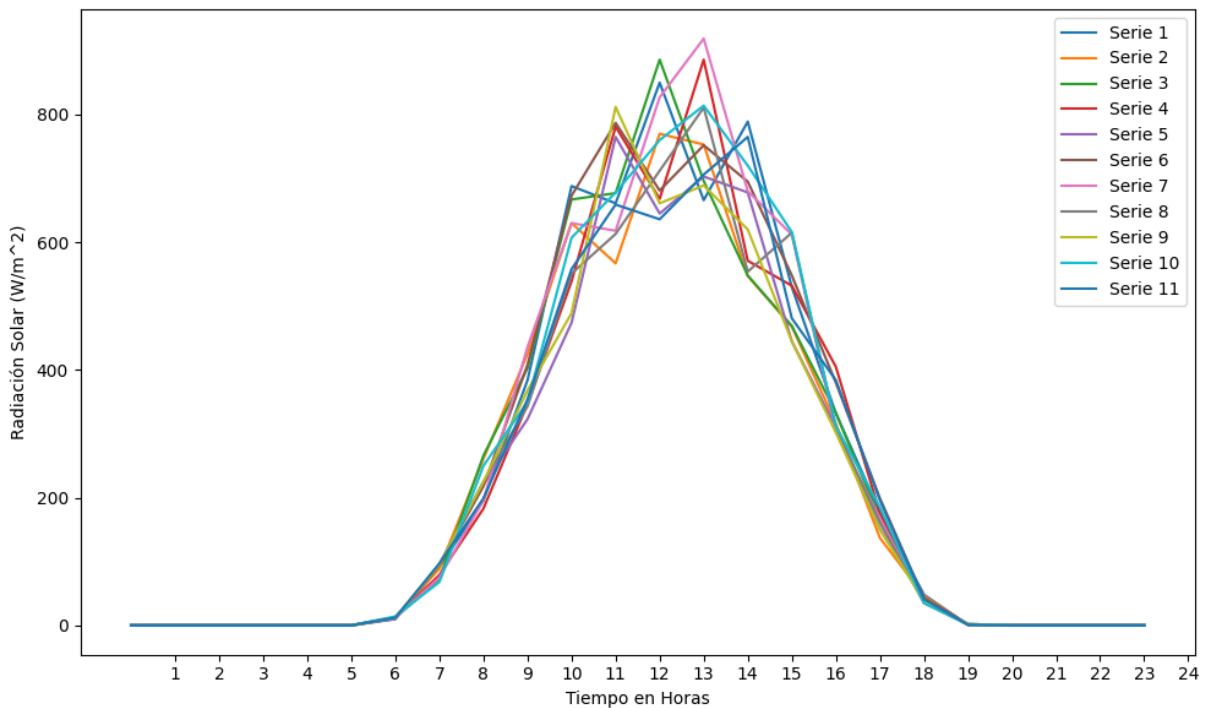


Figura 4.9. Comportamiento de la radiación solar obtenidos a partir de la simulación de Montecarlo.

La simulación de Montecarlo ayuda a crear datos aleatorios el cual nos facilita para rellenar la base de datos a ser estudiada, con la ayuda de la verificación de valores nulos (ver la Figura 4.11) nuestra base de datos no presenta ningún inconveniente con esta problemática por ende se realiza una decodificación de los datos como se muestra en la Figura 4.10.

	AÑO	MES	DIA	FECHA	HORA	RADIACION
8542	2023	4	6	15	22	0
8543	2023	4	6	15	23	0
8544	2023	4	7	16	0	0
8545	2023	4	7	16	1	0
8546	2023	4	7	16	2	0
8547	2023	4	7	16	3	0
8548	2023	4	7	16	4	0
8549	2023	4	7	16	5	0
8550	2023	4	7	16	6	14
8551	2023	4	7	16	7	157
8552	2023	4	7	16	8	226
8553	2023	4	7	16	9	321
8554	2023	4	7	16	10	793
8555	2023	4	7	16	11	605
8556	2023	4	7	16	12	568
8557	2023	4	7	16	13	607
8558	2023	4	7	16	14	674
8559	2023	4	7	16	15	500
8560	2023	4	7	16	16	250
8561	2023	4	7	16	17	13
8562	2023	4	7	16	18	0
8563	2023	4	7	16	19	0
8564	2023	4	7	16	20	0
8565	2023	4	7	16	21	0
8566	2023	4	7	16	22	0
8567	2023	4	7	16	23	0
8568	2023	4	1	17	0	0
8569	2023	4	1	17	1	0
8570	2023	4	1	17	2	0

Figura 4.10. Preprocesamiento de la base de datos.

```
# Verificar valores nulos en el DataFrame
valores_nulos = datos.isnull().sum()

print("Valores nulos:")
print(valores_nulos)
```

```
Valores nulos:
AÑO          0
MES          0
DIA          0
HORA         0
RADIACION    0
dtype: int64
```

Figura 4.11. Verificación de valores nulos.

4.6. DETERMINACIÓN DE VARIABLES

En cualquier investigación conlleva una recopilación de datos que son gestionados a través de las denominadas "bases de datos". Estos datos están estructurados en forma de "variables", las cuales pueden ser variables dependientes e independientes. Para esta investigación las variables independientes proporcionan la información necesaria para prever la variable dependiente, contribuyendo así a la comprensión y predicción más precisa de la radiación solar.

A continuación, en la Tabla 4.1, se determinan las variables dependientes e independientes:

Tabla 4.1. Determinación de variables dependientes e independientes.

Variabes	Parámetros	Unidad de medida
Dependiente	Radiación Solar	W/m ²
Independientes	Año, mes, día, hora	Tiempo

4.7. COMPORTAMIENTO DE LA RADIACIÓN SOLAR EN LA ZONA DE ESTUDIO

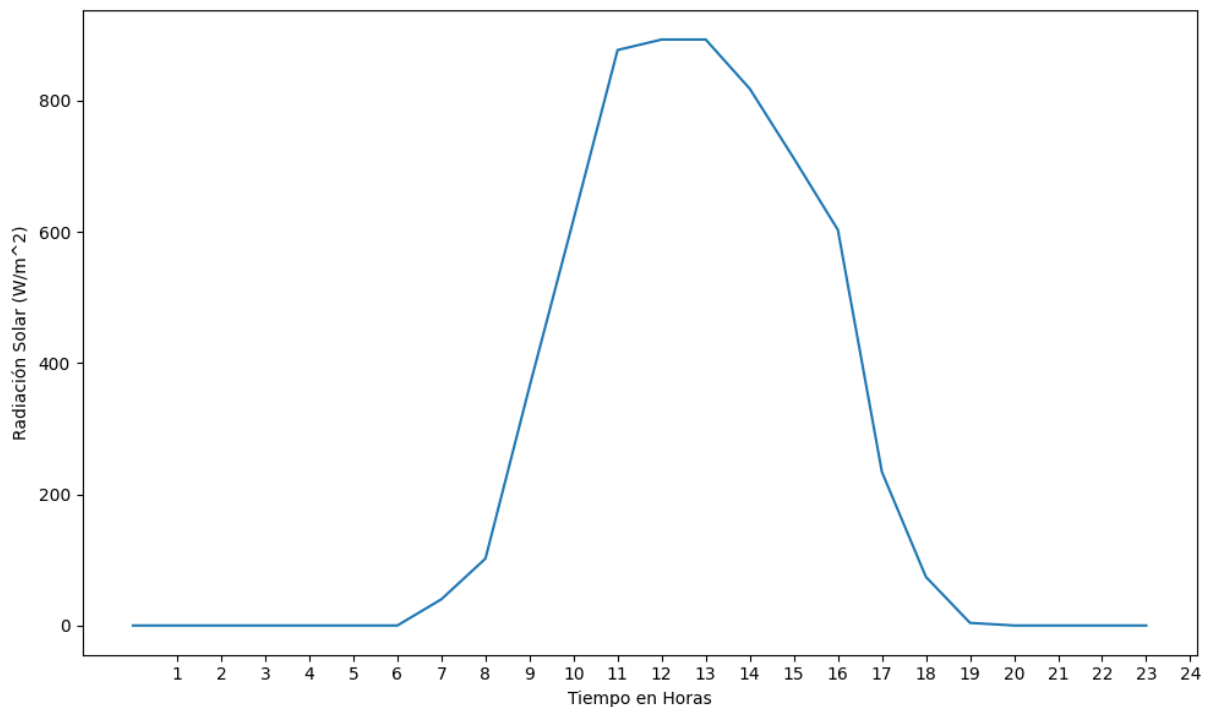


Figura 4.12. Comportamiento de la radiación solar del día 25/04/2022.

En la Figura 4.12, se puede apreciar el comportamiento de la radiación solar en el transcurso de un día específicamente del 25 de abril del 2022, el cual proporciona una tendencia ascendente hasta el mediodía cuando el sol está en su posición más alta llegando a un valor pico de 877

W/m^2 , posteriormente la disminución progresiva en la tarde y noche, esto se debe a reducción natural de la radiación solar a medida que el sol se pone.

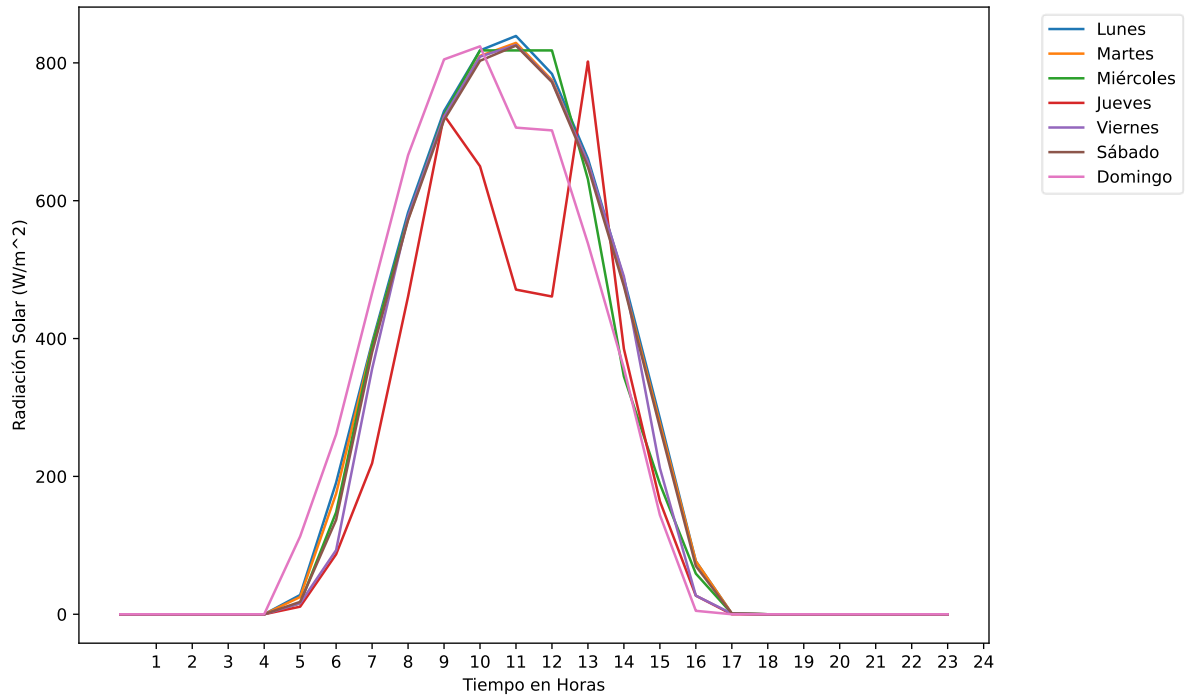


Figura 4.13. Comportamiento de la radiación solar durante la primera semana de mayo del 2022.

Para el análisis de datos semanales en la Figura 4.13, se observa el comportamiento de la radiación solar durante la primera semana del mes de mayo del 2022, el cual nos ayuda a comprender la influencia del clima y las estaciones en la estación meteorológica de la Universidad de Piura en el cual nos proporciona tendencias muy importantes a la hora de la generación de energía solar.

