UNIVERSIDAD NACIONAL DE MOQUEGUA

ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS E INFORMÁTICA



"Predicción diaria de radiación solar en la provincia de Ilo con modelo Transformer"

Tesis

Presentada por:

Bach. Raul Donato Cruz Huacac

Para optar el Título Profesional de:

Ingeniero de Sistemas e Informática

Ilo - Perú

2024

UNAM
UNIVERSIDAD NACIONAL DE MOQUEGUA

ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS E INFORMÁTICA

Folio **Nº 016**

ACTA DE SUSTENTACIÓN

I. Modalidad:

Tesis : [X]
Trabajo de Suficiencia Profesional : [ ]
Otro: ..... : [ ]

II. Documento de aprobación para la sustentación:

Resolución de Facultad N°235-2024-FACIA-UNAM

Fecha: 26 noviembre 2024

III. El jurado ha dictaminado la (el) Tesis, TITULADO:

Prediccion Diaria de Radiacion Solar en la Provincia de Ilo con Modelo Transformer

Bachiller: Raul Donato Cruz Huacac

Asesor Principal: Hugo Euler Tito Chura

Co-Asesor: .....

IV. Se dictamina:

Como: Aprobado por Unanimidad la sustentación de manera:

Presencial [X] Virtual [ ]

Con nota: Quince (15) por lo que; Sí [X] No [ ]

Opta el Título Profesional de INGENIERO DE SISTEMAS E INFORMÁTICA.

Siendo las 13:30 horas, del 4 de diciembre del año 2024 se procede con la firma en conformidad de lo dictaminado por el Jurado.

Carlos A. Seco Salgado
Presidente

Victor Yanz Mamani
Primer Miembro

Juan Carlos Flores Perca
Segundo Miembro

Raul Donato Cruz Huacac
Bachiller

Dedicatoria

A mi madre GENARA

A mis hermanos

En memoria de mi padre ROSALIO CONCEPCIÓN.

Agradecimientos

Mi agradecimiento especial a la Universidad Nacional de Moquegua, la cual me abrió sus puertas para formarme profesionalmente.

Me gustaría mostrar mi más profundo agradecimiento a mi asesor de tesis M.Sc. Hugo Euler Tito Chura, por su apoyo constante que fueron fundamentales para realización de la presente investigación y su confianza en mí para seguir adelante y superar los desafíos.

Mi más sincero agradecimiento a los miembros del jurados, quienes me ayudaron con sus comentarios y sugerencias para mejorar este trabajo.

Resumen

La predicción de radiación solar es uno de los temas más relevantes en la actualidad, ya que proporciona la información para aprovechar la energía y mitigar los efectos de radiación solar. En esta tesis de investigación, se implementó el modelo predictivo con una de las arquitecturas avanzadas de deep learning (DL) denominada Transformer; en el cual ha sido aplicada con datos reales de series de tiempo, se utilizó datos diarios de radiación solar de un lugar especifico de la provincia de Ilo, con las siguientes coordenadas de latitud -17.63185 y longitud -71.34108, obtenidos del repositorio del proyecto POWER de la NASA; para analizar y modelar con datos históricos de radiación solar se aplicó la metodología de CRISP-DM, cuyos resultados alcanzados en la etapa de evaluación del modelo óptimo, mostró su rendimiento con valores de $RMSE = 0{,}1559\ kwh/m^2$ y $R^2 = 99{,}12\,\%$, permitió lograr estos valores con la utilización de unos de los bloques de Transformer, como en este caso, el Transformer de solo decodificador, dado que este bloque permite predecir futuro en series de tiempo. Se comparó con otros modelos tradicionales: media móvil integrada autorregresiva (ARIMA), como así también con modelos clásicos de DL: redes neuronales recurrentes (RNN), memoria a corto plazo (LSTM), y unidad recurrentes cerradas (GRU), donde el modelo propuesto muestra la superioridad en predicción de radiación solar, siendo así, el mas efectivo en tareas de predicción.

Palabras Claves: series de tiempo, radiación solar, predicción, deep learning, transformer.

Abstract

Solar radiation prediction is one of the most relevant topics nowadays, since it provides the information to take advantage of the energy and mitigate the effects of solar radiation. In this research thesis, the predictive model was implemented with one of the advanced architectures of deep learning (DL) called Transformer, which has been applied with real time series data, using daily solar radiation data from a specific location in the province of Ilo, with the following coordinates of latitude -17.63185 and longitude -71. 34108, obtained from the repository of the NASA POWER project; to analyze and model with historical solar radiation data, the CRISP-DM methodology was applied, whose results reached in the evaluation stage of the optimal model, showed its performance with values of $RMSE = 0{,}1559\ kwh/m^2$ and $R^2 = 99{,}12\,\%$, allowed to achieve these values with the use of one of the Transformer blocks, as in this case, the Transformer of only decoder, since this block allows predicting future in time series. It was compared with other traditional models: autoregressive integrated moving average (ARIMA), as well as with classical DL models: recurrent neural networks (RNN), short term memory (LSTM), and closed recurrent unit (GRU), where the proposed model shows superiority in solar radiation prediction, being thus, the most effective in prediction tasks.

Keywords: time series, solar radiation, prediction, deep learning, transformer.

Índice general

Introducción **8**

1. Planteamiento del problema **10**

1.1. Descripción del problema 10

1.2. Formulación del problema 11

1.3. Objetivos 12

1.3.1. *Objetivo general* 12

1.3.2. *Objetivos específicos* 12

1.4. Justificación 13

1.5. Hipótesis 13

1.5.1. *Hipótesis general* 13

1.5.2. *Hipótesis especifica* 14

1.6. Variables 14

1.6.1. *Variable independiente* 14

1.6.2. *Variable dependiente* 14

2. Marco Teórico **15**

2.1. Antecedentes 15

2.2. Bases teóricas 18

2.2.1. *Radiación solar* 18
2.2.1.1. Principios básicos 18
2.2.1.2. Medición de radiación solar 19
2.2.1.3. Tipos de radiación solar 19
2.2.1.4. Importancia e inconvenientes de la radiación solar 20
2.2.2. *Series de tiempo* 21
2.2.2.1. Componentes de los datos de series de tiempo 22
2.2.3. *Modelos clásicos para series de tiempo* 23
2.2.3.1. ARIMA 23
2.2.4. *Neural Network Artificial* 26
2.2.5. *Deep Learning* 30
2.2.5.1. RNN 31
2.2.5.2. LSTM 33
2.2.5.3. GRU 36
2.3. Transformer 37
2.3.1. *Arquitectura del modelo* 38
2.3.2. *Atención* 39
2.3.3. *Atención a los productos de puntos escalados* 41
2.3.4. *Atención multicabezal* 41
2.3.5. *Codificación posicional* 43
2.3.6. *Atención enmascarada* 44
2.3.7. *Hiperparámetros del modelo Transformer* 45
2.4. Regularización para Deep Learning 45

3. Metodología **48**

3.1. Tipo y nivel de investigación 48
3.2. Población y muestra 48
3.3. Instrumentos 49
3.4. Procedimientos 49
3.4.1. *Conjunto de datos* 51
3.4.1.1. Descripción de las variables 52
3.4.2. *Preprocesamiento de datos* 53
3.4.2.1. Estandarización 54
3.4.2.2. Normalización Mín-Máx 54
3.4.2.3. División de datos 54
3.4.3. *Implementación del modelo Transformer* 55
3.4.4. *Métricas de evaluación del modelo* 60
3.5. Medio informático 61

4. Resultados y discusión **62**
4.1. Resultados de modelización Transformer 62
4.2. Resultados de evaluación del modelo 63
4.3. Resultados de comparación con otros modelos de predicción 64
4.4. Resultados de predicción diaria de RS 66
4.5. Discusión de resultados 66

Conclusiones **71**

Bibliografía **73**

Apéndices **81**

Índice de figuras

1.1. *Promedio mensuales de radiación solar en la provincia de Ilo 2021-2022* . 12

2.1. *Tipos de radiación solar según su forma al llegar a la tierra* 20

2.2. *Tipos de radiación solar según los tipos de luz* 21

2.3. *Ejemplo de visualización de datos de series de tiempo* 23

2.4. *Ejemplos de componentes de los datos de series de tiempo* 24

2.5. *Arquitectura básica de redes neuronales* 27

2.6. *Diagrama de Venn que describe el Deep Learning como subcampo del Machine Learning y que a su vez un subcampo de Artificial Intelligence* . . . 31

2.7. *Diagrama de RNN* 32

2.8. *Diagrama de LSTM* 34

2.9. *Diagrama de GRU* 36

2.10. *Estructura de Codificador-Descodificador de la aquitectura Transformer* . 40

2.11. *(izquierda) Atención a los productos de puntos escalados, (derecha) Atención multicabezal* 42

2.12. *Ejemplo matriz de enmascaramiento para el bloque decodificador* 44

2.13. *Error del conjunto de entrenamiento frente al conjunto de prueba* 47

3.1. *Ubicación del punto en la ciudad de Ilo para el estudio en la investigación* 50

3.2. *Modelo de proceso CRISP-DM* 51

3.3. *Evolución diaria de radiación solar en la provincia de Ilo* 53

3.4. *Arquitectura del transformer de sólo decodificador para predicción del radiación solar* 56

4.1. *Pérdidas en el entrenamiento y validación del modelo Transformer* 64

4.2. *Comparación de métricas RMSE y R^2 con otros modelos en predicción de RS* 67

4.3. *Comparación de los valores predichos y reales de Radiación Solar (con $w = 7$ días)* 70

Índice de tablas

2.1. *Funciones de activación mas comunes* 28

2.2. *Tipos de hiperparámetros de Transformers* 45

3.1. *Explicación de las variables de conjunto de datos* 52

3.2. *Descripción estadistico de radiación solar diaria* 52

3.3. *División de datos para encontrar modelo de predicción* 55

3.4. *Búsqueda de parámetros para el modelo transformer* 57

4.1. *Definición de hiperparámetros para el modelo Transformer en la predicción de RS* 63

4.2. *Desempeño del modelo transformer con distintos valores de w utilizada en la predicción* 65

4.3. *Parámetros óptimos para modelos de predicción* 66

Introducción

La radiación solar recibida en la superficie de la Tierra depende de las condiciones climáticas y de las características geográficas de la región (Guermoui et al., 2018). Por otra parte, la previsión de la radiación solar se basa en parámetros meteorológicos como la irradiancia solar, la temperatura atmosférica, la velocidad y presión del viento, la humedad y la nubosidad. Debido a la fluctuación de estas variables meteorológicas, la radiación solar varía con el tiempo. En consecuencia, la predicción precisa de la radiación solar es excepcionalmente difícil (Naveen et al., 2023).

En tareas de predicción de series de tiempo, han sido utilizados ámpliamente con diversos modelos de predicción: modelos tradicionales ARIMA, modelos de Machine Learning: Regresión Vectorial de Soporte (SVR), Redes Neuronales Artificiales (ANN) e incluso modelos híbridos con SVR-ARIMA, ANN-ARIMA y modelos de Deep Learning: RNN, LSTM y GRU. Ahora, con la aparición del modelo Transformer publicado por Google en 2017 (Vaswani et al., 2023), ha sido diseñado para utilizar mecanismos de atención, para manejar datos secuenciales con el fin de abordar el problema del aprendizaje de secuencias en el Procesamiento del Lenguaje Natural, como la traducción automática. Por consiguiente, el modelo Transformer se puede utilizarse en tareas de previsión de series de tiempo (Wu et al., 2020; Lezmi & Xu, 2023).

Por lo tanto, el objetivo de esta tesis es implementar el modelo Transformer para

predicción en series de tiempo, con datos de radiación solar en la provincia de Ilo, evaluar su precisión y comparar su desempeño con otros modelos de predicción como: ARIMA, RNN, LSTM, y GRU.

El desarrollo de esta tesis esta organizado de la siguiente manera: en el primer capítulo presenta "Planteamiento del problema" que muestra los motivos y razones para esta investigación, segundo capítulo "Marco teórico" que describe temas sobre radiación solar, series de tiempo, ANN y las arquitecturas de DL, tercer capítulo "Metodología" que muestra los procedimientos y la implementación del modelo para predicción de radiación solar, en cuarto capítulo "Resultados y discusión" en la que se muestra los resultados de los objetivos trazados en esta investigación y su discusión respectivas, y finalmente, la "Conclusión y Recomendación" dando los resúmenes en base a los resultados obtenidos de nuestro modelo propuesto.

Capítulo 1

Planteamiento del problema

1.1. Descripción del problema

En la actualidad, nuestro planeta por efectos de cambios climáticos y el deterioro de la capa de ozono, se vienen registrando niveles altos de radiación ultravioleta, y el Perú no es exento a sus consecuencias, pues junto a los países vecinos a la línea ecuatorial, son los que reciben más altos niveles de radiación solar. (CEPRIT, 2017)

Según un estudio del SENAMHI (2023), se vienen incrementando los niveles del Índice Ultravioleta Solar (IUV) en todo el país, alcanzándose niveles entre "muy alta" y "extremadamente alta".

Como así también, se comprueba, que en la provincia de Ilo, ubicada en la costa sur del Perú, recibe niveles altos de radiación solar, principalmente en estación de verano, y en estos últimos años, las cifras van en incremento tal como se puede apreciar en la Figura 1.1.

En Huang et al., (2021) señala que la radiación solar es la fuente principal de energía de la Tierra y desempeña un papel importante en el balance de radiación de la superficie,

los ciclos hidrológicos, la fotosíntesis de la vegetación, los fenómenos meteorológicos y climáticos extremos. Por tanto, predicción de la radiación solar es un tópico muy importante tanto para la industria solar y la investigación climática.

Al tratarse el tema de radiación solar, también es de mucha importancia para la población, tanto para los gobiernos locales como también para las empresas, que tienen la responsabilidad en tomar medidas de protección efectivas y salvaguardar el bienestar de su población, como así también, el aprovechamiento de la energía solar. Para lograrlo este propósito, deberá contar con la información por anticipado de la posible radiación solar.

Existen varias alternativas que pueden responder a este problema y dar la información adecuada para la toma de decisiones; para ello, se requiere contar con buenos modelos matemáticos, que puedan hacer predicciones con errores mínimos.

En el estado del arte, al tratarse de predicción en series de tiempo, mayormente recurren en usar los modelos tradicionales como ARIMA.

Por este motivo, se ha propuesto en utilizar unas de las arquitecturas avanzadas de deep learning para predecir la radiación solar diaria en la provincia de Ilo, de tal manera logre en obtener buenos resultados en problemas de series de tiempo.

1.2. Formulación del problema

De acuerdo a la problemática descrito anteriormente se plantea lo siguiente: ¿ Se puede predecir la radiación solar en la provincia de Ilo aplicando con modelo Transformer? ¿ Cuál es su desempeño del modelo Transformer frente a los modelos del estado del arte en la predicción diaria de radiación solar en la provincia de Ilo?

Figura 1.1

Promedio mensuales de radiación solar en la provincia de Ilo 2021-2022



*N*ota. Son valores procesados de data histórica NASA-POWER.

1.3. Objetivos

1.3.1. *Objetivo general*

Predecir diariamente la radiación solar en la provincia de Ilo aplicando un modelo Transformer.

1.3.2. *Objetivos específicos*

1. Diseñar una arquitectura Transformer y determinar el modelo óptimo para predicción diaria de la radiación solar en la provincia de Ilo.

2. Evaluar los resultados del modelo Transformer.

3. Comparar los modelos del estado del arte con el modelo propuesto en la predicción diaria de radiación solar.

1.4. Justificación

Brahma & Wadhvani (2020) afirman que el interés y la necesidad son cada vez mayores de energía verde, esto han hecho que la investigación sobre la predicción de la irradiación solar haya cobrado mayor importancia en los últimos tiempos.

Por otro lado, en Garnacho et al. (2020) señalan que "las quemaduras solares, fotosensibilidad, fotodermatosis, inmunodepresión, el fotoenvejecimiento y la fotocarcinogénesis son algunos de los efectos adversos más importantes"(p. 2), cuales podrían generar consecuencias graves en el ser humano.

En definitiva, el impacto de la radiación solar es un tema complejo y polifacético que tiene importantes implicaciones para diversos aspectos de nuestras vidas. Por lo tanto, el desarrollo de modelos de predicción precisos y sólidos puede ayudarnos a comprender mejor y mitigar los efectos de la radiación solar, y apoyar la toma de decisiones en una amplia gama de dominios.

Además, frente a los avances de ciencia y tecnología, los modelos de deep learning ofrecen una capacidad avanzada para modelar y predecir en problemas de series de tiempo, superando muchas de las limitaciones de los enfoques tradicionales.

1.5. Hipótesis

1.5.1. *Hipótesis general*

Con un modelo Transformer se puede predecir la radiación solar en la provincia de Ilo.

1.5.2. *Hipótesis especifica*

Hipótesis especifica N 1: Se puede diseñar una arquitectura Transformer para predecir la radiación solar en la provincia de Ilo.

Hipótesis especifica N 2: El modelo propuesto tiene menor tasa de error frente a otros modelos del estado del arte en predicción diaria de radiación solar en la provincia de Ilo.

1.6. Variables

1.6.1. *Variable independiente*

Data histórica de radiación solar.

1.6.2. *Variable dependiente*

Modelo Transformer.

Capítulo 2

Marco Teórico

2.1. Antecedentes

Existen varios trabajos de investigación relacionados al tema con métodos de Machine Learning y Deep Learning. A continuación citamos algunos trabajos recientes.

Allal et al. (2024), en su investigación " Machine Learning Algorithms for Solar Irradiance Prediction: A Recent Comparative Study", entrenaron y probaron dos algoritmos de Deep Learning como MLP y LSTM frente a nueve modelos de Machine Learning (*Decision Tree, Random Forest, XGBoost, LighGBM, CatBoost, KNN, SVM, Stacking*, y *Boosting*). Utilizando con datos meteorológicos de Izmir (Turquía) para predicción de irradiación solar, con una proporción de 70 % para entrenamiento, 20 % para validación, y 10 % para la prueba; los resultados determinan que el modelo MLP es el mejor, con un error en $RMSE$ de 0.78(kwh/m^2), seguido por *Random Forest* con $RMSE$ de 2.56(kwh/m^2).

Park et al. (2023), en su investigación "Predictive Model for Solar Insolation Using the Deep Learning Technique", analizaron comparativamente el rendimiento predictivo

con tres modelos: modelos de regresión lineal, ANN, y LSTM para la predicción de la insolación horaria, con conjunto de entrenamiento del 90 % y 10 % para la validación. Al comparar estos modelos, LSTM muestra la superioridad, con un rendimiento predictivo fiable con CV($RMSE$) de 15.8 % y R^2 de 0.9745.

Paxi (2023), en su tesis "Predicción de series de tiempo de radiación solar en el Perú aplicando deep learning, caso: Moquegua y Tacna", implementó un modelo de LSTM con el aumento de datos (DA+LSTM) para predicción de radiación solar, al utilizar con los datos históricos de radiación solar en la ciudad de Moquegua, obtiene mejores resultados, con un menor valor en $RMSE$ de 0.1947(kwh/m^2), de este modo, demuestra que al hacer el uso de esta técnica reduce significativamente el error de predicción frente al modelo LSTM (sin aumento de datos).

Yusuf et al. (2022), en su investigación "Solar Radiation Forecast by Using Machine Learning Method for Gaziantep Province", utilizaron el método de regresión lineal para la estimación de radiación solar en la provincia de Gaziantep (Turquía) situada en la confluencia de la región mideterránea y la región sureste de Anatolia. Sus resultados alcanza el coeficiente de determinación (R^2) el 98 % y el valor de raíz de error cuadrático medio ($RMSE$) con 0.028(kwh/m^2).

Huang et al. (2021), en su investigación "Solar Radiation Prediction Using Different Machine Learning Algorithms and Implications for Extreme Climate Events" construyeron doce modelos de Machine Learning (*Multiple linear regression, RBF neural network, K-nearest neighbor, Decision tree, Extreme learning machine, Support vector machine regression, Gaussian process regression, AdaBoost, Gradient boosting regression tree, XGBoost*, y *Random forest*) y modelo *Stacking* para predecir y comparar los valores diarios y mensuales de radiación solar, en las que destacan a *Stacking* y *XGBoost* como mejores

modelos en la predicción de radiación solar, los resultados alcanzados de los modelos *XGBoost* y *Stacking* en la predicción diaria de radiación solar en términos de $RMSE$ son 2.043 MJ/m^2 y 1.940 MJ/m^2 respectivamente.

