PROYECTO INDIVIDUAL

Presentado a

LA UNIVERSIDAD DE LOS ANDES

FACULTAD DE INGENIERÍA

DEPARTAMENTO DE INGENIERÍA MECÁNICA

Para obtener el título de

INGENIERO MECÁNICO

Por

Diego Alejandro Rojas Sánchez

Predicción de la radiación solar mediante modelos de forecasting estadísticos tradicionales y de machine learning con datos históricos de la planta solar del edificio Santo Domingo de la Universidad de los Andes.

Asesor

Andrés Leonardo Gonzales Mancera

Profesor, Universidad de los Andes

Índice

Índice de figuras

Índice de tablas

1. Introducción

El aumento de la implementación de energías renovables es el claro ejemplo de cómo la energía es un recurso vital del ser humano. Por lo cual, se necesitan cada vez más procesos de adquisición energética y distribución eficientes y optimizados para suplir las necesidades energéticas que se tienen en la actualidad.

Además, desde la idea de "Smart grids", donde cada usuario se puede volver su propio generador eléctricos, ha generado la necesidad de predecir el comportamiento de la red en diferentes regiones con condiciones específicas al alcance de cualquier persona. Por ello, la energía solar ha sido una de estas energías renovables más comunes para el alcance de este objetivo, ya que se ha vuelto de fácil acceso, fácil implementación y un proceso de comercialización más activo y técnico que apoya a todas las comunidades.

Por lo tanto, en busca de la optimización de las redes eléctricas, el campo de la gestión energética ha tenido un constante crecimiento en busca de herramientas cada vez más precisas de análisis de los flujos eléctricos. Así, los procesos de predicciones (Forecasting) de la irradiancia solar, la temperatura o la producción de la energía se vuelven relevantes.

Desde esta perspectiva, surge este proyecto con el objetivo de diseñar pronósticos en el tiempo de la irradiancia solar sobre el edificio santo domingo de la universidad de los andes como primer prototipo para la implementación futura en otras plantas solares. Sin embargo, estos proceso de pronósticos temporales ha sido un campo de investigación que ya ha se ha desarrollado desde tiempos pasados. Por lo tanto, se pretende acoger moleos ya implementados y crear modelos a base de la tecnología actual como es la inteligencia artificial con información de las plantas solares de la universidad. Esto con el fin, de comparar los métodos tradicionales y los métodos actuales para evidenciar sus ventajas y desventajas.

2. Objetivos

En busca de analizar los efectos de cada uno de los métodos tradicionales y de inteligencia artificial de forecasting se plantearon los siguientes objetivos:

2.1. Objetivo general

Generar modelos estadísticos y de machine learning para la estimación (Forecasting) de la irradiancia global sobre los paneles del edificio Santo Domingo de la Universidad de los Andes.

2.2. Objetivos específicos

- Identificar y seleccionar variables relevantes para la estimación de la irradiación solar, tales como la hora del día, la fecha, la ubicación geográfica, entre otras.

- Desarrollar modelos estadísticos para la estimación de la irradiancia global sobre los paneles del edificio Santo Domingo, utilizando técnicas como regresión lineal, regresión múltiple, series de tiempo, entre otras.

- Implementar modelos de machine learning para la estimación de la irradiancia global

- Validar los modelos desarrollados utilizando técnicas de validación cruzada y métricas de desempeño adecuadas, como el coeficiente de determinación (R2), el error cuadrático medio (MSE), el error absoluto medio (MAE), entre otros.

2.3. Alcance y productos finales

Al definir el objetivo general de este proyecto como la comparación de la calidad de la clasificación frente al modelo realizado, se establecieron los siguientes alcances del proyecto:

- Implementación y evaluación de los modelos de forecasting.
- Gráficas de los resultados de la función de pérdida frente a las iteraciones de la red neuronal.
- Tablas de las métricas alcanzadas por cada modelo.
- Parámetros usados por la red neuronal (Cantidad de capas, cantidad de neuronas, tipo de capas, tasa de aprendizaje, etc)
- Comparación final de los métodos generados.

3. Descripción de la problemática y justificación del trabajo

La energía eléctrica es una necesidad actual para la conformación y avance de la sociedad. Sin embargo, a causa del crecimiento exponencial de la población y el cambio climático es una nueva restricción. Por ello, la generación de energía se ve envuelta en un paradigma relacionado a la generación energética sin efectos contaminantes a la atmosfera.

Sin embargo, más del 80% de la energía consumida en el mundo proviene de generaciones contaminantes [1] y como nos comenta la UPME para Colombia “En la composición de la oferta interna 2021 no se observan cambios estructurales. Los fósiles mantienen su participación del 75,7% en 2020 y 2021. ” [2]siendo Colombia uno de los países con la red energética más limpia al depender mayormente de la generación hidráulica como se evidencia en la figura 1.

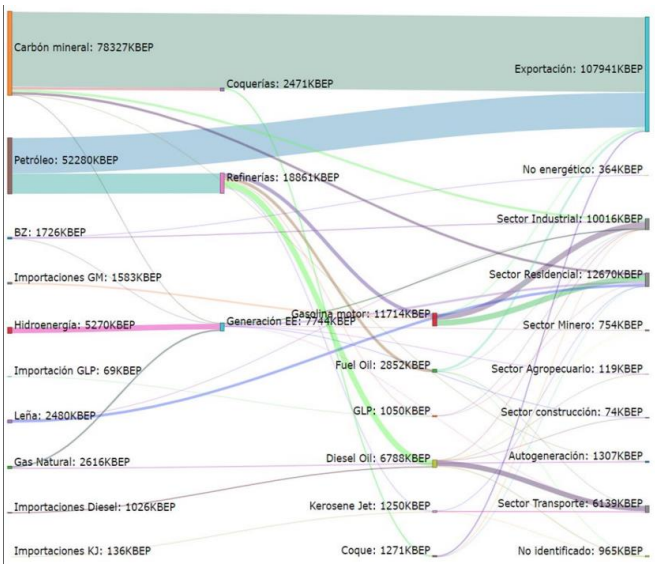

Figura 1. Balance energético colombiano [2]

Por lo anterior, los procesos de investigación relacionados a la generación de energía eléctrica están enfocados a la producción energética limpia y renovable. En este sentido, la energía solar y eólica han sido los procesos de mayor exploración en los últimos años. Sin embargo, estas energías tienen características y restricciones muy específicas que no dejan la evolución continua de la explotación energética. Por ejemplo, la energía eólica no puede ser producida en cualquier lugar sin fuentes claras de vientos.

Así mismo, la energía solar es una tecnología de baja eficiencia, por lo cual, es de gran importancia tener una optimización de la producción de energía, una gestión de la demanda energética y una planificación de la red eléctrica para la reducción de costos. Por lo tanto, los procesos de forecasting se vuelven relevantes en para el diseño de las plantas solares.

4. Marco teórico, conceptual e histórico

4.1. Marco Teórico y conceptual

4.1.1. Forecasting

El forecast consiste en la estimación y monitorización de variables futuras para un producto utilizando datos históricos. Por ello, desde un enfoque energético solar, el forecasting va desde las variables de entrada a los paneles como la irradiancia y la temperatura, como en la producción neta final [3] . Por lo tanto, el forecasting de la irradiancia puede tomar los siguientes enfoques: La optimización de la energía al conocer con anticipación la cantidad de energía que se puede esperar de los paneles solares y poder planificar mejor las operaciones en busca de tomar decisiones informadas sobre el uso de la energía producida [4].

La gestión de la demanda energética al predecir cuánta energía solar se generará en un momento dado con el forecasting de la irradiancia, lo que permite la distribución de la energía producida para gestionar la demanda energética [4]. Por ejemplo, si se espera una alta producción de energía solar, se puede programar en estos espacios los picos de alta demanda, lo que puede reducir el consumo de energía de la red.

La planificación de la red eléctrica se puede realizar mediante los resultados del forecasting de la irradiancia solar, pues, como nos explica la UPME, “este es un estudio técnico, cuya finalidad es proveer información objetiva que sirva de soporte a la toma de decisiones de inversión en infraestructura de abastecimiento energético y facilite la construcción de consensos sobre los proyectos y apuestas prioritarias del sector” [5]. Ya que, se permite anticipar los flujos de energía solar y planificar la distribución de la energía a los usuarios. Además, el forecasting de la irradiancia solar también es útil para evitar situaciones de congestión en la red eléctrica y planificar la expansión de esta [4].

Finalmente, el forecasting de la irradiancia solar también puede ayudar a reducir los costos de producción de energía solar. Pues, al conocer con anticipación la cantidad de energía solar que se generará, se pueden evitar situaciones en las que la generación de energía solar excede la demanda, lo que puede resultar en costos adicionales para la distribución de la energía [4].

4.1.2. Métodos tradicionales

Basado en el Review de forecasting de la irradiancia solar titulado "Review of solar irradiance forecasting methods and a proposition for small-scale insular grids" [6] se determinaron los 3 modelos de mayor uso para este propósito.

4.1.2.1. Media móvil integrada autorregresiva (ARIMA)

Este modelo se basa en un análisis estadístico que utiliza datos de series de tiempo para predecir tendencias futuras. Este modelo predice una serie de tiempo determinada en función de sus propios valores pasados, es decir, sus propios retrasos y los errores de pronóstico retrasados [7]. De esta manera, el modelo se describe por la ecuación 1, donde se evidencia los términos autorregresivos (p), un grado de diferenciación (d) y el orden del promedio móvil (q).

$$y'_t = d + \alpha_1 y'_{t-1} + \cdots + \alpha_p y'_{t-p} + e_t + \theta_1 e_{t-1} + \cdots + \theta_q e_{t-q} \quad [1]$$

Los anteriores atributos se relacionan a las características del modelo ARIMA que son la estacionalidad, un método autorregresivo y el promedio móvil. En primer lugar, la estacionalidad es el efecto de una observación que se disipa a medida que pasa el tiempo, por ello, una serie de tiempo se debe diferenciarse para volverse estacionaria [8].

En segundo lugar, Nogan explica que la característica autorregresiva "se define simplemente como el modelo lineal o no lineal entre el valor actual de la serie con sus valores anteriores (los llamados retrasos), y hay un número ilimitado de retrasos en el modelo. El supuesto básico de este modelo es que el valor actual de la serie depende de sus valores anteriores. Este es el modelo de memoria larga porque el efecto se disipa lentamente con el tiempo" [8]. Finalmente, la sección de promedio móvil identifica el error en el modelo de su valor actual en comparación con los valores anteriores.