4.8. PROGRAMACIÓN EN PYTHON

4.8.1. Uso de las librerías

Para la programación de los modelos de aprendizaje automático se utilizará diferentes librerías (pandas, keras, matplotlib, numpy, etc.) tal como se muestra en la Figura 4.14, estas librerías que nos ofrece Python cumplen un papel importante en el creación y evaluación de modelos de predicción utilizando aprendizaje automático.

Para la manipulación de datos numéricos y tabular respectivamente se utiliza Pandas y Numpy, por otra parte, Matplotlib juega un papel importante en la visualización de los datos permitiendo crear gráficos detallados y Keras se utiliza para simplificar la construcción y entrenamiento de los modelos de aprendizaje automático.

```

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
import keras
from keras.models import Sequential
from keras.layers import Dense, Activation, Dropout
from sklearn.metrics import mean_squared_error

```

Figura 4.14. Librerías usadas en la programación de los modelos de aprendizaje automático.

4.8.2. Lectura de datos

En esta sección se incorpora la base de datos de la radiación solar comprendida entre abril del 2022 hasta abril de 2023 como se muestra en la Figura 4.15, en el cual consta con las variables de entrada (año, mes, día, hora y radiación solar), esta base de datos deberá ser un formato xlsx para su mejor manipulación.

```

# Lectura de datos
data = pd.read_excel('DATA_RADIACION.xlsx')

data

```

	AÑO	MES	DIA	FECHA	HORA	RADIACION
0	2022	4	lunes	25	00:00:00	0
1	2022	4	lunes	25	01:00:00	0
2	2022	4	lunes	25	02:00:00	0
3	2022	4	lunes	25	03:00:00	0
4	2022	4	lunes	25	04:00:00	0
...
8587	2023	4	lunes	17	19:00:00	0
8588	2023	4	lunes	17	20:00:00	0
8589	2023	4	lunes	17	21:00:00	0
8590	2023	4	lunes	17	22:00:00	0
8591	2023	4	lunes	17	23:00:00	0

8592 rows x 6 columns

Figura 4.15. Lectura de la base de datos de estudio.

4.8.3. Datos de entrenamiento

En esta parte de la investigación es fundamental dividir los datos como se muestra en la Figura 4.16, en el cual el 20 % de los datos finales es utilizado para la validación de los modelos de aprendizaje automático mientras que el 80% de los datos restantes serán tomados para el entrenamiento de los modelos de aprendizaje automático.

```
#DIVISIÓN DE DATOS
#Dividir los datos de entrenamiento y de test

#Tomamos el 20% de los datos finales como conjunto de prueba y el resto como conjunto de entrenamiento
size = int(0.2*data_final.shape[0])
data_test = data_final[-size:]
data_train = data_final[:-size]
```

Figura 4.16. Porcentaje de datos de entrenamiento.

4.8.4. Predicción

Para la predicción de la radiación solar de los tres modelos de aprendizaje automático (ARIMA, GRU y redes neuronales), se debe tomar en cuenta el tiempo de la predicción en este caso el pronóstico que se está realizando es para un mes ya que estos modelos de predicción son muy útiles a corto plazo, para esto se debe tomar en cuenta los siguientes pasos como se muestra en la Figura 4.17.

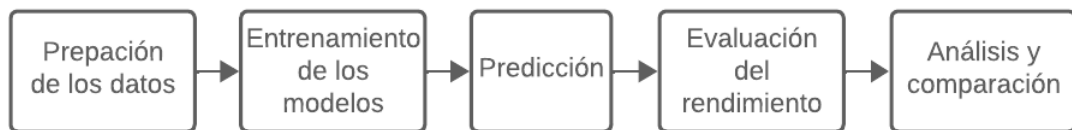


Figura 4.17. Proceso general de los modelos de predicción.

4.9. METODOLOGÍA DE LOS MODELOS DE PREDICCIÓN

4.9.1. Regresión lineal múltiple

La regresión lineal simple es una herramienta fundamental en estadísticas y análisis de datos debido a su simplicidad y utilidad en diversos contextos, por ende, se aplica este modelo para realizar el pronóstico de la radiación solar, por tal motivo se emplea la siguiente metodología representada en un diagrama de bloques como se muestra en la Figura 4.18.

La metodología da comienzo con la identificación de la variable independiente (X), seguida de la recopilación de datos relevantes que representen la relación entre (X) y la variable dependiente (Y). Después, se realiza el preprocesamiento de datos, que incluye la limpieza, manejo de valores atípicos y transformaciones si es necesario. A continuación, se divide el

conjunto de datos para el entrenamiento y prueba. La construcción del modelo implica ajustar la ecuación de regresión lineal utilizando el conjunto de entrenamiento para encontrar los coeficientes (β_0 y β_1). Posteriormente, se evalúa el modelo utilizando métricas como el MAPE, MAE, RMSE, entre otros y se realiza una predicción utilizando el modelo ajustado en nuevos datos o en el conjunto de prueba.

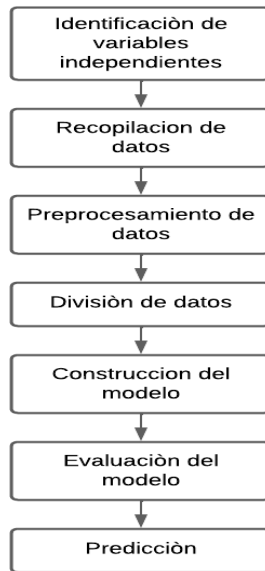


Figura 4.18. Metodología de la regresión lineal simple.

4.9.1.1. Parámetros a considerar de la regresión lineal simple

Tabla 4.2. Parámetros de la regresión lineal simple.

Parámetro	Descripción
Variable Independiente	La variable X se utiliza para predecir la variable Y.
Variable Dependiente	La variable Y se busca predecir basándose en la variable independiente.
Coefficiente de intercepto (β_0)	Representa el valor esperado de la variable dependiente cuando la variable independiente es 0.
Coefficiente de pendiente (β_1)	Indica la tasa de cambio en la variable dependiente por unidad de cambio en la variable independiente.
Error residual	Diferencia entre el valor observado de Y y el valor predicho del modelo.
Conjunto de entrenamiento y prueba	Esta parte es fundamental ya que este permite dividir los datos para el entrenamiento y la evaluación del modelo.
Métricas de evaluación	Indicadores como el MAPE, RMSE, R^2 miden la precisión y el rendimiento del modelo.

4.9.2. Unidades Recurrentes Cerradas (GRU)

El modelo de Unidades Recurrentes Gated (GRU) es una variante de las Redes Neuronales Recurrentes (RNN) que aborda problemas como el desvanecimiento del gradiente. Similar a las LSTM, las GRU pueden retener información a lo largo del tiempo en secuencias.

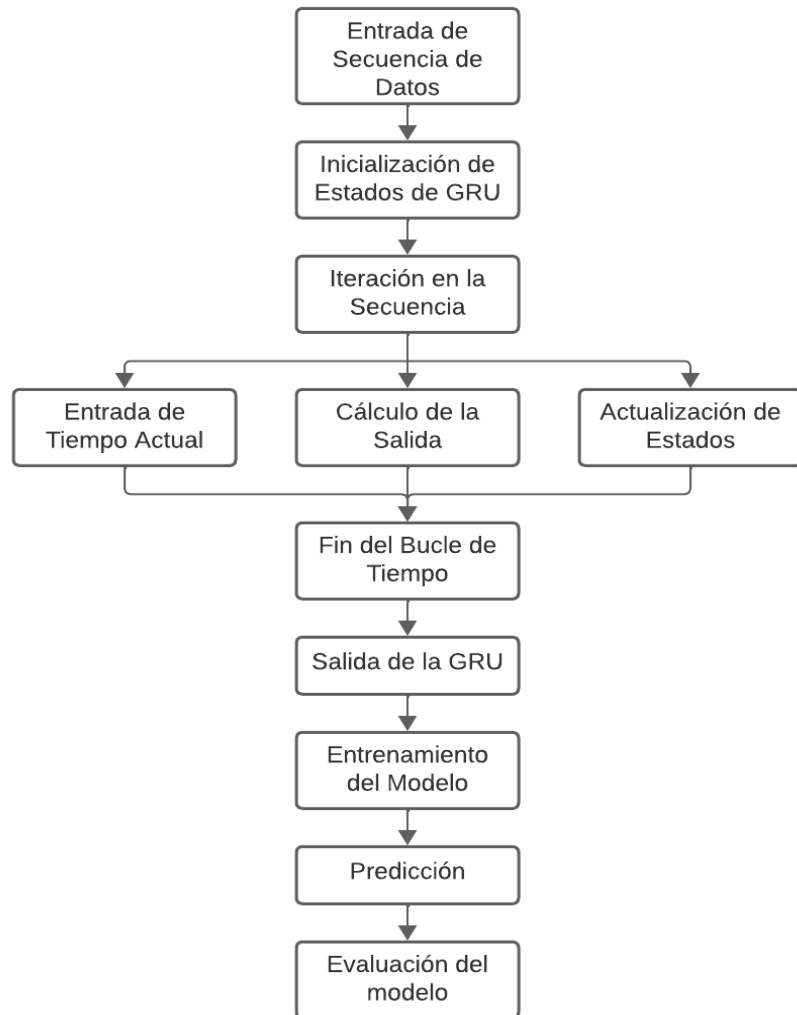


Figura 4.19. Metodología del modelo unidades recurrentes cerradas (GRU).

En la Figura 4.19, se observa la metodología del modelo GRU, realizada en un diagrama de bloques, comienza con la entrada de una secuencia de datos, seguido por la inicialización de los estados de la GRU incluyendo la memoria recurrente. Luego, se itera sobre la secuencia de datos a lo largo del tiempo, con cada iteración incluyendo la entrada actual, el cálculo de la salida, y la actualización de los estados de la GRU. Después de iterar sobre toda la secuencia, se obtiene la salida final de la GRU. El modelo se entrena mediante la retro propagación a través del tiempo (BPTT) y la actualización de los pesos de la red. Finalmente, el modelo entrenado se utiliza para realizar predicciones y evaluación del modelo mediante métricas ya establecidas.

4.9.2.1. Parámetros de las unidades recurrentes cerradas (GRU)

En la Tabla 4.3, se describen los parámetros que se debe considerar al momento de generar un modelo GRU.

Tabla 4.3. Parámetros de la arquitectura de las unidades recurrentes cerradas (GRU).

Parámetro	Descripción
Unidades Gru	Cantidad de unidades o celdas GRU en la capa.
Número De Capas Gru	Número de capas GRU apiladas.
Dropout Rate	Proporción de neuronas desactivadas en Dropout.
Unidades Densas	Número de neuronas en cada capa densa.
Funciones De Activación	Funciones utilizadas en capas GRU y densas.
Activación Function Gru	Función de activación para las puertas de la unidad GRU.
Uso De Sesgos En Gru	Indica si se utilizan sesgos en las puertas de la unidad GRU.
Uso De Sesgos En Densas	Indica si se utilizan sesgos en capas densas.
Épocas	Número de veces que todo el conjunto de datos se pasa a través del modelo.
Optimizador	Algoritmo de optimización utilizado para ajustar los pesos del modelo durante el entrenamiento.

- **Arquitectura del modelo GRU**

La arquitectura del modelo sigue una estructura secuencial utilizando capas de Unidad Recurrente con Puertas (GRU) para capturar patrones temporales en los datos. Se incorporan capas de Dropout para mitigar el sobreajuste. El modelo consta de dos capas GRU, seguidas por capas densas con Dropout, culminando en una capa densa de salida. Con un total de 1435 parámetros, todos entrenables, la arquitectura está diseñada para abordar eficazmente la tarea de predicción temporal específica de este estudio, tal como se muestra en la Figura 4.20.

Layer (type)	Output Shape	Param #
gru (GRU)	(None, 60, 10)	390
dropout (Dropout)	(None, 60, 10)	0
gru_1 (GRU)	(None, 10)	660
dense (Dense)	(None, 32)	352
dropout_1 (Dropout)	(None, 32)	0
dense_1 (Dense)	(None, 1)	33
=====		
Total params: 1,435		
Trainable params: 1,435		
Non-trainable params: 0		

Figura 4.20. Arquitectura del modelo GRU.