Mohammadi & Moazenzadeh (2021), en su investigación "Performance Analysis of Daily Global Solar Radiation Models in Peru by Regression Analysis", evaluaron y compararon el rendimiento de los modelos empíricos frente a su propuesta modelos de análisis de regresión lineal múltiple (RLM) en la estimación de la radiación solar diaria de 13 estaciones meteorológicas peruanas (Tumbes, Cusco, Arequipa, Lima, Loreto, San Ramón, Puno, Tacna, San Martín, Lambayeque, Junín, Cajamarca, y Huánuco), determinan a RLM como mejor modelo. Según el $RMSE$ de los modelos propuestos, los peores y mejores de desempeño en la predicción, se ha logrado en San Martín ($RMSE$ = 508,8 J/cm^2/día) y Tacna ($RMSE$ = 223,2 J/cm^2/día) respectivamente.

Brahma & Wadhvani (2020), en su investigación "Solar Irradiance Forecasting Based on Deep Learning Methodologies and Multi-Site Data", desarrollaron modelos de predicción basados en metodologías de Deep Learning para predecir la irradiación solar diaria en dos lugares de India, destacando que los modelos de LSTM bidireccional y LSTM basados en la atención, obtuvieron buenos resultados en la predicción de irradiación solar diaria.

Guermoui et al. (2018), en su investigación "Support vector regression methodology for estimating global solar radiation in Algeria", utilizaron los modelos de Máquina de Soporte Vectorial para estimar la radiación solar global diaria, logrando alcanzar el resultado de $RMSE$ = 1,59 (MJ/m^2) y R^2 = 97,4 %. Según estos resultados obtenidos, destacan que la regresión por vectores soporte tiene la alta capacidad predictiva.

2.2. Bases teóricas

En esta sección, introduciremos bases teóricas sobre investigación.

2.2.1. *Radiación solar*

La radiación solar (RS) es la energía emitida por el sol, la cual se propaga en todas direcciones a través del espacio mediante ondas electromagnéticas. Emitida por la superficie del sol, esta energía influye en los procesos atmosféricos y climáticos. También es responsable, directa e indirectamente de fenómenos comunes, como la fotosíntesis de las plantas, el mantenimiento del planeta a una temperatura compatible con la vida y la formación del viento, esencial para generar energía eólica. (Iberdrola, s.f)

2.2.1.1. Principios básicos

Según Energy.gov (s.f) todos los lugares de la tierra reciben luz solar al menos una parte del año. La cantidad de radiación solar que llega a cualquier punto de la superficie terrestre varía en función de:

- La situación geográfica.
- La hora del día.
- Estación del año.
- Paisaje local.
- El clima local.

2.2.1.2. Medición de radiación solar

Las mediciones de la energía solar suelen expresarse como radiación total sobre una superficie horizontal, o como radiación total sobre una superficie que sigue al sol. Los datos de radiación para los sistemas eléctricos solares (fotovoltaicos[1]) suelen representarse como kilovatios-hora por metro cuadrado (kWh/m2). Las estimaciones directas de la energía solar también pueden expresarse en vatios por metro cuadrado (W/m2). (Energy.gov, s.f)

2.2.1.3. Tipos de radiación solar

Según Stein (2024) hay dos tipos de radiación solar:

Tipos de radiación solar según su forma al llegar a la tierra. Son tres tipos de RS, según su forma, que penetra en la superficie de ta tierra:

- **Radiación solar directa**. Esta radiación solar penetra en la atmósfera y llega a la superficie sin dispersarse del todo.
- **Radiación solar difusa**. Es la radiación que llega a la superficie de la Tierra tras dispersarse en la atmósfera.
- **Radiación solar reflejada**. El efecto albelo[2] es un fenómeno por el que la superficie de la Tierra refleja la radiación solar.

Tipos de radiación solar según los tipos de luz. Son tres tipos de luz que llegan a la superficie terrestre.

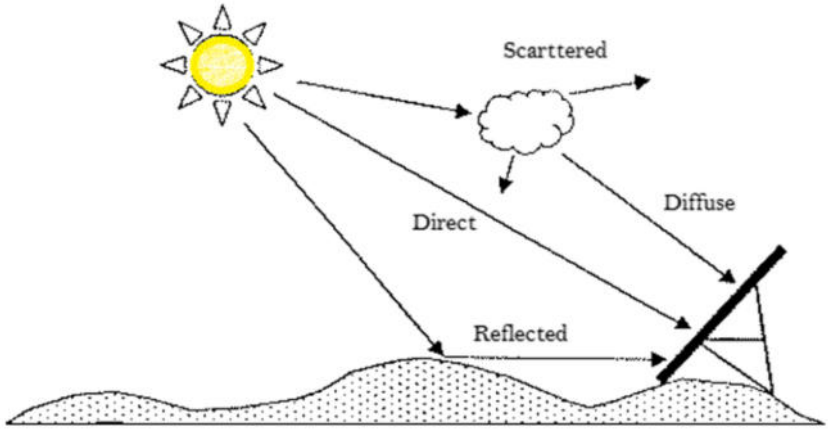
- **Rayos infrarrojos**. Este tipo de luz tiene una longitud de onda más larga que la luz visible, por lo que refleja más que los rayos UV y la luz visible.

[1]Los materiales y dispositivos fotovoltaicos convierten la luz solar en energía eléctrica

[2]Se trata de la cantidad de radiación electromagnética que se refleja en comparación con la que se absorbe.

Figura 2.1

Tipos de radiación solar según su forma al llegar a la tierra



*N*ota. Tomada de El Mghouchi (2022).

- **Rayos visibles**. La radiación solar visible o luz visible incluye los colores claros visibles para el ojo humano.
- **Rayos ultravioleta**. La radiación solar ultravioleta o luz ultravioleta tiene longitudes de onda más corta que la luz visible y puede ser perjudicial para el ser humano y otros seres vivos si se expone a ella durante mucho tiempo.

2.2.1.4. Importancia e inconvenientes de la radiación solar

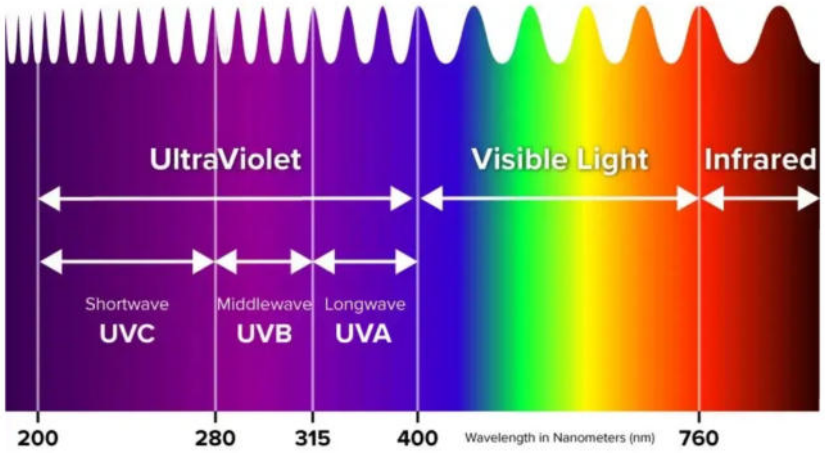
De acuerdo Stein (2024) la importancia e inconvenientes de la radiación solar son:

Importancia de la radiación solar. La radiación solar es muy importante por las siguientes razones:

- Proporciona luz.
- Equilibra la temperatura de la Tierra, proporciona calor.
- Emite energía que puede almacenarse y convertirse en electricidad.

Figura 2.2

Tipos de radiación solar según los tipos de luz



*N*ota. Tomada de Hepacart (2017).

- Es necesaria para la fotosíntesis.

Inconvenientes de la radiación solar. Son varios inconvenientes que provoca la radiación solar:

- Puede ser perjudicial para la piel humana, especialmente cuando se expone durante demasiado tiempo.
- Altera la respuesta inmunitaria.
- Provoca problemas oculares.
- Es perjudicial para la vida microscópica.

2.2.2. *Series de tiempo*

Una serie de tiempo se define como una secuencia de valores, ordenados cronológicamente, y observados a lo largo del tiempo. Aunque el tiempo es una variable que se mide de forma continua, los valores de una series de tiempo se muestrean a intervalos

constantes (frecuencia de muestreo fija) (Torres et al., 2021).

Los conjuntos de datos de series de tiempo registran observaciones de la misma variable en distintos momentos del tiempo, como un ejemplo se puede apreciar en la Figura 2.3.

2.2.2.1. Componentes de los datos de series de tiempo

Los datos de series de tiempo están formados por distintos componentes que caracterizan los patrones y el comportamiento de los datos a lo largo del tiempo.

De acuerdo a Data Science Wizards (2023), los principales componentes de los datos de series de tiempo son los siguientes:

Tendencias: Se trata de una representación de un movimiento ascendente o descendente a lo largo de un periodo prolongado.

Estacionalidad: Cuando en una serie de tiempo se encuentra patrones que son predecibles y se producen en un intervalo de tiempo concreto, se denomina estacionalidad.

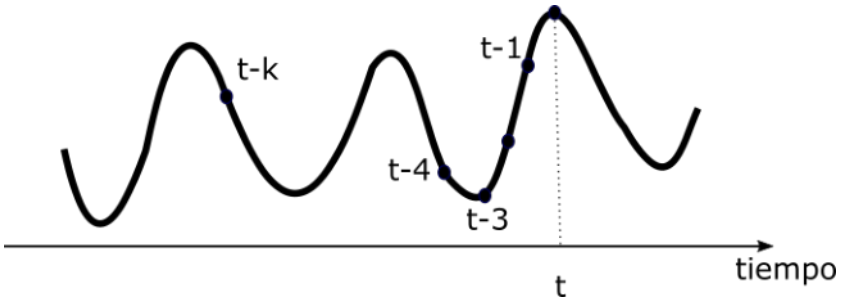
Cíclicos: Los componentes cíclicos captan fluctuaciones u oscilaciones que no están fijadas a un marco temporal concreto.

Irregular: El componente irregular o aleatorio, a menudo denominado ruido, representa las fluctuaciones imprevisibles de las series de tiempo. Este componente hace que los datos de las series de tiempo sean únicos y difíciles de modelar con precision

Como ejemplo, se visualizan en la Figura 2.4 los cuatro componentes de los datos de series de tiempo.

Figura 2.3

Ejemplo de visualización de datos de series de tiempo



2.2.3. *Modelos clásicos para series de tiempo*

Uno de los modelos mas importante y utilizados para series de tiempo es el modelo de ARIMA. Además, existen varios enfoques para la modelización de series de tiempo. Los modelos estadisticos tradicionales, como la media movil, el suavizado exponencial y el ARIMA, son lineales en el sentido de que las predicciones de los valores futuros se limitan a ser funciones lineales de las observaciones pasadas (Zhang, 2003).

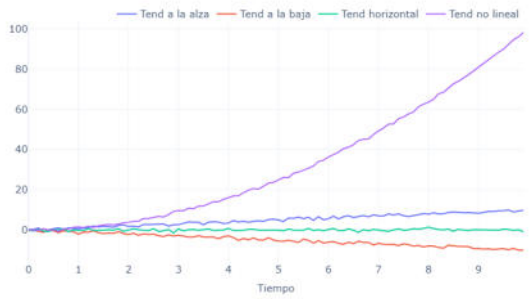
2.2.3.1. ARIMA

Box-Jenkins ARIMA (media móvil integrada autorregresiva) es otro enfoque popular para modelar sistemas dinámicos. ARIMA modela la variable observada x_t y asume que x_t puede descomponerse en componentes de tendencia, estacionales e irregulares. En lugar de modelizar estos componentes por separado, Box y Jenkins tuvieron la idea de diferenciar la series de tiempo x_t para eliminar la tendencia y la estacionalidad. La serie resultante se trata como datos de series de tiempo estacionarias y se modela utilizando la combinación de sus valores de series de tiempo retardadas (AR) y la media móvil de error de prevención retardados (MA). Un modelo ARIMA suele especificarse mediante una tupla (p, d, q), en la que p y q definen los órdenes de AR y MA, y d especifica el orden de la operación de diferenciación. (Wu et al., 2020)

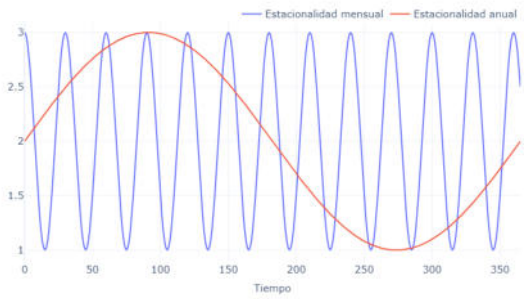
Figura 2.4

Ejemplos de componentes de los datos de series de tiempo

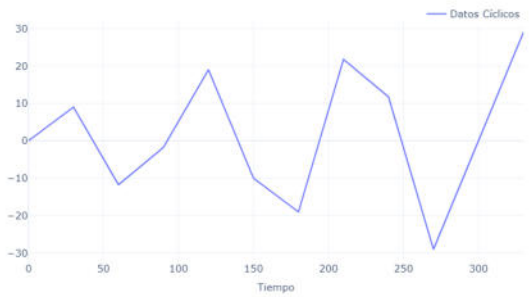
a) Tendencias



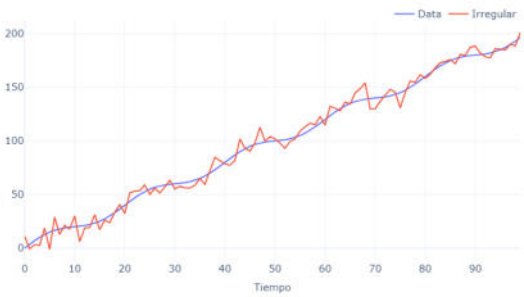
b) Estacional



c) Cíclico



d) Irregular



Según Siami et al. (2018), explica que las siglas ARIMA(p, d, q) recoge los elementos clave del modelo:

- AR: Autoregresión. Modelo de regresión que utiliza las dependencias entre una observación y una serie de observaciones retardadas (p).
- I: Integrado. Hacer estacionaria la serie de tiempo midiendo las diferencias de las observaciones en distintos momentos (d).
- MA: Media móvil. Enfoque que tiene en cuenta la dependencia entre las observaciones y los términos de error residuales cuando se utiliza un modelo de media móvil para las observaciones retardadas (q).

Una forma simple de un modelo AR de orden p, es decir, AR(p), se puede escribir como un proceso lineal dado por:

$$x_t = c + \sum_{i=1}^{p} \phi_i x_{t-i} + \epsilon_t \tag{2.1}$$

Donde x_t es la variable estacionaria, c es constante, ϕ_i son coeficientes de autocorrelación en los retardos $1, 2, \cdots, p$ y ϵ_t los residuos, son las series de ruido blanco Gaussiano con media cero y varianza σ^2. Un modelo MA de orden q, es decir, MA(q), puede escribirse de la forma:

$$x_t = u - \sum_{i=0}^{q} \theta_i \epsilon_{t-i} \tag{2.2}$$

Donde u es la expectativa de x_t (normalmente se asume igual a cero), los términos θ_i son los pesos aplicados a los valores actuales y anteriores de un término estocástico en la series de tiempo, y $\theta_0 = 1$. Suponemos que ϵ_t es una serie de ruido blanco Gaussiano con media cero y varianza σ^2.

Al combinar estos dos modelos se forma un modelo ARMA de orden (p, q):

$$x_t = c + \sum_{i=1}^{p} \phi_i x_{t-i} + \epsilon_t - \sum_{i=0}^{q} \theta_i \epsilon_{t-i} \tag{2.3}$$

Donde $\phi_i \neq 0$, $\theta_i \neq 0$, y $\sigma_\epsilon^2 > 0$. Los parámetros p y q se denominan órdenes AR y MA respectivamente.

Por lo tanto, en Rubio & Alba (2022) un modelo ARIMA(p, d, q) se expresa a menudo como:

$$\phi(\beta)(1 - \beta)^d x_t = c + \theta(\beta)\epsilon_t \tag{2.4}$$

donde

$$\phi(\beta) = 1 - \sum_{i=1}^{p} \phi_i \beta^i \text{ y } \theta(\beta) = 1 - \sum_{i=1}^{q} \theta_i \beta^i \tag{2.5}$$

Los operadores de desplazamiento hacia atrás para AR(p) y MA(q) se define como

$\phi(\beta)y_t = c + \epsilon_t$ y $y_t = u + \theta(\beta)\epsilon_t$, con $c = u - \phi u$, donde u y ϵ son media y el ruido blanco respectivamente, y $E(\epsilon_t) = 0$.

La forma general de un modelo ARIMA estacional se denota como ARIMA$(p, d, q) \times (P, D, Q)S$, donde p es el orden AR no estacional, d es la diferenciación no estacional, q es el orden MA no estacional, P es el orden AR estacional, D es la diferenciación estacional, Q es el orden MA estacional y S es el periodo de tiempo del patrón estacional repetitivo, respectivamente (Siami et al., 2018).

2.2.4. *Neural Network Artificial*

Las redes neuronales son redes de neuronas artificiales interconectadas. Su estructura se inspira en gran medida en la red neuronal del cerebro. Una red neuronal se utiliza generalmente para crear modelos de machine learning supervisado para la clasificación, similar a un modelo de regresión logística y es útil en los casos en que la regresión logística puede no proporcionar una precisión razonable. Las redes neuronales son la base de muchas aplicaciones complejas y algoritmos de machine learning. (Rebala et al., 2019)

La arquitectura básica de redes neuronales artificiales esta compuesta por tres capas: capa de entrada, oculta y la capa de salida, tal como se ve en la Figura 2.5

Para entender su funcionamiento básico del modelo de red neuronal, en Lindholm et al. (2022) detalla lo siguiente:

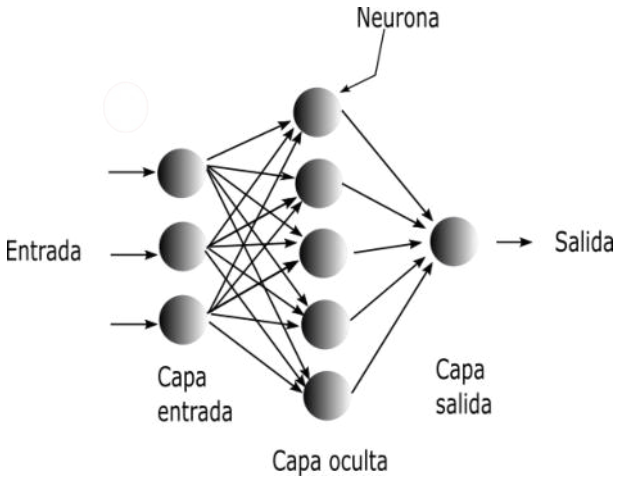
Se introduce el concepto de funciones paramétricas no lineales para modelar la relación entre las variables de entrada x y de salida y. Denotaremos esta relación no lineal en su forma de predicción como:

$$\hat{y} = f_\theta(x_1, \cdots, x_m) \tag{2.6}$$

donde la función f es parametrizada por θ. Esta función no lineal puede parametrizarse

Figura 2.5

Arquitectura básica de redes neuronales



de muchas formas. En una red neuronal, la estrategia consiste en varias capas de modelos de regresión lineal y funciones de activación no lineales.

La relación no lineal entre $x = [1\ x_1\ x_2 \cdots x_m]^T$ y $\hat{y}$, se introduce una función escalar no lineal denominada función de activación $h : \mathbb{R} \to \mathbb{R}$.

$$\hat{y} = h(W_1x_1 + W_2x_2 + \cdots + W_mx_m + b) \tag{2.7}$$

Las funciones de activaciones mas comunes para ANN, según el cuadro2.1 son: función logística, tangente hiperbólico, y unidad lineal rectificada.

