Escuela Superior
de Ingeniería y Tecnología
Universidad de La Laguna

Trabajo de Fin de Grado

Técnicas de machine learning para la predicción de la temperatura de punto de rocío aplicado al mantenimiento predictivo de un telescopio astronómico

Machine learning techniques for dew point temperature forecasting applied to predictive maintenance of an astronomical telescope

Daniel García Hernández

La Laguna, 25 de mayo de 2023

D. **Christopher Expósito Izquierdo**, con N.I.F. 78.851.649-J profesor Ayudante Doctor adscrito al Departamento Ingeniería Informática y de Sistemas de la Universidad de La Laguna como tutor

D. **Adrián Calzadilla González**, con N.I.F. 78.616.940-L ingeniero de software en el Instituto Astrofísico de Canarias como cotutor

C E R T I F I C A N

Que la presente memoria titulada:

”*Técnicas de machine learning para la predicción de la temperatura de punto de rocío aplicado al mantenimiento predictivo de un telescopio astronómico*”

ha sido realizada bajo su dirección por don **Daniel García Hernández**, con N.I.F. 79.070.891-L.

Y para que así conste, en cumplimiento de la legislación vigente y a los efectos oportunos firman la presente en La Laguna a 25 de mayo de 2023

Agradecimientos

En primer lugar, quiero agradecer a mi tutor académico, Christopher, por su guía experta y apoyo a lo largo de este proyecto.

También quiero expresar mi agradecimiento a mis tutores del Instituto de Astrofísica de Canarias (IAC), Fran y Adrián, por su orientación, dedicación y por brindarme la oportunidad de explorar de cerca el fascinante mundo de los telescopios.

A mi familia, quiero agradecerles por su amor incondicional, apoyo constante y comprensión a lo largo de mi carrera universitaria.

A mis amigos, por su amistad, compañerismo y por estar a mi lado, brindándome alegría y distracción cuando más lo necesitaba.

Finalmente, quiero expresar mi gratitud a todas aquellas personas que de una u otra manera me apoyaron y alentaron durante este emocionante viaje académico.

Resumen

En este trabajo se realiza una comparativa de modelos para la predicción de la temperatura del punto de rocío con el objetivo de implementar uno de ellos en el Sistema de Control de Telescopios Nocturnos (Control TTNN) del Instituto de Astrofísica de Canarias (IAC). Se realiza un preprocesamiento de los datos y se evaluan diferentes enfoques de aprendizaje automático, tanto univariable como multivariable, y se analizan sus resultados.

El preprocesamiento de los datos efectuado incluye una detección y suavizado de anomalías, un tratamiento de valores faltantes y una selección de características, así como de un análisis de las propiedades de las variables contempladas. Se obtiene que el preprocesamiento proporciona una mejora efectiva en las predicciones, obteniendo tras las dos primeras etapas una Raíz del Error Cuadrático Medio (RMSE) un 13.24 % menor que en el conjunto original.

Utilizando 10 muestras diferentes del conjunto de pruebas, se calcula la media de los resultados obtenidos por cada modelo. Los modelos son configurados con regresores externos y se ajustan los hiperparámetros durante la fase de entrenamiento y validación. Los resultados muestran que el modelo Gradient Boosting Regressor (GBR) de Scikit-Learn obtiene el mejor rendimiento, logrando un RMSE de 4.81º. Además, otras métricas respaldan la superioridad del GBR, incluyendo las referentes al intervalo de predicción, dado que obtiene el menor Error Medio de Cobertura (11.26) y el Porcentaje de Cobertura más alto (66.25). En contraste, el modelo Extreme Gradient Boosting (XGBoost) muestra un RMSE de 7.21º, lo cual representa una diferencia de 2.40 grados en comparación con el GBR. Basándose en estos resultados, se selecciona el modelo GBR para su implementación en el Sistema de Control de TTNN.