De esta manera, el valor de p, d y q necesitan ser estimados con base a la base de datos del pasado de la irradiancia. Según Kotu, la forma más comunes de estimar dichos parámetros es a través de la Estimación de Máxima Verosimilitud. La cual, es similar a la estimación de mínimos cuadrados para la ecuación de regresión, excepto que MLE encuentra los coeficientes del modelo de tal manera que maximiza las posibilidades de encontrar los datos reales [9].

4.1.2.2. Suavización exponencial

El suavizado exponencial genera una predicción dando más importancia a los datos más nuevos y menos importancia a los datos más antiguos. Este método procura eliminar el ruido de los datos por lo cual, "suaviza" los resultados que permite que los patrones y las tendencias sean más visibles [10], cómo se puede evidenciar en la ecuación 2.

$$y_{t+1} = \alpha y_t + (1 - \alpha) * y_{t-1} \quad [2]$$

Donde, se busca entrenar el modelo hasta encontrar un valor de Alpha que mejor genere la predicción de 0 a 1. Así, el suavizado exponencial simple es un modelo de pronóstico que amplía el promedio móvil básico agregando pesos a los retrasos anteriores.

4.1.2.3. heteroscedástico condicional autorregresivo (ARCH)

Los modelos ARCH aparecen en los años 80 con el fin de agrupar los efectos de la volatilidad en los mercados financieros de le época. Por ello, su influencia ha crecido en el transcurso del tiempo, ya que la volatilidad de los comportamientos no solo financieros han sido el detonante de usar este método en un sinfín de campos de investigación [11].

Por ello, las principales características que debe tener un sistema para ser aplicado este modelo son: la ausencia de la estructura regular dinámica en la media, el exceso de curtosis, la simetría de los datos, agrupamiento de la volatilidad en intervalos de tiempos y la característica asimétrica de la volatilidad.

De esta manera, se creó el modelo que se puede expresar en la ecuación 3. Donde se tiene una dependencia entre la varianza de los datos pasados con la varianza del dato futuro y el valor esperado de estos. Así, la predicción depende del valor esperado y de los datos pasado cumpliendo la relación de la varianza.

$$y_t - \mu = \sum_{h=1}^{p} \phi_h (y_{t-h} - \mu) + \sum_{l=1}^{q} \nu V_{t-l} - V_t$$
$$\sigma^2 = \alpha_0 + \sum_{i=1}^{m} \alpha_i V_{t-i}^2 + \sum_{j=1}^{r} \beta_i \alpha_{t-j}^2 \qquad [3]$$

En donde, se hace uso de los datos pasados para entrenar y encontrar los valores de phi, Alpha y beta para generar una predicción posterior.

4.1.3. Inteligencia Artificial

La inteligencia artificial se crea bajo la premisa de generar maquinas que puedan suplir tareas que necesitan inteligencia humana bajo una lógica matemática. Por lo tanto, es una tecnología que tiene sus inicios aplicados en los años 50 con la definición del test de Turing creado por Alan Turing, el cual determina que si una inteligencia artificial, que se hiciera pasar por humano frente a otro humano, este no podría identificar si es humano o no. Así mismo, en el año 1957 se crea el perceptrón como la primera red neuronal artificial por Frank Rosenblatt y que abrió las puertas para el desarrollo de estas tecnologías año por año [12]. De esta manera, llega en los años 2020's el boom de la inteligencia artificial, donde se tienen asistentes en varios campos de la vida diaria de las personas como son DALLE para la generación de imágenes o Chat GPT como la revolución en los motores de búsqueda.

Las inteligencias artificiales están buscando simular ciertas características y capacidades cognitivas, como el razonamiento, el aprendizaje, la percepción, y la toma de decisiones, con el objetivo de automatizar y mejorar diversos aspectos de la vida cotidiana de los humanos. Por lo cual, se han creado una gran gama de campos de investigación como son

el aprendizaje de máquina, el Deep learning y el data science bajo idea de redes conectadas de neuronas que simulan las neuronas del cerebro humano. En la figura 2, se puede observar los diferentes proceso o caminos de investigación que se pueden tener en este campo de investigación de la inteligencia artificial.

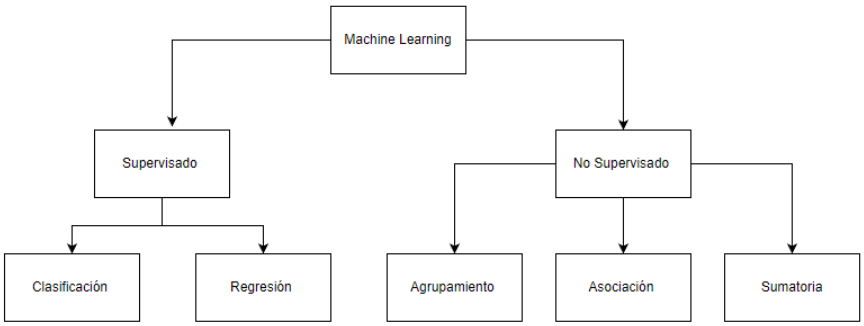

Figura 2. Métodos de inteligencia artificial.mp

Para comprender las diferencias anteriores, se debe entender que la inteligencia artificial se basa en 3 partes fundamentales. Una primera, es el entrenamiento de la red neuronal, donde con cierta información de entrada la red neuronal comprende las características particulares para generar una salida. De segundas, se encuentra la validación de la red, donde se utiliza la red entrenada, para introducirle una nueva información y observar si los resultados son los deseados generando porcentajes de error que se optimizan para reentrenar la red. Los dos pasos anteriores se realizan iterativamente hasta un punto de parada y no generar sobre ajustes de la red (donde la red no puede generalizar sus resultados a una nueva información de entrada desconocida). Finalmente se tiene un proceso de prueba, en el cual se testea el funcionamiento de la red entrenada y validad con información totalmente nueva para evidenciar si la red cumple o no su objetivo.

De esta manera, se entiende que un proceso supervisado como aquel entrenamiento en el que se tienen datos de entrada y datos de salida, donde la red entiende cuales son los resultados que debe entregar. Por el otro lado, se tienen los procesos no supervisados, en donde se entrena la red con datos de entrada, pero no se tienen datos de salidas. Por lo cual, la red se entrena basado en la idea de seguir caminos que probabilísticamente sean los que cumplan de mejor manera el objetivo final.

Todo lo anterior, se fundamenta basado en las estructuras de las redes neuronales, por lo cual existen dos métodos de gran importancia para la inteligencia artificial: las redes neuronales artificiales y las redes neuronales convolucionales.

4.1.3.1. Red neuronal Artificial (ANN)

Entender las redes neuronales artificiales, es entender como algoritmos de procesamiento que reconocen patrones para la resolución de problemas analíticos basados en entradas específicas de manera autónoma [13] da pie a la integración de la naturaleza con las máquinas para fundamentar este nuevo campo de estudio. Pues, su

unidad básica son las neuronas que descienden de la idea del comportamiento neuronal biológico. Las neuronas de las redes ANN como las neuronas biológicas, tiene información de entrada y genera una salida a partir de operaciones matemáticas generando combinaciones lineales.

La idea de red se basa en la forma como se interconectan dichas neuronas para el procesamiento de información robusto simulando la red interconectada del cerebro humano como se evidencia en la figura 3. Esta estructura, se basa en 3 partes: la capa de entrada, las capas ocultas y la capa de salida las cuales se componen de neuronas de esta jerarquía.

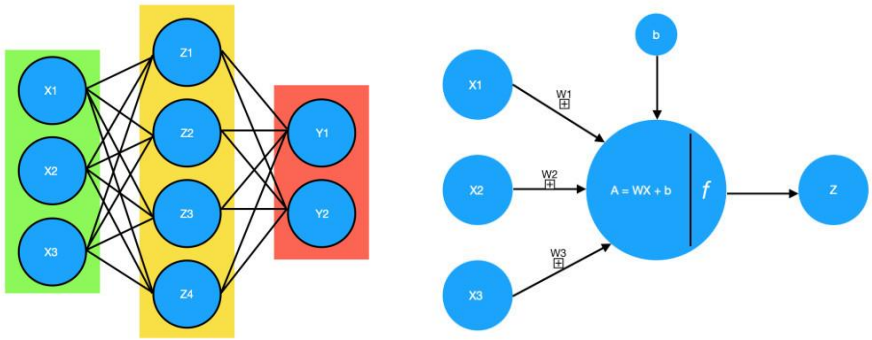

Figura 3. Estructura de una red neuronal [14]

La capa de entrada se basa de un conjunto de neuronas que reciben los datos de entrada de la red y contiene la cantidad de neuronas equivalentes a las dimensiones de dichos datos de entrada. Las capas ocultas, es un conjunto de neuronas que pueden conformar diferentes capas entre la capa de entrada y la de salida que procesan y transforman los datos de entrada. Finalmente, la capa de salida genera las predicciones deseadas de la red neuronal la cual representa el campo o las categorías de destino, donde una neurona representa una categoría.

Las capas de neuronas se conectan entre sí por los pesos numéricos que la red define. En general, una red aprende realizando cambios de los valores de los pesos hasta que se cumpla el criterio de parada predefinido. Este proceso de actualización de pesos es el que se realiza en el entrenamiento de la red, en donde se busca encontrar los valores ideales de estos. Sin embargo, estos valores son modelos lineales, los cuales siempre no son la mejor opción para generar procesos de predicción, de esta manera, se crea el concepto de bias y las funciones de activación. El bias es el corrimiento de esa función lineal que se acople mejor como entrada de la función de activación para generar el mejor resultado. Mientras que las funciones de activación son funciones no lineales a la suma ponderada de sus entradas. Esta función introduce la capacidad de las redes neuronales para aprender y modelar relaciones complejas en los datos [15].

Finalmente, una red neuronal artificial se compone del proceso de optimización que garantice los mejores valores de los pesos entre neuronas. Por ello, se tiene el concepto

de “forward propagation”, funciones de pérdidas y el “Backprogatation”. En primer lugar, el forward propagation es el paso de los datos de entrada por la red neuronal hasta obtener un resultado final. Este proceso, se basa en el cálculo matemático de cada neurona y función de activación que son transformados linealmente por los pesos de la red, los cuales inicialmente son aleatorios pero que después de la optimización se van reemplazando.

En segundo lugar, se tienen las funciones de pérdida, las cuales son un grupo de funciones matemáticas que comparan los resultados reales vs los resultados obtenidos por la red neuronal. Con esto, se obtiene un porcentaje de error de la red y con el cual se realiza el proceso de backpropagation. Este último concepto, es un método de optimización de redes neuronales que hace uso de funciones de optimización. El cual, genera una optimización de la función de pérdida con el objetivo de minimizar el error de salida de esta, de esta manera, se obtiene un valor óptimo de los pesos de la última capa para así empezar una propagación inversa. Esta propagación inversa, se enfoca en transformar los pesos de la red neuronal desde la capa final hasta la capa de entrada tomando las funciones inversas de activación transformando la entrada como salida de la neurona. Con lo anterior, se genera un proceso de entrenamiento y validación de la red eficiente.