Vectorizacion sobre unidades. El modelo de red neuronal puede escribir de forma mas compacta utilizando la notacion matricial, donde los parametros de cada capa se apilan en una matriz de pesos W y un vector de desplazamiento b como:

$$W^{(1)} = \begin{bmatrix} w_{11}^{(1)} & \cdots & w_{1m}^{(1)} \\ w_{21}^{(1)} & \cdots & w_{2m}^{(1)} \\ \vdots & \ddots & \vdots \\ w_{U1}^{(1)} & \cdots & w_{Um}^{(1)} \end{bmatrix}, b^{(1)} = \begin{bmatrix} b_1^{(1)} \\ b_2^{(1)} \\ \vdots \\ b_U^{(1)} \end{bmatrix}$$

$$W^{(2)} = \begin{bmatrix} w_1^{(2)} \cdots w_U^{(2)} \end{bmatrix}, b^{(2)} = \begin{bmatrix} b^{(2)} \end{bmatrix}$$

Tabla 2.1

Funciones de activación mas comunes

Función	**Ecuación**	**Rango de salida**
sigmoidea	$h(x) = \sigma(x) = \dfrac{1}{1+e^{-x}}$	$\in (0,1)$
tanh	$h(x) = tanh(x) = \dfrac{e^{x}-e^{-x}}{e^{x}+e^{-x}}$	$\in (-1,1)$
ReLU	$h(x) = f(x) = \begin{cases} 0, & \text{si } x < 0 \\ x, & \text{si } x \geq 0 \end{cases}$	$\in [0,\infty)$

Nota. Las funciones de activaciones según Freeman et al. (2018)

El modelo completo se puede escribirse como:

$$q = h(W^{(1)}x + b^{(1)}) \tag{2.8}$$

$$\hat{y} = W^{(2)}q + b^{(2)} \tag{2.9}$$

donde $x = [x_1 \cdots x_m]^T$ y $q = [q_1 \cdots q_U]^T$. Se observa que la función de activación h en la Ecuación 2.8 actúa elemento a elemento sobre el vector de entrada y da como resultado un vector de salida de la misma dimensión. Las dos matrices de pesos y los dos desplazamientos son los parámetros del modelo, que puede escribirse como

$$\theta = \begin{bmatrix} vec(W^{(1)})^T & b^{(1)T} & vec(W^{(2)})^T & b^{(2)T} \end{bmatrix}^T \tag{2.10}$$

donde el operador *vec* toma todos los elementos de la matriz y los coloca en un vector. En general, la Ecuación 2.9 describe un modelo de regresion no lineal de forma $\hat{y} = f_\theta(x)$.

Para encontrar valores adecuados para los parametros θ, se resuelve un problema de

optimizacion de la forma

$$\hat{\theta} = \arg\,\min_\theta J(\theta) \text{ donde } J(\theta) = \frac{1}{n}\sum_{i=1}^{n} L(x_i, y_i, \theta) \tag{2.11}$$

La función de coste se denomina $J(\theta)$ y la función de pérdida como $L(x_i, y_i, \theta)$. La forma funcional de la función de pérdida depende del problema en cuestión, principalmente si se trata de un problema de regresión o de clasificación.

Para problema de regresión, típicamente se usa la pérdida de error cuadrático

$$L(x, y, \theta) = \Big(y - f(x;\theta)\Big)^2 \tag{2.12}$$

donde $f(x;\theta)$ es la salida de red neuronal.

En todos los algoritmos de optimización numérica, los parámetros se actualizan de forma iterativa. En el deep learning, se utiliza varias versiones de búsqueda basada en el gradiente:

1. Elige una inicialización θ_0

2. Actualiza los parámetros como:

$$\theta_{t+1} \longleftarrow \theta_t - \gamma \nabla_\theta J(\theta_t) \text{ para } t = 0, 1, 2, \cdots \tag{2.13}$$

3. Termina cuando se cumple algún criterio, y toma la última θ_t como $\hat{\theta}$

Backpropagation. El algoritmo de retropropagación es un ingrediente importante en casi todos los procedimiento de entrenamiento de redes neuronales.

Los parámetros de este modelo son todas las matrices de pesos y todos los vectores de desplazamiento. Por lo tanto, en cada iteración de nuestro algoritmo de búsqueda basado en el gradiente tal como señala en la Ecuación 2.13, también tenemos que encontrar el gradiente de la función de coste con respecto a todos los elementos de estas matrices y

vectores. En resumen, queremos encontrar

$$dW^{(l)} = \nabla_{\partial W^{(l)}} J(\theta) = \begin{bmatrix} \frac{\partial J(\theta)}{\partial W_{1l}^{(l)}} & \cdots & \frac{\partial J(\theta)}{\partial W_{1,U^{(l-1)}}^{(l)}} \\ \vdots & \ddots & \vdots \\ \frac{\partial J(\theta)}{\partial W_{U^{(l)},1}^{(l)}} & \cdots & \frac{\partial J(\theta)}{\partial W_{U^{(l)},U^{l-1}}^{(l)}} \end{bmatrix} \text{ y} \tag{2.14}$$

$$db^{(l)} = \nabla_{\partial b^{(l)}} J(\theta) = \begin{bmatrix} \frac{\partial J(\theta)}{\partial b_1^{(l)}} \\ \vdots \\ \frac{\partial J(\theta)}{\partial b_{U^{(l)}}^{(l)}} \end{bmatrix} \tag{2.15}$$

cuando se calculan estos gradientes, se actualizan las matrices de pesos (Ecuación 2.16) y los vectores de desplazamiento (Ecuación 2.17).

$$W_{t+1}^{(l)} \longleftarrow W_t^{(l)} - \gamma dW_t^{(t)} \tag{2.16}$$

$$b_{t+1}^{(l)} \longleftarrow b_t^{(l)} - \gamma db_t^{(l)} \tag{2.17}$$

Para calcular todos estos gradientes de forma eficaz, la retropropagación aprovecha la estructura del modelo en lugar de calcular ingenuamente las derivadas con respecto a cada parámetro por separado. para ello, la retropropagación utiliza la regla de la cadena del cálculo.

2.2.5. *Deep Learning*

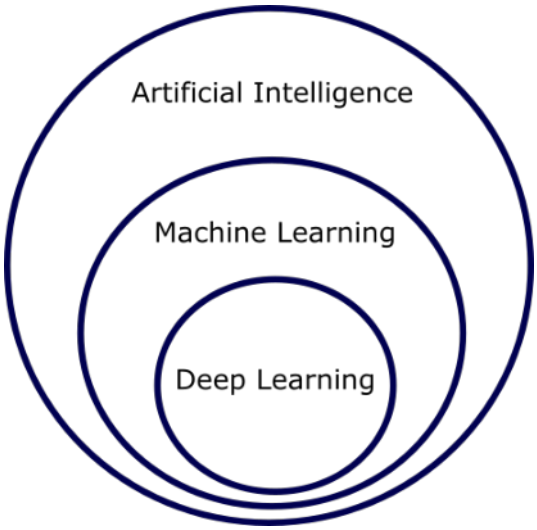
El Deep Learning es un subcampo del Machine Learning, que a su vez, un subcampo de la inteligencia artificial AI (Rosebrock, 2017), tal como se ilustra en la Figura 2.6.

El deep learning pertenece a la familia de los algoritmos de redes neuronales artificiales.

El Deep Learning moderno proporciona un marco muy potente para aprendizaje supervisado. Añadiendo mas capas y mas unidades dentro de una capa, una red profunda puede representar funciones de complejidad creciente (Goodfellow et al., 2016).

Figura 2.6

Diagrama de Venn que describe el Deep Learning como subcampo del Machine Learning y que a su vez un subcampo de Artificial Intelligence



A continuación se describen los modelos clásicos de Deep Learning: *Recurrent Neural Networks* (RNN), *Long Short-term Memory* (LSTM) y *Gated Recurrent Units* (GRU); dado que estas arquitecturas son diseñadas para datos secuenciales.

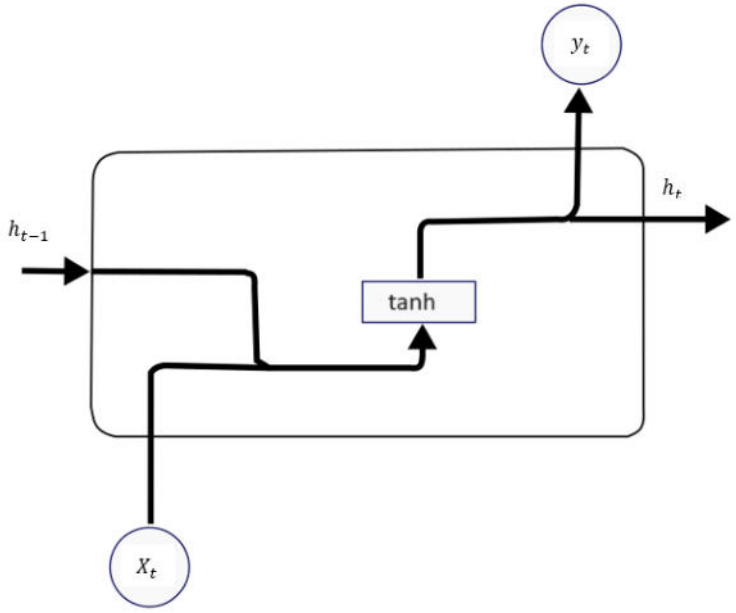
2.2.5.1. RNN

Las RNN están diseñadas específicamente para tratar datos secuenciales, como secuencias de palabras en problemas relacionados con la traducción automática, datos de audio en el reconocimiento de voz o series temporales en problemas de previsión. Todos estos problemas presentan una característica común, y es que los datos tienen una dependencia temporal entre si. Las redes neuronales feed-forward tradicionales no pueden tener en cuenta estas dependencias, y las RNNs surgen precisamente para abordar este problema. (Torres et al., 2021)

En la Figura 2.7 muestra el flujo de trabajo de única cédula RNN.

Figura 2.7

Diagrama de RNN



De acuerdo a Shi et al. (2022) tiene capa de cálculo simple con activación tanh, h_t es la función del estado anterior de la cédula h_{t-1} y X_t como la entrada actual.

Matemáticamente, se da de la siguiente forma:

$$h_t = \tanh\left(W_{xh}.\left(X_t \oplus h_{t-1}\right) + b_h\right) \tag{2.18}$$

$$y_t = \sigma\left(W_{hy}^T h_t + b_y\right) \tag{2.19}$$

Donde $X_t \in \mathbb{R}^m$ es el vector de entrada de m características en el tiempo t, $W_{xh} \in \mathbb{R}^{n\times(m+n)}$ y $W_{hy} \in \mathbb{R}^{n\times n}$ son matrices de parámetros; n es el número de neuronas en la capa RNN, $b_h \in \mathbb{R}^n$ y $b_y \in \mathbb{R}^n$ son los vectores de sesgo para el estado interno y la salida, respectivamente; σ es la función de activación (sigmoidea); h_t es el estado interno (oculto) y $X_t \oplus h_{t-1}$ es la concatenación de los vectores X_t y h_{t-1}.

Cabe mencionar, que las RNN se enfrentan a una serie de retos y obstáculos, según en Indrajitbarat (2023) señala que uno de los mayores retos de las RNN es el problema de

la desaparición o explosión de gradientes. Esto ocurre cuando los gradientes de la función de pérdida con respecto a los parámetros de la RNN se vuelven muy pequeños o muy grandes a medida que se propagan en el tiempo. Esto puede dificultar en el entrenamiento eficaz de la RNN, ya que las actualizaciones de los parámetros serán muy pequeñas o muy grandes en que la red no aprenderá eficazmente. Otro problema de las RNN es que pueden ser difíciles de entrenar. Esto se debe a que las RNN suelen entrenarse mediante retropropagación, lo que puede resultar caro desde el punto de vista computacional y llevar mucho tiempo. Además, las RNN pueden ser sensibles a la elección de los hiperparámetros, como la tasa de aprendizaje y el número de capas.

Para resolver este desafío y obstáculos de las RNN se han desarrollado varias técnicas, una de ellas consiste en utilizar LSTM y GRU, dado que estas células están diseñadas para resolver el problema de los gradientes que desaparecen o exploten.

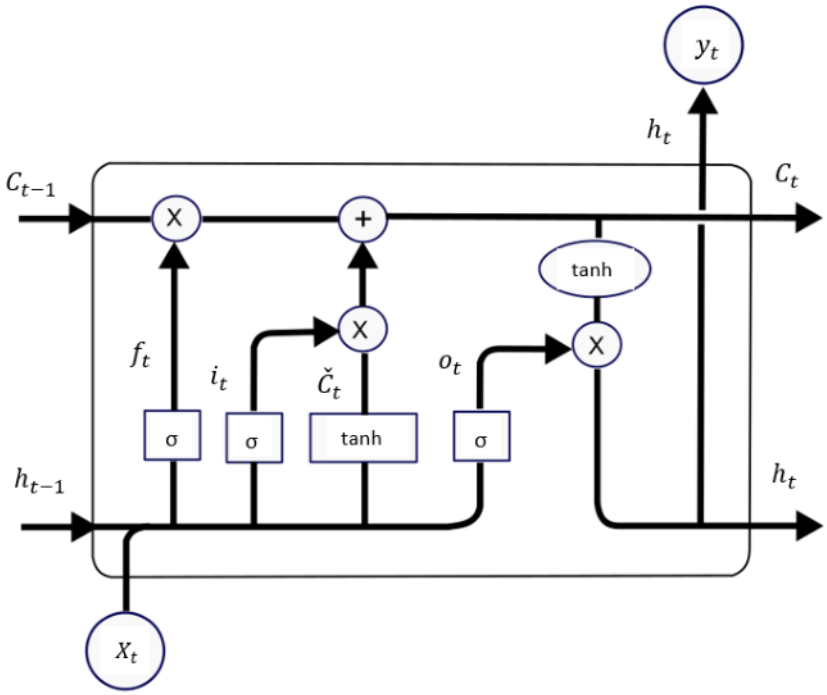
2.2.5.2. LSTM

Las RNN básicas estándar sufren el problema del gradiente evanescente, que consiste en que el gradiente disminuye a medida que aumenta el numero de capas. De hecho, para las RNNs profundas con un numero elevado de capas, el gradiente se vuelve prácticamente nulo, lo que impide el aprendizaje de la red. Por este motivo, estas redes tienen una memoria a corto plazo y no obtienen buenos resultados cuando se trata de secuencias largas que requieren memorizar toda la información contenida en la secuencia completa. Las redes recurrentes con memoria a corto plazo (LSTM) surgen para resolver el problema del gradiente evanescente. (Torres et al., 2021)

La estructura del LSTM se muestra en la Figura 2.8. A diferencia de la RNN estándar, que solo tiene una función de activación en el bucle, manteniendo intactos los datos

Figura 2.8

Diagrama de LSTM



de entrada iniciales, la LSTM contiene múltiples funciones y operaciones de activación dentro de cuatro capas interactivas. En una estructura LSTM, la sección por la que pasa selectivamente la información se denomina (Puerta) , y el camino por el que fluye la información en linea recta a lo largo de toda la cédula LSTM se denomina (Estado de cédula). Cada puerta se compone de funciones como sigmoide y tanh y cálculos (Park et al., 2023).

A continuación, según en la Figura 2.8 se muestra las ecuaciones de las tres puertas diferentes en una cédula LSTM: una puerta de olvido (*forget gate*), una puerta de entrada (*input gate*) y una puerta de salida (*output gate*).

Forget Gate

$$f_t = \sigma\Big(W_t \cdot \Big(X_t \oplus h_{t-1}\Big) + b_t\Big) \tag{2.20}$$

donde t es tiempo de paso, f_t olvida la puerta en t, $X_t \in \mathbb{R}^m$ como entrada, h_t estado oculto anterior, $W_t \in \mathbb{R}^{n\times(m+n)}$ matriz de pesos entre la puerta de olvido y la puerta de entrada y $b_t \in \mathbb{R}^n$ como sesgo de conexión en t.

Input Gate

$$i_t = \sigma\Big(W_i \cdot \big(X_t \oplus h_{t-1}\big) + b_i\Big) \tag{2.21}$$

$$\tilde{C}_t = \tanh\Big(W_c \cdot \big(X_t \oplus h_{t-1}\big) + b_c\Big) \tag{2.22}$$

donde i_t puerta de entrada en t, $W_i \in \mathbb{R}^{n\times(m+n)}$ matriz de pesos de operador sigmoide entre la puerta de entrada y la puerta de salida, $b_i \in \mathbb{R}^n$ vector de sesgo en i con respecto a W_i, $\tilde{C}_t$ valor generado por tanh, $W_c \in \mathbb{R}^{n\times(m+n)}$ matriz de pesos de operador tanh entre la información del estado de cédula y salida de la red y $b_c \in \mathbb{R}^n$ vector de sesgo en c con respecto a W_c.

Cell State

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t \tag{2.23}$$

donde C_t información del estado de la cédula, f_t olvida la puerta en t, i_t puerta de entrada en t, C_{t-1} pasos anteriores y $\tilde{C}_t$ valor generado por tanh.

Output Gate

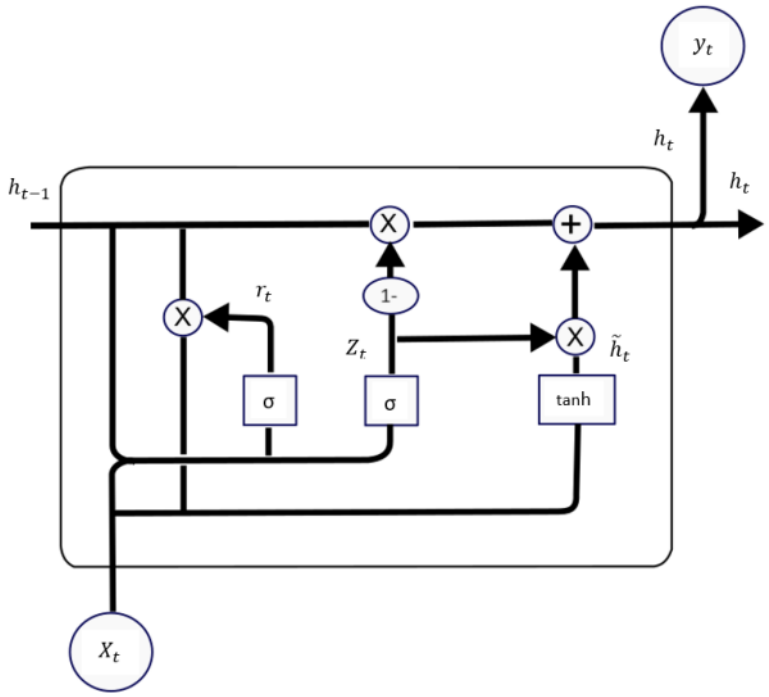
$$o_t = \sigma\Big(W_o \cdot \big(X_t \oplus h_{t-1}\big) + b_o\Big) \tag{2.24}$$

$$h_t = o_t \cdot \tanh(C_t) \tag{2.25}$$

donde o_t puerta de salida en t, $W_o \in \mathbb{R}^{n\times(m+n)}$ matriz de pesos de puerta salida, $b_o \in \mathbb{R}^n$ vector de sesgo con respecto a W_o y h_t la salida de LSTM.

Figura 2.9

Diagrama de GRU



2.2.5.3. GRU

La GRU puede considerarse una versión simplificada de LSTM en la que los estados de las cédulas no se utilizan explícitamente. La principal simplificación es que los vectores de estado se fusionan en un único vector (Brahma & Wadhvani, 2020).

GRU y LSTM están diseñados para resolver el problema del gradiente evanescente en la RNN, GRU tiene menos parámetros que LSTM y, por tanto, es mas rápido de entrenar y es menos probable que sobreajuste los datos. Además, GRU tiene la arquitectura mas sencilla que LSTM, con solo dos puertas (*reset* y *update*) en comparación con las tres puertas de LSTM. (Bousnguar et al., 2023)

En la siguiente Figura 2.9, se puede apreciar la arquitectura de GRU; en el diagrama se ve que toma la entrada actual X_t y el estado oculto anterior h_{t-1}, y tiene menos

operaciones en comparación con el LSTM; tiene los siguientes componentes: r_t la puerta de reinicio (*reset gate*), Z_t la puerta de actualización (*update gate*), h_t el estado oculto (*hidden state*) y $\tilde{h}_t$ el estado oculto del candidato (*candidate hidden state*). Las ecuaciones utilizadas para calcular de cada componente de una GRU son las siguientes:

Gating

$$r_t = \sigma\Big(W_r \cdot \Big(X_t \oplus h_{t-1}\Big) + b_r\Big) \tag{2.26}$$

$$Z_t = \sigma\Big(W_z \cdot \Big(X_t \oplus h_{t-1}\Big) + b_z\Big) \tag{2.27}$$

Candidate Hidden State

$$\tilde{h}_t = \tanh\Big(W_h \cdot \Big(X_t \oplus h_{t-1} \cdot r_t\Big) + b_h\Big) \tag{2.28}$$

Hidden State

$$h_t = (1 - Z_t) \cdot h_{t-1} + Z_t \cdot \tilde{h}_t \tag{2.29}$$

donde $X_t \in \mathbb{R}^m$ es el vector de entrada de m características en el momento t, $W_r, W_z, W_h \in \mathbb{R}^{n\times(m+n)}$ son matrices de parámetros, n el número de neuronas en la capa de GRU, $b_r, b_z, b_h \in \mathbb{R}^n$ son vectores de sesgos, σ es la función de activación sigmoide.