- **Selección del optimizador**

Para este modelo de predicción se seleccionó el optimizador Adam el cual es comúnmente utilizado durante el entrenamiento del modelo, como se puede observar en la Figura 4.21. Adam (Adaptive Moment Estimation) es un algoritmo de optimización que combina las ideas del descenso de gradiente estocástico (SGD) con momentum y el método de adaptación de la tasa de aprendizaje.

```
#ADAM LOS MEJORES RESULTADOS
model, history = basic_GRU(optimizador = 'adam', unidades = 10, epocas = 10)
#Guardamos el modelo para no tener que entrenar otra vez para probar
model.save('MODELOS GRU/model.h5')
```

Figura 4.21. Selección del optimizador para el modelo GRU.

- **Selección de unidades GRU**

Las unidades en una capa GRU (Gated Recurrent Unit) se refieren al número de nodos o neuronas en esa capa, es un hiperparámetro clave que influye en la capacidad de aprendizaje y complejidad del modelo. La elección del número óptimo de neuronas debe equilibrar la necesidad de capturar patrones complejos con la prevención del sobreajuste y la consideración de los recursos computacionales disponibles, para este caso se trabaja con 10 neuronas como se puede observar en la Figura 4.21.

- **Selección de la época**

La cantidad de épocas y la cantidad de capas en el modelo GRU (Unidad recurrente cerrada) son dos hiperparámetros importantes que afectan el rendimiento y la capacidad de aprendizaje del modelo. En la Tabla 4.4, se aprecia el comportamiento de las épocas y a su vez la selección del número de las capas.

Tabla 4.4. Comportamiento del número de épocas.

Optimizador	Época	Neurona	Loss	Error absoluto medio
ADAM	1	10	0,1396	0,3463
	3	10	0,0443	0,1712
	5	10	0,0128	0,0716
	7	10	0,0102	0,0619
	10	10	0,0087	0,0546

4.9.3. Redes Neuronales

Si el objetivo es utilizar una red neuronal para predecir valores futuros o continuaciones en base a datos históricos, series temporales u otras secuencias, un diagrama de bloques puede ayudar a visualizar la metodología de trabajo con una red neuronal para predecir valores futuros, tal como se puede apreciar en la Figura 4.22.

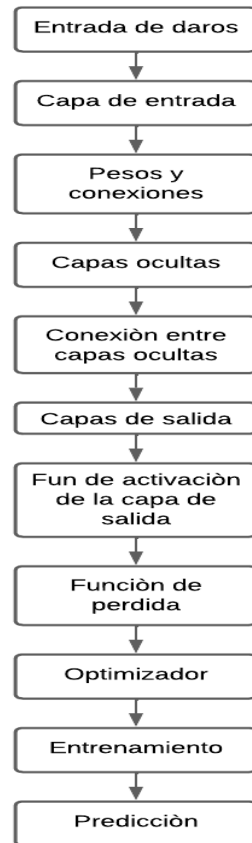


Figura 4.22. Diagrama de bloques de la metodología de una red neuronal.

En la metodología de un modelo de redes neuronales, inicia con los datos de entrada se alimentan a través de una capa inicial, donde cada característica se representa mediante

neuronas. Las conexiones entre estas neuronas tienen pesos ajustables. Luego, los datos atraviesan múltiples capas ocultas, donde se aplican funciones de activación a las entradas ponderadas. La información se transforma progresivamente hasta llegar a la capa de salida, que produce la predicción final. Durante el entrenamiento, se utiliza una función de pérdida para calcular la discrepancia entre la predicción y la verdad conocida, y se optimizan los pesos mediante retro-propagación del error. Este proceso busca minimizar la función de pérdida, mejorando así la capacidad del modelo para realizar predicciones precisas. La arquitectura específica puede variar según el problema y las funciones de activación utilizadas, entre otros factores. La evaluación del modelo se realiza en un conjunto de prueba utilizando métricas relevantes, como el error absoluto medio (MAE) o el error porcentual absoluto medio (MAPE). Con el modelo entrenado, se realizan predicciones futuras basadas en nuevos datos temporales.

4.9.3.1. Parámetros de las redes neuronales

En la Tabla 4.5, se muestra los parámetros que debemos tomar en cuenta a la hora que simular nuestro modelo de aprendizaje automático.

Tabla 4.5. Parámetros de las redes neuronales.

Parámetro	Descripción
Número De Capas	Cantidad de capas en la red neuronal.
Neuronas Por Capa	Número de neuronas en cada capa.
Funciones De Activación	Funciones aplicadas a la salida de cada capa (ReLU, Sigmoid, Tanh).
Inicialización De Pesos	Método para inicializar los pesos de la red.
Tasa De Aprendizaje	Magnitud de ajustes a los pesos durante el entrenamiento.
Regulación (Dropout)	Porcentaje de neuronas desactivadas aleatoriamente durante el entrenamiento.
Batch Normalization	Normalización de las actividades en cada capa.
Épocas	Número de veces que todo el conjunto de datos se pasa a través de la red durante el entrenamiento.
Tamaño Del Lote	Cantidad de muestras de datos utilizadas en cada paso de actualización de pesos.
Optimizador	Algoritmo de optimización utilizado para ajustar los pesos de la red durante el entrenamiento.

- **Selección de optimizador**

Para la selección del optimizador de la red neuronal, utilizaremos 10 neuronas y 10 épocas para saber cuan preciso es el optimizador, utilizando diferentes métricas, como se puede observar en la Tabla 4.6.

Tabla 4.6. Optimizadores para la red neuronal.

Optimizador	Época	Neurona	Loss	Error absoluto medio
ADAM	10	10	0,0142	0,0838
ADELTA	10	10	0,1370	0,2266
RMSPROP	10	10	0,0161	0.0923

A continuación, en la Figura 4.23, se puede observar el comportamiento de la radiación solar con los optimizadores, en el cual por simplicidad se puede decir que el optimizador RMSPROP se acerca a los valores reales de la radiación solar.

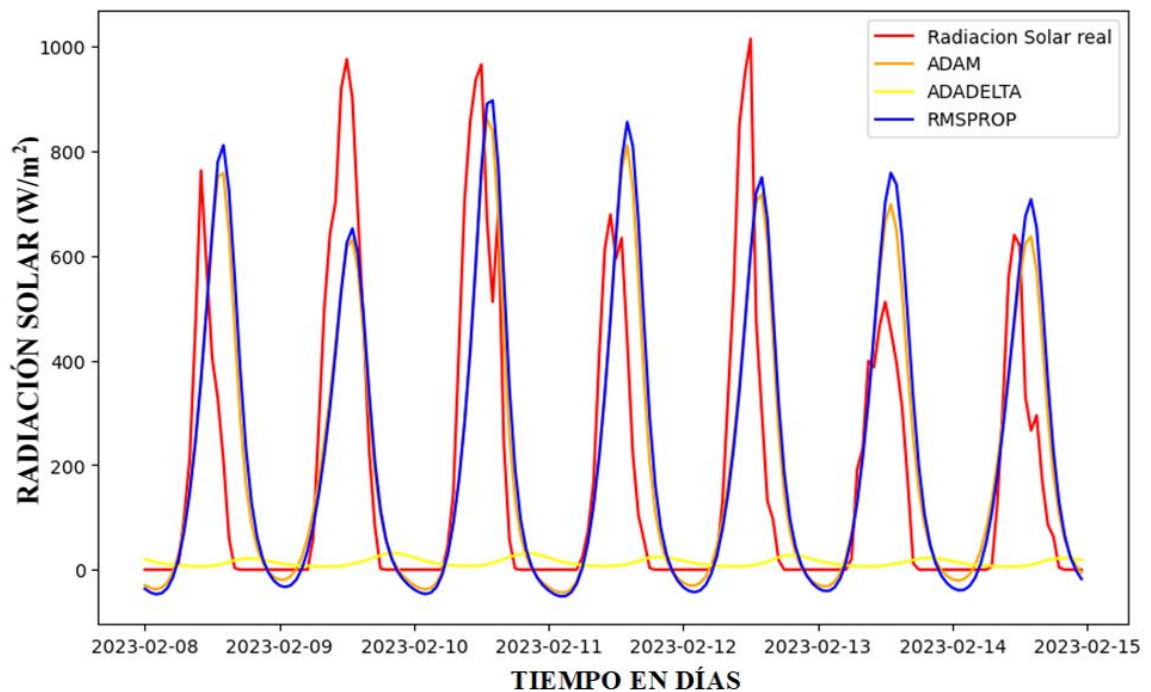


Figura 4.23. Comportamiento de la radiación solar con los optimizadores.

- **Selección de Neurona y Épocas**

Con la selección del optimizador se procede a la selección del número de época y neurona para la red neuronal, como se muestra en la Tabla 4.7.

Tabla 4.7. Selección del número de neuronas y épocas para la red neuronal.

Optimizador	Época	Neurona	Loss	Error absoluto medio
RMSPROP	1	15	0,0711	0,2101
	3	15	0,0290	0,1385
	5	15	0,0168	0,0964
	10	15	0,0138	0,0848
	15	15	0,0108	0,0720

Dependiendo del número de épocas y neuronas que se puede trabajar el error va disminuir, en este caso se trabaja con 15 épocas y 15 neuronas, el cual se aproxima mucho a la curva de la radiación solar real, tal como se muestra en la Figura 4.24.

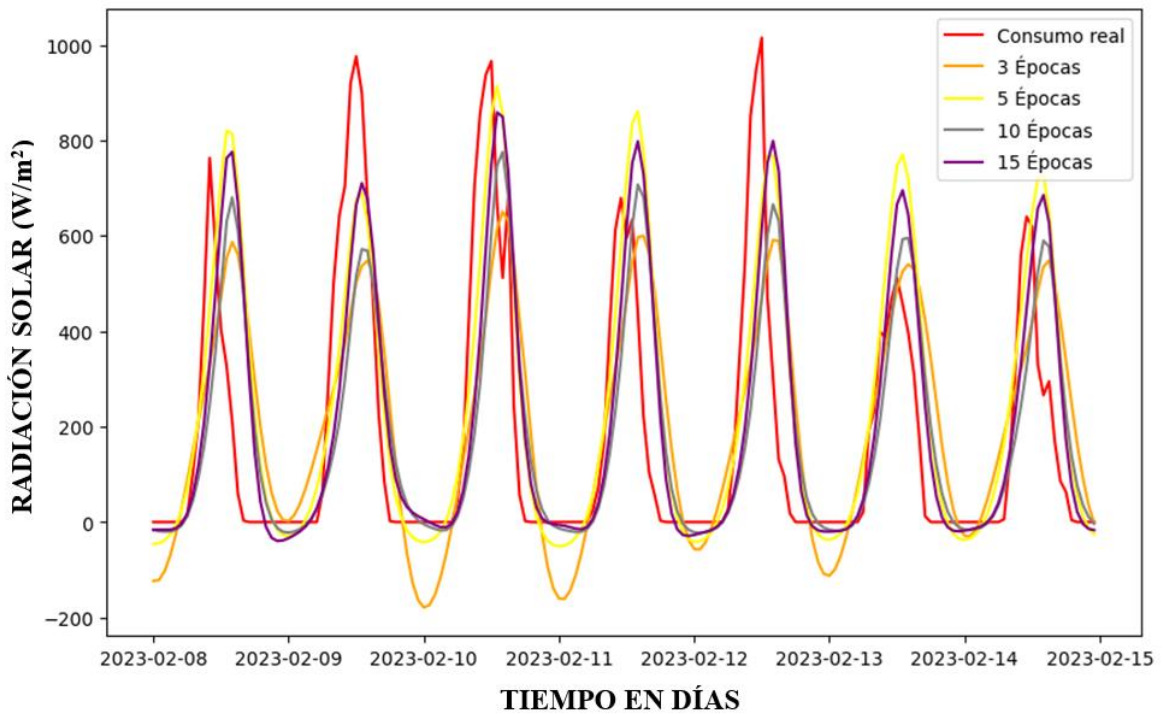


Figura 4.24. Comportamiento de la radiación solar en relación al número de época.

4.10. COMPARACIÓN DE LOS MODELOS DE APRENDIZAJE AUTOMÁTICO

En la Tabla 4.8, se compara los modelos de aprendizaje automático

Tabla 4.8. Comparación de los modelos de aprendizaje automático [22].