Palabras clave: Telescopio, Temperatura de punto de rocío, Mantenimiento predictivo, ROS, Aprendizaje automático, Preprocesamiento de datos

Abstract

In this study, a comparison of models is conducted to predict the dew point temperature with the objective of implementing one of them in the Nighttime Telescope Control System (Control TTNN) at the Institute of Astrophysics of the Canary Islands (IAC). Data preprocessing is performed, evaluating different machine learning approaches, both univariate and multivariate, and analyzing their results.

The data preprocessing includes anomaly detection and smoothing, treatment of missing values, feature selection, and an analysis of the variables' properties. It is found that preprocessing effectively improves predictions, achieving a 13.24 % lower Root Mean Square Error (RMSE) than that of the original dataset after the first two stages.

Using 10 different samples from the test set, the average results obtained by each model are calculated. The models are configured with external regressors, and the hyperparameters are tuned during the training and validation phase. The results indicate that the *Gradient Boosting Regressor* (GBR) model from *Scikit-Learn* performs the best, achieving an RMSE of 4.81º. Furthermore, other metrics support the superiority of GBR, including those related to the prediction interval, as it has the lowest Mean Coverage Error (11.26) and the highest Coverage Percentage (66.25). In contrast, the *Extreme Gradient Boosting* (XGBoost) model exhibits an RMSE of 7.21º, representing a difference of 2.40 degrees compared to GBR. Based on these findings, the GBR model is selected for implementation in the TTNN Control System.

Keywords: *Telescope, Dew point temperature, Predictive maintenance, ROS, Machine learning, Data preprocessing*

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

1.1. Instituto Astrofísico de Canarias

El Instituto de Astrofísica de Canarias (IAC)[1] es un centro de investigación en astronomía y astrofísica, con sede en la isla de Tenerife, España. Su objetivo es promover y realizar investigaciones de vanguardia en el campo de la astrofísica, así como fomentar la divulgación científica y el desarrollo tecnológico en esta área.

El IAC se especializa en la investigación en diversos temas de la astrofísica, incluyendo la formación y evolución de galaxias, el estudio de estrellas y planetas, la cosmología y la física de altas energías. También lleva a cabo una importante actividad en el campo de la instrumentación astronómica, diseñando y construyendo instrumentos de observación y telescopios avanzados. Entre sus proyectos más destacados se encuentran los siguientes:

- Gran Telescopio Canarias (GTC). Es un telescopio óptico-infrarrojo con un espejo primer de 10,4 metros de diámetro, lo que lo convierte en el más grande del mundo en su categoría. El IAC es uno de los socios fundadores de este proyecto, que se inauguró en 2009.

- Camelot-2. Es un sistema que utiliza técnicas de aprendizaje automático y una base de datos de espectros estelares de alta calidad para identificar y clasificar las características de los espectros estelares. Ha sido utilizado en numerosos estudios astronómicos y ha demostrado ser un sistema eficaz y preciso para la clasificación automática de espectros estelares.

- European Solar Telescope (EST). Un telescopio solar de nueva generación que permite estudiar la física del Sol con una resolución sin precedentes. Es un proyecto promovido por la Asociación europea para telescopios solares (EAST)[2] que involucra a 29 socios y 9 instituciones de 15 países diferentes.

- Proyecto de Anisotropía del Fondo Cósmico de Microondas (CMB, por sus siglas en inglés). Busca estudiar las pequeñas variaciones en la temperatura del CMB[3] en diferentes regiones del cielo (véase Figura 1.1). Estas variaciones son extremadamente pequeñas, pero son de gran importancia para entender la estructura y

[1] Instituto de Astrofísica de Canarias (IAC)

[2] EAST

[3] El fondo cósmico de microondas es una radiación electromagnética que llena todo el universo observable y se detecta como una radiación de microondas de baja intensidad que se emite en todas las direcciones del cielo. Es considerado uno de los principales vestigios del Big Bang.

evolución del universo, dado que permiten estudiar la evolución del universo a lo largo del tiempo.