2.3. Transformer

Los Transformer son una clase de modelos de deep learning que se definen por algunos rasgos arquitectónicos. Fueron introducidos por primera vez en el ahora famoso articulo **"Attention is All you Need"** por investigadores de Google en 2017 (Vaswani et al., 2023). El articulo ha acumulado una cantidad enorme de 38000 citas en sólo 5 años (Amatriain et al., 2023).

2.3.1. *Arquitectura del modelo*

La arquitectura de un Transformer consta de dos bloques, un Codificador y un Decodificador que funciona conjuntamente.

De acuerdo a Lezmi & Xu (2023) cada bloque de la arquitectura Transformer se compone de la siguiente manera:

Codificador: En un modelo Transformer, la parte codificadora consiste en una pila de N bloques codificadores idénticos, todos ellos conectados uno detrás de otro (conexión en serie). Por lo tanto, N controla la profundidad de la red.

Cada bloque del Codificador como se muestra en el lado izquierdo de la Figura 2.10, se compone de dos subcapas:

Capa de mecanismo autoatención multicabezal

Se ha visto que el mecanismo multicabezal implementa h cabezales que reciben una versión (diferente) proyectada linealmente de las consultas (Q), claves (K) y valores (V), cada uno para producir h salidas en paralelo que luego se utilizan para generar un resultado final.

Capa de red feed-forward totalmente conectada de dos capas

La primera capa toma ReLU como función de activación y la segunda toma la función de activación Lineal, de modo que:

$$\mathrm{FFN}(x) = \mathrm{ReLU}(W_1 x + b_1)W_2 + b_2 \tag{2.30}$$

donde W_1, b_1 y W_2, b_2 son los pesos y los parámetros de sesgo de dos capas respectivamente.

Esta red feed-forward totalmente conectada actúa como las capas ocultas de una red neuronal clásica.

Decodificador: Como se muestra en la parte derecha de la Figura 2.10, la parte decodificadora también consiste en una pila de N bloques decodificadora idénticos y cada bloque tiene tres componentes:

Una capa de autoatención multicabezal enmascarada

Esta capa es similar a la del bloque codificador, su propósito es también capturar la relación entre los elementos de la entrada de secuencia del decodificador.

Una capa de atención multicabezal

Esta parte sigue siendo diferente de la capa de autoatención multicabezal del bloque codificador y de la capa de autoatención multicabezal enmascarada del bloque decodificador. En esta capa de atención multicabezal, la consulta, la clave y el valor proceden de fuentes distintas, es decir: la consulta procede del componente anterior del bloque descodificador y la clave y el valor son la salida del codificador. Este mecanismo se encarga de buscar la relación entre la entrada del codificador y la entrada del descodificador.

Una red feed-forward totalmente conectada de dos capas

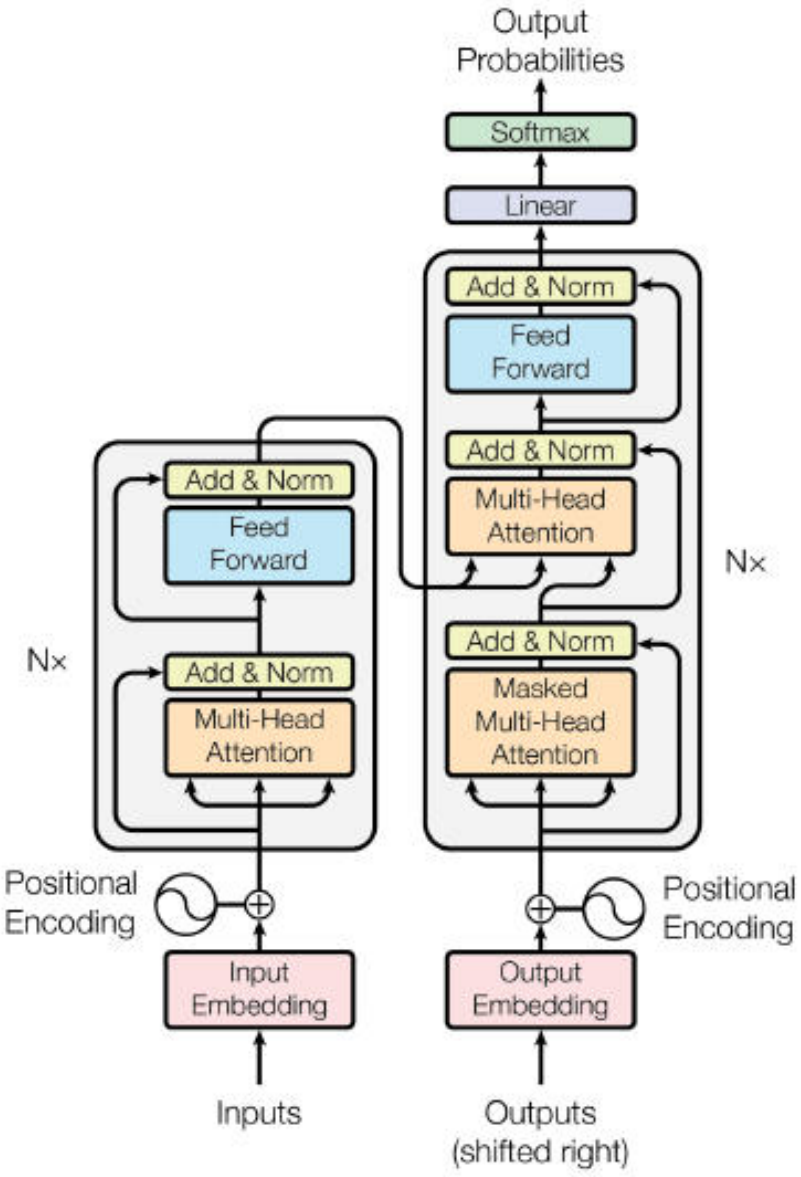
Esta parte es idéntica al de bloque codificador.

2.3.2. *Atención*

Una función de atención puede describirse como la asignación de una consulta y un conjunto de pares clave-valor a una salida, donde la consulta, las claves, los valores y la salida son todos vectores. La salida se calcula como una suma ponderada de los valores,

Figura 2.10

Estructura de Codificador-Descodificador de la aquitectura Transformer



*N*ota. Tomado de Vaswani et al. (2023).

donde el peso asignado a cada valor se calcula mediante una función de compatibilidad de la consulta con la clave correspondiente. (Vaswani et al., 2023)

2.3.3. *Atención a los productos de puntos escalados*

Según en la Figura 2.11 de lado izquierdo. la entrada consiste en consultas y claves de dimensión d_k, y valores de dimensión d_v (Vaswani et al., 2023).

Se calcula los productos escalares de la Q con todas las K, se divide cada uno por $\sqrt{d_k}$ y luego se aplica una función $softmax$ para obtener los pesos de los valores.

El producto punto es una forma sencilla de medir la relevancia de consulta con respecto a la clave, pero esta operación requiere que tanto la consulta como la clave tengan la misma dimensión (Lezmi & Xu, 2023).

Formalmente, el vector de atención para n consultas Q, claves K y valores V se calcula como:

$$\text{Attention}(Q, K, V) = \text{softmax}_k\Big(\frac{QK^T}{\sqrt{d_k}}\Big)V \tag{2.31}$$

donde $Q \in \mathbb{R}^{n\times d_k}, K \in \mathbb{R}^{n\times d_k}, V \in \mathbb{R}^{n\times d_v}$ y podemos observar que los productos escalares QK^T se reducen en $\sqrt{d_k}$.

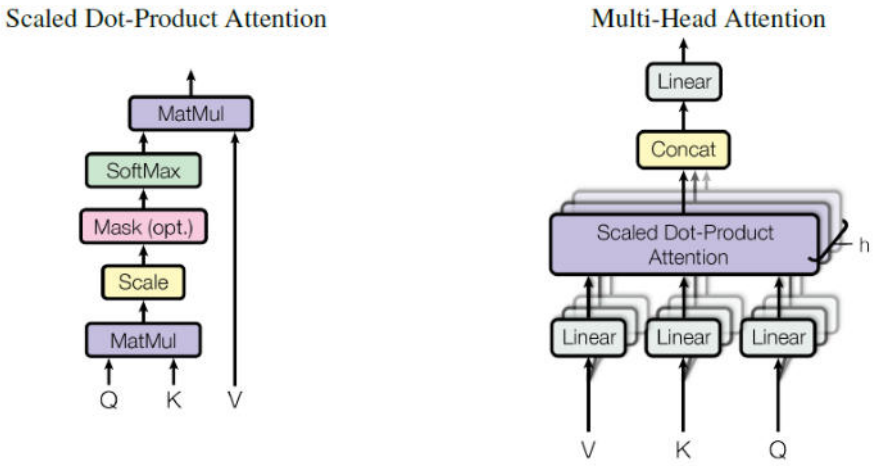
Cabe precisar, para que el álgebra funcione, las matrices de n K y V deben ser iguales, pero para la matriz de n consultas Q pueda variar. Del mismo modo, la dimensionalidad de las K y Q debe coincidir, pero la de los V puede variar.

2.3.4. *Atención multicabezal*

En lugar de realizar una única función de atención con claves,valores y consultas en d_{model}-dimensionales, nos pareció beneficioso proyectar linealmente las consultas, claves y valores h veces con diferentes proyecciones lineales aprendidas a d_k, d_k y d_v dimensiones,

Figura 2.11

(izquierda) Atención a los productos de puntos escalados, (derecha) Atención multicabezal



*N*ota. Tomado de Vaswani et al. (2023).

respectivamente (Vaswani et al., 2023).

El mecanismo de atención multicabezal como se puede observar en la Figura 2.11 de lado derecho, se realiza el proceso de atención de producto escalado de puntos varias veces de forma independiente para obtener diferentes matrices de puntuación de atención y luego se las fusiona.

Sea h el número de cabezas del mecanismo de autoatención multicabeza. Dada una entrada X, tenemos (Lezmi & Xu, 2023):

$$
\begin{aligned}
Q_i &= XW_i^Q \\
K_i &= XW_i^K \quad \text{para } i = 1, \ldots, h \\
V_i &= XW_i^V
\end{aligned}
$$

donde $W_i^Q \in \mathbb{R}^{d_{input} \times d_{head}}, W_i^K \in \mathbb{R}^{d_{input} \times d_{head}}, W_i^V \in \mathbb{R}^{d_{input} \times d_{head}}$. A continuación, cada

cabeza es una matriz de puntuación de la atención:

$$\text{head}_i = \text{Attention}(Q_i, K_i, V_i)$$

luego se concatena todas estas matrices de puntuación de la atención:

$$\text{Self-MultiHead}(X) = \text{Concat}(head_1, \cdots, head_h)W^O \tag{2.32}$$

donde $W^O \in \mathbb{R}^{h.d_{head} \times d_{model}}$ también es un parámetro aprendible para la transformación lineal.

2.3.5. *Codificación posicional*

En Vaswani et al. (2023), los autores diseñaron una codificación posicional utilizando funciones trigonométricas seno y coseno de frecuencias diferentes:

$$\text{PE} = \begin{cases} p_{i,2j} &= \sin\left(\frac{i}{10000^{\frac{2j}{d_{input}}}}\right) \\ p_{i,2j+1} &= \cos\left(\frac{i}{10000^{\frac{2j}{d_{input}}}}\right) \end{cases}$$

para $i = 0, \cdots, l-1, j = 0, \cdots, [(d-1)/2]$

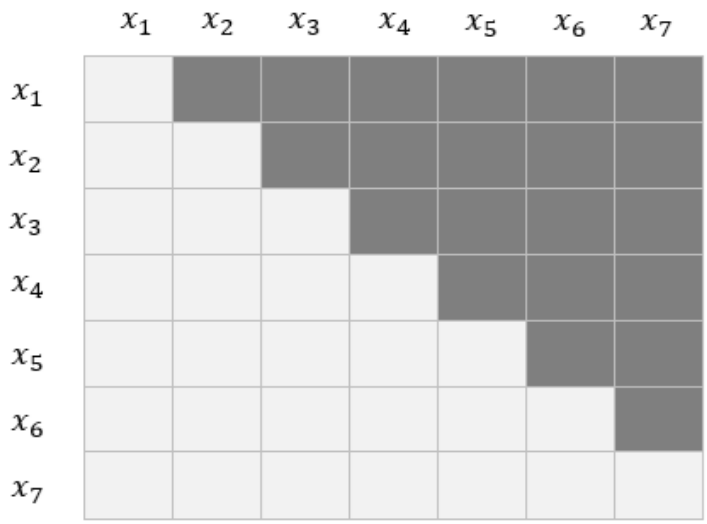
Si la longitud de la frase viene dada por la l y la dimensión/profundidad de la incrustación por d, la codificación posicional PE es una matriz bidimensional de la misma dimensión, es decir, PE$\in \mathbb{R}^{l \times d}$. Cada posición puede representarse con una ecuación en términos de i, que está a lo largo de l, y j que está a lo largo de d (Kamath et al., 2022).

Como se muestra en la Figura 2.10, las dos matrices, es decir, la incrustación de palabras X y la codificación posicional PE se suman para generar la representación de entrada $\tilde{X} = X + \text{PE} \in \mathbb{R}^{l \times d}$

Las incrutaciones posicionales sirven para que el Transformer conozca la posición de los vectores de entrada.

Figura 2.12

Ejemplo matriz de enmascaramiento para el bloque decodificador



*N*ota. La matriz consiste con valores 0 en su triángulo inferior y $-\infty$ en el resto.

2.3.6. *Atención enmascarada*

En los casos que el modelo no debe prestar atención a los tokens futuros (para evitar la fuga de datos durante el entrenamiento), se aplica una máscara de atención triangular. Esta máscara garantiza que cada token sólo pueda prestar atención a los tokens anteriores de la secuencia (DataSpeckle, 2023).

Este proceso implica, no utilizar la información futura, se debe ocultar la secuencia de los próximos elementos durante su entrenamiento. Para ello, tiene que añadir una mascara de relleno a la entrada del decodificador, como se muestra en la Figura 2.12. A este proceso se llama enmascaramiento prospectivo (Lezmi & Xu, 2023).

Su cálculo para atención al producto punto escalado enmascarado, la Ecuación 2.31 tendría la nueva forma:

$$\text{Attention}(Q, K, V) = \text{softmax}_k\Big(\frac{QK^T}{\sqrt{d_k}} + M\Big)V \tag{2.33}$$

Tabla 2.2

Tipos de hiperparámetros de Transformers

Hiperparámetros de modelo	**Hiperparámetros del algoritmo**
Número de capas	Tasa de aprendizaje
Tamaño oculto	Tamaño de lote
Número de cabezas de atención	Número de épocas de entrenamiento
Dimensiones del vector de incrustación	Decaimiento del peso

donde $M \in \mathbb{R}^{n \times n}$, es la matriz que precisamente se utiliza para enmascarar las similitudes que incluyen marcos futuros.

2.3.7. *Hiperparámetros del modelo Transformer*

En el contexto de los Transformers, el ajuste de los hiperparámetros es esencial para optimizar el rendimiento del modelo y garantizar su buena generalización a los datos no vistos.

De acuerdo a Restack (2024), los hiperparámetros se clasifican en dos tipos principales: parámetros que definen la arquitectura del modelo Transformer y los parámetros que afectan al proceso de entrenamiento.

En la Tabla 2.2 se muestran los hiperparámetros de Transformer.

2.4. Regularización para Deep Learning

La regularización es una técnica utilizada en ML y DL, que sirven para evitar *overfitting* y mejorar el rendimiento de generalización de un modelo (Analytics Vidhya, 2024).

En la Figura 2.13 detalla sobre *overfitting*, el modelo aprende demasiado bien con

datos de entrenamiento pero su rendimiento es deficiente con datos no vistos.

Para prevenir este problema de *overfitting* en el modelamiento de Deep Learning, se utilizan varias técnicas de regularización.

De acuerdo Lemberger (2017), los mecanismos de regularización mas comunes en Deep Learning son las siguientes:

- **Penalización L^2 sobre los pesos** consiste en añadir términos cuadráticos a la función de pérdida, lo que tiende a suavizar el paisaje de optimización.

- **Dropout** consiste en eliminar aleatoriamente algunos eslabones de la red neuronal durante el entrenamiento, impidiendo así que el modelo se adapte lo suficiente a los datos entrenados.

- **Early stopping** consiste en monitorizar el error de predicción en un conjunto de validación mientras se realiza el descenso de gradiente y detenerlo cuando el error da validación empieza a aumentar.

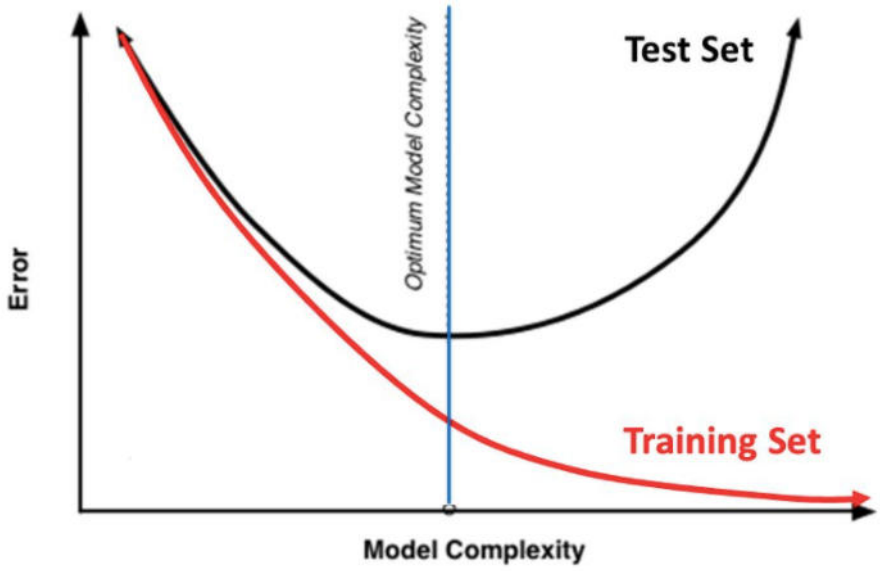
- **Data augmentation** se utiliza en problemas de reconocimiento de imágenes. El conjunto de datos original se aumenta con imágenes artificiales, distorsionadas (el tamaño, la orientación, el tono) de las imágenes originales.

- **Implicit regularization** reduce eficazmente la clase de $\mathcal{F}$[3] como consecuencia de la dinámica de cómo el algoritmo explora $\mathcal{F}$ al optimizar los parámetros w de la red.

[3]Son clase de funciones llamada clase de hipótesis, que denota implícitamente mediante una arquitectura de red neuronal junto con uno o mas procedimientos de regularización.

Figura 2.13

Error del conjunto de entrenamiento frente al conjunto de prueba



*N*ota. Tomado de Analytics Vidhya (2024).

Capítulo 3

Metodología

3.1. Tipo y nivel de investigación

De acuerdo a Ñaupas et al. (2018), "Es el proceso de investigación de las técnicas y de la tecnología que se basan en los resultados de las investigaciones básicas y de las investigaciones aplicadas" (p. 136). Por lo tanto, en este estudio se considera tipo y nivel de investigación:

- **Tipo**: Aplicada-Tecnológico
- **Nivel**: Predictivo

3.2. Población y muestra

La población de radiación solar son datos históricos, con gran cantidad de datos en series temporales de *Prediction Of Worldwide Energy Resource* (POWER) que existe desde el año 1981 hasta la actualidad.

La muestra para esta investigación, se realiza por "conveniencia" (Ñaupas et al.,

2018, p. 342), ha sido seleccionado como un punto en la ciudad de Ilo, con las siguientes coordenadas: latitud: -17.63185, longitud: -71.34108 tal como se visualiza en la Figura 3.1, periodo de 01 de enero del 2018 al 09 de septiembre del 2023. Los datos son series de tiempo con las variables año, mes, día y la irradiancia descendente de onda corta expresada en (Kwh/m^2/día).

3.3. Instrumentos

Para el modelo de predicción diaria de radiación solar, se construye a partir de datos históricos en tiempo de series. La data es obtenida de la plataforma de NASA's POWER, que son conjuntos de datos basados de satélite.