CARACTERÍSTICAS	REGRESIÓN LINEAL SIMPLE	GRU	REDES NEURONALES
TIPO DE MODELO	Paramétrico, Lineal	Red neuronal recurrente (RNN)	Diversos tipos, incluyendo RNN, LSTM y otros
USO PRINCIPAL	Modelar Relaciones Lineales	Secuencia de datos temporales	Datos complejos, no lineales
MANEJO DE TENDENCIA	Modela tendencia lineal	Capacidad limitada para manejar tendencias complejas	Puede aprender patrones más complejos y no lineales
DEPENDENCIAS TEMPORALES	No explícitas, lineales	Captura dependencias temporales a través de las unidades recurrentes	Captura dependencias temporales complejas y a largo plazo
HIPERPARAMETROS	No aplica	Menos hiperparámetros para ajustar	Mayor cantidad de hiperparámetros para ajustar
INTERPRETACIÓN	Fácil de interpretación de la relación lineal.	Intermediario en interpretación, basado en conexiones recurrentes	Menos interpretable, basado en la complejidad de las capas y conexiones
RENDIMIENTO EN DATOS PEQUEÑOS	Bueno con relaciones lineales y pequeños conjunto de datos	Puede funcionar bien en datos pequeños a medianos	Puede necesitar grandes conjuntos de datos para un rendimiento óptimo
CAPACIDAD DE GENERALIZACIÓN	Limitada a patrones lineales.	Buena capacidad para generalizar, pero puede sobreajustarse en datos pequeños	Puede capturar patrones complejos y generalizar bien con suficientes datos
VENTAJA	Simplicidad, fácil de interpretar, rápido entrenamiento.	Captura dependencias temporales, adecuado para datos secuenciales	Puede modelar patrones complejos y no lineales
DESVENTAJA	Limitado a relaciones lineales, no maneja bien los patrones no lineales.	Puede sufrir de problemas de desvanecimiento/exploración de gradientes	Necesita grandes cantidades de datos, más complejidad de ajuste

5. ANÁLISIS DE RESULTADOS

En esta sección de la investigación se presenta los resultados obtenidos de los modelos de aprendizaje automático, para posteriormente compararlos mediante métricas ya establecidas y así determinar cuál de los modelos de aprendizaje automático es el adecuado para predecir.

5.1. CASO DE ESTUDIO

Para la implementación de los modelos de aprendizaje automático con el objetivo de pronosticar la radiación solar, se utilizó software de código abierto en este caso el lenguaje de programación Python. La base de datos se obtuvo de la Estación Meteorológica Convencional Miraflores en la Universidad Nacional de Piura. La base de datos de un año de información comprendida entre abril del 2022 hasta abril del 2023, con un tiempo de muestra de una hora obteniendo un total de 8592 datos de la radiación solar, en el cual el 20 % de los datos serán utilizados para la validación de los modelos de aprendizaje automático, mientras que el 80 % de los datos se utilizarán para el entrenamiento de los modelos de predicción. Para determinar la precisión de los modelos de aprendizaje automático, se evalúa a cada una de los modelos mediante métricas ya establecidas anteriormente como: Error Medio Absoluto (MAE), Error Cuadrático Medio (MSE), Error Porcentual Absoluto Medio (MAPE) y el R^2 .

5.1.1. Análisis de predicción para 1 día

La Figura 5.1, proporciona la comparación de los resultados obtenidos de los modelos de aprendizaje automático con respecto a la radiación real, el análisis que se realiza corresponde al día 2023/02/06. Para más detalle revisar el anexo 8.

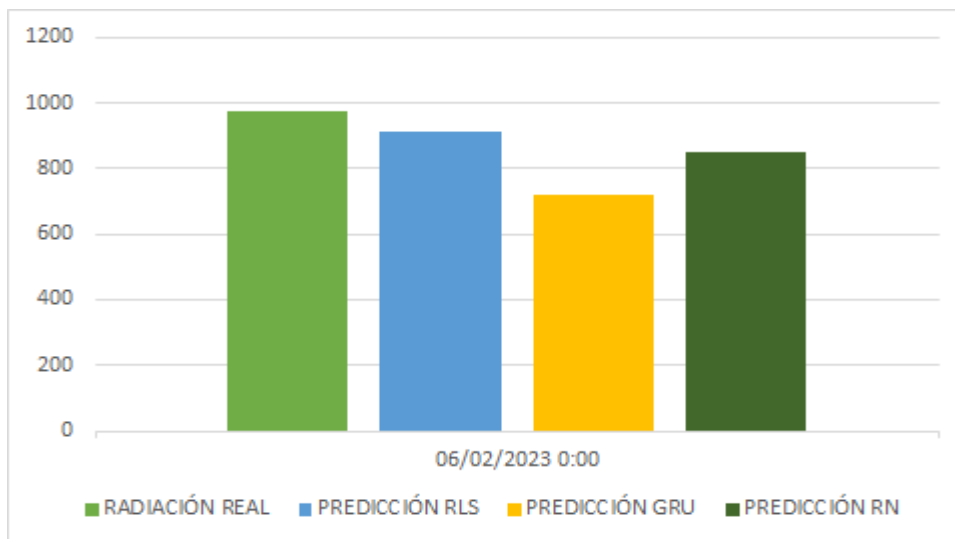


Figura 5.1. Resultados obtenidos de los modelos de predicción durante el 2023/02/06.

A continuación, en la Figura 5.2, se realiza un análisis del comportamiento de la radiación solar del día 2023/02/06. El perfil rojo representa la radiación solar real y el perfil azul representa la predicción, utilizando diferentes técnicas de aprendizaje automático como: a) regresión lineal simple, b) unidades recurrentes cerradas (GRU), c) redes neuronales.

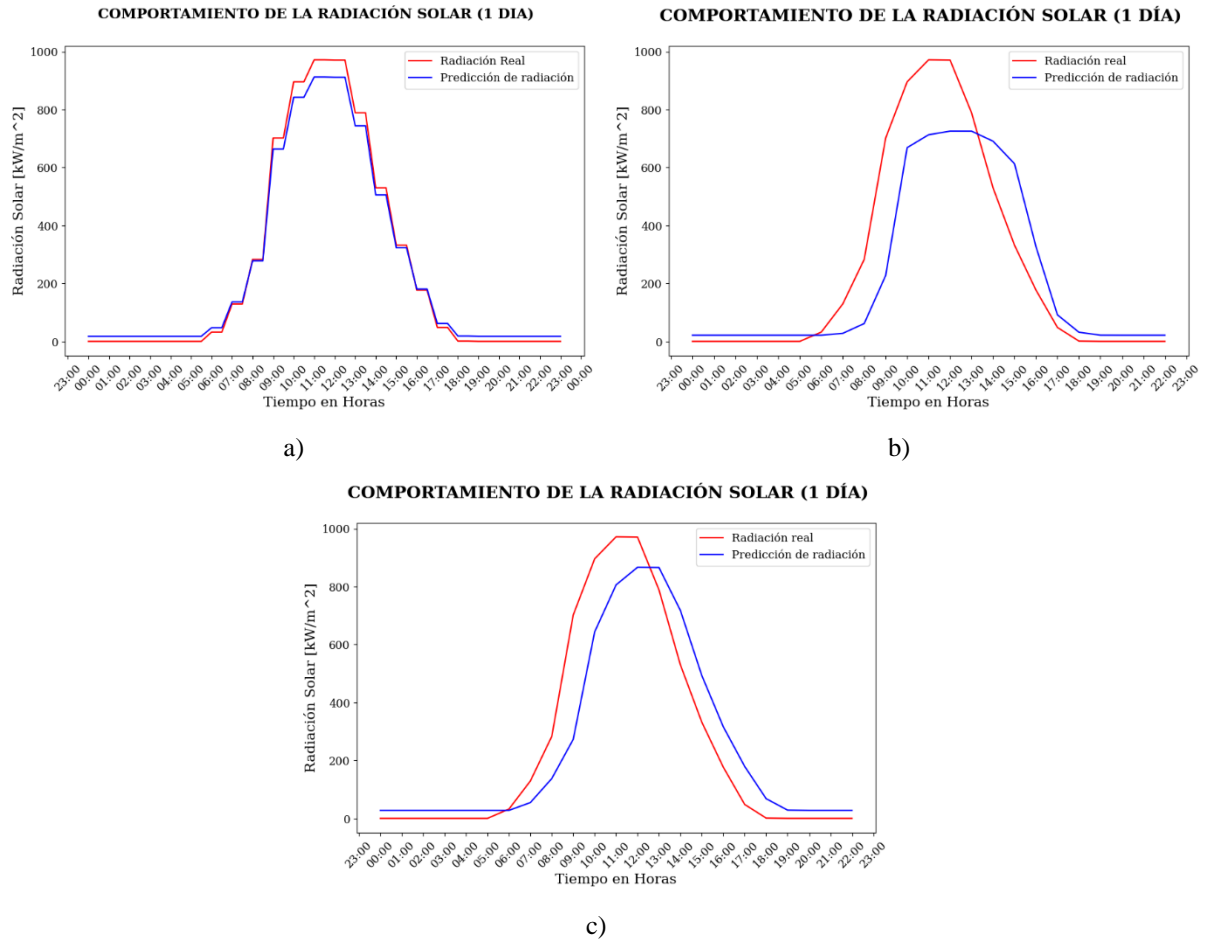


Figura 5.2. Técnicas de predicción de la radiación solar del día 2023/02/06: a) Regresión lineal simple, b) Unidades recurrentes cerradas (GRU), c) Redes neuronales.

En la Tabla 5.1, proporciona los resultados de las métricas de evaluación implementadas: Error Absoluto Medio (MAE), Error Cuadrático Medio (MSE) y el R^2 , para cada uno de las técnicas de aprendizaje automático para para predecir la radiación solar.

Tabla 5.1. Métricas de evaluación del modelo Regresión Lineal Simple.

Técnica	Error absoluto medio (MAE)	R ²	Error cuadrático medio (MSE)
Regresión lineal simple	22,773	0,9930	769,974
Unidades recurrentes cerradas (GRU)	107,884	0,7875	26177,015
Redes neuronales	96,371	0,8486	18650,726

La comparación de tres los modelos de pronóstico de la radiación solar, evaluados mediante métricas de rendimiento, la regresión lineal simple exhibe un rendimiento general superior, con un bajo error absoluto medio (MAE) de 22.773, un alto coeficiente de determinación R² de 0.9930 y un error cuadrático medio (MSE) de 769.974. Por otro lado, el modelo de unidades recurrentes cerradas (GRU) muestra un rendimiento menos favorable, con un MAE de 107.884, un R² de 0.7875 y un MSE de 26177.015. Las redes neuronales presentan resultados intermedios, con un MAE de 96.371, un R² de 0.8486 y un MSE de 18650.726. La Tabla 5.1, valida que la mejor técnica de aprendizaje automático para la predicción de la radiación solar para un día, es el modelo de la regresión lineal simple.

5.1.2. Análisis de predicción para 1 semana

El análisis que se realiza esta comprendido del 2023/02/13 hasta 2023/02/19, para comparar los resultados obtenidos de los modelos de predicción, como se muestra en la Figura 5.3.

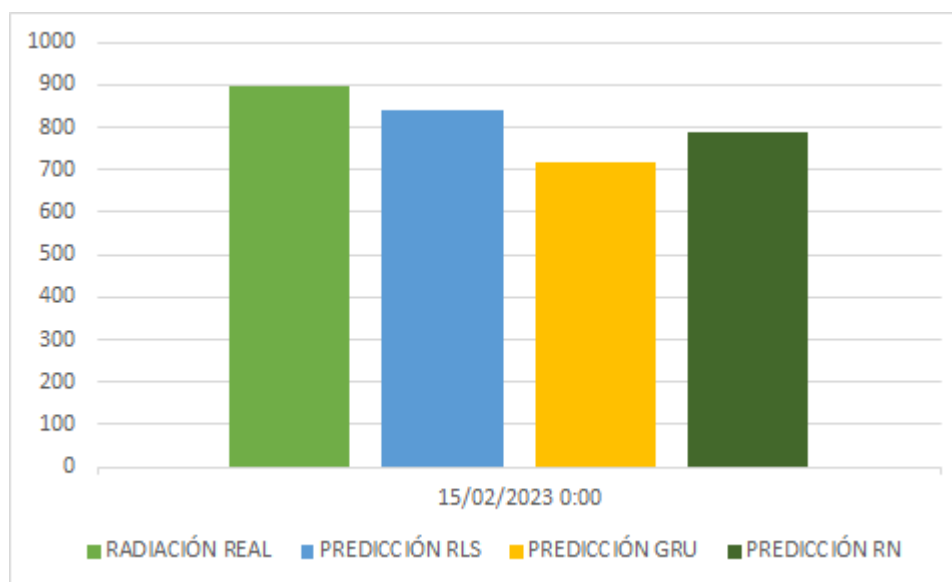


Figura 5.3. Resultados obtenidos de los modelos de predicción durante el 2023/02/13 al 2023/02/19.