4.1.3.2. Funciones de activación

Como se evidencio anteriormente, estas funciones son de gran importancia en las redes neuronales para generar resultados no linealizados y por lo tanto mejores. De esta manera, es de importancia especificar las principales funciones de activación y el objetivo principal que cumplen:

- Rectificador (ReLU):
  Esta función de activación es una de la más utilizadas debido a que permite generar un aprendizaje rápido en las redes neuronales a causa de su simplicidad matemática. Pues, esta función se mantiene los valores negativos de entrada en 0 y los valores positivos de entrada quedan igual como se puede evidenciar en la figura 4 y en la ecuación 4.

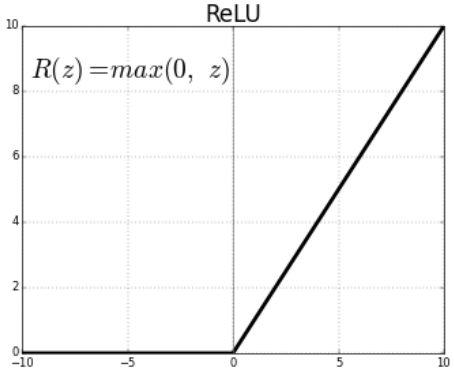

Figura 4. Función ReLu [16]

$$f(x) = \begin{cases} o, x < 0 \\ x, x > 0 \end{cases} \quad [4]$$

Esta función, genera una desactivación de neuronas al enviar valores a 0, por lo cual des complejiza el modelo diseñado. Sin embargo, este efecto de la red se puede determinar como un método de regulación de la red para mantener los resultados en rangos óptimos (Dropout's). Por lo cual, este efecto se debe tener en cuenta según el modelo de aplicación. A causa de este efecto, esta función ocasionalmente mejora los modelos que tienen imágenes como entradas a la red [17], pues, se descartan datos innecesarios para el aprendizaje de la red a causa de pixeles irrelevantes en las imágenes.

- Sigmoide

  Esta función se caracteriza por su desplazamiento en el eje y que mantiene las salidas en valores entre 0 y 1 como se observa en la figura 5. Esto genera, que los resultados entregados por esta función se distribuyan entre valores cercanos a 1 o cercanos a 0, por lo cual, se toma como una función de probabilidad.

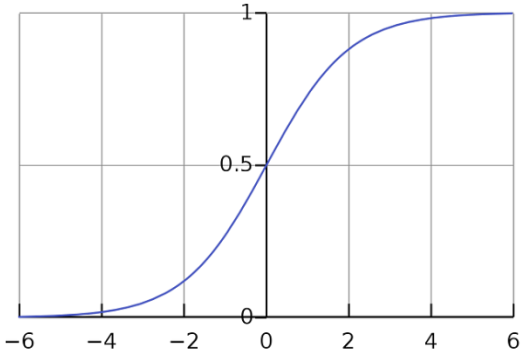

Figura 5. Función sigmoide [16]

$$f(x) = \frac{1}{1 - e^{-x}} \quad [5]$$

De esta manera, es una función tiene principal uso en la clasificación de clases binarias al generar probabilidades de pertenencia de una clase frente a la otra. Por lo tanto, es una función que se utiliza principalmente en las neuronas de la última capa, pues, al usarse en capas intermedias genera una saturación del gradiente que afecta el aprendizaje de una red interconectada [17].

4.1.3.3. Funciones de perdidas

Las funciones de pérdidas determinan la desviación de los resultados generados por la red frente a los resultados deseados. Por lo cual, son la función objetivo que tienen los métodos de optimización para minimizar dicha desviación y generar resultados cada vez mejores. Por lo tanto, estas funciones deben tener propiedades de derivación para generar los procesos de optimización sin indeterminaciones en el modelo.

- Error Cuadrático Medio ('Mean Square Error ', MSE)

Este método calcula la diferencia entre el valor real y el valor calculado de cada dato objetivo. De esta manera, se genera un promedio de dichas diferencias por cada dato objetivo y tener un error global del modelo como se evidencia en la ecuación 6, por lo tanto, es un método que se utiliza de gran medida para procesos de regresión [18].

$$MSE = \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n} \quad [6]$$

- Entropía cruzada Categórica ('Categorical Cross Entropy')
  Este método genera un cálculo entre las distancias de los valores reales y los valores calculados por medio de distribuciones de probabilidad como se evidencia en la ecuación 7. De esta manera, es una función que se utiliza para procesos de clasificación, donde la capa de laida entrega valores probabilísticos de cada clase como la función softmax [18].

$$CCE = -(y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)) \quad [7]$$

4.1.3.4. Funciones de optimización

- Descenso estocástico del gradiente (SGD)
  Como nos explica Guzmán: "Es una aproximación estocástica de la optimización del descenso de gradiente. Tiene como objetivo reducir el error de la neurona paso a paso hasta llegar a un error mínimo" [19]
- Adam
  Como nos explica Guzmán: "Es una combinación de la función SGD con momentum y la función de propagación de raíz cuadrática media" [19]

4.1.3.5. Métricas de evaluación

Para procesos de forecasting, se hacen uso métricas de regresión lineal que califica cuanto se equivocó el modelo:

- Error medio absoluto (EMA)

Esta métrica de regresión, que se observa en la ecuación 8, mide el valor medio de la diferencia absoluta entre el valor real y el valor predicho [20].

$$EMA = \frac{1}{n}\sum_{t=1}^{n}|y_t - y_t'| \quad [8]$$

- Error cuadrático medio (ECM)

El error cuadrático medio (ECM) de la ecuación 9, calcula el valor medio de la diferencia al cuadrado entre el valor real y el predicho para todos los puntos de datos. Este modelo, al ser una potencia cuadrada, el impacto de los errores es mayor. De esta manera, las ventajas del ECM frente al EMA [20]:

1. El ECM destaca grandes errores entre los pequeños.

2. El ECM es diferenciable, lo que ayuda a encontrar los valores mínimos y máximos utilizando los métodos matemáticos de manera más efectiva.

$$ECM = \frac{1}{n}\sum_{t=1}^{n}(y_t - y_t')^2 \quad [9]$$

- raíz del error cuadrático medio (RECM)

La raíz del ECM genera valores absolutos más pequeños, lo que es útil para los cálculos informáticos [20].

4.2. Marco Histórico

El deseo de generar predicciones a futuro ha sido una preocupación desde hace tiempo para los sistemas eléctricos. Por ello, modelos de forecasting presentados anteriormente como el ARIMA, la suavización exponencial y el ARCH se han creado desde los años 80's y 90's. Sin embargo, empezar la aplicación de estos métodos al diseño de sistemas fotovoltaicos es un campo de desarrollo moderno. De esta manera, se puede observar trabajos enfocados al desarrollo de modelos de forecasting de condiciones climáticas o de producción de las plantas solares generados en los últimos años. Como ejemplo, se tiene en el 2013 el libro de desarrollo del forecasting "Weather Modeling and Forecasting of PV systems operation" [21]en el cual se recopila una serie de métodos de forecasting que van desde el diseño de métodos estadísticos hasta métodos de inteligencia artificial. Sin embargo, los modelos se describen de manera teórica y no existe una aplicación especifica de estos métodos. Por ello, en el 2014 se empezó a realizar trabajos de redes neuronales con base a datos históricos de plantas solares para encontrar la potencia eléctrica de esta.

Así mismo, para simplificar lo evidenciado en el libro "Weather Modeling and Forecasting of PV systems operation" [21] y especificar desarrollos de forecasting a tiempos intra diarios, se construyó un review que resume los modelos más utilizados en dichas predicciones titulado "Review of solar irradiance forecasting methods and a proposition for small-scale insular grids" [6]. De esta manera, se empieza a observar caminos distintos de investigación relacionado a los objetivos deseados, por lo tanto, se comienza a tener modelos de forecasting intra diarios para evidenciar el comportamiento de una planta y poder gestionar de mejor manera la energía eléctrica. Y una segunda que es la proyección a rangos temporales lejanos para la planificación cada vez más realista de una planta solar. Así, empiezan a existir en el 2016, investigaciones relacionadas al efecto de los modelos de forecasting para la evolución de la energía solar, como es el informe del laboratorio nacional de energía renovable de estados unidos titulado "Pronóstico de generación de energía eólica y solar: mejorando la operación del sistema" [4].

Con todo lo anterior y como consecuencia de la evolución de las redes neuronales, la inteligencia artificial y el Deep learning. Se crean modelos de redes neuronales para medir no solo la producción de la planta solar, sino también modelos de predicción de variables meteorológicas como la temperatura y la radiación. Así, aparecen trabajos como "Predicción

de la energía generada por un panel solar basado en su temperatura y radiación" [22] en el 2019 y "Deep learning models for solar irradiance forecasting: A comprehensive review" [23] en el 2021. Con lo cual, se empiezan a generar investigaciones relacionados a evidenciar los efectos de cada tipo de modelo de redes neuronales como son modelos no supervisados en el trabajo de grado de Johanna Michelle Romero Rodríguez titulada "Modelo para predicción de potencia de paneles fotovoltaicos utilizando técnicas de clasificación no supervisada y redes neuronales artificiales" [24].

Esto demuestra, que este campo de investigación es de rápido avance, ya que en menos de una década se han generado una gran cantidad de proyectos e investigaciones en busca de diferentes formas de generar un forecasting cada vez más detallado y preciso. Así mismo, demuestra la necesidad que se tiene en la actualidad para tener modelos predictivos exactos y precisos con el objetivo de optimizar la distribución energética, gestionar la demanda, planificar redes eléctricas y lo más importante reducir costos de construcción, generación y distribución de las plantas solares.

5. Definición y especificación del trabajo

5.1. Definición

En consecuencia, de la necesidad de buscar modelos cada vez más precisos para el desarrollo de un a predicción a futuro de variables esenciales para la producción de una planta solar como es la radiación solar con la que se obtiene la energía eléctrica. Se estableció, un proyecto basado en la comparación de modelos tradicionales estadísticos de forecasting y modelos novedosos de machine learning basado en datos históricos de la radiación solar. Por lo cual, se tomo como base la planta solar del edificio santo domingo de la universidad de los Andes, para tener una adquisición de datos sencilla.

A partir de lo anterior, se estipularon métricas de evaluación de los modelos para generar una comparación argumentada, como fueron el error medio absoluto y la raíz del valor medio cuadrado. De esta manera, evidenciar las características de cada modelo y observar las aplicaciones en las que se pueden enfocar cada uno de los modelos.