3.4. Procedimientos

En este estudio, para análisis y la modelación de datos, se utiliza la metodología CRISP-DM (*Cross Industry Standard Process for Data Mining*) (Schröer et al., 2021).

Esta metodología consiste de seis fases iterativas diseñado para incorporar la minería de datos en sus prácticas empresariales más amplias. En la Figura 3.2 se muestran los siguientes fases:

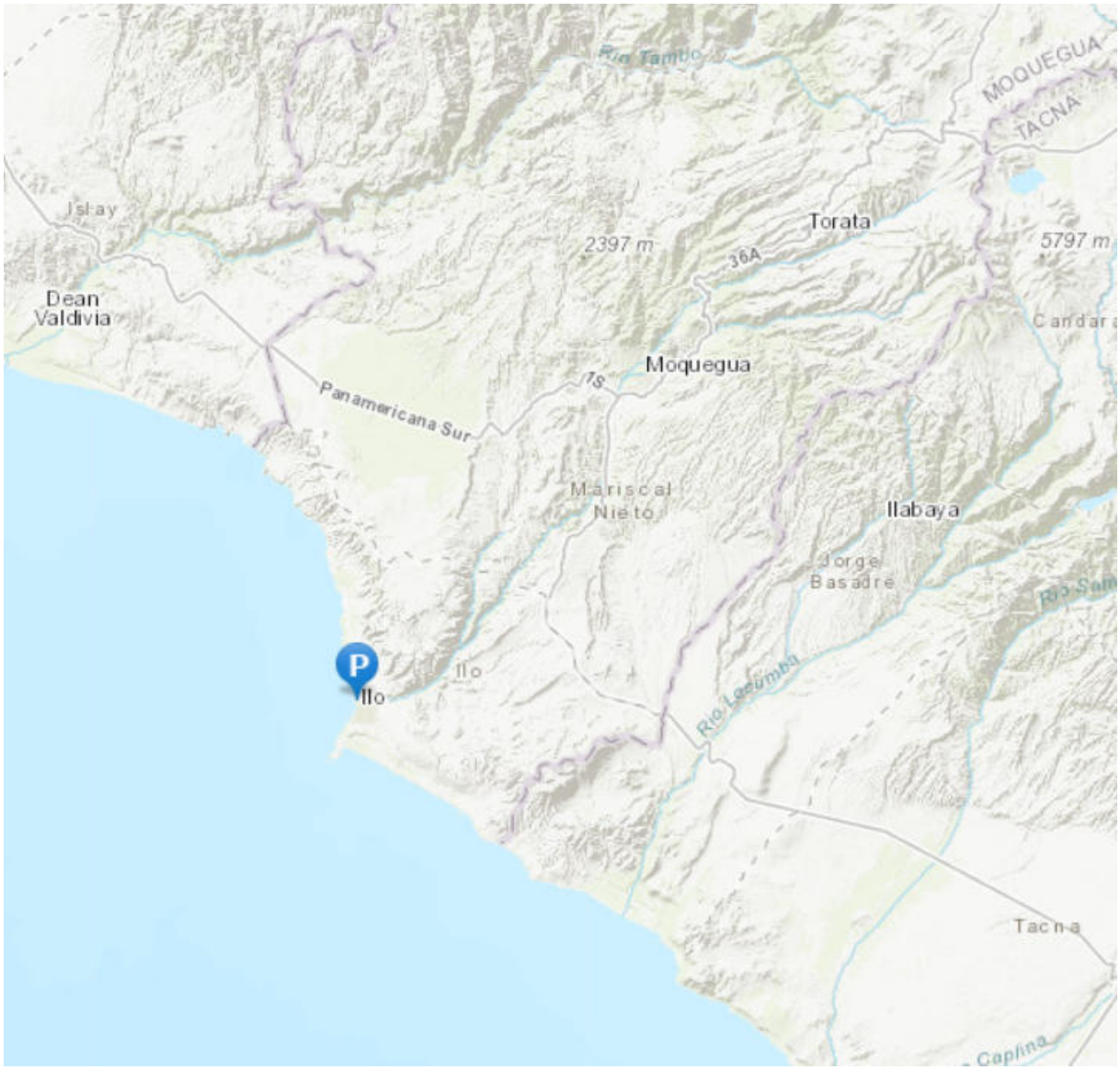
1. **Comprensión de la empresa**. Consiste la definición explícita del objetivo de la minería de datos.

2. **Comprensión de los datos**. Consiste en recopilar datos, explorarlos y describirlos.

3. **Preparación de los datos**. Métodos y enfoques (transformación, selección, limpieza)

Figura 3.1

Ubicación del punto en la ciudad de Ilo para el estudio en la investigación

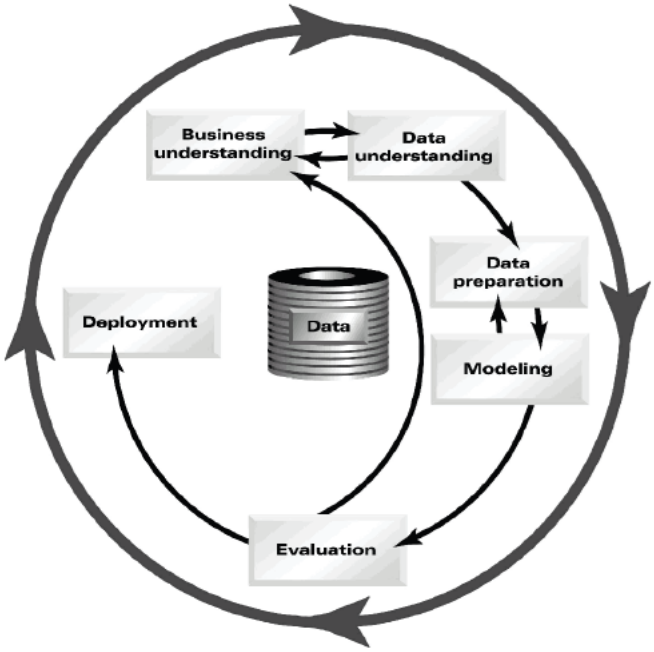


.

*N*ota. Adaptado del Visor de Acceso a Datos POWER por el proyecto POWER de la NASA (https://power.larc.nasa.gov/data-access-viewer/)

Figura 3.2

Modelo de proceso CRISP-DM



*N*ota. Tomado de SPSS Inc. (2005).

4. **Modelización**. Consiste en seleccionar la técnica de modelización, construir el caso de prueba y evaluación de modelo.

5. **Evaluación**. Definición de métricas, visualización del modelo y las métricas.

6. **Despliegue**. En esta fase se trata de elaboración de un informe final ó de un componente de software y la revisión del proyecto.

3.4.1. *Conjunto de datos*

El conjunto de datos son series temporales de radiación solar en la provincia de Ilo, que compone de cuatro columnas, las tres primeras columnas etiquetadas como año, mes y día; se combinaron en una sola característica denominada (año-mes-día), la columna

Tabla 3.1

Explicación de las variables de conjunto de datos

Comentario	**Variable**	**Explicación**
Combinado de la variable tiempo	año	Número de años en la fila
	mes	Número de meses en la fila
	día	Número de días en la fila
Objetivo	RS	RS en kwh/m^2 por día

Tabla 3.2

Descripción estadistico de radiación solar diaria

Estadística	**valores**
Total de observaciones	2077
Rango	2018-01-01 al 2023-09-08
Mínimo	1.90
Máximo	8.79
Media	5.56
Desviación Estándar	1.63

radiación descendente de onda corta es la variable de objetivo, tal como se puede observar en la Tabla 3.1.

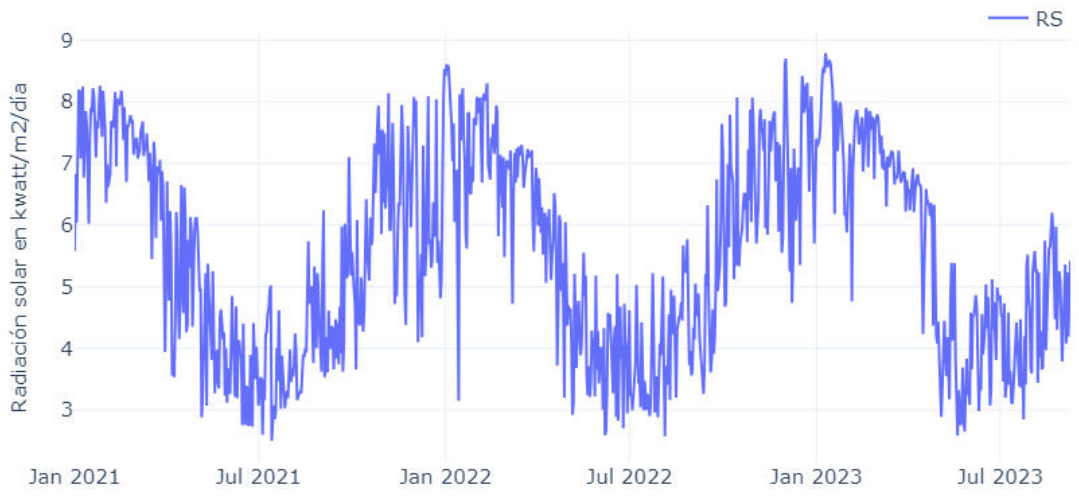
3.4.1.1. Descripción de las variables

La descripción de la variable en cuestión se muestra en la Tabla 3.2. Cabe precisar, que las muestras de radiación solar son completos, sin datos faltantes.

La Figura 3.3 muestra la data histórica de radiación solar diaria en la provincia de

Figura 3.3

Evolución diaria de radiación solar en la provincia de Ilo



Ilo de 01-01-2021 al 08-09-2023; en el cual, se observa que los niveles altos de radiación solar se registran en estación de verano, especialmente en mes de enero, los niveles bajos en los meses de junio y julio que corresponde a la estación de invierno; siendo la serie de RS como estacionario e irregular.

3.4.2. *Preprocesamiento de datos*

De acuerdo Skillfloor (2023), el preprocesamiento de datos es un paso indispensable en el proceso de ML, ya que la calidad y la idoneidad de los datos afectan directamente a la precisión y la fiabilidad del modelo.

En esta investigación, para tratamientos de datos se utiliza la técnica de estandarización y la normalización.

3.4.2.1. Estandarización

La estandarización es otro método de escalado de características en el que los valores se centran en torno a la media con una desviación estándar unitaria. Esto significa que la media del atributo se convierte en cero y la distribución resultante tiene una desviación estándar unitaria. La ecuación se muestra a continuación:

$$y_i = \frac{x_i - u}{\sigma} \tag{3.1}$$

donde

y_i es el nuevo valor, x_i el valor original, u media de la variable x_i y σ desviación estándar de la variable x_i

3.4.2.2. Normalización Mín-Máx

La normalización Mín-Máx es una técnica de escalado que transforma las características en un rango específico, normalmente [0,1]. Matemáticamente expresado de la siguiente forma:

$$x_{normalized} = \frac{x - min(x)}{max(x) - min(x)} \tag{3.2}$$

donde $x_{normalized}$ es el valor normalizado, x es el valor original, $min(x)$ es el valor mínimo de las característica en todos los puntos de datos y $max(x)$ es el valor máximo de la característica en todos los puntos de datos.

3.4.2.3. División de datos

Para este estudio, los datos de radiación solar en la provincia de Ilo, cuenta con un total de 2077 muestras, son datos que dependen del tiempo. Se dividen en tres conjuntos de datos (entrenamiento, validación y prueba) en periodos de tiempo sucesivos, manteniendo el orden cronológico tal como se puede apreciar en la Tabla 3.3.

Para conjunto de datos de entrenamiento, en la que se utiliza para entrenar al modelo, se selecciona de 01 de enero del 2018 al 23 de diciembre del 2021 con un total de 1453 muestras que equivale el 70 % del total. Para el conjunto de validación, en la que se utiliza para mitigar el efecto de sobreajuste en la etapa de entrenamiento, cuyas muestras para este conjunto van de 24 de diciembre del 2021 al 31 de octubre del 2022 un total de 312 que representa el 15 % del total y para conjunto de datos de prueba, que se utiliza para medir el rendimiento de los modelos de predicción; las muestras para este conjunto se toma de 01 de noviembre del 2022 al 08 de septiembre del 2023, siendo un total de 312 muestras que representa el 15 %.

Tabla 3.3

División de datos para encontrar modelo de predicción

Conjunto	Rango	Porcentaje
Entrenamiento	01-01-2018 al 23-12-2021	70 %
Validación	24-12-2021 al 31-10-2022	15 %
Prueba	01-11-2022 al 08-09-2023	15 %

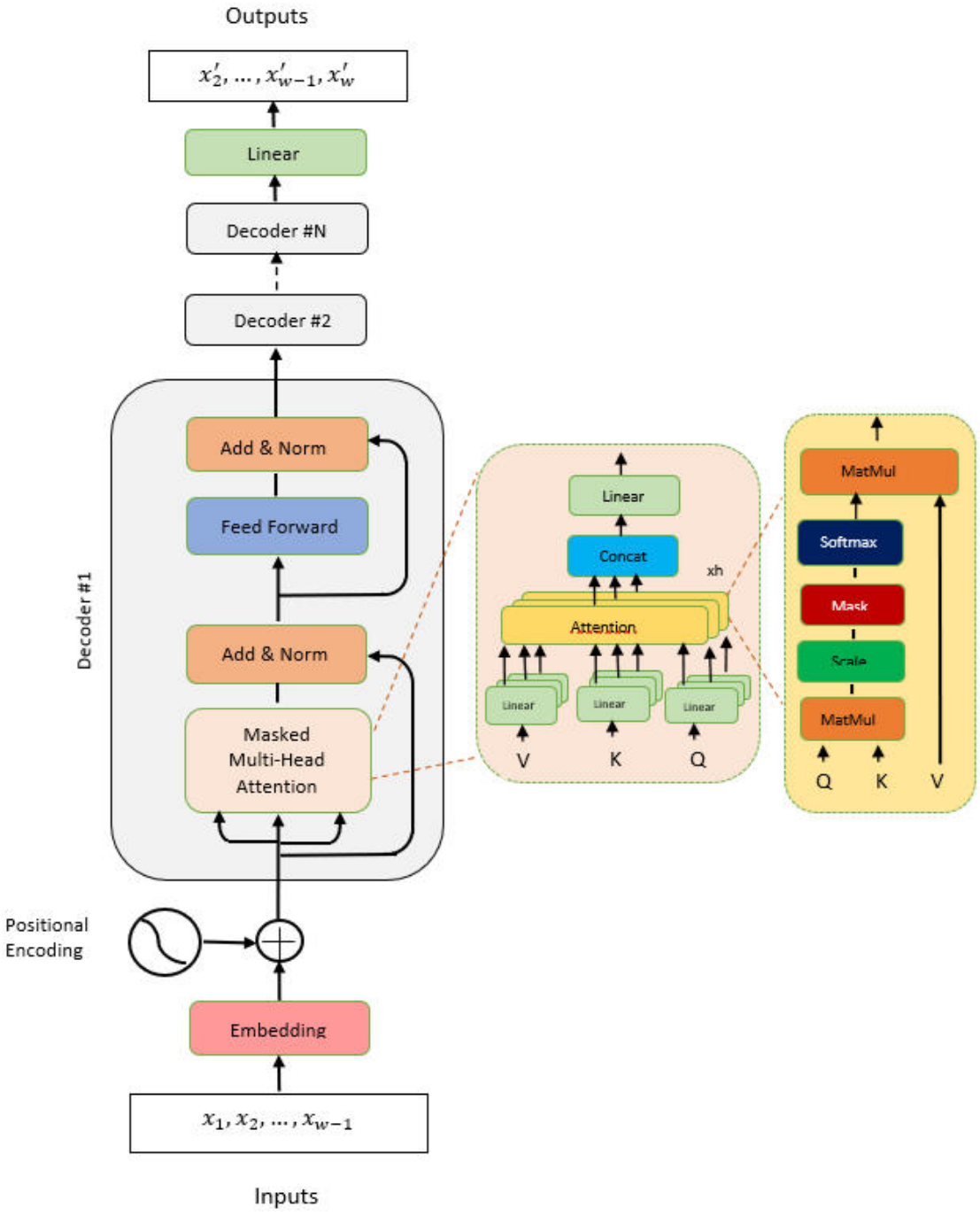
3.4.3. *Implementación del modelo Transformer*

La arquitectura de Transformer original se compone normalmente de un codificador y decodificador. Sin embargo, en este estudio, consideramos en utilizar una arquitectura Transformer de solo Decodificador tal como se puede apreciar en la Figura 3.4, con esta arquitectura nos permitirá predecir los valores futuros de radiación solar a partir de sus valores históricos con modelos univariantes [1], es decir solo con la variable objetivo.

[1]Los modelos univariantes significa que solo se utiliza la misma variable de interés para predecir la respectiva variable objetivo.

Figura 3.4

Arquitectura del transformer de sólo decodificador para predicción del radiación solar



Para el experimento, se ha utilizado diferentes predicciones de RS con el propósito de encontrar el mejor modelo de predicción, para ello se crea cuatro conjuntos (uno para predicciones con diferentes ventanas de retrospección y tres relacionados a los parámetros del modelo Transformer). A continuación de definen los valores:

- Predicciones con las entradas de varios pasos de tiempo (días) w cuyos valores son $w = \{7, 14, 30, 60\}$.
- Número de bloques del Transformer con valores $N = \{1, 2, 3\}$
- Multicabezal $h = \{2, 3, 4\}$.
- Dimensión del modelo $d_{model} = \{16, 32\}$

En la Tabla 3.4 se definen los valores relacionados a los parámetros del modelo Transformer.

Tabla 3.4

Búsqueda de parámetros para el modelo transformer

(a) Parámetros de arquitectura del modelo

Hiperparámetros	**valores**
Layers(N)	1,2,3
head (h)	2,3,4
d_{model}	16,32
d_ff	32

(b) Parámetros de entrenamiento

Hiperparámetros	**valores**
lr	0.00005
Dropout	0.1
Optimizer	Adam

Para implementar el algoritmo, se utiliza la librería Pytorch. El algoritmo Transformer desarrollado se muestra en el Listing 1. El algoritmo comienza dividiendo la data en tres conjuntos: Entrenamiento, validación y prueba, generada con la función *split_dataset*

(línea 1), su código python de esta función se puede apreciar en Apéndice A. En la siguiente línea se divide cada conjunto en entradas y objetivos con la función *data_to_dataset* (línea 2), su código python en Apéndice B. Después de la configuración y ejecución para el entrenamiento (linea 3), se construye el modelo Transformer de solo Decodificador, definiendo los parámetros de su arquitectura (*núm de bloques N, dimensión de la entrada D, núm de heads H*) y del entrenamiento (línea 4), ver su código en Apéndice C.

Una vez construida el modelo Transformer de sólo decodificador, se debe compilar y analizar con datos de entrenamiento y validación (líneas 6-21). Al compilar un modelo, se debe especificar un algoritmo de optimización (Línea 5) y la función de pérdida (línea 16). El *Adam* y MSE se utilizan como el algoritmo de optimización y la función de pérdida respectivamente. Y finalmente, el modelo Transformer predice con datos de prueba (líneas 35-43), para que posteriormente sea evaluada con varias métricas de predicción.