En la Figura 5.4, se realiza un análisis del comportamiento de la radiación solar de una semana (2023/02/13 al 2023/02/19). El perfil rojo representa la radiación solar real y el perfil azul representa la predicción, utilizando diferentes técnicas de aprendizaje automático como: a) regresión lineal simple, b) unidades recurrentes cerradas (GRU), c) redes neuronales.

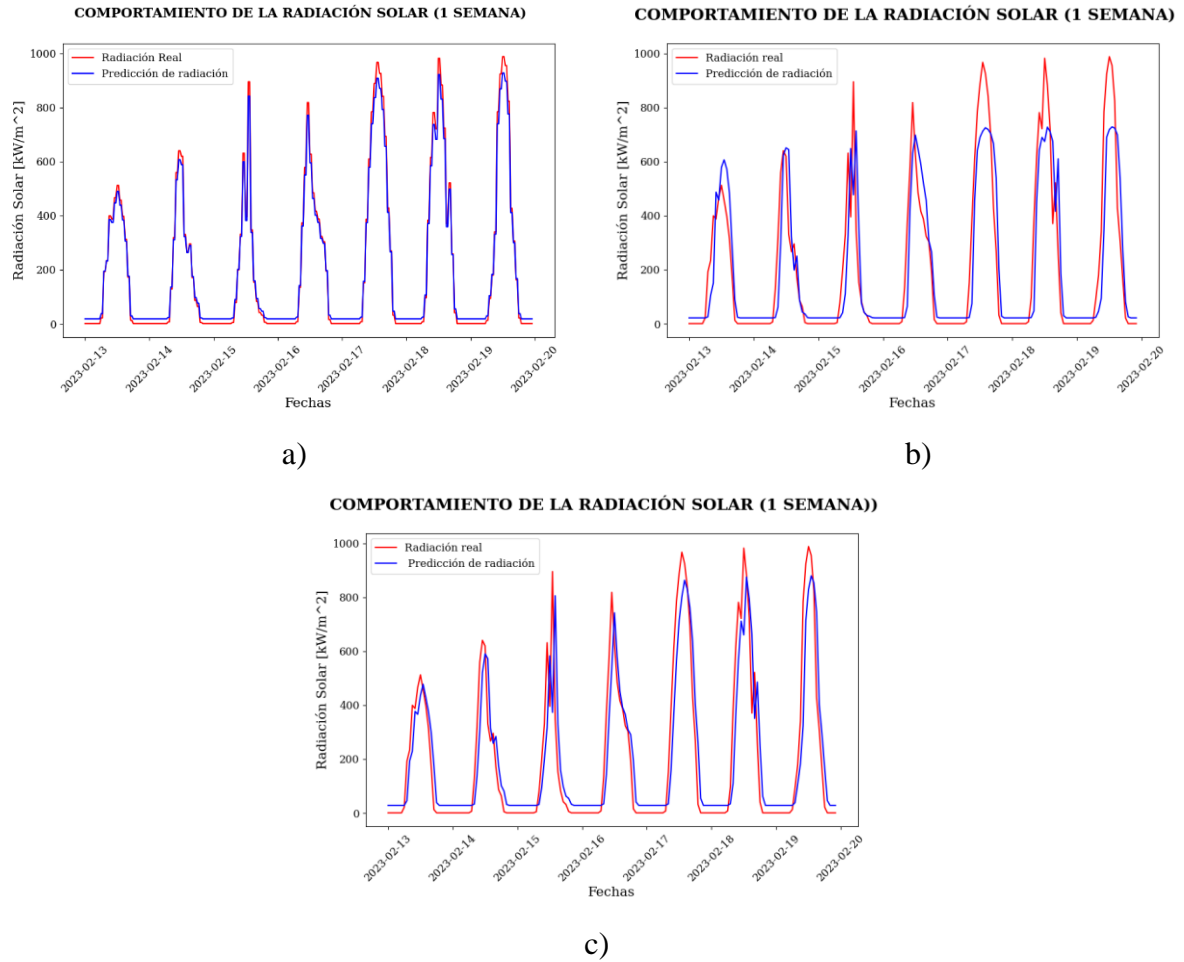


Figura 5.4. Técnicas de predicción de la radiación solar de una semana (2023/02/13 al 2023/02/19): a) Regresión lineal simple, b) Unidades recurrentes cerradas (GRU), c) Redes neuronales.

La Tabla 5.2, proporciona los resultados de las métricas de evaluación implementadas: Error Absoluto Medio (MAE), Error Cuadrático Medio (MSE) y el R^2 , para cada uno de las técnicas de aprendizaje automático para para predecir la radiación solar. El análisis corresponde al lapso de una semana (2023/02/13 al 2023/02/19).

Tabla 5.2. Métricas de evaluación para el modelo GRU.

Técnica	Error absoluto medio (MAE)	R ²	Error cuadrático medio (MSE)
Regresión lineal simple	19,379	0,9937	525,963
Unidades recurrentes cerradas (GRU)	88,450	0,7843	18032,076
Redes neuronales	84,443	0,8486	16082,952

La regresión lineal simple presenta el mejor rendimiento con un error absoluto medio (MAE) de 19.379, un coeficiente de determinación R² de 0.9937 y un error cuadrático medio (MSE) de 525.963. Por otro lado, las unidades recurrentes cerradas (GRU) muestran un rendimiento inferior con un MAE de 88.450, un R² de 0.7843 y un MSE de 18032.076. Las redes neuronales se sitúan entre ambos modelos, con un MAE de 84.443, un R² de 0.8486 y un MSE de 16082.952. La Tabla 5.2, valida que el mejor modelo es la regresión lineal simple supera a los modelos más complejos en esta evaluación específica.

5.1.3. Análisis de predicción para 1 mes

Se realiza un análisis de todo el mes de marzo del 2023, para comparar los resultados obtenidos de cada modelo de aprendizaje automático tal se muestra en la Figura 5.5. Para más detalle revisar el anexo 10.

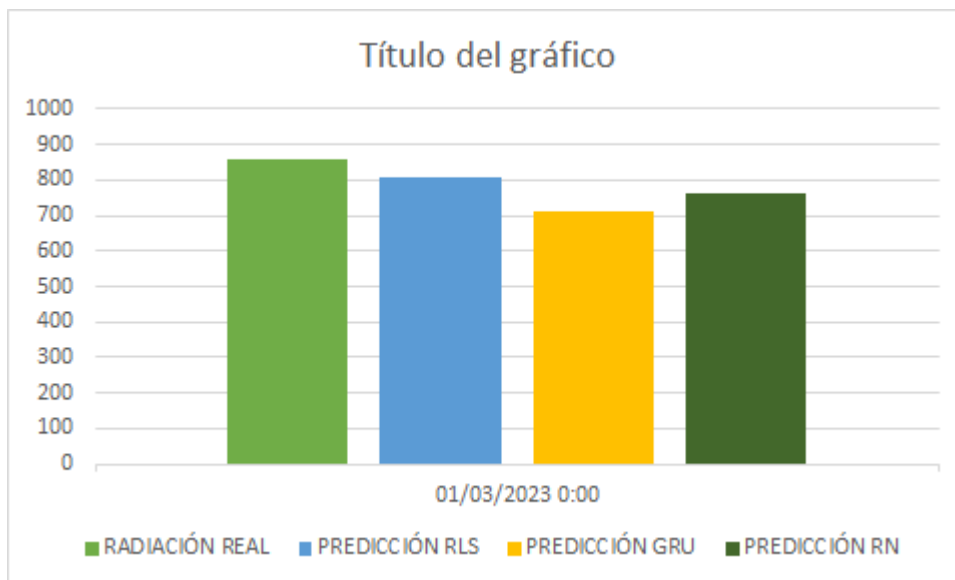


Figura 5.5. Resultados obtenidos de los modelos de predicción durante todo el mes de marzo del 2023.

La Figura 5.6, se procede a realizar un análisis del comportamiento de la radiación solar de todo el mes de marzo correspondiente al 2023. El perfil rojo representa la radiación solar real y el perfil azul representa la predicción, utilizando diferentes técnicas de aprendizaje automático como: a) regresión lineal simple, b) unidades recurrentes cerradas (GRU), c) redes neuronales.

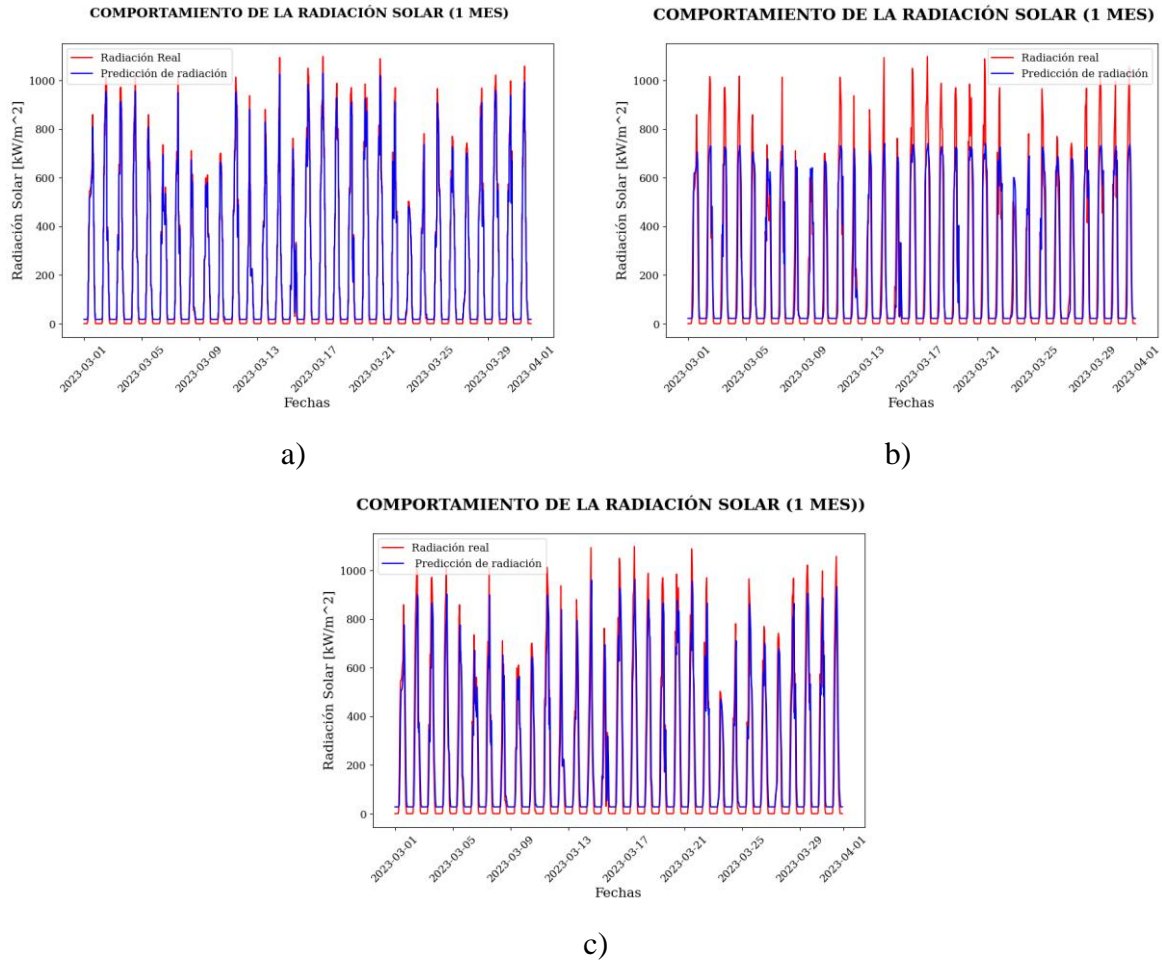


Figura 5.6. Técnicas de predicción de la radiación solar del mes de marzo del 2023: a) Regresión lineal simple, b) Unidades recurrentes cerradas (GRU), c) Redes neuronales.