5.2. Especificaciones

A partir de la intención del proyecto, se estipuló a partir de la búsqueda bibliográfica que los modelos más utilizados para el forecasting en diferentes campos de aplicación son: la media móvil integrada autorregresiva (ARIMA), la suavización exponencial y el modelo heteroscedástico condicional autorregresivo (ARCH). Además, se evidenciaron que las arquitecturas de redes neuronales utilizadas se conforman por pocas cantidades de capas ocultas para evitar así problemas de sobreajuste en la red.

Por ello, se evidenciaron restricciones para el desarrollo del proyecto como se pueden observar:

- Identificar que intervalo de tiempo es el ideal para el diseño de los modelos, ya que los datos pueden extraerse en intervalos de tiempo de cinco minutos, media hora, por horas o por día.

- Especificar los periodos de tiempos que se van a utilizar a consecuencia de la falta de información por parte de la base de datos entregada por la planta solar. Con la cual, no se puede utilizar una información continua desde el primer dato hasta el último.
- Capacidad computacional de los equipos a utilizar para el diseño del proyecto, ya que tradicionalmente los modelos de machine learning hacen uso de grandes cantidades de recursos computacionales.

6. Metodología de trabajo

Para el desarrollo del proyecto, se estipulo tres rutas principales de desarrollo: la investigación previa de los modelos de forecasting utilizados en los antecedentes, la adquisición de la base de datos y el preprocesamiento de los datos para ser manipulados en el entorno de Python y finalmente el diseño de los modelos de forecasting con su respectiva evaluación. Por lo tanto, en la figura 6 se puede observar el plan detallado que se llevo a cabo para el desarrollo del proyecto.

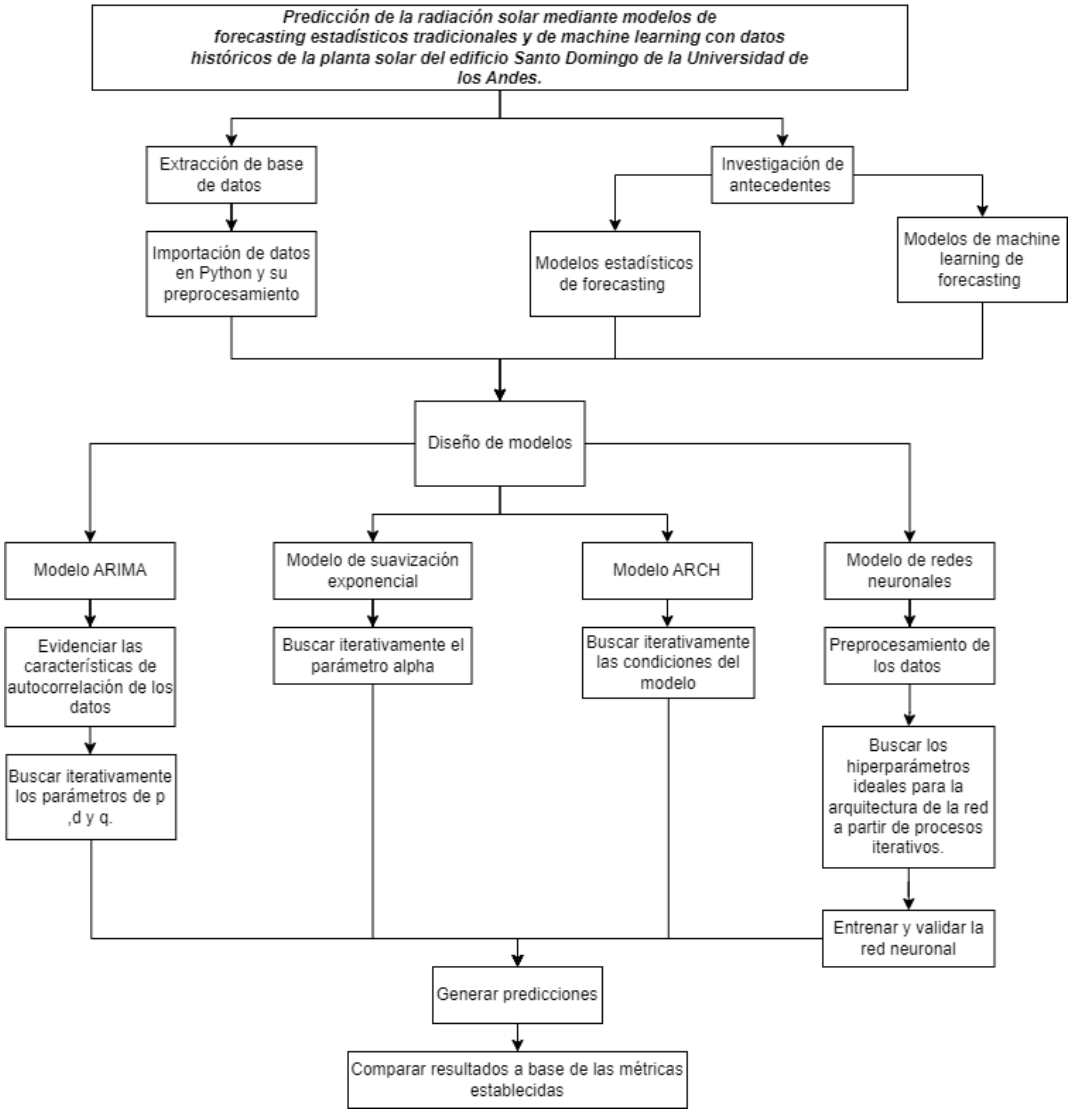

Figura 6. Metodología de trabajo

Con lo anterior, se puede evidenciar que el proyecto se manejó en un sistema paralelo, donde se diseñaron los 4 modelos a implementar de manera paralela basado en los datos previamente procesados para su uso en Python. De esta manera, se logró estipular métricas de evaluación constantemente e ir mejorando los modelos hasta encontrar predicciones de bajo error.

6.1. Análisis del contexto actual

Con el aumento de las demandas energéticas en el país y a partir de los proyectos de ampliación de la malla eléctrica colombiana con base en energías renovables de la UPME. El diseño de pronósticos sobre la irradiancia solar sobre los paneles solares se vuelve cada vez más relevantes. De esta manera, se empezó el proyecto en busca de entender de mejor manera estos procesos actuales de predicciones y las investigaciones que se estaban llevando a cabo sobre la radiación solar. Pues, se entiende que, a partir de esta variable, los paneles solares producen su potencia eléctrica que al no tener grandes cantidades de radiación solar la producción de la planta es ineficiente.

Con ello, se encontró el libro "Weather Modeling and Forecasting of PV systems operation" [21] y el "Review of solar irradiance forecasting methods and a proposition for small-scale insular grids" [6]. En los cuales, se evidenciaba una recolección de grandes cantidades de modelos con los que se podía realizar un forecasting y con los cuales se identificaron los modelos ARIMA, suavización exponencial y el ARCH como modelos de usos tradicionales. Así mismo, se identificaba que, en estos referentes, los modelos de machine learning ya se proyectaban como soluciones para este campo eléctrico.

Por lo cual, se evidencio los proyectos desarrollado en los últimos tiempos para observar el progreso de este campo. Con esto, se identificó que se han realizados modelos de forecasting de manera separada sin una comparación de los modelos. Por ello, se desarrolló el proyecto enfocado en la comparación de los modelos. De esa esta manera, buscar bases sólidas sobre los modelos a aplicar para ciertas aplicaciones o enfoques deseados de los proyectos solares futuros.

6.1.1. Métodos tradicionales

6.1.1.1. Media móvil integrada autorregresiva (ARIMA)

Para el desarrollo de la media móvil integrada autorregresiva (ARIMA) en busca del forecasting de la radiación solar a base de datos históricos se dividió en 2 partes el diseño del modelo. Primero, se evidencia la estacionalidad de los datos con el valor estadístico p-value y la autocorrelación de los datos de la radiación solar de la planta solar del edificio Santo Domingo. Segundo, se realiza la calibración de los parámetros del modelos (p, d y q) para realizar la correspondiente predicción.

Para ello, se utilizaron las librerías y las funciones que se observan en la tabla 1. Con las cuales, se importaron los datos, se procesaron, se analizaron y se utilizaron para generar predicciones futuras.

Tabla 1. Librerías y funciones de Python utilizadas para el diseño del modelo ARIMA

Librerías	Función	Descripción

Pandas	Read_excel(URL)	Importa el nombre, en una lista, de los archivos internos de la carpeta introducida en la URL
statsmodels	tsa.stattools.adfuller()	Prueba de Dickey-Fuller aumentada es una prueba de raíz unitaria para una muestra de una serie de tiempo para observar los valores de p-value, ADF Statistic y los Critical Values
	Api.sm.graphics.tsa.plot_acf()	Gráfica la función de autocorrelación de los datos (tipo dataframe) de entrada.
	Api.sm.tsa.arima.ARIMA	Genera simbólicamente el modelo ARIMA de los datos de entrada con los valores p,d y q de entrada
matplotlib	Pyplot.plt()	Función para graficar funciones
pmdarima	auto_arima()	Iterar parámetros p, d y q del modelo ARIMA para buscar el mínimo AIC
Scikit-learn	Fit()	Crea numéricamente el modelo ARIMA de entrada
	set_index()	Convertir una predicción en un dataframe
	forecast()	Predice la radiación futura la cantidad de periodos especificado con el modelo ARIMA creado previamente

Con lo anterior, se utilizó la función de pandas "Read_excel(URL)", en el cual se importó los datos de excel .xlsx en un tipo de dato DataFrame. Posteriormente, se realizó el análisis previo de los datos. En el cual, se busca el p-value con la función "tsa.stattools.adfuller()" para confirmar la estacionalidad de los datos. Cabe resaltar, que si este p-value es superior a 0.05, se debe diferenciar los datos restando los datos por los mismos datos con un corrimiento en el tiempo.

Posteriormente, se obtiene la función de autocorrelación de los datos con la función "Api.sm.graphics.tsa.plot_acf()". De esta manera, si la función de autocorrelación se aproxima a 0 rápidamente confirma la estacionalidad del modelo, ya que se tiene una dependencia entre datos pasados y futuros por cierta cantidad de tiempo definida Así, en dichos resultados se puede observan la cantidad de periodos de tiempo en el eje x y en el eje y el porcentaje de autocorrelación, por lo tanto al caer al 25% la dependencia de un periodo de tiempo específico, este ya no tiene una influencia significativa en periodos futuros. Esto ayuda a entender que el modelo depende de datos pasados, pero no es una dependencia eterna que no deje predecir datos futuros por la ausencia de estos.