Listing 1 *Algoritmo de Transformer para entrenamiento y prueba*

Input: *data histórica de radiación solar*

Output: *train_losses, val_losses, valores predichos por el modelo*

1: *(train, val, test) ⟵ split_dataset(data,train_split,val_split)*

2: *(train_d, val_d, test_d) ⟵ data_to_dataset(train, val, test, batch_size)*

3: *device ⟵ torch.device("cuda:0" if torch.cuda.is_available() else "cpu")*

4: *transformer ⟵ $Transformer$(N, D, H, dim_ff, inp_feature, out_feature, drouput).to(device)*

5: *optimizer ⟵ torch.optim.Adam(transformer, parameters(), lr)*

6: *epoch ⟵ n*

7: *n_iter ⟵ len(train_d)*

8: *losses, val_losses ⟵ [ ],[ ]*

9: ***for*** *e in tqdm(range(epoch))* ***do***

```
transformer.train()
sum_train_loss ⟵ 0
for x, y in train_d do
    S ⟵ x.shape[1]
    mask ⟵ create_look_ahead_mask(S)
    out, _ ⟵ transformer(x, mask)
    loss ⟵ torch.nn.MSELoss()(out, y)
    sum_train_loss + ⟵ loss.item()
    loss.backward()
    optimizer.setp()
end for
 losses.append(sum_train_loss/n_iter)
#Evaluación con conjunto de validación
transformer.eval()
sum_val_loss ⟵ 0
for i, (x,y) in enumerate(val_d) do
    S ⟵ x.shape[1]
    mask ⟵ create_look_ahead_mask(S)
    out, _ ⟵ transformer(x, mask)
    loss ⟵ torch.nn.MSELoss()(out, y)
    sum_val_loss + ⟵ loss.item()
end for
val_losses.append(sum_val_loss/(i+1))
end for
```

34: #Predicción con conjunto de prueba

35: test_preds ⟵ [ ]

36: transformer.eval()

37: ***for*** *(x, y) in test_d* ***do***

38: *S ⟵ x.shape[-2]*

39: *mask ⟵ create_look_ahead_mask(S)*

40: *y_pred, _ ⟵ transformer(x, mask)*

41: *test_preds.append(y_pred.detach().cpu().numpy())*

42: ***end for***

43: test_preds ⟵ np.vstack(test_preds)

3.4.4. *Métricas de evaluación del modelo*

Para su evaluación y la comparación de los modelos de predicción, se utiliza las siguientes métricas de predicción: Error cuadrático medio (MSE), error absoluto medio (MAE), raíz de error cuadrático medio ($RMSE$), error porcentual absoluto medio ($MAPE$) y coeficiente de determinación (R^2); a continuación se muestran las siguientes fórmulas correspondientes:

$$MSE = \frac{1}{n}\sum_{t=1}^{n}(y_t - \hat{y}_t)^2 \tag{3.3}$$

$$MAE = \frac{1}{n}\sum_{t=1}^{n}|y_t - \hat{y}_t| \tag{3.4}$$

$$RMSE = \sqrt{\frac{1}{n}(y_t - \hat{y}_t)^2} \tag{3.5}$$

$$MAPE = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{y_t - \hat{y}_t}{y_t}\right| \times 100 \tag{3.6}$$

$$R^2 = 1 - \frac{\sum_{t=1}^{n}(y_t - \hat{y}_t)^2}{\sum_{t=1}^{n}(\bar{y} - y_t)^2} \tag{3.7}$$

donde y_t es la radiación solar actual en en tiempo t, $\hat{y}_t$ es el valor predicho por el modelo en tiempo t y $\bar{y}$ el promedio de la radiación solar.

Los posibles valores de estas métricas que puedan dar, son los siguientes: para MSE, MAE y $RMSE$ el mejor valor es 0 y peor valor es $+\infty$, para $MAPE$ el mejor valor es 0 % y peor valor es $+\infty$, y para R^2 el mejor valor es 1 (100 %) y peor valor es 0.

3.5. Medio informático

Todos los experimentos realizados en única GPU NVIDIA GeForce RTX 3050 de 8 GB. Los códigos fuente están implementadas en Python 3.8.0 y pytorch 2.1.2

Capítulo 4

Resultados y discusión

Después de procesar la data histórica de radiación solar, la implementación del modelo Transformer de solo Decodificador, y la realización de las pruebas correspondientes, presentamos los resultados en cuatro partes: resultados de modelización Transformer, resultados de evaluación del modelo, resultados de comparación con otros modelos de predicción y resultados de predicción diaria de radiación solar en la provincia de Ilo; posteriormente se dará discusión en base a los resultados obtenidos.

4.1. Resultados de modelización Transformer

En la etapa de entrenamiento con conjunto de datos de entrenamiento y la validación, se hizo el uso de la técnica *Grid Search* (Búsqueda de Cuadrícula) con la finalidad de encontrar el mejor modelo de predicción, obteniendo así un total de 72 modelos de Transformer de solo Decodificador, y se eligió el que tenga el error mínimo. Los parámetros óptimos del modelo, se resume en la Tabla 4.1, según esta tabla, muestra los parámetros de la arquitectura de Transformer basado de sólo Decodificador, que compone de 2 bloques (Layers), 4 heads (multicabezal), 32 como dimensión de la capa interna de red

Tabla 4.1

Definición de hiperparámetros para el modelo Transformer en la predicción de RS

Hiperparámetros	**Valores**
Layers	2
head	4
d_{model}	32
d_ff	32
lr	0.00005
batch_size	32
Dropout	0.1
Optimizer	Adam
Epochs	20

feed-forward (d_ff) y 32 como la dimensión del modelo o la entrada (d_{model}).

En la Figura 4.1, muestra el resultado de entrenamiento con valores óptimos.

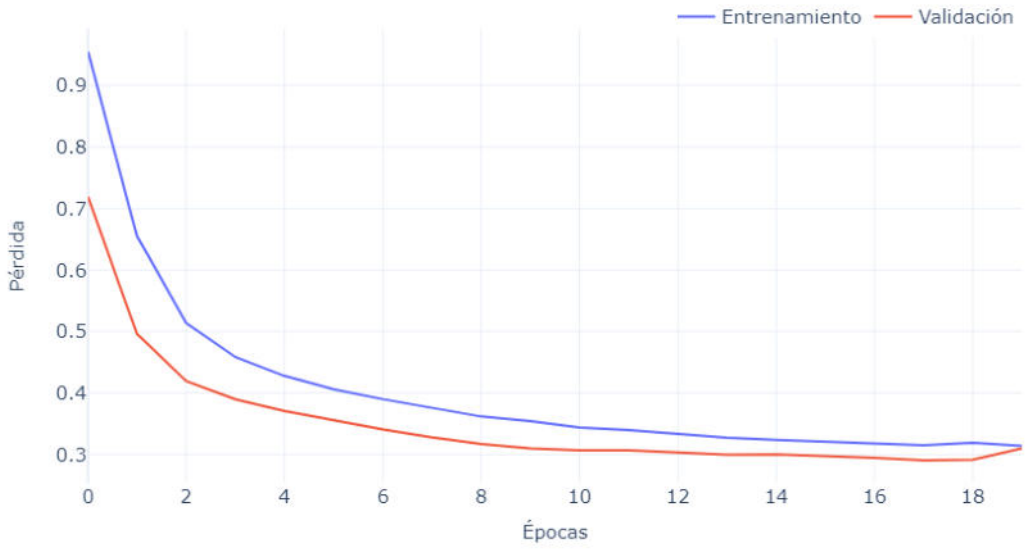
4.2. Resultados de evaluación del modelo

En la etapa de prueba, con modelo entrenado y sus respectivos parámetros óptimos, se evaluó con datos no vistos durante su entrenamiento, es decir, con conjunto de prueba; para conocer su rendimiento en la predicción de radiación solar, se ha utilizado varias métricas, en las cuales se obtiene los siguientes resultados: $MSE = 0{,}0243$, $MAE = 0{,}0901$, $RMSE = 0{,}1559$, $MAPE = 1{,}52\,\%$ y $R^2 = 0{,}9912$.

Los resultados de predicción con diferentes tamaños de ventana retrospectiva w, se observa en la Tabla 4.2, el modelo obtiene buenos resultados con ventana de $w = 7$ y 14

Figura 4.1

Pérdidas en el entrenamiento y validación del modelo Transformer



días; cuando se incrementa ventana ($w > 14$ días) el modelo tiene menos precisión en la predicción.

4.3. Resultados de comparación con otros modelos de predicción

Para resaltar el rendimiento del modelo Transformer de solo Decodificador en la predicción diaria de RS, se comparó con modelos tradicionales ARIMA, los modelos clásicos de deep learning: RNN, LSTM, y GRU. Cabe señalar, que los parámetros óptimos de estos modelos clásicos de deep learning en mención, se obtienen al probar con diferentes ajustes de hiperparametros (número de capas, tasas de aprendizaje y tamaños de lote), con el 85 % en datos de entrenamiento y 15 % para evaluar a los modelos, estos datos han sido normalizados con el uso de la técnica (*Min-Max*) entre los valores de $[0, 1]$. Tras

Tabla 4.2

Desempeño del modelo transformer con distintos valores de w utilizada en la predicción

Pasos de tiempo (w)	**MSE** $(kwh/m^2)^2$	**MAE** (kwh/m^2)	**RMSE** (kwh/m^2)	**MAPE** (%)	**R^2**
7 días	**0.0243**	**0.0901**	**0.1559**	**1.52**	**0.9912**
14 días	0.0343	0.1412	0.1853	2.51	0.9875
30 días	0.158	0.3626	0.3975	7.24	0.9422
60 días	0.2334	0.3709	0.4729	6.56	0.9089

varios ensayos en la etapa de entrenamiento, se ha encontrado los parámetros óptimos. Para modelo RNN de 4 capas ocultas (40,40,40,40) unidades, con activación *tanh*, optimizador *momentum*, tasa de aprendizaje de 0.01, $w = 3$ días, $batch_size = 256$ y *drouput* 0.2 en la primera y segunda capa; para modelo LSTM de 4 capas ocultas (50,40,50,40) unidades, *dropout* de 0.2, con optimizador *adam*, $w = 2$ días, $batch_size = 40$, y para el modelo GRU también de 4 capas ocultas (40,50,30,40) unidades, *dropout* de 0.2 con optimizador *adam*, $w = 2$ días, $batch_size = 40$. Los tres modelos de deep learning se entrenó con $epochs = 20$ y de un paso único (*single-step*).

Para el modelo ARIMA con datos de entrenamiento, se hizo con el uso de *auto_arima*, para encontrar los parámetros óptimos (1,1,1)(0,0,0), y para su evaluación, con conjunto de prueba, se hizo con la técnica de predicciones multietapa con re estimación; es decir, que el modelo se reajusta en cada iteración antes de realizar cada predicción.

En la Tabla 4.3 se resume los parámetros óptimos de los modelos tradicionales (ARIMA, RNN, LSTM, y GRU).

El rendimiento de cada modelo mencionados previamente y con sus respectivos parámetros óptimos, se visualiza en la Figura 4.2. Según esta figura, muestra los resultados ba-

Tabla 4.3

Parámetros óptimos para modelos de predicción

Modelo	**Parámetros**
Arima	$(p = 1, d = 1, q = 1)(P = 0, D = 0, Q = 0)$
RNN	$Layers = (40, 40, 40, 40), batch_size = 256, optim = momentum, droupot = 0{,}2$
LSTM	$Layers = (50, 40, 50, 40), batch_size = 40, optim = Adam, droupot = 0{,}2$
GRU	$Layers = (40, 50, 30, 40), batch_size = 40, optim = Adam, droupot = 0{,}2$

sados en la métricas de $RMSE$ y R^2, en las cuales, refleja su rendimiento de cada modelo en la predicción de radiación solar diaria.

4.4. Resultados de predicción diaria de RS

Utilizando el conjunto de datos de prueba, con el modelo óptimo de Transformer de solo Decodificador, se realizó predicción diaria de radiación solar.

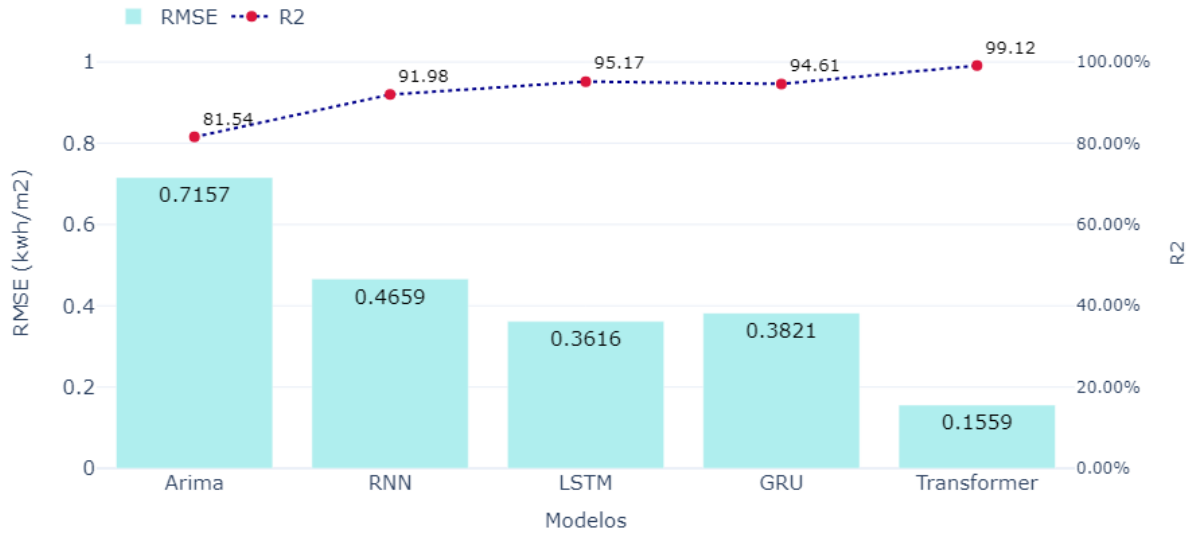
En la Figura 4.3, se visualiza los resultados de los valores predichos por el modelo, con un valor óptimo de ventana de $w = 7$ días y la comparación con valores reales de radiación solar.

4.5. Discusión de resultados

A partir de los resultados alcanzados en la predicción de RS, el modelo propuesto (Transformer de solo decodificador) tiene la capacidad predictiva con datos de radiación solar, con valores de $RMSE = 0{,}1559\ kwh/m^2$, $MAPE = 1{,}52\,\%$ y $R^2 = 99{,}12\,\%$, estos datos indican que nuestro modelo tiene un alto rendimiento en la predicción de RS;

Figura 4.2

Comparación de métricas RMSE y R^2 con otros modelos en predicción de RS



además, según (Lewis, 1982; Moreno et al., 2013), el valor de $MAPE$ es muy inferior del 10 %, por el cual, indica que nuestro modelo propuesto es excelente, debido que su predicción es muy precisa (Figura 4.3). Por lo tanto, la hipótesis general que planteó al inicio de la investigación ha sido aceptada, dado que el modelo Transformer puede predecir la RS en la provincia de Ilo.

Respecto a la arquitectura de Transformer, conocido básicamente por sus dos bloques (Codificador-Decodificador), se pudo diseñar con el bloque decodificador y cada bloque decodificador consta de una capa de atención multicabezal enmascarada seguida de una capa de normalización y una red neuronal feed-forward seguida de una capa de normalización, y al final de N bloques se conecta con la capa lineal, de este modo permite construir el modelo para predicción de RS en la provincia de Ilo. Esta claro, con el uso de este bloque, el modelo aprende con menos cantidad de parámetros que la utilización

en su conjunto. Además, observamos que la optimización del modelo que consiste en encontrar los mejores hiperparámetros para su arquitectura del Transformer como en su entrenamiento, se ha logrado con la utilización del método búsqueda de cuadricula (*grid search*) en la que define el modelo Transformer con sus respectivos valores óptimos (Tabla 4.1). En consecuencia, apoya a la hipótesis especifica 1, dado que si puede diseñar una arquitectura Transformer para predecir la RS en la provincia de Ilo.

Con relación al desempeño del modelo Transformer de solo decodificador, según en la Figura 4.2 observamos que al utilizar solamente con datos históricos de RS como entrada de información, el modelo propuesto es superior en cuanto su precisión en la etapa de prueba frente a otros modelos clásicos como ARIMA, RNN, LSTM, y GRU. Además, se evidencia que el modelo LSTM es mejor que a otros modelos clásicos, pero ámpliamente superado por el modelo propuesto que se encuentra por debajo de 56.9 % de error.

En esta línea, un reciente estudio (Paxi, 2023) utilizó LSTM con la técnica de aumento de datos (DA+LSTM) para predicción de RS, con datos de la ciudad de Moquegua, resaltando como el mejor modelo frente a otros modelos relacionados en la predicción de RS con un resultado de $RMSE = 0{,}1947\ kwh/m^2$. Al hacer la comparación con el modelo Transformer de solo decodificador en términos al valor de $RMSE$ es menor que de DA+LSTM, con una reducción del error de 19,93 %, esto implica que nuestro modelo propuesto predice mejor que DA+LSTM, utilizando solamente con valores pasados de RS y sin aumento de datos.

Otro estudio como de (Mohammadi & Moazenzadeh, 2021) con su modelo propuesto RLM obtiene mejores resultados en la predicción diaria de RS en Tacna, utilizando la precipitación y la humedad relativa como entrada de información al modelo, obtiene el

$RMSE$ con un menor valor de 223.2 J/cm^2 (0.620 kwh/m^2)[1], en términos de $RMSE$ nuestro modelo propuesto reduce en un 74.85 % de error. (Allal et al., 2024) destacan al MLP como el mejor modelo de DL en la predicción de irradiación solar, basándose en datos meteorológicos como la entrada de información, sin embargo, en este estudio, al contar solo con datos históricos de RS, el modelo Transformer de solo decodificador es mas preciso. En (Park et al., 2023) muestran al LSTM como mejor modelo de predicción, al hacer la comparación en términos de R^2, nuestro modelo propuesto sigue siendo superior. (Yusuf et al., 2022) logra una buena precisión con un modelo básico de ML denominada regresión lineal, en términos de R^2 nuestro modelo propuesto es superior con una diferencia de 1.12 %. En (Huang et al., 2022) con el modelo Stacking logra su mejor rendimiento en la predicción diaria y mensual, respecto en la predicción diaria de radiación solar, su resultado en términos de $RMSE$ con un valor de 1.940 MJ/m^2 que equivale a 0.5389 kwh/m^2, nuestro modelo propuesto reduce en un 71.07 % de error. (Brahma & Wadhvani, 2020) muestra su mejor resultado con los modelos LSTM bidireccional y LSTM basado en atención, en términos de $RMSE$ y R^2, nuestro modelo propuesto es superior. (Guermoui et al., 2018) al utilizar el modelo de regresión vectorial de soporte logra un resultado de $RMSE$ con un valor de 1.59 MJ/m^2 (0.442kwh/m^2), el presente estudio reduce en un 64.7 % de error. Por consiguiente, respalda a la hipótesis especifica 2, dado que el modelo propuesto tiene menor tasa de error frente a otros modelos del estado del arte en la predicción diaria de RS en la provincia de Ilo.

[1]Considerando que $1MJ/m^2 = 0{,}27778$ kwh/m^2

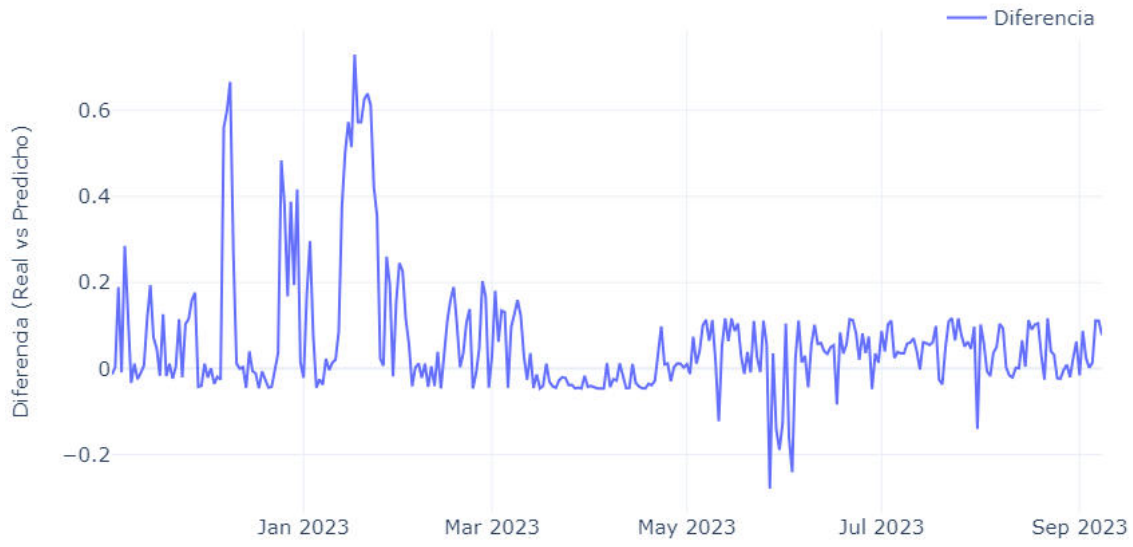
Figura 4.3

Comparación de los valores predichos y reales de Radiación Solar (con $w = 7$ días)

a) Predicho por el modelo vs Valores actuales de RS



b) Diferencia entre valores Predicho y Actuales de RS



Nota. El gráfico b) muestran los errores, es decir, la diferencia entre valores predicho por el modelo y los reales.

Conclusiones

En esta investigación demuestra claramente, que el modelo Transformer tiene la capacidad de predecir la radiación solar diaria en la provincia de Ilo con un error mínimo. Esto abre la posibilidad de utilizar esta potente herramienta para obtención de información de forma anticipada y que permita tomar decisiones correctas.