En la Tabla 5.3, proporciona los resultados de las métricas de evaluación implementadas: Error Absoluto Medio (MAE), Error Cuadrático Medio (MSE) y el R2, para cada uno de las técnicas de aprendizaje automático para para predecir la radiación solar. El análisis corresponde a todo el mes de marzo del 2023.

Tabla 5.3. Métricas de evaluación para el modelo de Redes Neuronales.

Técnica	Error absoluto medio (MAE)	R ²	Error cuadrático medio (MSE)
Regresión lineal simple	20,6142	0,9937	592,233
Unidades recurrentes cerradas (GRU)	93,954	0,7720	21512,167
Redes neuronales	93,878	0,7912	19705.396

En términos de rendimiento, la regresión lineal simple continúa siendo la opción más efectiva entre los tres modelos evaluados. La regresión lineal presenta un bajo Error Absoluto Medio (MAE) de 20.6142, un alto coeficiente de determinación R² de 0.9937, y un Error Cuadrático Medio (MSE) de 592.233, indicando una precisa capacidad predictiva y una explicación sustancial de la variabilidad en los datos. En contraste, las unidades recurrentes cerradas (GRU) y las redes neuronales exhiben mayores MAE y MSE, junto con R² más bajos. La GRU tiene un MAE de 93.954, R² de 0.7720 y MSE de 21512.167, mientras que las redes neuronales tienen un MAE de 93.878, R² de 0.7912 y MSE de 19705.396. Estos resultados refuerzan la superioridad de la regresión lineal simple en términos de precisión y capacidad explicativa en esta evaluación específica.

6. CONCLUSIONES Y RECOMENDACIONES

6.1. CONCLUSIONES

En virtud de lo expuesto en el estado de arte, se identifica los métodos de aprendizaje automático para la predicción de la radiación solar, cabe destacar que los modelos de pronóstico basados en técnicas de aprendizaje automático como: regresión lineal simple, GRU y redes neuronales, muestran un potencial significativo para realizar pronósticos precisos. Un aspecto destacable es la importancia de la disponibilidad y flexibilidad de software de código abierto que facilite la implementación exitosa de modelos predicción.

La construcción y evaluación de modelos de pronóstico, tales como: regresión lineal simple, GRU, y redes neuronales, ha permitido una comprensión más profunda del comportamiento de la radiación solar a través del análisis de datos históricos. La diversidad de enfoques, desde métodos tradicionales como regresión lineal hasta modelos más complejos como GRU y redes neuronales, ha revelado diferentes perspectivas y potencialidades para mejorar la precisión en los pronósticos.

Mediante las métricas de evaluación del rendimiento y precisión, el que mejor destaca en los tres escenarios de análisis, es el modelo de regresión lineal simple debido a su consistente desempeño superior en diversas métricas. Con un bajo Error Absoluto Medio ($MAE = 19,379$), alto Coeficiente de Determinación ($R^2 = 0,9937$) cercano a 1, y el Error Cuadrático Medio ($MSE = 525,963$) más bajo en todas las instancias, esta técnica demuestra ofrecer predicciones más precisas y un ajuste robusto. La estabilidad en su rendimiento sugiere que la sencillez y linealidad de la regresión lineal simple son apropiadas y eficientes para tipo de análisis.

En conclusión, la Regresión Lineal Simple destaca como la opción más acertada para predicciones a corto plazo, evidenciada por su bajo Error Absoluto Medio (MAE) y alto Coeficiente de Determinación (R^2), por ende estas métricas validan un ajuste sólido y precisión en el corto plazo, por otro lado, para predicciones a largo plazo, las Unidades Recurrentes Cerradas (GRU) y las Redes Neuronales podrían ser consideradas, aunque tienen un MAE y un R^2 más bajos que la Regresión Lineal Simple, pueden capturar patrones temporales más complejos y no lineales, lo que podría ser beneficioso para predicciones a largo plazo.

6.2. RECOMENDACIONES

Se recomienda utilizar otros modelos de aprendizaje automático como: Redes Neuronales Convolucionales (CNN) para datos de series temporales, o modelos de series temporales específicos como ARIMA o Prophet. Estos enfoques pueden capturar patrones más complejos en los datos y ofrecer perspectivas adicionales en comparación con modelos más simples.

El uso de otros softwares libres puede ser beneficiosos tales como: KNIME, Rstudio, Orange o H2O.ai. Cada una tiene características únicas y puede ofrecer diferentes enfoques para el análisis de datos y la construcción de modelos de aprendizaje automático.

Es posible llevar a cabo este tipo de investigación en entornos de software privativos como MATLAB, ya que es conocido por su versatilidad en aplicaciones científicas y de ingeniería, también ofrece funciones y herramientas avanzadas para análisis de datos y modelado.

También es recomendable implementar el algoritmo de las series temporales con la simulación de Montecarlo, ya que este algoritmo nos permitió rellenar los datos faltantes de nuestra base de datos y así tener los datos adecuados para el estudio.

7. REFERENCIAS

- [1] J. A. ; Segovia *et al.*, “Meteorological Variables Forecasting System Using Machine Learning and Open-Source Software,” *Electronics* 2023, Vol. 12, Page 1007, vol. 12, no. 4, p. 1007, Feb. 2023, doi: 10.3390/ELECTRONICS12041007.
- [2] D. G. Castañeda Cachimuel and F. A. Fabara Vargas, “PREDICCIÓN DE LA RADIACIÓN SOLAR UTILIZANDO REDES NEURONALES PARA EL DIMENSIONAMIENTO DE PLANTAS FOTOVOLTAICAS EN LA PROVINCIA DE PICHINCHA,” 2023.
- [3] Gilberto Rubén Aguirre Salas, “Series temporales investigando variaciones aleatorias de radiación solar,” 2018.
- [4] Ismael Francisco Mendaña, ““Predicción de Radiación Solar a partir de Modelos Físicos y Aprendizaje Automático,”” 2018.
- [5] “Diferencias entre energías renovables y no renovables | ENEL.” Accessed: Nov. 23, 2023. [Online]. Available: <https://www.enel.com.co/es/historias/a202011-diferencia-energia-renovable-no-renovable.html>
- [6] “PM Consulting Energy | Comercialización de energía.” Accessed: Nov. 29, 2023. [Online]. Available: <https://www.thepmconsulting.com/es/comercializacion-de-energia-esp/>
- [7] “Tipos de Energía: Renovables y no Renovables.” Accessed: Nov. 23, 2023. [Online]. Available: <https://www.remica.es/tipos-de-energia/>
- [8] “ENERGÍAS RENOVABLES y NO RENOVABLES: ejemplos y resumen.” Accessed: Nov. 23, 2023. [Online]. Available: <https://www.ecologiaverde.com/energias-renovables-y-no-renovables-ejemplos-y-resumen-1248.html>
- [9] “Capacidad de energía renovable en América Latina y Caribe por país 2021 | Statista.” Accessed: Nov. 13, 2023. [Online]. Available: <https://es.statista.com/estadisticas/1076835/energia-renovable-capacidad-america-latina-por-pais/>
- [10] “▷ ¿Qué son las energías renovables? ♻️ | Twenergy.” Accessed: Nov. 13, 2023. [Online]. Available: <https://twenergy.com/energia/energias-renovables/que-son-las-energias-renovables-516/>

- [11] Luis. Jutglar Banyeras, “Energía solar,” 2004.
- [12] TADEO RICARDO ARGEÑAL, SAÚL NATÁN CASTRO, JOSÉ MANUEL DÁVILA, and FABIÁN LÓPEZ, “Radiación Solar.” Accessed: Nov. 29, 2023. [Online]. Available: <https://es.slideshare.net/jmanueldavila/radiacin-solar>
- [13] A. Javier Segarra Poma, “UBICACIÓN ÓPTIMA GEORREFERENCIADA DE CENTRALES DE GENERACIÓN FOTOVOLTAICA CONSIDERANDO RESTRICCIONES DE RADIACIÓN SOLAR Y TEMPERATURA,” 2022.
- [14] JOSÉ IGNACIO MOYA HERNÁNDEZ, “EVALUACIÓN DE MÉTODOS SUPERVISADOS DE APRENDIZAJE AUTOMÁTICO EN PRONÓSTICOS SOLARES INTRA-HORARIO,” 2023.
- [15] R. H. Inman, H. T. C. Pedro, and C. F. M. Coimbra, “Solar forecasting methods for renewable energy integration,” *Prog Energy Combust Sci*, vol. 39, no. 6, pp. 535–576, Dec. 2013, doi: 10.1016/J.PECS.2013.06.002.
- [16] “¿Qué es el aprendizaje automático? | Glosario | HPE LAMERICA.” Accessed: Nov. 13, 2023. [Online]. Available: <https://www.hpe.com/lamerica/es/what-is/machine-learning.html>
- [17] C. Camacho, “Regresión lineal simple,” 2020.
- [18] J. Dagnino S., “REGRESIÓN LINEAL,” *Revista Chilena de Anestesia*, vol. 43, no. 2, Apr. 2014, doi: 10.25237/REVCHILANESTV43N02.14.
- [19] Ebac, “Regresión Lineal: qué es, para qué sirve, por qué es importante, tipos y ejemplos de uso.” Accessed: Feb. 14, 2024. [Online]. Available: <https://ebac.mx/blog/regresion-lineal>
- [20] Castillo Fiallos Jessica Nataly and Bustamante Freire Fernando Santiago, “Unidades Recurrentes Cerradas (GRU) vs Redes Neuronales Artificiales en la predicción de la generación Eléctrica de la Central Hidroeléctrica Illuchi.,” 2023.
- [21] “¿Qué es la red neuronal recurrente?” Accessed: Nov. 29, 2023. [Online]. Available: <https://forum.huawei.com/enterprise/es/%25C2%25BFqu%25C3%25A9-es-la-red-neuronal-recurrente/thread/667225225659564032-667212895009779712>
- [22] W. Rivas, A. Bertha, and M. Olivo, “Redes neuronales artificiales aplicadas al reconocimiento de patrones,” 2017.




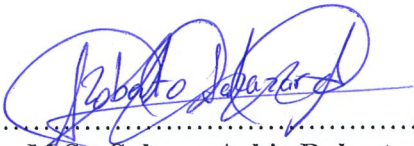
- [23] Miguel Bravo, Alexander Molero, Jorge Rojas, and Angel Gomez, “Arquitectura básica de una red neuronal.” Accessed: Jan. 24, 2024. [Online]. Available: https://www.researchgate.net/figure/Figura-1-Arquitectura-basica-de-una-red-neuronal-Haykin-1998-Partiendo-de-la-neurona_fig1_314151933
- [24] Pedro Larrañaga, Iñaki Inza, and Abdelmalik Moujahid, “Redes Neuronales,” 2019.
- [25] Antonio J. Serrano, Emilio Soria, and Jose D. Martín, “Redes Neuronales Artificiales,” 2023.
- [26] Carrillo Carrillo Manuel Francisco and Tapia Guerrero Jipson Danilo, “ESCENARIOS DE PRODUCCIÓN DE ENERGÍA ELÉCTRICA EN UNA PEQUEÑA CENTRAL HIDROELÉCTRICA UTILIZANDO SERIES SINTÉTICAS,” 2023.
- [27] Nicolás Arrijoja Landa Cosio, “Métricas en regresión.” Accessed: Nov. 26, 2023. [Online]. Available: <https://medium.com/@nicolasarrijoja/m%C3%A9tricas-en-regresi%C3%B3n-5e5d4259430b>
- [28] Nicolás Arrijoja Landa Cosio, “Métricas en regresión. La regresión es importante y conocer... | by Nicolás Arrijoja Landa Cosio | Medium.” Accessed: Nov. 29, 2023. [Online]. Available: <https://medium.com/@nicolasarrijoja/m%C3%A9tricas-en-regresi%C3%B3n-5e5d4259430b>
- [29] “Métrica de evaluación para modelos de regresión - Analytics Vidhya.” Accessed: Nov. 26, 2023. [Online]. Available: <https://www.analyticsvidhya.com/blog/2021/10/evaluation-metric-for-regression-models/>
- [30] EMENESES DEVELOPERS, “Aprendizaje Automático – Métricas de Regresión.” Accessed: Jan. 31, 2024. [Online]. Available: <https://emenesesdevelopers.com/2022/07/02/aprendizaje-automatico-metricas-de-regresion/>
- [31] Esteban Madrigal, “Métricas de precisión más comunes para Modelos de Regresión.” Accessed: Jan. 31, 2024. [Online]. Available: <https://www.growupcr.com/post/metricas-precision>
- [32] “▷ Coeficiente de determinación (R cuadrado).” Accessed: Nov. 26, 2023. [Online]. Available: <https://www.probabilidadyestadistica.net/coeficiente-de-determinacion-r-cuadrado/>