Con lo anterior, se puede extraer el valor de q, el cual es la cantidad de periodos de datos pasados con dicha influencia significativa. Mientras que el valor de d, se obtiene basado en la cantidad de diferenciaciones que se genere. De esta manera, se itera el valor p para obtener el mejor valor de predicción con la función "Api.sm.tsa.arima.ARIMA()" y "Fit()" las cuales entregan un resumen del modelo y se observa el valor AIC (Criterio de información de Akaike) que se desea tener el mínimo

de este. Sin embargo, este proceso iterativo se puede realizar con la función "auto_arima()" en donde se especifican los rangos máximos y mínimos de iteración del modelo ARIMA con los datos de radiación solar. De esta manera, el modelo determina cual es la combinación de parámetros con el valor más pequeño de AIC.

Finalmente, se construye el modelo con las funciones "Api.sm.tsa.arima.ARIMA()" y "Fit()" con un DataFrame de entrada de los datos pasados y se hace uso de la función "forecast(k)" donde se especifica la cantidad k de periodos de tiempos futuros que se desean predecir. De esta manera, se obtiene una predicción de k periodos en el futuro basados en los datos pasados de entrada. Por lo anterior, para evidenciar la generalización del modelos, los datos se dividieron en 2 partes, una primera para calibrar el modelo con el que se encontraba los valores de p-d-q y una segunda para usarlos como datos pasados para la predicción futura.

6.1.1.2. Suavización exponencial

Para el desarrollo del modelo de suavización exponencial en busca del forecasting de la radiación solar a base de datos históricos se utilizaron las librerías y las funciones que se observan en la tabla 2. Con las cuales, se importaron los datos, se procesaron y se utilizaron para generar predicciones futuras.

Tabla 2. Librerías y funciones de Python utilizadas para el diseño del modelo suavización exponencial

Librerías	Función	Descripción
Pandas	Read_excel(URL)	Importa el nombre, en una lista, de los archivos internos de la carpeta introducida en la URL
statsmodels	tsa.api.SimpleExpSmoothing()	Genera simbólicamente el modelo de suavización exponencial de los datos de entrada con los valores p,d y q de entrada
matplotlib	Pyplot.plt()	Función para graficar funciones
Scikit-learn	Fit()	Crea numéricamente el modelo de suavización exponencial de entrada
	set_index()	Convertir una predicción en un dataframe y ser graficado
	forecast()	Predice la radiación futura la cantidad de periodos especificado con el modelo de suavización exponencial creado previamente

Con lo anterior, se utilizó la función de pandas "Read_excel(URL)", en el cual se importó los datos de excel .xlsx en un tipo de dato DataFrame. Posteriormente, se realizó el análisis previo de los datos. De esta manera, se hace uso de la función "tsa.api.SimpleExpSmoothing()" para crear el modelo con los datos de entrada y la función "Fit()" con el valor Alpha que se va a utilizar para crear el modelo numéricamente, además, se hace uso de la función "forecast(k)" donde se especifica la cantidad k de periodos de tiempos futuros que se desean predecir. Por ello, para

encontrar el mejor valor Alpha, se itera desde 0 hasta 1 de a pasos de 0.01 para observar que valor es el mejor. Para determinar qué modelo es mejor que el otro, se hace uso del error medio absoluto para generar dicha comparación entre modelos.

6.1.1.3. heteroscedástico condicional autorregresivo (ARCH)

Para el desarrollo del modelo de suavización exponencial en busca del forecasting de la radiación solar a base de datos históricos se utilizaron las librerías y las funciones que se observan en la tabla 2. Con las cuales, se importaron los datos, se procesaron y se utilizaron para generar predicciones futuras.

Tabla 3. Librerías y funciones de Python utilizadas para el diseño del modelo suavización exponencial

Librerías	Función	Descripción
Pandas	Read_excel(URL)	Importa el nombre, en una lista, de los archivos internos de la carpeta introducida en la URL
arch_model	arch_model()	Genera simbólicamente el modelo ARCH de los datos de entrada con los valores p,d y q de entrada
matplotlib	Pyplot.plt()	Crea numéricamente el modelo ARCH de entrada
Scikit-learn	Fit()	Convertir una predicción en un dataframe y ser graficado
	set_index()	Función para graficar funciones
	forecast()	Predice la radiación futura la cantidad de periodos especificado con el modelo ARCH creado previamente

Con lo anterior, se utilizó la función de pandas “Read_excel(URL)”, en el cual se importó los datos de excel .xlsx en un tipo de dato DataFrame. Posteriormente, se realizó el análisis previo de los datos. De esta manera, se hace uso de la función “arch_model()” para crear el modelo con los datos de entrada y la función “Fit()”, además, se hace uso de la función “forecast(k)” donde se especifica la cantidad k de periodos de tiempos futuros que se desean predecir. En comparación a los modelos anteriores, la función “arch_model()” itera sus parámetros internos para encontrar a criterio propio la mejor combinación y entregar una predicción correspondiente.

6.1.2. Red Neuronal Artificial

Para el desarrollo de la red neuronal artificial para el forecasting de la radiación solar a base de datos históricos, se realizó la misma metodología de desarrollo. La cual consistía en preprocesar los datos, diseñar la red ANN y evaluar el resultado final. Para ello, se utilizaron las librerías y las funciones que se observan en la tabla 4.

Tabla 4. Librerías y funciones utilizadas para la construcción de la red artificial

Librerías	Función	Descripción
Pandas	Read_excel(URL)	Importa el nombre, en una lista, de los archivos internos de la carpeta introducida en la URL
	.iloc()	Localizar un dato en específico en el DataFrame

	.pop	Recortar el DataFrame entre rangos especificos
sklearn	Preprocessing. MinMaxScaler()	Escala los valores de un DataFrame en valores entre 0 y 1
Numpy	Reshape()	Redimensiona un vector a las dimensiones introducidas
	Save()	Guarda la estructura de la red en un archivo .h5
	Save_weights	Guarda los pesos de la red entrenada en un archivo .h5
Keras	Models. Sequential()	Crea una red neuronal base
	Models.add()	Adhiere una nueva capa a la red
	Layers.Dense()	Importa una capa densa a la red
	Layers.Dropout()	Importa una capa de regularización Dropout a la red
	Models.compile()	Crea el modelo optimizador
	Categorical_crossentropy	Determina que la función de perdidas es una función categorical cross entropy
Tensorflow.keras	Optimizer.Adam	Determina como optimizador, el método de optimización Adam
	metrics	Determina las métricas a optimizar
Scikit-learn	Fit()	Entrena la red
	Predict()	Predice la clase que pertenece la imagen de entrada según el entrenamiento de la red

Con lo anterior, se utilizó la función de pandas “Read_excel(URL)”, en el cual se importó los datos de excel .xlsx en un tipo de dato DataFrame. Posteriormente, se realizó el análisis previo de los datos, en donde se modelo el comportamiento del concepto día y noche con una función seno(2*pi/dia) y cos(2*pi/día) donde “dia” son los segundos de un día y la posición del mes del año con estas mismas funciones, pero con los segundos de un año. Estas expresiones obtenidas bajo una simplificación para el cálculo de la posición solar frente a un punto de referencia terrestre se realizan con el fin de darle más información al modelo, el cual, relacione el instante temporal con una posición específica del día y del año.

Así mismo, se realiza una selección de los datos que se van a utilizar a causa de la no continuidad de la base de datos de la planta solar. Para así, realizar un escalamiento de los datos entre 0 y 1 con la función “MinMaxScaler()”, con el fin de facilitar el proceso de entrenamiento y evitar gastos de recursos computacionales.

Con lo anterior, se generan los pares de datos para realizar una red neuronal de aprendizaje supervisado. En el cual, se impone una variable que va a ser la cantidad de días que se desean predecir en términos del rango temporal de los datos, por ejemplo, si los datos son de cada 5 minutos; esta variable se compondrá por 12 intervalos que completan una hora por 24 horas de un día por la cantidad de días a predecir. Con esta variable, se realiza el

proceso iterativo de crear un vector x_train y un vector y_train, en el cual, el DataFrame de la base de datos se va partiendo y guardando en la variable x_train desde la variable del ciclo hasta el periodo de tiempo especificado con la variable establecida anteriormente. Así mismo, con el DataFrame de únicamente de los datos de radiación solar, se guarda el valor de la posición que la variable me especifica en el vector y_train. Este proceso se realiza de la misma manera para crear los vectores de validación y de testeo del modelo, por ello, en un comienzo se debe partir en 3 los datos totales de la planta solar en un porcentaje de 80-10-10 para entrenamiento, validación y testeo respectivamente.

A partir de los vectores creados anteriormente, se empieza a crear la red neuronal artificial iniciando con la función "models.Sequential()" que establece la construcción de la red neuronal base. Posteriormente, se pone la primera capa, la capa de entrada con la función "model.add(layers.Dense())". Cabe resaltar, que se observó mejor resultado, como capa de entrada, una capa densa que una capa LSTM (que en registros teóricos también se utiliza como capa de entrada). Para la capa de entrada, se introducía un parámetro con las dimensiones del vector.

Posteriormente, se construía una serie de capas ocultas con la función "model.add(layers.Dense())", donde se especificaba una función de activación tipo "relu" que no demandaba tanto recurso computacional. Finalmente, la capa de salida se construye con la función: "Layers.Dense(1)" al tener un único valor de salida que es la predicción futura.

Al terminar la estructura de la red, se crea el optimizador del modelo a través de la función "model.compile()" en el cual, se introduce como parámetros la función de pérdida (mean_squared_error), el optimizador Adam que evidencio mejores resultados y la métrica a optimizar accuracy.

Con lo anterior, se utiliza la función "model.fit()" para empezar el proceso de entrenamiento y el cálculo de los pesos de la red. Al terminar el entrenamiento, el modelo se puede guardar en una carpeta con la función "model.save(URL+Nombre_archivo)" y "model.save_weights(URL+Nombre_archivos_pesos)". Con esto, se importa en un nuevo script el modelo a través de las funciones "load_model(URL_modelo)" y "cnn.load_weights(URL_pesos_modelo)".

Por medio de este entrenamiento, se empieza el proceso de evaluación del modelo con el vector de test y la función "predict(x_test)" que predice el valor siguiente de cada sub vector que se encuentra en el vector x_test. Por medio de los resultados obtenidos, se va generando una evaluación de los resultados, con las predicciones y el vector test al calcular el error medio absoluto y la raíz del valor medio cuadrático.

El diseño de la red neuronal partió de un proceso iterativo en el cual se construía una red, se evaluaba y se testeaba, en donde, se buscaba tener las mejores métricas de evaluación. Por lo cual, el proceso que se realizó fue la iteración de la cantidad de neuronas de las capas ocultas en potencias de 2 para mantener un recurso computacional menor al poder discretizar la información en bytes. Este proceso, se realizó desde 4 neuronas hasta 16 384

neuronas, en los cuales se calculaban las métricas de evaluación para evidenciar cual de estos valores era el mejor. Posteriormente, se elegia el mejor valor como parámetro de la capa y se repetía el proceso incluyendo una nueva capa e iterándola. De esta manera, se fue introduciendo capa tras capa con el número de neuronas óptimo que garantizaba métricas de altos valores hasta obtener los mínimos errores. Este proceso terminaba, cuando no se veía aumento en el valor de las métricas o se generaba una disminución de estas.