Se logró diseñar una arquitectura Transformer de solo Decodificador para predicción de radiación solar como un modelo univariante.

Se evaluó al modelo óptimo de Transformer de solo decodificador y se determina que los valores alcanzados muestra su mayor capacidad predictiva y la mejor eficiencia con un valor menor en $RMSE$ y alto valor de índice R^2.

Se comparó el modelo óptimo de Transformer de solo decodificador con otros modelos de predicción de series de tiempo y se concluye que el modelo propuesto es superior en la predicción frente a otros modelos tradicionales como ARIMA, RNN, LSTM, y GRU, utilizando solo con la información pasada de radiación solar en la provincia de Ilo.

Recomendaciones

Las recomendaciones para el uso del modelo Transformer en las tareas de predicción de series temporales son las siguientes:

Cuando la data de series de tiempo son de tipo estacionaria, no es necesario utilizar técnica de aumento de datos, ya que es más simple de aprender para el modelo Transformer cuando se aplica sólo con el bloque decodificador.

Caso, cuando se trata de datos de series no estacionarias, con bastante ruidos (irregular), talvéz sea necesario en utilizar la técnica de aumento de datos, que permita alcanzar los buenos resultados; dado que este tipo de datos son muy difíciles de aprender para cualquier modelos de deep learning.

Para su entrenamiento del modelo, se recomienda estandarizar la data, es decir, con media y desviación estándar.

Y finalmente, se recomienda aplicar el modelo Transformer de solo Decodificador para tareas de predicción, relacionadas a series de tiempo como en áreas de salud pública, agricultura, financiero, astronomía, meteorología, desarrollo empresarial y otros.

Bibliografía

Allal Z., Noura H. & Chahine K. (2024). Machine Learning Algorithms for Solar Irradiance Prediction: A Recent Comparative Study, *e-Prime - Advances in Electrical Engineering, Electronics and Energy 7(2024) 100453*
https://doi.org/10.1016/j.prime.2024.100453

Amatriain X., Sankar A., Bing J., Bodigutla P. K., Hazen T. J. & Kazi M. (2023). Transformer models: an Introduction and catalog, *arXiv:*2302.07730[cs.CL]
https://doi.org/10.48550/arXiv.2302.07730

Analytics Vidhya. (Jun 18, 2024). *Regularization in Deep Learning with Python Code*
https://www.analyticsvidhya.com/blog/2018/04/fundamentals-deep-learning-regularization-techniques/

Berbić J., Ocvirk E., Carević D. & Lončar G. (2017). Application of neural networks and support vector machine for significant wave height prediction, *Oceanologia* (2017) 59,331-349
https://doi.org/10.1016/j.oceano.2017.03.007

Bousnguar H., Battou A. & Najdi L. (2023). Gated Recurrent units (GRU) for Time Series Forecasting in Higher Education, *International Journal of Engineering Research*

& Technology (IJERT) vol. 12, Issue 03 (March 2023)
http://www.ijert.org

Brahma B. & Wadhvani R. (2020). Solar Irradiance Forecasting Based on Deep Learning Methodologies and Multi-Site Data, *Symmetry*, doi:10.3390/sym12111830
www.mdpi.com/journal/symmetry

Burkov A. (2020) *Machine Learning Engineering*, Draft

Büyükşahin, Ü. Ç., & Ertekin, Ş. (2019). Improving forecasting accuracy of time series data using a new ARIMA-ANN hybrid method and empirical mode decomposition. *Neurocomputing*, 361, 151-163
https://doi.org/10.1016/j.neucom.2019.05.099

Centro de Prevención de Riesgos del Trabajo CEPRIT (2017). *La Radiación Solar: ¿Un riesgo laboral?*
http://www.essalud.gob.pe/downloads/ceprit/JULIO_2017_CEPRIT.pdf

Data Science Wizards. (Jun 23, 2023). *Introduction to Time Series Analysis-I.*
https://medium.com/@datasciencewizards/introduction-to-time-series-analysis-i-82d614d462b0

DataSpeckle (Sept 15, 2023). *What is an Attention Mask?.*
https://www.linkedin.com/pulse/what-attention-mask-dataspeckle

El Mghouchi Y. (2022). Best Combinations of Inputs for ANN-Based Solar Radiation Forecasting in Morocco, *Technology and Economics of Smart Grids and Sustainable Energy*, doi: 10.1007/s40866-022-00152-z

Energy.gov (s.f). *Solar Radiation Basics*
https://www.energy.gov/eere/solar/solar-radiation-basics

Fernandes R. & Antonelli M. (2018). *Machine Learning A Practical Approach on the Statistical Learning Theory*, Springer
https://doi.org/10.1007/978-3-319-94989-5

Freeman B., Taylor G., Gharabaghi B. & Thé J. (2018). Forecasting air quality time series using deep learning. *Journal of the Air & Waste Management Association, 68:8, 866-886*, DOI:10.1080/10962247.2018.1459956
https://doi.org/10.1080/10962247.2018.1459956

Garnacho G., Salido R. & Moreno J. (2020). Efectos de la radiación solar y actualización en fotoprotección, *An Pediatr*,92(6):377.e1-377.e9

Goodfellow I., Bengio Y. & Courville A. (2016). *Deep Learning*, MIT Press
http://www.deeplearningbook.org

Guermoui M., Rabehi A., Gairaa K. & BenkacialiS. (2018). Support vector regression methodology for estimating global solar radiation in Algeria, *The European Physical Journal Plus*, doi 10.1140/epjp.i2018-11845-y

HEPACART (May 31, 2017). *far-UV VS. Near-UV*. Hepacart
https://www.hepacart.com/blog/far-uv-vs.-near-uv

Huang L., Kang J., Wan M., Fang L., Zhang C. & Zeng Z. (2021). Solar Radiation Prediction Using Different Machine Learning Algorithms and Implications for Extreme Climate Events, *Front. Earth Sci.*9:596860, doi:10.3389/feart.2021.596860

Huang K., Hussian A., Wang Q. & Zhang R. (2019). *Deep Learning Fundamentals, Theory and Applications*, Springer

Iberdrola (s.f). Solar radiation: what is its impact on the planet and on humans?. https://www.iberdrola.com/social-commitment/solar-radiation

Indrajitbarat (Aug 13, 2023). *Recurrent Neural Networks (RNNs): Challenges and Limitations*, Medium. https://medium.com/@indrajitbarat9/recurrent-neural-networks-rnns-challenges-and-limitations-4534b25a394c

Kamath U., Graham K. & Emara W. (2022). *Transformer for Machine Learning: A Deep Dive*, CRC Press

Kumari P. & Toshniwal D. (2021). Long short term memory-convolutional neural network based deep hybrid approach for solar irradiance forecasting, *Elsevier, Applied Energy* 295(2021) 117061

Lawton G. (2023). *Transformer model*, Enterprise AI.com https://www.techtarget.com/searchenterpriseai/definition/transformer-model

Lemberger P. (2017). On Generalization and Regularization in Deep Learning: An Introduction for Data Scientists, *arXiv:1704.01312v2 [stat.ML]*

Lewis C. D. (1982). *Industrial and business forecasting methods*, London: Butterworths.

Lezmi E. & Xu J. (2023). Time Series Forecasting with Transformer Model and Application to Asset Management, *Electronic copy available at:* https://ssm.com/abstract=4375798

Lindholm A., Wahlström N., Lindsten F. & Schön T. (2022). *Machine Learning A First Course for Engineers and Scientists*, Cambridge University Press. http://www.cambridge.org.

Mohammadi, B. & Moazenzadeh, R. (2021). Performance Analysis of Daily Global Solar Radiation Models in Peru by Regression Analysis, *Atmosphere* 2021, 12 389 https://doi.org/10.3390/atmos12030389

Moreno J., Palmer A., Abab A. & Cajal B. (2013). Using the R-MAPE index as a resistant measure of forecast accuracy, *Psicothema*, doi:10.7334/psicothema2013.23

Naveen Krishnan, K. Ravi Kumar & Chandrapal Singh Inda, (2023). How solar radiation forecasting impacts the utilization of solar energy: A critical review, *Journal of Cleaner Production, Vol 388, ISSN 0959-6526* https://doi.org/10.1016/j.jclepro.2023.135860

Ñaupas Paitán H., Valdivia Dueñas M. R., Palacios Vilela J. J. y Romero Delgado H. E. (2018). *Metodología de la investigación Cuantitativa-Cualitativa y Redacción de la Tesis*, Ediciones de la U

Ordoñez L. E., León D. A., Bucheli V. A. & Ordoñez H. A. (2020). Solar Radiation Prediction on Photovoltaic System Using Machine Learning Techniques. *Revista Facultad de Ingeniería, vol.*29(54), e11751

Park J., Hong S. H., Yeon S. H., Seo B. M. & Lee K. H. (2023). Predictive Model for Solar Insolation Using the Deep Learning Technique, *International Journal of Energy Research, Volume* 2023, *Article ID* 3525651, 17 *pages* https://doi.org/10.1155/2023/3525651

Paxi W. J. (2023). *Predicción de Series de Tiempo de Radiación Solar en el Perú Aplicando Deep Learning, caso: Moquegua y Tacna*. [Tesis para optar el título], Universidad Nacional de Moquegua

Rebala G., Ravi A. & Churiwala S. (2019). *An Introduction to Machine Learning*, Springer.
https://doi.org/10.1007/978-3-030-15729-6

Restack (Sept 9, 2024). *Hyperparameter Tuning Transformer*
https://www.restack.io/p/hyperparameter-tuning-answer-transformers-cat-ai

Rosebrock A. (2017). *Deep Learning for Computer Vision with Python*, pyimagesearch

Rubio L. & Alba K. (2022). Forecasting Selected Colombian Shares Using a Hybrid ARIMA-SVR Model, *Mathematics*. https://doi.org/10.3390/math10132181

Servicio Nacional de Meteorología e Hidrología del perú SENAMHI (8 de febrero de 2023). *Incremento de los niveles de radiación UV a nivel nacional*.
https://www.gob.pe/institucion/senamhi/noticias/698048-incremento-de-los-niveles-de-radiacion-uv-a-nivel-nacional

Schröer C., Kruse F. & Gómez J. M. (2021). A Systematic Literature Review on Applying CRISP-DM Process Model, *Procedia Computer Science*, 181(2021):526-534
https://www.sciencedirect.com/science/article/pii/S1877050921002416

Shi J., Jain M., & Narasimhan G. (2022). Time Series Forecasting (TSF) Using Various Deep Learning Models. *arXiv:2204.11115*
https://doi.org/10.48550/arXiv.2204.11115

Siami S., Tavakoli N. & Siami A. (2018). A Comparison of ARIMA and LSTM in Forecasting Time Series. *2018 17th IEEE International Conference on Machine Learning and Applications*. DOI 10.1109/ICMLA.2018.00227

Skillfloor (Aug 7, 2023). *The Role of Data Preprocessing in Machine Learning* https://skillfloor.medium.com/the-role-of-data-preprocessing-in-machine-learning-d9ad9db54d49

SPSS Inc. (2005).*Clementine 11.1 User's Guide*,Chicago: SPSS Inc.

Stein Z. (Sept 29, 2024). *Solar Radiation*. Carbon Collective https://www.carboncollective.co/sustainable-investing/solar-radiation

Torres J., Hadjout D., Sebaa A., Martinez-Álvares F. & Troncoso A. (2021). Deep Learning for Time Series Forecasting: A Survey. *Big Data*, 9:1,3-21 doi:10.1089/big.2020.0159

Varoquaux G. & Colliot O. (2023). Evaluating machine learning models and their diagnostic value. *Hal Id:* hal-03682454 https://hal.science/hal-03682454v4

Vaswani A., Shazeer N.,Parmar N., Uszkoreit J., Jones L., Gomez A. N.,Kaiser L. & Polosukhin I. (2023). Attention Is All You Need, *arXiv:1706.03762v7[cs.CL]* https://doi.org/10.48550/arXiv.1706.03762

Wu N., Green B., Ben X. & O'Banion S. (2020). Deep Transformer Models for Time Series Forecasting: The Influenza Prevalence Case, *arXiv: 2001.08317v1[cs,LG]*

Yusuf K., Emre B. & Gulizar T. (2022). Solar Radiation Forecast by Using Machine Learning Method for Gaziantep Province, *Journal of Scientific Reports-A*, n°51,127-135

Zhang G. P. (2003). Time series forecasting using a hybrid ARIMA and neural network model, *Neurocomputing, Vol 50, pp 159-175, ISSN 0925-2312*
https://doi.org/10.1016/S0925-2312(01)00702-0.

Apéndices

Apéndice A

Código python: División de datos

splitdata.py

```
1  def split_dataset_into_seq(dataset, start_index=0, end_index=
       None, history_size=7, step=1):
2      '''split the dataset to have sequence of observations of
           length history size'''
3      data = []
4      start_index = start_index + history_size
5      if end_index is None:
6          end_index = len(dataset)
7      for i in range(start_index, end_index):
8          indices = range(i - history_size, i, step)
9          data.append(dataset[indices])
10     return np.array(data)
11
12 def split_dataset(data, TRAIN_SPLIT=0.7, VAL_SPLIT=0.5,
       save_path=None):
```

```
13      '''split the dataset into train, val and test splits'''
14      data_mean = data.mean(axis=0)
15      data_std = data.std(axis=0)
16      data = (data - data_mean) / data_std
17      stats = (data_mean, data_std)
18      data_in_seq = split_dataset_into_seq(data, start_index=0,
            end_index=None, history_size=7, step=1)
19      train_data,val_data=np.split(data_in_seq,[int(TRAIN_SPLIT*
            len(data_in_seq))])
20      val_data,test_data=np.split(val_data,[int(VAL_SPLIT*len(
            val_data))])
21
22      return train_data, val_data, test_data
```

Apéndice B

Código python: División de datos con entradas y objetivos

dataset.py

```
def split_fn(chunk):
    """to split the dataset sequences into input and targets
       sequences"""
    inputs = torch.tensor(chunk[:, :-1, :], device=device)
    targets = torch.tensor(chunk[:, 1:, :], device=device)
    return inputs, targets

def data_to_dataset(train_data, val_data, test_data, batch_size
    =32, target_features=list(range(1))):
    '''
    split each train split into inputs and targets
    convert each train split into a tf.dataset
    '''
    x_train, y_train = split_fn(train_data)
    x_val, y_val = split_fn(val_data)
```

```
14      x_test, y_test = split_fn(test_data)
15      y_train = y_train[:, :, target_features]
16      y_val = y_val[:, :, target_features]
17      y_test = y_test[:, :, target_features]
18      train_dataset = torch.utils.data.TensorDataset(x_train,
            y_train)
19      val_dataset = torch.utils.data.TensorDataset(x_val, y_val)
20      test_dataset = torch.utils.data.TensorDataset(x_test, y_test
            )
21      train_loader = torch.utils.data.DataLoader(train_dataset,
            batch_size=batch_size)
22      val_loader = torch.utils.data.DataLoader(val_dataset,
            batch_size=batch_size)
23      test_loader = torch.utils.data.DataLoader(test_dataset,
            batch_size=batch_size)
24      return train_loader, val_loader, test_loader
```

Apéndice C

Código python: Construcción del modelo Transformer de solo Decodificador

modelTD.py

```
class MultiHeadAttention(nn.Module):
    '''Multi-head self-attention module'''
    def __init__(self, D, H):
        super(MultiHeadAttention, self).__init__()
        self.H = H # number of heads
        self.D = D # dimension
        self.wq = nn.Linear(D, D*H)
        self.wk = nn.Linear(D, D*H)
        self.wv = nn.Linear(D, D*H)
        self.dense = nn.Linear(D*H, D)

    def concat_heads(self, x):
        '''(B, H, S, D) => (B, S, D*H)'''
        B, H, S, D = x.shape
```

```
15          x = x.permute((0, 2, 1, 3)).contiguous()  # (B, S, H, D)
16          x = x.reshape((B, S, H*D))    # (B, S, D*H)
17          return x
18
19      def split_heads(self, x):
20          '''(B, S, D*H) => (B, H, S, D)'''
21          B, S, D_H = x.shape
22          x = x.reshape(B, S, self.H, self.D)     # (B, S, H, D)
23          x = x.permute((0, 2, 1, 3))  # (B, H, S, D)
24          return x
25
26      def forward(self, x, mask):
27          q = self.wq(x)  # (B, S, D*H)
28          k = self.wk(x)  # (B, S, D*H)
29          v = self.wv(x)  # (B, S, D*H)
30
31          q = self.split_heads(q)  # (B, H, S, D)
32          k = self.split_heads(k)  # (B, H, S, D)
33          v = self.split_heads(v)  # (B, H, S, D)
34
35          attention_scores = torch.matmul(q, k.transpose(-1, -2))
              #(B,H,S,S)
36          attention_scores = attention_scores / math.sqrt(self.D)
37
```

```
38          # add the mask to the scaled tensor.
39          if mask is not None:
40              attention_scores += (mask * -1e9)
41
42          attention_weights = nn.Softmax(dim=-1)(attention_scores)
43          scaled_attention = torch.matmul(attention_weights, v)  #
                (B, H, S, D)
44          concat_attention = self.concat_heads(scaled_attention) #
                (B, S, D*H)
45          output = self.dense(concat_attention)  # (B, S, D)
46
47          return output, attention_weights
48
49  # Positional encodings
50  def get_angles(pos, i, D):
51      angle_rates = 1 / np.power(10000, (2 * (i // 2)) / np.
            float32(D))
52      return pos * angle_rates
53
54  def positional_encoding(D, position=100, dim=3, device=device):
55      angle_rads = get_angles(np.arange(position)[:, np.newaxis],
            np.arange(D)[np.newaxis, :],D)
56      # apply sin to even indices in the array; 2i
57      angle_rads[:, 0::2] = np.sin(angle_rads[:, 0::2])
```

```
58      # apply cos to odd indices in the array; 2i+1
59      angle_rads[:, 1::2] = np.cos(angle_rads[:, 1::2])
60      if dim == 3:
61          pos_encoding = angle_rads[np.newaxis, ...]
62      elif dim == 4:
63          pos_encoding = angle_rads[np.newaxis,np.newaxis,  ...]
64      return torch.tensor(pos_encoding, device=device)
65
66  # function that implement the look_ahead mask for masking future
        time steps.
67  def create_look_ahead_mask(size, device=device):
68      mask = torch.ones((size, size), device=device)
69      mask = torch.triu(mask, diagonal=1)
70      return mask  # (size, size)
71
72  class TransformerLayer(nn.Module):
73      def __init__(self, D, H, hidden_mlp_dim, dropout_rate):
74          super(TransformerLayer, self).__init__()
75          self.dropout_rate = dropout_rate
76          self.mlp_hidden = nn.Linear(D, hidden_mlp_dim)
77          self.mlp_out = nn.Linear(hidden_mlp_dim, D)
78          self.layernorm1 = nn.LayerNorm(D, eps=1e-9)
79          self.layernorm2 = nn.LayerNorm(D, eps=1e-9)
80          self.dropout1 = nn.Dropout(dropout_rate)
```

```
81          self.dropout2 = nn.Dropout(dropout_rate)
82          self.mha = MultiHeadAttention(D, H)
83
84      def forward(self, x, look_ahead_mask):
85          attn, attn_weights = self.mha(x, look_ahead_mask)  # (B,
                S, D)
86          attn = self.dropout1(attn) # (B,S,D)
87          attn = self.layernorm1(attn + x) # (B,S,D)
88          mlp_act = torch.relu(self.mlp_hidden(attn))
89          mlp_act = self.mlp_out(mlp_act)
90          mlp_act = self.dropout2(mlp_act)
91          output = self.layernorm2(mlp_act + attn)  # (B, S, D)
92
93          return output, attn_weights
94
95  class Transformer(nn.Module):
96      '''Transformer Decoder Implementating several Decoder Layers
97      '''
98      def __init__(self, num_layers, D, H, hidden_mlp_dim,
            inp_features, out_features, dropout_rate):
99          super(Transformer, self).__init__()
100         self.sqrt_D = torch.tensor(math.sqrt(D))
101         self.num_layers = num_layers
102         self.input_projection = nn.Linear(inp_features, D)
```

```
103         self.output_projection = nn.Linear(D, out_features)
104         self.pos_encoding = positional_encoding(D)
105         self.dec_layers = nn.ModuleList([TransformerLayer(D, H,
                hidden_mlp_dim, dropout_rate=dropout_rate) for _ in
                range(num_layers)])
106         self.dropout = nn.Dropout(dropout_rate)
107
108     def forward(self, x, mask):
109         B, S, D = x.shape
110         attention_weights = {}
111         x = self.input_projection(x)
112         x *= self.sqrt_D
113         x += self.pos_encoding[:, :S, :]
114         x = self.dropout(x)
115
116         for i in range(self.num_layers):
117             x, block = self.dec_layers[i](x=x,look_ahead_mask=
                    mask)
118             attention_weights['decoder_layer{}'.format(i + 1)] =
                    block
119
120         x = self.output_projection(x)
121
122         return x, attention_weights
```

"UNIVERSIDAD NACIONAL DE MOQUEGUA"

VICERRECTORADO DE INVESTIGACIÓN

FORMATO DE AUTORIZACIÓN DE PUBLICACIÓN EN EL REPOSITORIO INSTITUCIONAL UNAM

Formulario de autorización para la publicación electrónica en la página web del Repositorio Institucional de la Universidad Nacional de Moquegua, de conformidad con la Ley N° 30035 Ley que regula el Repositorio Nacional Digital de Ciencia, Tecnología e Innovación de Acceso Abierto y Resolución de Comisión Organizadora N° 041-2022-UNAM que aprueba el Reglamento que regula el Repositorio Institucional Digital de la UNAM.