- [33] “▷ Correlación: qué es, tipos, fórmula, interpretación,...” Accessed: Nov. 26, 2023. [Online]. Available: https://www.probabilidadyestadistica.net/correlacion/#google_vignette
- [34] “▷ Correlación: qué es, tipos, fórmula, interpretación,...” Accessed: Nov. 29, 2023. [Online]. Available: <https://www.probabilidadyestadistica.net/correlacion/>
- [35] “¿Qué es la correlación estadística y cómo interpretarla? - Máxima Formación.” Accessed: Nov. 26, 2023. [Online]. Available: <https://www.maximaformacion.es/blog-dat/que-es-la-correlacion-estadistica-y-como-interpretarla/>
- [36] H. A. Delgado Rodríguez, “Richard Stallman - Creador del proyecto GNU Software Libre,” *Diseño Web akus.net*, 2019, Accessed: Nov. 13, 2023. [Online]. Available: <https://disenowebakus.net/richard-stallman.php>
- [37] “Lenguajes de programación más usados en el mundo ACTUALIZADO.” Accessed: Nov. 26, 2023. [Online]. Available: <https://www.yeeply.com/blog/lenguajes-de-programacion-mas-usados/>
- [38] “Python: El lenguaje del futuro - BAOSS.” Accessed: Nov. 13, 2023. [Online]. Available: <https://www.baoss.es/python-el-lenguaje-del-futuro/>
- [39] “Librerías de Python, ¿qué son y cuáles son las mejores?” Accessed: Nov. 26, 2023. [Online]. Available: <https://immune.institute/blog/librerias-python-que-son/>
- [40] Larkin Alonso, “¿Qué es TensorFlow y para qué sirve?” Accessed: Nov. 26, 2023. [Online]. Available: <https://www.incentro.com/es-ES/blog/que-es-tensorflow>
- [41] “Scikit-Learn: Descubre la biblioteca de Python dedicada al Machine Learning.” Accessed: Nov. 26, 2023. [Online]. Available: <https://datascientest.com/es/scikit-learn-decubre-la-biblioteca-python>
- [42] Chacón José Luis, “Introducción a Pandas, la librería de Python para trabajar con datos.” Accessed: Nov. 26, 2023. [Online]. Available: <https://profile.es/blog/pandas-python/>
- [43] “📖 Lista de las mejores librerías de Python para el 2022.” Accessed: Nov. 26, 2023. [Online]. Available: <https://www.crehana.com/blog/transformacion-digital/librerias-python/>
- [44] Gonzalez Ligdi, “Librería NumPy - 🤖 Aprende IA.” Accessed: Nov. 26, 2023. [Online]. Available: <https://aprendeia.com/libreria-de-python-numpy-machine-learning/>

- [45] Guevara Alban Gladys Patricia, Verdesoto Arguello Alexis Eduardo, and Castro molina Nelly Esther, “Metodologías de investigación educativa (descriptivas, experimentales, participativas, y de investigación-acción),” 2020.
- [46] D. A. Neill and L. Cortez Suárez, “Procesos y Fundamentos de la Investigación Científica,” 2020.
- [47] C. Ramos-Galarza, “Diseños de investigación experimental,” *CienciAmérica*, vol. 10, no. 1, pp. 1–7, Feb. 2021, doi: 10.33210/ca.v10i1.356.
- [48] “Historia – Universidad Nacional de Piura.” Accessed: Nov. 29, 2023. [Online]. Available: <https://www.unp.edu.pe/historia/>
- [49] “SENAMHi - Piura.” Accessed: Nov. 29, 2023. [Online]. Available: <https://www.senamhi.gob.pe/main.php?dp=piura&p=estaciones>

ANEXOS

Anexo 1: Informe Anti-Plagio

 UNIVERSIDAD TÉCNICA DE COTOPAXI FACULTAD DE CIENCIAS DE LA INGENIERÍA Y APLICADAS CARRERA DE INGENIERÍA EN ELECTRICIDAD 	
Facultad:	Ciencias de la Ingeniería y Aplicadas
Carrera:	Ingeniería en Electricidad
Nombre del docente evaluador que emite el informe:	Ing. M.Sc. Salazar Achig Roberto Edgar
Documento evaluado:	Proyecto de investigación presentado previo a la obtención del Título de Ingeniero Eléctrico
Autores del documento:	Muguicha Hinojoza Cristhian Ismael
Programa de similitud utilizado:	Sistema Compilatio
Porcentaje de similitud según el programa utilizado:	6 %
Observaciones: Calificación de originalidad atendiendo a los siguientes criterios: <ul style="list-style-type: none"> • El documento cumple criterios de originalidad, sin observaciones. • El documento cumple criterios de originalidad, con observaciones. • El documento no cumple criterios de originalidad. 	X
Fecha de realización de informe:	2024-02-27
Captura de pantalla del documento analizado: <div data-bbox="231 1411 1372 1702" style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;">  <p>INFORME DE ANÁLISIS mugister</p> <p>GRUPO_23_MUGUICHA</p> <p>6% Textos sospechosos</p> <p>6% Similitudes < 1% similitudes entre comillas 0% entre las fuentes mencionadas 0% Idiomas no reconocidos</p> <p>Nombre del documento: GRUPO_23_MUGUICHA.pdf ID del documento: 8445039bf125620a0d05736d2644863006dbbf00 Tamaño del documento original: 6,31 MB</p> <p>Depositante: EDGAR ROBERTO SALAZAR ACHIG Fecha de depósito: 18/2/2024 Tipo de carga: Interface fecha de fin de análisis: 18/2/2024</p> <p>Número de palabras: 14.635 Número de caracteres: 97.886</p> </div>	
 Ing. M.Sc. Salazar Achig Roberto Edgar C.C: 050284761-9 TUTOR	

Anexo 2: Programación del modelo de regresión lineal.

```
# Importar bibliotecas
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score, mean_absolute_error
from sklearn.preprocessing import MinMaxScaler
from datetime import datetime
```

```
# Importar datos
data = pd.read_excel('DATA_RADIACION.xlsx')
print(data)
```

```
# División de datos
division = int(0.2 * datos_concatenado.shape[0])
datos_test = datos_concatenado[-division:]
datos_train = datos_concatenado[:-division]
```

```
# Entrenamiento del modelo
time_step = 1
X_train = []
Y_train = []
m = len(datos_train_scaled)

for i in range(time_step, m):
    X_train.append(datos_train_scaled[i - time_step:i, 0])
    Y_train.append(datos_train_scaled[i, 0])

X_train, Y_train = np.array(X_train), np.array(Y_train)
X_train = X_train.reshape(-1, time_step)
```

```
# Modelo Lineal
modelo = LinearRegression()
modelo.fit(X_train, Y_train)
```

```
# Datos de prueba
x_test = datos_test.values
x_test_scaled = sc.transform(x_test)
```

```
# Organizar datos de prueba
X_test = []
for i in range(time_step, len(x_test_scaled)):
    X_test.append(x_test_scaled[i - time_step:i, 0])

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

# Realizar predicciones
prediccion = modelo.predict(X_test)
prediccion1 = sc.inverse_transform(prediccion.reshape(-1, 1))

from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error

# Crear y entrenar el modelo
modelo = LinearRegression()
modelo.fit(X_train, Y_train)

# Realizar predicciones en datos de prueba
prediccion = modelo.predict(X_test)
```

Anexo 3: Programación del modelo GRU

```
# Importancio de Librerias
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
import keras
from keras.models import Sequential
from keras.layers import Dense, Activation, Dropout
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score
import plotly.express as px
import os
from datetime import datetime
```

```
# Lectura de datos
data = pd.read_excel('DATA_RADIACION.xlsx')
```

```
data
```

```
#DIVISIÓN DE DATOS
#Dividir los datos de entrenamiento y de test

#Tomamos el 20% de los datos finales como conjunto de prueba y el resto como conjunto de entrenamiento.
size = int(0.2*data_final.shape[0])
data_test = data_final[-size:]
data_train = data_final[:-size]
```

```
#Tenemos que entrenar la red por bloques de datos
#Tomamos bloques de 1 datos consecutivos
time_step = 1
X_train = [] #Lista de conjuntos de 60 datos
Y_train = []
m = len(data_train_scaled)
for i in range(time_step, m):
    X_train.append(data_train_scaled[i-time_step:i, 0]) #Guardamos 60 valores
    Y_train.append(data_train_scaled[i, 0]) #Buscamos predecir el dato time_step + 1, no la var potencia
X_train, Y_train = np.array(X_train), np.array(Y_train) #Usamos np por reshape y optimizacion
X_train.shape, Y_train.shape

X_train = np.reshape(X_train, (X_train.shape[0], X_train.shape[1], 1)) #Agregamos una nueva dimension
X_train.shape, Y_train.shape
```

```

from keras.layers import Dense, Activation, Dropout

def basic_GRU(optimizador, unidades, epocas):

    input_dim = (X_train.shape[1],X_train.shape[2])
    output_dim = 1
    num_unidades = unidades

    model = Sequential()

    model.add(GRU(units=num_unidades, return_sequences = True, input_shape=(input_dim))) # 256, return_sequences = True
    # model.add(GRU(units=256, recurrent_dropout = 0.2)) #, return_sequences = True
    model.add(Dropout(0.5))
    model.add(GRU(units=num_unidades, input_shape=(input_dim)))
    #model.add(Dense(128))
    model.add(Dense(32, activation='relu'))
    # model.add(Dense(32))
    model.add(Dropout(0.5))
    model.add(Dense(1,activation='sigmoid'))
    model.compile(optimizer= optimizador, loss='mse', metrics = ['mean_absolute_error'])
    history = model.fit(X_train, Y_train, epochs = epocas, batch_size = 128)
    return model, history

```

#ADAM LOS MEJORES RESULTADOS

```

model, history = basic_GRU(optimizador = 'adam', unidades = 10, epocas = 10)
#Guardamos el modelo para no tener que entrenar otra vez para probar
model.save('MODELOS GRU/model.h5')

```

```

model = keras.models.load_model('MODELOS GRU/model.h5')

```

```

prediccion = predDataTest(model)

x_test = data_test.values
x_test = sc.transform(x_test)
## x_test
X_test = []
for i in range(time_step, len(x_test)):
    X_test.append(x_test[i-time_step: i, 0])
X_test= np.array(X_test)
X_test = np.reshape(X_test, (X_test.shape[0], X_test.shape[1], 1))
X_test.shape
prediccion = model.predict(X_test)
prediccion = sc.inverse_transform(prediccion)
len(prediccion)