Cabe resaltar, que el proceso anterior, se realizaba con valores comunes de batch size, tasa de aprendizaje y optimizador. Sin embargo, al obtener la red, se realizaba un proceso iterativo para encontrar los valores que mejor se acoplaban a las redes.

7. Trabajo realizado

7.1. Descripción del resultado final

Los resultados generados se basan principalmente en el diseño de los modelos de forecasting estadísticos y los modelos de redes neuronales. Para los modelos computacionales, el proceso de entrenamiento y validación del funcionamiento del modelo, se realizaron en un computador Dell de 64 GB de memoria RAM, un disco duro de estado sólido de 1 TB de espacio y un procesador core i7.

7.1.1. Métodos tradicionales

7.1.1.1. Media móvil integrada autorregresiva (ARIMA)

En primer lugar, se evidencio la estacionalidad de los datos para observar si estos son predecibles o no. Por lo cual, se busca estadísticamente el valor del p-value para afirmar o rechazar la hipótesis de estacionalidad de los datos. Por lo tanto, al obtener un valor p-value de 0.0001 se confirma la estacionalidad de los datos y no se deben diferenciar para buscar dicha estacionalidad, de esta manera, el valor d es igual a 0. Así mismo, se evidencia la autocorrelación de los datos en la figura 7.

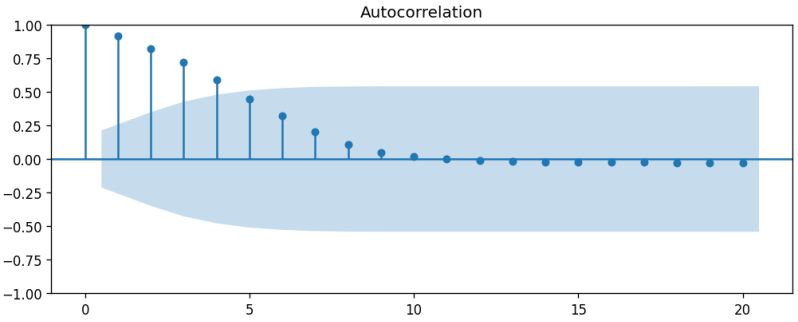

Figura 7. Función de autocorrelación de los datos de radiación solar sobre la planta solar del edificio Santo Domingo

Con lo anterior, se puede observar la cantidad de datos pasados de los cuales el dato futuro depende. Por ello, cuando la dependencia baja del 25%, ya estos datos no se deben tener en cuenta en el modelo, por lo cual, se puede seleccionar los valores de q en los que se va a iterar el modelo.

A partir de la función "pm.auto_arima()" se encontró que el mejor modelo ARIMA es con p= 3, q=1 y d=0. De esta manera, se realizó el testeo del modelo con los datos seleccionados para tal fin, el cual cuenta con 2 días (un día para predecir el siguiente). Con lo cual, se observó que al predecir el día 2 completo a partir del día 1, el modelo no generaba ninguna aproximación y tendía a generar una función logarítmica. Por ello, se realizó un proceso iterativo, en el cual, se predecía solo un intervalo de tiempo en el futuro y se guardaba dicha predicción en un vector. Así, para el siguiente intervalo de tiempo, se utilizaba los datos del día 1 más el valor del sensor de radiación solar del intervalo predicho anteriormente como datos pasados para predecir este nuevo intervalo futuro. De esta manera, se iba generando en cada iteración la predicción futura de un intervalo de tiempo usando los datos inmediatamente pasados de la radiación solar de la planta solar.

Con esto en mente, se quiso evidenciar la capacidad de predicción del modelo para poder observar cuantos intervalos de tiempo podía predecir sin la necesidad de utilizar los datos pasados inmediatos. Con lo cual, en la figura 8 se observa la tendencia del error absoluto medio del modelo ARIMA (3,1,0) frente a los intervalos de tiempo que está prediciendo.

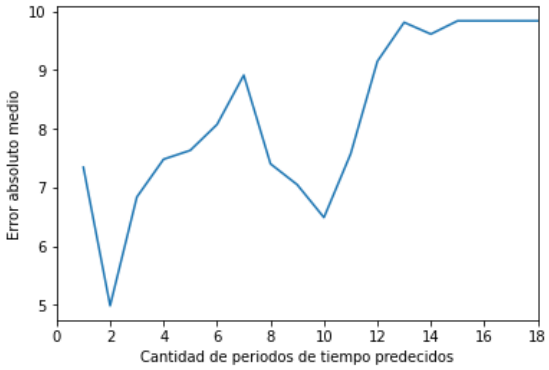

Figura 8. Tendencia del EMA frente a los intervalos de tiempo de la predicción del modelo ARIMA

Con lo anterior, se puede observar que el modelo tiene el mínimo error, cuando se predice 2 intervalos de tiempo futuros. De esta manera, se observa que este modelo es un modelo de corta memoria y necesita ser retroalimentado constantemente (cada dos periodos de tiempo).

Con lo cual, para entrar a los códigos de desarrollo, se puede ingresar al siguiente link:

https://github.com/DiegoRojasXD/Codigos_Tesis_IMEC/tree/main/ARIMA

7.1.1.2. Suavización exponencial

En segundo lugar, se hace uso del proceso iterativo para observar que con una relación Alpha de 0.6 genera una predicción de bajo error con una equivalencia entre la predicción anterior y el valor real anterior.

Con esto en mente, se quiso evidenciar la capacidad de predicción del modelo al igual que el modelo ARIMA y poder observar cuantos intervalos de tiempo podía predecir sin la necesidad de utilizar los datos pasados inmediatos. Con lo cual, en la figura 9 se observa la tendencia del error absoluto medio del modelo frente a los intervalos de tiempo que está prediciendo.

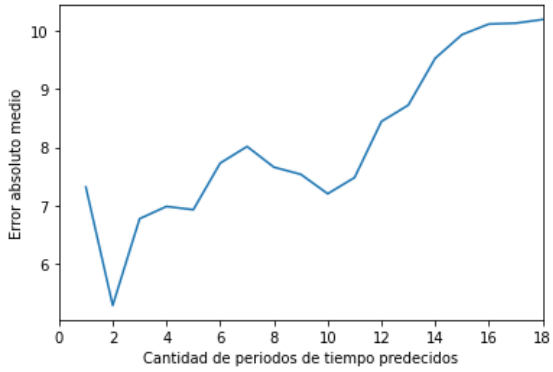

Figura 9. Tendencia del EMA frente a los intervalos de tiempo de la predicción del modelo Suavización exponencial

Con lo anterior, se puede observar que el modelo tiene el mínimo error, cuando se predice 2 intervalos de tiempo futuros. De la misma manera, se observa que este modelo también es un modelo de corta memoria y necesita ser retroalimentado cada dos periodos de tiempo.

Con lo cual, para entrar a los códigos de desarrollo, se puede ingresar al siguiente link:

https://github.com/DiegoRojasXD/Codigos_Tesis_IMEC/tree/main/ETS

7.1.1.3. heteroscedástico condicional autorregresivo (ARCH)

En tercer lugar, se hace uso de la función "arch_model" para encontrar el mejor modelo ARCH con los datos de entrada específicos. De esta manera, se evidencio la capacidad del modelo al mismo tiempo que la función iba encontrando el mejor modelo para dichos valores. Con lo cual, en la figura 10 se observa la tendencia del error absoluto medio del modelo frente a los intervalos de tiempo que está prediciendo.

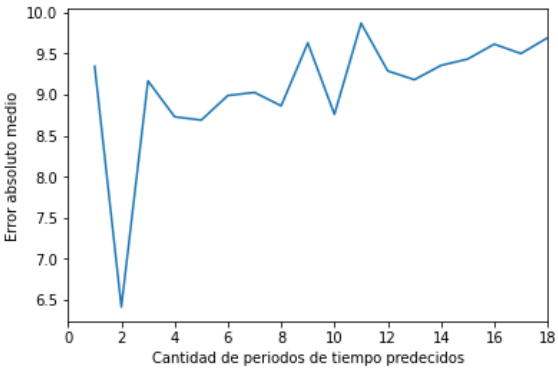

Figura 10. Tendencia del EMA frente a los intervalos de tiempo de la predicción del modelo ARCH

Al igual que en los anteriores modelos, se puede observar que el modelo tiene es de corta memoria y necesita ser retroalimentado cada dos periodos de tiempo.

Con lo cual, para entrar a los códigos de desarrollo, se puede ingresar al siguiente link:

https://github.com/DiegoRojasXD/Codigos_Tesis_IMEC/tree/main/ARCH

7.1.2. Red Neuronal Artificial

Finalmente, los procesos iterativos que se pueden observar en la figura 11 de manera de ejemplo, se encontró una arquitectura que cumpliera una clasificación estadísticamente correcta.

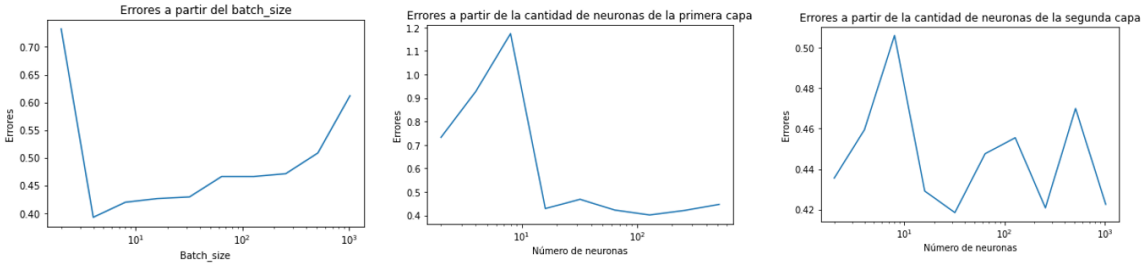

Figura 11. Ejemplo de procesos iterativos realizados

Con ello, se obtuvo la gráfica de las pérdidas generadas a través del proceso de entrenamiento de la red como se evidencia en la figura 12 para un modelo de predicción de 6 días.