1. DATOS DEL AUTOR

Nombres y Apellidos : Raul Donato Cruz Huacac
DNI/Carnet de Extranjería/Pasaporte N° : 41014085
Correo electrónico : radon7_c@hotmail.com
Teléfono/Celular : 953998533

2. GRADO, TITULO O ESPECIALIZACIÓN (Marque con una X)

Bachiller	Título Profesional	Magister	Doctor	Segunda Especialidad
	X			

3. TIPO DE PUBLICACIÓN (Marque con una X)

a) Articulo ( )
b) Tesis de pregrado (X)
c) Tesis de Maestría ( )
d) Tesis de Doctorado ( )
e) Monografía ( )
f) Reporte de caso científico o hallazgos ( )
g) Libro ( )
h) Parte de un libro ( )
i) Revisión ( )
j) Conferencia ( )
k) Presentación ( )
l) Artículo preliminar ( )
m) Articulo antes de ser publicado ( )
n) Artículo de prensa ( )
o) Documentos internos ( )
p) Imagen ( )
q) Documentos técnicos ( )
r) Ponencias presentadas en eventos científicos (Jornada, Congreso, Simposio Conferencias, etc.) ( )
s) Informes de suficiencia profesional ( )
t) Patentes producidas por académicos/as ( )

4. TIPO DE TRABAJO DE INVESTIGACIÓN (Marque con una X)

a) Tesis (x)
b) Trabajo de Investigación ( )
c) Trabajo de suficiencia profesional ( )
d) Trabajo académico ( )

5. DATOS DE LA PUBLICACION

a) Título : PREDICCIÓN DIARIA DE RADIACIÓN SOLAR EN LA PROVINCIA DE ILO CON MODELO TRANSFORMER
b) Campo del conocimiento OCDE (Indicar el código de clasificación OCDE de la obra : https://catalogos.concytec.gob.pe/vocabulario/ocde_ford.html) : .....
c) Fecha de publicación : 2024

6. DATOS DEL ASESOR:

a) Nombres y Apellidos : Hugo Euler Tito Chura
b) DNI/Pasaporte/Cédula : 01315850
c) ORCID : 0000-0001-8269-0655

FIRMA : .....

DOCUMENTO DE IDENTIDAD : 41014085
NOMBRES Y APELLIDOS : Raul Donato Cruz Huacac

"UNIVERSIDAD NACIONAL DE MOQUEGUA"

VICERRECTORADO DE INVESTIGACIÓN

LICENCIAS

Licencia Estándar

Bajo los siguientes términos, autorizo el depósito de mi trabajo de investigación en el Repositorio Institucional Digital de la UNAM.

Con la autorización de depósito da mi trabajo de investigación, otorgo a la Universidad Nacional de Moquegua, una Licencia No Exclusiva para reproducir. distribuir, comunicar al público, transformar (únicamente mediante su traducción a otros idiomas) y poner a disposición del público mi trabajo de investigación (incluido el resumen), en formato físico o digital, en cualquier medio, conocido o por conocerse, a través de los diversos servicios provistos por la Universidad, creados o por crearse, tales como el Repositorio Institucional Digital de la UNAM, Colección de Tesis, entre otros, en el Perú y en el extranjero, por el tiempo y veces que considere necesarias, y libre de remuneraciones.

En virtud de dicha Licencia, la UNIVERSIDAD NACIONAL DE MOQUEGUA, podrá reproducir mi trabajo de investigación en cualquier tipo de soporte y en más de un ejemplar, sin modificar su contenido, solo con los propósitos de seguridad, respaldo y preservación.

Declaro que el trabajo de investigación es una creación de mi autoría y exclusiva titularidad, o coautoría con titularidad compartida, y me encuentro facultado a conceder la presente licencia y, asimismo, garantizo que dicho trabajo de investigación no infringe derechos de autor de terceras personas.

La UNIVERSIDAD NACIONAL DE MOQUEGUA consignara el nombre del/los autor/es del trabajo de investigación, y no le hará ninguna modificación más de la permitida en la presente licencia.

Autorizo el depósito (marque con una X).

(X) **SI autorizo que se deposite inmediatamente.**

FIRMA : .....

DOCUMENTO DE IDENTIDAD : 41014085

NOMBRES Y APELLIDOS : Raul Donato Cruz Huacac

FECHA : 19 de diciembre del 2024



ANEXO VIII

DECLARACIÓN JURADA DE ORIGINALIDAD DE TRABAJO DE INVESTIGACIÓN / TESIS / TRABAJO DE SUFICIENCIA PROFESIONAL

Yo, Raul Donato Cruz Huacac ,
bachiller de la Escuela Profesional de:
Ingeniería de Sistemas e Informática ,
de la Universidad Nacional de Moquegua, identificado(a) con DNI Nº. 41014085 , autor(a) del Trabajo de Investigación / Tesis / Trabajo de suficiencia profesional denominado:
PREDICCIÓN DIARIA DE RADIACIÓN SOLAR EN LA PROVINCIA DE ILO CON MODELO TRANSFORMER

Declaro bajo juramento que:

1. El trabajo de investigación es auténtico, siendo resultado de mi trabajo personal, que no se ha copiado, que no se ha utilizado ideas, formulaciones, citas integrales e ilustraciones diversas, sacadas de cualquier tesis, obra, artículo, memoria, etc., (en versión digital o impresa), sin mencionar de forma clara y exacta su origen o autor, tanto en el cuerpo del texto, figuras, cuadros, tablas u otros que tengan derechos de autor. Por tanto, el trabajo de investigación no ha sido plagiado total ni parcialmente.
2. El trabajo de investigación no ha sido autoplagiada, es decir, no ha sido publicada ni presentada anteriormente para obtener algún grado académico previo o título profesional.
3. Los datos presentados en los resultados son reales, no han sido falseados, ni duplicados, ni copiados y por tanto los resultados que se presentan en el trabajo de investigación constituirán aporte a la realidad investigada.

Soy consciente de que, el hecho de no respetar los derechos de autor y hacer plagio, es objeto de sanciones universitarias y/o legales, por lo que asumo cualquier responsabilidad que pudiera derivarse de irregularidades en el trabajo de investigación, así como de los derechos sobre la obra presentada.

Moquegua, 19 de diciembre del 2024



Nombres y Apellidos: Raul Donato Cruz Huacac
DNI: 41014085

Huella digital

CERTIFICADO DE ANÁLISIS
magister

Borrador Tesis - Raul Donato Cruz

2% Textos sospechosos

0% Similitudes
0% similitudes entre comillas
0% entre las fuentes mencionadas
2% Idiomas no reconocidos

Nombre del documento: Borrador Tesis - Raul Donato Cruz.pdf
ID del documento: 4e11105e7ecbfe5bf1df78cb0ead54b111f063c0
Tamaño del documento original: 2,05 MB

Depositante: HUGO EULER TITO CHURA
Fecha de depósito: 10/7/2024
Tipo de carga: interface
fecha de fin de análisis: 10/7/2024

Número de palabras: 16.449
Número de caracteres: 84.999

Ubicación de las similitudes en el documento:

Fuente considerada como idéntica

N°	Descripciones	Similitudes	Ubicaciones	Datos adicionales
1	**Borrador Tesis - Raul Donato Cruz.pdf** | Borrador Tesis - Raul Donato Cruz #e121a9 El documento proviene de mi biblioteca de referencias 1 fuente similar	100%		Palabras idénticas: 100% (16.449 palabras)

Fuentes principales detectadas

N°	Descripciones	Similitudes	Ubicaciones	Datos adicionales
1	repositorio.uchile.cl https://repositorio.uchile.cl/bitstream/2250/198592/1/Prediccion-de-curvas-de-luz-de-supernovas-m... 106 fuentes similares	3%		Palabras idénticas: 3% (1282 palabras)
2	openaccess.uoc.edu https://openaccess.uoc.edu/bitstream/10609/147433/4/javicorralesTFM0123memoria.pdf 103 fuentes similares	3%		Palabras idénticas: 3% (1180 palabras)
3	addi.ehu.es http://addi.ehu.es/bitstream/10810/53945/2/TFM.pdf 98 fuentes similares	3%		Palabras idénticas: 3% (1179 palabras)
4	idus.us.es https://idus.us.es/bitstream/11441/143923/1/jiménez Navarro, Manuel Jesús Tesis.pdf 92 fuentes similares	3%		Palabras idénticas: 3% (1107 palabras)
5	uvadoc.uva.es https://uvadoc.uva.es/bitstream/10324/57345/1/TFM-G1631.pdf 92 fuentes similares	3%		Palabras idénticas: 3% (1132 palabras)

Fuentes con similitudes fortuitas

N°	Descripciones	Similitudes	Ubicaciones	Datos adicionales
1	Documento de otro usuario #4d1322 El documento proviene de otro grupo	< 1%		Palabras idénticas: < 1% (20 palabras)
2	repositorio.unam.edu.pe https://repositorio.unam.edu.pe/server/api/core/bitstreams/8d3bd616-251b-49bd-b415-ae651e4c1...	< 1%		Palabras idénticas: < 1% (21 palabras)
3	link.springer.com | Forecasting approaches in a higher education setting | Educat... https://link.springer.com/article/10.1007/s10639-021-10684-z	< 1%		Palabras idénticas: < 1% (20 palabras)
4	fastercapital.com | Analisis de series temporales exploracion del vinculo entre el ... http://fastercapital.com/es/contenido/Analisis-de-series-temporales--exploracion-del-vinculo-entre-...	< 1%		Palabras idénticas: < 1% (16 palabras)
5	habr.com | Анализ временных рядов, применение нейросетей (1 часть) / Хабр https://habr.com/ru/articles/693562/	< 1%		Palabras idénticas: < 1% (11 palabras)

Fuentes mencionadas (sin similitudes detectadas)

Estas fuentes han sido citadas en el documento sin encontrar similitudes.

1 https://power.larc.nasa.gov/data-access-viewer/
2 https://doi.org/10.48550/arXiv.2302.07730
3 https://www.analyticsvidhya.com/blog/2018/04/fundamentals-deep-learning-regularization
4 http://www.ijert.org
5 http://www.essalud.gob.pe/downloads/ceprit/JULIO

DECENIO DE LA IGUALDAD DE OPORTUNIDADES PARA MUJERES Y HOMBRES
"AÑO DE LA RECUPERACIÓN Y CONSOLIDACIÓN DE LA ECONOMÍA PERUANA"

RESOLUCIÓN DE CONSEJO UNIVERSITARIO N° 136-2025-CU-UNAM

Moquegua, 20 de marzo de 2025

VISTOS:

El Acuerdo de Sesión Ordinaria de Consejo Universitario de fecha 19 de marzo de 2025; Proveído N° 872-2025 de fecha 14 de marzo de 2025 emitido por el Rectorado; Informe N° 124-2025-UNAM-R/SG de fecha 14 de marzo de 2025 emitido por la Secretaría General; Carta N° 56-2025-UNAM/SEGE/GYT/SVR con fecha de recepción 14 de marzo de 2025 emitido por el Área de Grados y Títulos; Sobre el otorgamiento del Título Profesional de Ingeniero de Sistemas e Informática, y;

CONSIDERANDO:

Que, la Constitución Política del Perú en su artículo 18° prevé que cada Universidad es autónoma en su régimen normativo, de gobierno, académico, administrativo y económico, y el artículo 8° de la Ley N° 30220, Ley Universitaria, establece que la referida autonomía inherente a las Universidades, se ejerce de conformidad con la Constitución y las Leyes de la República e implica las organizaciones de su sistema académico, económico y administrativo, así como la administración de sus bienes y rentas, elaborar su presupuesto y aplicar sus fondos con la responsabilidad que impone la Ley;

Que, el artículo 58° de la Ley Universitaria, Ley N° 30220, concordado con el artículo 14° del Estatuto de la Universidad Nacional de Moquegua, aprobado con Resolución de Comisión Organizadora N° 578-2021-UNAM, establecen que el **Consejo Universitario es el máximo órgano de gestión, dirección y ejecución académica y administrativa de la universidad;**

Que, el artículo 44° de la Ley N° 30220, Ley Universitaria, señala que las universidades otorgan los grados académicos de Bachiller, Maestro, Doctor y los Títulos Profesionales que correspondan, a nombre de la Nación (...);

Que, el numeral 45.2 del artículo 45° de la Ley N° 30220, Ley Universitaria, precisa **los requisitos mínimos para la obtención del título profesional, como son: el grado de Bachiller y la aprobación de una tesis o trabajo de suficiencia profesional**. Asimismo, establece que el título profesional solo se puede obtener en la universidad en la cual se haya obtenido el grado de bachiller;

Que, con Resolución de Comisión Organizadora N° 1153-2021-UNAM, se aprobó el Reglamento de Grados y Títulos de la Universidad Nacional de Moquegua, el mismo que fue modificado a través de las Resoluciones de Comisión Organizadora N° 410, 453 y 890-2022; N° 880-2023 y N° 081, 484-2024-UNAM/CO, el cual establece los requisitos y norma los procedimientos para la obtención de los grados académicos y títulos profesionales conferidos por esta Casa Superior de Estudios;

Que, mediante Carta N°056-2025-UNAM/SEGE/GYT/SVR, con fecha de recepción 14 de marzo de 2025, el Área de Grados y Títulos, remite a Secretaria General, los expedientes de dos (2) bachilleres de la Escuela Profesional de Ingeniería de Sistemas e Informática, quienes se encuentran aptos para la expedición del Título Profesional de Ingeniero de Sistemas e Informática por la modalidad de sustentación de Tesis, al haber cumplido satisfactoriamente con los requisitos legales y reglamentarios[1], exigidos en el Reglamento de Grados y Títulos de la Universidad Nacional de

[1] Artículo 122°. - Requisitos para la obtención del Título Profesional
Para obtener el Título Profesional el interesado deberá presentar ante su Escuela Profesional un expediente con la siguiente documentación:
a. Solicitud dirigida al Director de la Escuela Profesional o quien haga sus veces, indicando la fecha y número de comprobante de pago.
b. Dos (2) fotografías de frente, tamaño pasaporte a color con fondo blanco (con traje formal); para garantizar la seguridad ciudadana.

Moquegua, conforme lo señala la Dirección de la Escuela Profesional de Ingeniería de Sistemas e Informática, razón por la cual la Secretaría General incluye el expediente como punto de agenda para ser tratado en Sesión de Consejo Universitario de fecha 19 de marzo de 2025;

Que, se ha establecido como atribución del Consejo Universitario de la UNAM, la facultad de conferir los grados académicos y los títulos profesionales (...), por lo que en mérito al pedido del Secretario General y debatido el caso en Sesión Ordinaria de fecha 19 de marzo de 2025, se acordó por unanimidad **conferir a nombre de la Nación el Título Profesional de Ingeniero de Sistemas e Informática a dos (2) bachilleres de la Escuela Profesional de Ingeniería de Sistemas e Informática de la Facultad de Ingeniería y Arquitectura de la Universidad Nacional de Moquegua;**

De conformidad con el artículo 59° de la Ley Universitaria, Ley N° 30220, concordado con el artículo 16° del Estatuto de la Universidad Nacional de Moquegua, y estando a lo acordado en la Sesión Ordinaria de fecha 19 de marzo de 2025 de Consejo Universitario;

SE RESUELVE:

ARTÍCULO PRIMERO: CONFERIR, a nombre de la Nación el **TÍTULO PROFESIONAL DE INGENIERO DE SISTEMAS E INFÓRMATICA** por la modalidad de Sustentación de Tesis, a los bachilleres de la Escuela Profesional de Ingeniería de Sistemas e Informática de la Facultad de Ingeniería y Arquitectura de la Universidad Nacional de Moquegua, de acuerdo al siguiente detalle:

1. **RAUL DONATO CRUZ HUACAC**

   TESIS : "PREDICCIÓN DIARIA DE RADIACIÓN SOLAR EN LA PROVINCIA DE ILO CON MODELO TRANSFORMER", cuyo Asesor fue el Dr. Hugo Euler Tito Chura.

2. **ESTRELLA DE LOS ANGELES VELASQUEZ ZUÑIGA**

   TESIS : "DESARROLLO DE UN PROTOTIPO DE CHATBOT BASADO EN REDES NEURONALES RECURRENTES PARA EL PROCESO DE ADMISIÓN DE LA UNIVERSIDAD NACIONAL DE MOQUEGUA", cuyo Asesor fue el Ing. Honorio Apaza Alanoca.

ARTÍCULO SEGUNDO: AUTORIZAR, a la Secretaría General de la Universidad Nacional de Moquegua, expedir y registrar los Diplomas correspondientes para su otorgamiento.

ARTÍCULO TERCERO: ENCARGAR al Área de Grados y Títulos de la Universidad Nacional de Moquegua, para que realice las acciones pertinentes conducentes, relacionadas con la inscripción de los grados y títulos en el Registro de Grados y Títulos de la Superintendencia Nacional de Educación Superior Universitaria - SUNEDU.

ARTÍCULO CUARTO: NOTIFICAR, la presente Resolución a las instancias y dependencias administrativas y académicas de la Universidad Nacional de Moquegua, para su conocimiento, cumplimiento y demás fines pertinentes.

---

c. Dos ejemplares de la Tesis, del Trabajo de Suficiencia Profesional o Informe del Examen Profesional en físico, uno para la Biblioteca y otro para la Escuela Profesional, y un CD con la versión digital.
d. Formulario de autorización del graduado para la publicación de la Tesis, Trabajo de Suficiencia Profesional o Informe de Examen Profesional en el repositorio.
e. Declaración Jurada de originalidad de la Tesis, Trabajo de Suficiencia Profesional o Informe de Examen Profesional.

La Escuela Profesional es responsable de completar el expediente del bachiller en un plazo máximo de 3 días hábiles y derivar a la Oficina de Secretaría General, adjuntando los siguientes documentos:

a. Copia del acta de sustentación indicando la modalidad de obtención del título profesional.
b. Constancia de "no adeudo" de Bienes a la Universidad, bienes a los Laboratorios Básicos y de Especialidad, material bibliográfico; dichos documentos deberán ser refrendados por los responsables a cargo.

Domicilio Fiscal: Prolongación Av., Ancash S/N - Moquegua
RUC: 20449347448 - Teléfono: (053) 670168
Página Web: www.unam.edu.pe

///... **Resolución de Consejo Universitario N° 136-2025-CU-UNAM**

ARTÍCULO QUINTO: ENCARGAR, a la Oficina de Tecnología de la Información, la publicación de la presente Resolución en el portal de transparencia de la Universidad Nacional de Moquegua.

REGÍSTRESE, COMUNÍQUESE, CÚMPLASE Y ARCHÍVESE

UNIVERSIDAD NACIONAL DE MOQUEGUA UNAM RECTORADO

DR. HUGO RUBEN MARCA MAQUERA
Rector
Universidad Nacional de Moquegua

UNIVERSIDAD NACIONAL DE MOQUEGUA UNAM SECRETARÍA GENERAL

ABG. GUILLERMO S. KUONG CORNEJO
Secretario General
Universidad Nacional de Moquegua

Domicilio Fiscal: Prolongación Av., Ancash S/N - Moquegua
RUC: 20449347448 - Teléfono: (053) 670168
Página Web: www.unam.edu.pe