```

Anexo 4: Programación de las redes neuronales

```
# Importar las Librerías
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
import keras
from keras.models import Sequential
from keras.layers import Dense, Activation, Dropout
from sklearn.metrics import mean_squared_error
```

```
# Lectura de datos
data = pd.read_excel('DATA_RADIACION.xlsx')
data
```

```
#Tenemos que entrenar la red por bloques de datos
#Time_step es el tamaño del bloque
#Cada n datos genera una salida
#Tomamos bloques de 60 datos consecutivos
time_step = 1
X_train = [] #Lista de conjuntos de 60 datos
Y_train = []
m = len(data_train_scaled)
for i in range(time_step, m):
    X_train.append(data_train_scaled[i-time_step:i, 0]) #Guardamos 60 valores
    Y_train.append(data_train_scaled[i, 0]) #Buscamos predecir el dato time_step + 1, no la var potencia
X_train, Y_train = np.array(X_train), np.array(Y_train) #Usamos np por reshape y optimizacion
X_train.shape, Y_train.shape

X_train = np.reshape(X_train, (X_train.shape[0], X_train.shape[1], 1)) #Agregamos una nueva dimension
X_train.shape, Y_train.shape
```

```
#Llevamos de dos a 3 dimensiones nuestro arreglo
X_train = np.reshape(X_train, (X_train.shape[0], X_train.shape[1], 1)) #Agregamos una nueva dimension
X_train.shape, Y_train.shape
```

```
#IMPLEMENTACIÓN DEL MODELO
def NN_LSTM(optimizador, neuronas, epocas):
    dim_entrada = (X_train.shape[1],X_train.shape[2])
    dim_salida = 1
    num_neuronas = neuronas

    model = Sequential()

    model.add(LSTM(units=num_neuronas, return_sequences = True, input_shape=dim_entrada))
    model.add(Dropout(0.4))
    model.add(LSTM(units=num_neuronas))
    model.add(Dropout(0.2))
    model.add(Dense(units = dim_salida))
    # model.add(activation = 'linear')
    model.compile(optimizer= optimizador, loss='mse', metrics = ['mean_absolute_error']) #mean_squared_error

    #Entrenamiento
    history = model.fit(X_train, Y_train, epochs = epocas, batch_size = 128)
    return model, history
```

```
#rmsprop LOS MEJORES RESULTADOS
model, history = NN_LSTM(optimizador = 'rmsprop', neuronas = 15, epocas = 15)
#Guardamos el modelo para no tener que entrenar otra vez para probar
model.save('modelos/model.h5')
```

```
# Código para realizar el análisis diario, semanal y mensual
#Los primeros días de la prediccion
semana_estudio_test_pred = data_test_pred[
    data_test_pred.index.get_loc('2023-02-13 00:00:00'):
    data_test_pred.index.get_loc('2023-02-19 23:00:00')]

semana_estudio_test = semana_estudio_test_pred['RADIACION']
semana_estudio_pred = semana_estudio_test_pred['RADIACION_PRED']
# semana_estudio_pred
plt.figure(figsize=(10,6))
plt.plot(semana_estudio_pred,
         color='blue',
         label='Datos Predecidos')
plt.suptitle('COMPORTAMIENTO DE LA RADIACION SOLAR (1 DIA)',
            fontsize = 18,
            fontweight = 'bold',
            fontfamily='serif')
plt.legend()
plt.xlabel('Tiempo', fontsize=15, fontfamily='serif')
plt.ylabel('Radiacion Solar [kW/m^2]', fontsize=15, fontfamily='serif')
plt.legend()

# Guardar el gráfico en la carpeta 'graficos' como un archivo PNG
plt.savefig(os.path.join(carpeta_graficos, 'radiacion_predecida.png'))
```

Anexo 5: Resultados obtenidos del modelo Regresión Lineal Simple para 1 día.

DIA/HORA	RADIACIÓN_REAL	RLS_PREDICCIÓN
2023-02-06 00:00:00	0	17,615
2023-02-06 01:00:00	0	17,615
2023-02-06 02:00:00	0	17,615
2023-02-06 03:00:00	0	17,615
2023-02-06 04:00:00	0	17,615
2023-02-06 05:00:00	0	17,615
2023-02-06 06:00:00	32	47,080
2023-02-06 07:00:00	129	136,395
2023-02-06 08:00:00	283	278,196
2023-02-06 09:00:00	702	664,003
2023-02-06 10:00:00	896	842,634
2023-02-06 11:00:00	972	912,614
2023-02-06 12:00:00	971	911,693
2023-02-06 13:00:00	789	744,111
2023-02-06 14:00:00	530	505,629
2023-02-06 15:00:00	332	323,314
2023-02-06 16:00:00	1772	180,593
2023-02-06 17:00:00	48	61,812
2023-02-06 18:00:00	1	18,535
2023-02-06 19:00:00	0	17,615
2023-02-06 20:00:00	0	17,615
2023-02-06 21:00:00	0	17,615
2023-02-06 22:00:00	0	17,615
2023-02-06 23:00:00	0	17,615

**Anexo 6: Resultados obtenidos del modelo Unidades Recurrentes Cerradas (GRU)
simple para 1 día.**

DIA/HORA	RADIACIÓN_REAL	GRU_PREDICCIÓN
2023-02-06 00:00:00	0	18,928
2023-02-06 01:00:00	0	18,928
2023-02-06 02:00:00	0	18,928
2023-02-06 03:00:00	0	18,928
2023-02-06 04:00:00	0	18,928
2023-02-06 05:00:00	0	18,928
2023-02-06 06:00:00	32	18,928
2023-02-06 07:00:00	129	24,733
2023-02-06 08:00:00	283	56,111
2023-02-06 09:00:00	702	219,101
2023-02-06 10:00:00	896	663,696
2023-02-06 11:00:00	972	708,708
2023-02-06 12:00:00	971	720,998
2023-02-06 13:00:00	789	720,853
2023-02-06 14:00:00	530	686,482
2023-02-06 15:00:00	332	606,260
2023-02-06 16:00:00	1772	317,973
2023-02-06 17:00:00	48	85,805
2023-02-06 18:00:00	1	28,299
2023-02-06 19:00:00	0	19,086
2023-02-06 20:00:00	0	18,928
2023-02-06 21:00:00	0	18,928
2023-02-06 22:00:00	0	18,928
2023-02-06 23:00:00	0	18,928

Anexo 7: Resultados obtenidos del modelo de Redes Neuronales para 1 día.

DIA/HORA	RADIACIÓN_REAL	RN_ PREDICCIÓN
2023-02-06 00:00:00	0	27,368
2023-02-06 01:00:00	0	27,368
2023-02-06 02:00:00	0	27,368
2023-02-06 03:00:00	0	27,368
2023-02-06 04:00:00	0	27,368
2023-02-06 05:00:00	0	27,368
2023-02-06 06:00:00	32	27,368
2023-02-06 07:00:00	129	54,268
2023-02-06 08:00:00	283	137,156
2023-02-06 09:00:00	702	271,596
2023-02-06 10:00:00	896	634,320
2023-02-06 11:00:00	972	790,054
2023-02-06 12:00:00	971	847,704
2023-02-06 13:00:00	789	846,959
2023-02-06 14:00:00	530	705,570
2023-02-06 15:00:00	332	488,124
2023-02-06 16:00:00	1772	314,722
2023-02-06 17:00:00	48	178,778
2023-02-06 18:00:00	1	67,807
2023-02-06 19:00:00	0	28,205
2023-02-06 20:00:00	0	27,368
2023-02-06 21:00:00	0	27,368
2023-02-06 22:00:00	0	27,368
2023-02-06 23:00:00	0	27,368

Anexo 8: Comparación de los resultados obtenidos a partir de los modelos de aprendizaje automático en el análisis correspondiente al día (2023/20/06).

DIA/HORA	RADIACIÓN REAL	PREDICCIÓN RLS	PREDICCIÓN GRU	PREDICCIÓN RN
2023-02-06 00:00:00	0	17,615	18,928	27,368
2023-02-06 01:00:00	0	17,615	18,928	27,368
2023-02-06 02:00:00	0	17,615	18,928	27,368
2023-02-06 03:00:00	0	17,615	18,928	27,368
2023-02-06 04:00:00	0	17,615	18,928	27,368
2023-02-06 05:00:00	0	17,615	18,928	27,368
2023-02-06 06:00:00	32	47,080	18,928	27,368
2023-02-06 07:00:00	129	136,395	24,733	54,268
2023-02-06 08:00:00	283	278,196	56,111	137,156
2023-02-06 09:00:00	702	664,003	219,101	271,596
2023-02-06 10:00:00	896	842,634	663,696	634,320
2023-02-06 11:00:00	972	912,614	708,708	790,054
2023-02-06 12:00:00	971	911,693	720,998	847,704
2023-02-06 13:00:00	789	744,111	720,853	846,959
2023-02-06 14:00:00	530	505,629	686,482	705,570
2023-02-06 15:00:00	332	323,314	606,260	488,124
2023-02-06 16:00:00	172	180,593	317,973	314,722
2023-02-06 17:00:00	48	61,812	85,805	178,778
2023-02-06 18:00:00	1	18,535	28,299	67,807
2023-02-06 19:00:00	0	17,615	19,086	28,205
2023-02-06 20:00:00	0	17,615	18,928	27,368
2023-02-06 21:00:00	0	17,615	18,928	27,368
2023-02-06 22:00:00	0	17,615	18,928	27,368
2023-02-06 23:00:00	0	17,615	18,928	27,368

Anexo 9: Comparación de los resultados obtenidos a partir de los modelos de aprendizaje automático en el análisis correspondiente a una semana (2023/02/13 a 2023/02/19).

DIA/HORA	RADIACIÓN REAL	PREDICCIÓN RLS	PREDICCIÓN GRU	PREDICCIÓN RN
15/02/2023 0:00	0	17,615	17,474	27,368
15/02/2023 1:00	0	17,615	17,474	27,368
15/02/2023 2:00	0	17,615	17,474	27,368
15/02/2023 3:00	0	17,615	17,474	27,368
15/02/2023 4:00	0	17,615	17,474	27,368
15/02/2023 5:00	0	17,615	17,474	27,368
15/02/2023 6:00	0	17,615	17,474	27,368
15/02/2023 7:00	4	21,298	18,073	27,368
15/02/2023 8:00	78	89,436	33,818	30,717
15/02/2023 9:00	199	200,85	94,366	93,344
15/02/2023 10:00	331	322,393	289,878	197,956
15/02/2023 11:00	631	598,627	651,035	313,841
15/02/2023 12:00	395	381,323	436,937	574,719
15/02/2023 13:00	895	841,714	718,175	370,169
15/02/2023 14:00	347	337,126	324,495	789,282
15/02/2023 15:00	153	158,494	63,546	327,93
15/02/2023 16:00	82	93,119	34,983	157,926
15/02/2023 17:00	41	55,367	24,713	96,763
15/02/2023 18:00	31	46,159	22,705	61,877
15/02/2023 19:00	5	22,219	18,226	53,423
15/02/2023 20:00	0	17,615	17,474	31,555
15/02/2023 21:00	0	17,615	17,474	27,368
15/02/2023 22:00	0	17,615	17,474	27,368
15/02/2023 23:00	0	17,615	18,928	27,368

Anexo 10: Comparación de los resultados obtenidos a partir de los modelos de aprendizaje automático en el análisis correspondiente a todo el mes de marzo del 2023.

DIA/HORA	RADIACIÓN REAL	PREDICCIÓN RLS	PREDICCIÓN GRU	PREDICCIÓN RN
2023-03-01 00:00:00	0	17,615	17,474	27,368
2023-03-01 01:00:00	0	17,615	17,474	27,368
2023-03-01 02:00:00	0	17,615	17,474	27,368
2023-03-01 03:00:00	0	17,615	17,474	27,368
2023-03-01 04:00:00	0	17,615	17,474	27,368
2023-03-01 05:00:00	0	17,615	17,474	27,368
2023-03-01 06:00:00	11	27,743	17,474	27,368
2023-03-01 07:00:00	171	175,068	19,172	36,587
2023-03-01 08:00:00	413	397,897	73,993	173,558
2023-03-01 09:00:00	546	520,361	480,613	385,986
2023-03-01 10:00:00	547	521,282	618,803	501,960
2023-03-01 11:00:00	561	534,173	619,282	502,824
2023-03-01 12:00:00	609	578,370	625,912	514,895
2023-03-01 13:00:00	631	598,627	643,526	556,020
2023-03-01 14:00:00	859	808,566	651,035	574,719
2023-03-01 15:00:00	700	662,161	711,345	761,262
2023-03-01 16:00:00	417	401,580	672,604	632,658
2023-03-01 17:00:00	150	155,732	489,842	389,499
2023-03-01 18:00:00	27	42,476	61,967	155,325
2023-03-01 19:00:00	0	17,615	21,949	50,049
2023-03-01 20:00:00	0	17,615	17,474	27,368
2023-03-01 21:00:00	0	17,615	17,474	27,368
2023-03-01 22:00:00	0	17,615	17,474	27,368
2023-03-01 23:00:00	0	17,615	17,474	27,368