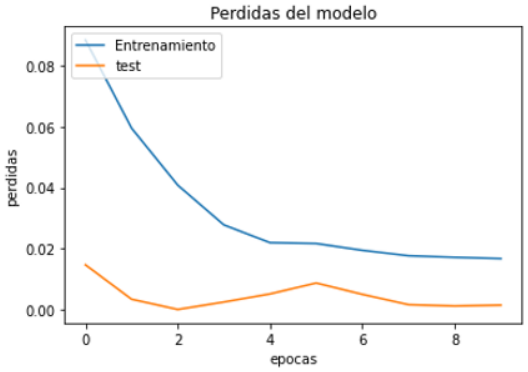

Figura 12. Pérdidas del modelo ANN

Con lo anterior, se evidencia una red que mantiene errores al finalizar su entrenamiento de 0.015, por lo tanto, la arquitectura final que se obtuvo se evidencia en la tabla 5. Para este modelo, se tiene un batch size de 128 y un learning rate de 0.0001. Cabe resaltar, que este código en Python se encuentra en el siguiente link:

Tabla 5. Arquitectura de la red ANN para el modelo de clasificación de la maduración del tomate

Capa	Dimensionamiento de salida	Cantidad de pesos
Capa LSTM	(None, 1024)	4 218 880
Capa densa	(None, 1024)	1 049 600

Capa densa	(None, 1)	1 025
Total, parámetros: 5 269 505		

Con lo cual, para entrar a los códigos de desarrollo, se puede ingresar al siguiente link:

https://github.com/DiegoRojasXD/Codigos_Tesis_IMEC/tree/main/Modelos_ANN

8. Validación del Trabajo

8.1. Metodología de prueba

Para la validación del trabajo realizado, se realizó un proceso de testeo de los modelos computacionales creados, de esta manera, se hace uso de los datos designados como x_test. Por medio de este testeo, se genera una gráfica comparativa entre los datos predichos frente a los datos reales registrados por el sensor. Con lo anterior, se obtiene el y_predict y con el y_test se obtienen el error medio absoluto y la raíz cuadrada del error medio cuadrático para cada modelo.

8.1.1. Métodos tradicionales

8.1.1.1. Media móvil integrada autorregresiva (ARIMA)

Por medio de los datos de testeo, se obtuvo la figura 13 en donde se evidencia una curva azul que da referencia a los datos del día 1 que son los utilizados como históricos para predecir los valores futuros de la radiación solar. De esta manera, en la curva verde, se observan dichos resultados de la predicción generada a través del proceso descrito anteriormente con una renovación de los históricos cada 2 periodos de tiempo predichos. Finalmente, se evidencia la curva naranja que representa los datos registrados por el sensor de radiación solar de la planta y con los cuales se compara el resultado del forecasting.

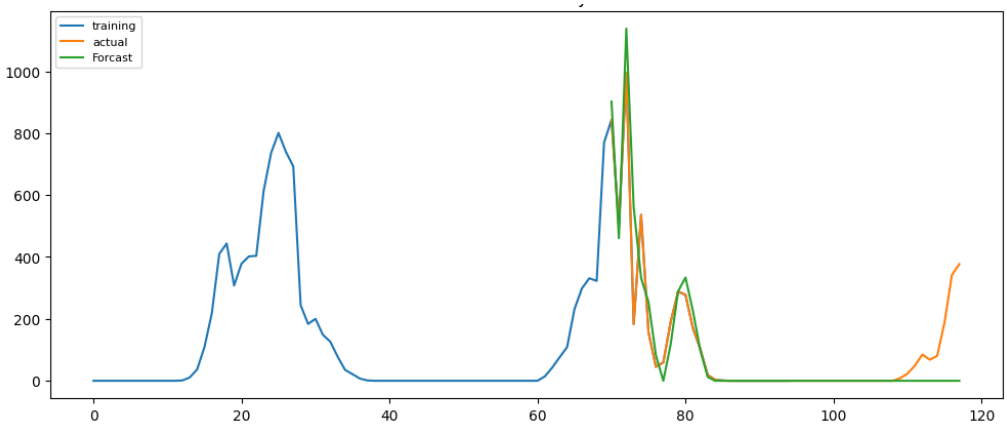

Figura 13. Forecasting del modelo ARIMA

A partir de este resultado, se generó el error medio absoluto y la raíz del valor medio cuadrado que se puede observar en la tabla 6.

Tabla 6. Métricas del modelo ARIMA

Error medio absoluto	Raíz del valor medio cuadrado
5.08%	9.41%

8.1.1.2. Suavización exponencial

De la misma manera, a partir del modelo de suavización exponencial se obtuvo el resultado observado en la figura 14. Así, con este resultado se generó el error medio absoluto y la raíz del valor medio cuadrado que se puede observar en la tabla 7.

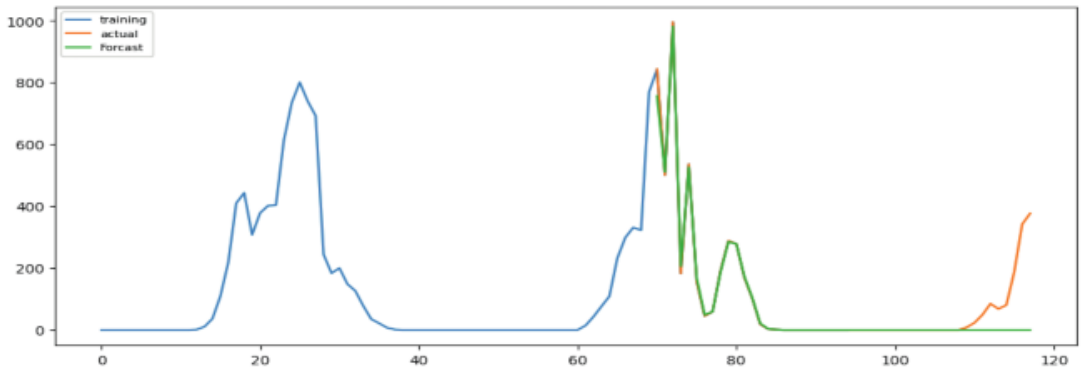

Figura 14. Forecasting del modelo de suavización exponencial

Tabla 7. Métricas del modelo suavización exponencial

Error medio absoluto	Raíz del valor medio cuadrado
2.91%	5.39%

8.1.1.3. heteroscedástico condicional autorregresivo (ARCH)

Finalmente, para terminar con los modelos tradicionales, se obtuvo el resultado observado en la figura 15 con el modelo ARCH. El cual, generó las métricas del error medio absoluto y la raíz del valor medio cuadrado que se puede observar en la tabla 8.

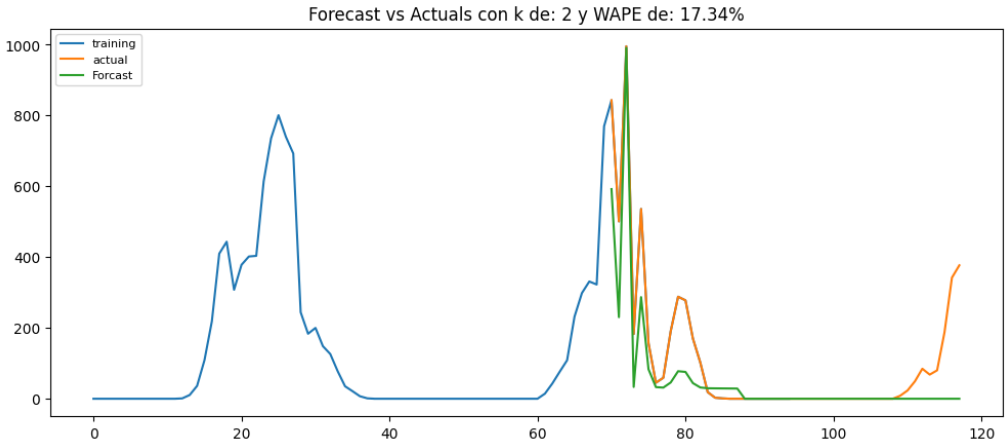

Figura 15. Forecasting del modelo ARCH

Tabla 8. Métricas del modelo ARCH

Error medio absoluto	Raíz del valor medio cuadrado
6.54%	12.11%

8.1.2. Red Neuronal Artificial

Por el otro lado, se tienen los resultados generados por el modelo de machine learning. En el cual, se resalta que, para realizar una comparación con los modelos anteriores, se realizó el entrenamiento de la red para predecir 1 día como se observa en la figura 16 y en la tabla 9 sus métricas.

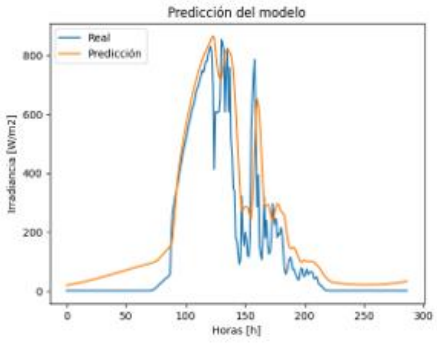

Figura 16. Forecasting del modelo de redes neuronales de 1 día de predicción

Tabla 9. Métricas del modelo de redes neuronales para la predicción de 1 día

Error medio absoluto	Raíz del valor medio cuadrado
21.10%	22.2%

Sin embargo, el modelo final que evidencia el potencial de las redes neuronales para el forecasting es en predicciones de largo tiempo como es 6 días. Este resultado se puede observar en la figura 17 y en la tabla 10.

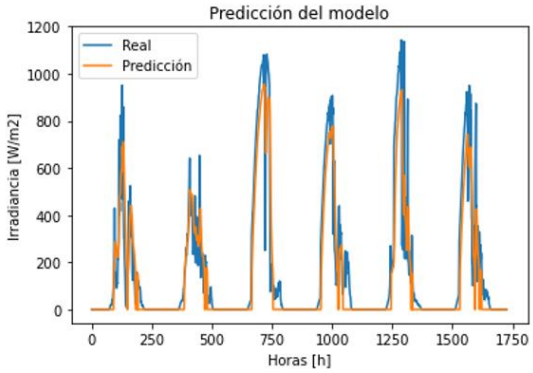

Figura 17. Forecasting del modelo de redes neuronales de 6 día de predicción

Tabla 10. Métricas del modelo de redes neuronales para la predicción de 6 día

Error medio absoluto	Raíz del valor medio cuadrado
8.32%	15.4%

Finalmente, en circustancias de ejemplo, en la figura 18 se puede observar uno de los potenciales de los modelos de forecasting a largo tiempo generado con redes neuronales. En esta figura, se puede observar los picos de potencia electrica disponible en kW de 15 días del mes de marzo del 2022 a partir de la radiación solar predecida. Con ello, se puede evidenicar que los modelos de forecasting sirven para poder anticiparce al comportamiento del sistema electrico en el futuro y preveer así condiciones de uso.

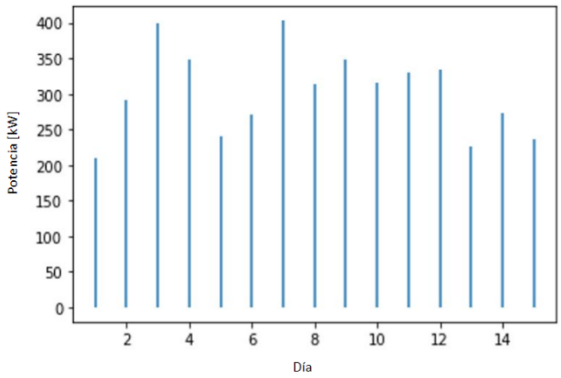

Figura 18. Potencia eléctrica disponible de los primeros 15 días del mes de marzo del 2022

En conjunto, en la tabla 11 se pueden observar todas las métricas de cada modelo obtenidas y evidenciar que el modelo de suavización exponencial es el que mejor comportamiento tiene a predicciones de corto plazo. Pero, que el modelo de media movial ARIMA también es una opción aceptable para este mismo fin. Además, se puede observar que los modelos de foerecasting por medio de redes neuronales, son de alta funcionalidad para predicciones a periodos largos de tiempo.

Tabla 11. Comparación métricas de evaluación de modelos

Modelo	Error medio absoluto	Raíz del valor medio cuadrado
ARIMA	5.08%	9.41%
Suavización exponencial	2.91%	5.39%

ARCH	6.54%	12.11%
Red neuronal de 1 día	21.1%	22.2%
Red neuronal de 6 días	8.32%	15.4%

A partir de los resultados anteriores. Se puede evidenciar que los modelos estadísticos generan predicciones de corto plazo pero que garantizan un alto porcentaje de precisión en la predicción. Por el otro lado, el modelo de machanie learning genera predicciones más globales, donde solo se identifican con precisión los puntos máximos de irradiancia solar. Por lo tanto, se puede observar que los modelos estadísticos son mejores para predicciones a corto plazo, pero son muy limitantes para análisis financieros y estadísticos del comportamiento futuro de la planta. Mientras, que el modelo de machine learnign es un modelo que puede acoplarse a varias estructuras de datos que pueden generar pronósticos a largo plazo que sirve para la planeación de proyectos de energía solar.

9. Trabajo adicional

Este trabajo creció bajo la restricción de una base de datos especifica de la planta solar de la universidad de los andes. Por lo cual, se espera en un futuro poder aplicar estos modelos a diferentes plantas solares y condiciones de radiación diferentes para evaluar la capacidad de generalización de los modelos.

Así mismo, se espera poder utilizar estas herramientas para el desarrollo de nuevas tecnologías base para la producción de energía solar optimizada. Por lo cual, se espera que, a partir de esta comparación, se de a entender las limitaciones de cada modelo y poder aplicarlos para las condiciones que mejor operan. Discusión

10. Conclusiones

Los modelos estadísticos de forecasting de la irradiancia solar tienen un gran potencial para la optimización del aprovechamiento de la energía solar. Al predecir con precisión la cantidad de energía que se generará en un momento dado, se pueden tomar decisiones informadas sobre la gestión de la energía. Así mismo los modelos de machine learning para el forecasting de la irradiancia solar tienen potencial en la planificación de la red eléctrica al evidenciar el comportamiento de la planta solar a tiempos futuros largos. De esta manera, los modelos de forecasting de la irradiancia solar tienen una gran proyección para la reducción de costos. Por lo cual, se puede enlistar las siguientes conclusiones:

- Al ser la radiación solar una variable de gran importancia para la para la producción energética de las plantas solares. Por lo tanto, la estimación precisa de la irradiancia global es fundamental para optimizar la eficiencia energética y el rendimiento de los paneles solares.
- El forecasting es una herramienta de gran importancia para la planificación y el diseño de sistemas de energía solar, ya que permite anticipar las fluctuaciones en la radiación solar y tomar medidas para optimizar el rendimiento de los paneles solares.

- El desarrollo de las redes neuronales artificiales muestra el potencial del aprendizaje automático en sistemas de series de tiempo al poder aprender patrones complejos. Lo cual, puede generar modelos con una mayor precisión en la predicción.
- Los modelos tradicionales proporcionan una aproximación inicial del objetivo deseado y pueden ser eficientes en términos de recursos computacionales y pronósticos de corto plazo.
- La elección del método más adecuado depende de los factores que se generan alrededor de la aplicación deseada considerado las limitaciones y ventajas de cada método para seleccionar el enfoque más apropiado.

11. Agradecimientos

Agradezco al docente Andrés Gonzales Mancera por apoyarme en el desarrollo del proyecto desde su experiencia y conocimiento. De la misma manera, agradezco al ingeniero Luis Fernando Vera por apoyarme en la gestión de los computadores de la sala de simulación de ingeniería mecánica de la Universidad de los Andes para el desarrollo del proyecto. Así mismo, quiero agradecer al ingeniero Jorge Enrique Carrillo Diaz, quien me apoyo con los requisitos técnicos de mi computadora y me permitió. Finalmente, agradezco a mi familia por el apoyo constante en el proceso de desarrollo de este proyecto.

12. Referencias

[1] Naciones Unidas, «Energías renovables: energías para un futuro más seguro,» [En línea]. Available: https://www.un.org/es/climatechange/raising-ambition/renewable-energy#:~:text=Los%20combustibles%20f%C3%B3siles%20dan%20cuenta,de%20fuentes%20de%20energ%C3%ADa%20renovables.. [Último acceso: 22 05 2023].

[2] UPME, «Balance energético colombiano,» UPME, [En línea]. Available: https://www1.upme.gov.co/DemandayEficiencia/Paginas/BECO.aspx.

[3] Solar Forecast Arbiter, «Solar Forecast Arbiter, A paradigm shift in forecast evaluation,» [En línea]. Available: https://forecastarbiter.epri.com/. [Último acceso: 22 05 2023].

[4] T. T. e. I. Chernyakhovskiy, «PRONÓSTICO DE GENERACIÓN DE ENERGÍA EÓLICA Y SOLAR: MEJORANDO LA OPERACIÓN DEL SISTEMA,» Laboratorio Nacional de Energía Renovable, Estados Unidos, 2016.

[5] UPME, «Proyecciones de demanda,» UPME, [En línea]. Available: https://www1.upme.gov.co/DemandayEficiencia/Paginas/Proyecciones-de-demanda.aspx. [Último acceso: 22 05 2023].

[6] H. Maïmouna Diagne, D. Mathieu , P. Lauret, J. Boland y N. Schmutz, «Review of solar irradiance forecasting methods and a proposition for small-scale insular grids,» HAL open science, Australia, 2013.

[7] «ARIMA Model - Complete Guide to Time Series Forecasting in Python,» Machine Learning Plus, [En línea]. Available: https://www.machinelearningplus.com/time-series/arima-model-time-series-forecasting-python/. [Último acceso: 22 05 2023].

[8] N. Hoang, «ARIMA Autoregressive Integrated Moving Average model family,» Nhu Hoang, 26 01 2020. [En línea]. Available: https://geniusnhu.netlify.app/project/2020-01-26-arima-autoregressive-intergreated-moving-average/. [Último acceso: 22 05 2023].

[9] V. Kotu, «Media móvil integrada autorregresiva, ARIMA es una de las técnicas de pronóstico más utilizadas en las empresas en la actualidad.,» Science direct, 2019. [En línea]. Available: https://www.sciencedirect.com/topics/mathematics/autoregressive-integrated-moving-average.

[10] J. Bean, «How to leverage the exponential smoothing formula for forecasting,» Zendesk Blog, 16 05 2022. [En línea]. Available: https://www.zendesk.com/blog/leverage-exponential-smoothing-formula-forecasting/.

[11] A. Novales, «Modelos ARCH univariantes y multivariantes,» Universidad Complutense de Madrid, 2013. [En línea]. Available: https://www.ucm.es/data/cont/media/www/pag-41460/Arch.pdf.

[12] National geographic, «Breve historia visual de la inteligencia artificial,» National geographic, 2020. [En línea]. Available: https://www.nationalgeographic.com.es/ciencia/breve-historia-visual-inteligencia-artificial_14419.

[13] C. E. Muñoz Domínguez y H. R. Casallas Ávila , «Diseño de una máquina para la clasificación de tomate chonto Lycopersicum esculentum mill,» Universidad de La Salle, Bogotá, 2020.

[14] V. Rodríguez, «Conceptos básicos sobre redes neuronales,» 30 10 2018. [En línea]. Available: https://vincentblog.xyz/posts/conceptos-basicos-sobre-redes-neuronales.

[15] IBM, «IBM Documentation,» IBM - Deutschland | IBM, s.f.. [En línea]. Available: https://www.ibm.com/docs/es/spss-modeler/saas?topic=nodes-neural-networks.

[16] E. Freire y S. Silva, «Redes neuronales,» Medium, 14 11 2019. [En línea]. Available: https://bootcampai.medium.com/redes-neuronales-13349dd1a5bb#:~:text=Funciones%20de%20activaci%C3%B3n,que%20permitir%C3%A1%2.

[17] D. Calvo, «Función de activación - Redes neuronales,» 7 12 2018. [En línea]. Available: https://www.diegocalvo.es/funcion-de-activacion-redes-neuronales/.

[18] I. Gavilán, «Catálogo de componentes de redes neuronales (III): funciones de pérdida,» [En línea]. Available: https://ignaciogavilan.com/catalogo-de-componentes-de-redes-neuronales-iii-funciones-de-perdida/.

[19] P. G. Juan Camilo , «Clasificación de la madurez del aguacate Hass para su cosecha,» UNIVERSIDAD DE LOS ANDES, Bogotá, Colombia, 2022.

[20] Fayrix, «Selección de métricas para los modelos de aprendizaje automático,» Fayrix | Offshore Custom Software Development Services, s.f.. [En línea]. Available: https://fayrix.com/machine-learning-metrics_es. [Último acceso: 23 05 2023].

[21] M. Paulescu, E. Paulescu, P. Gravila y V. Badescu, Weather modeling and forecasting of PV systems operation, New York: Springer, 2013.

[22] J. A. A. TOSCANO, PREDICCIÓN DE LA ENERGÍA GENERADA POR UN PANEL SOLAR BASADO EN SU TEMPERATURA Y RADIACIÓN, Bogotá: Universidad de los Andes, 2019.

[23] P. Kumari y D. Toshniwal, «Deep learning models for solar irradiance forecasting: A comprehensive review,» *Journal of Cleaner Production,* vol. 1, nº 128566, p. 26, 2021.

[24] J. M. R. Rodríguez, «Modelo para predicción de potencia de paneles fotovoltaicos utilizando técnicas de clasificación no supervisada y redes neuronales artificiales.,» Universidad del Norte, Barranquilla, 2020.