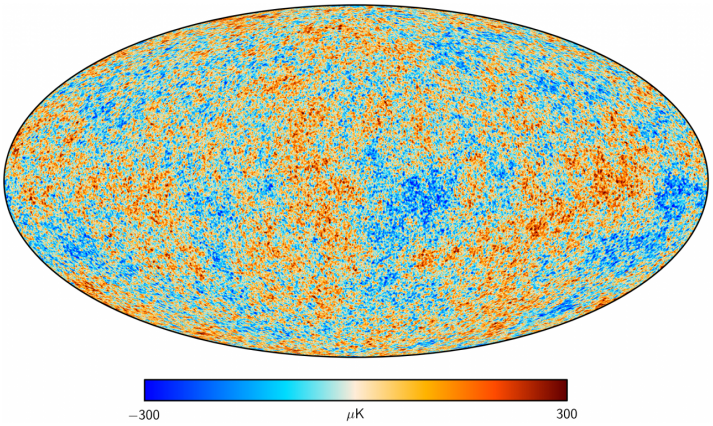

Figura 1.1: Mapa de las anisotropías del Fondo Cósmico de Microondas [21]

1.2. Proyecto

1.2.1. Sistema de control de los Telescopios Nocturnos del Observatorio del Teide

Este trabajo se plantea en el contexto del sistema de control de los Telescopios Nocturnos del Observatorio del Teide (TTNN) del IAC, implementado en los telescopios IAC80[4] y Carlos Sánchez (TCS)[5]. El sistema está formado por un conjunto de componentes hardware y software que permite la operación eficiente de los telescopios, ubicados en el Observatorio Astronómico del Teide, Tenerife, España (Figura 1.2). Este sistema tiene como objetivo principal controlar los movimientos y funciones del telescopio para realizar observaciones astronómicas de manera precisa y automatizada durante la noche.

Figura 1.2: Localización del IAC80 y Carlos Sánchez. Imágenes obtenidas mediante Google Earth

[4]El IAC-80 es un telescopio nocturno diseñado enteramente por el IAC y se utiliza actualmente para llevar a cabo investigaciones astronómicas en diferentes áreas, como la astrofísica estelar, la astrofísica galáctica y extragaláctica, y la búsqueda y seguimiento de asteroides y cometas.

[5]El Telescopio Carlos Sanchez (TCS es un telescopio nocturno reflector de 152 centímetros de diámetro diseñado para estudiar la física estelar y la astrofísica galáctica

Su funcionamiento implica el uso de controladores de movimiento para el posicionamiento y seguimiento del telescopio, así como una interfaz de usuario que permite a los astrónomos y operadores interactuar con el sistema y dar comandos de observación (véase Figura 1.3). El software de control procesa los comandos y ejecuta las acciones necesarias para el movimiento y operación del telescopio, incluyendo la adquisición de datos de los sensores y detectores integrados en el sistema.

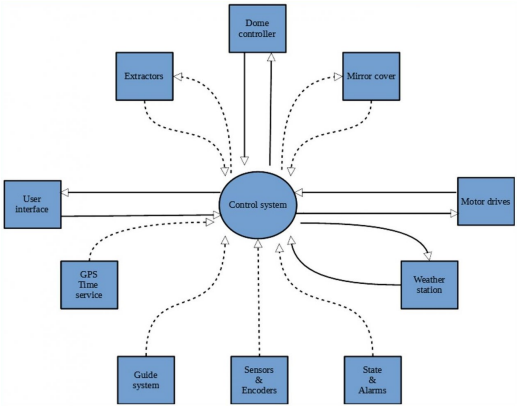

Figura 1.3: Esquema del sistema de control para telescopios nocturnos [22]

El propósito general del sistema es mejorar la eficiencia y precisión de las observaciones astronómicas realizadas con el telescopio. Esto incluye componentes para el control preciso del movimiento del telescopio para apuntar con precisión a objetos celestes, el seguimiento preciso de los objetos en movimiento durante las observaciones, la adquisición y procesamiento de datos de los sensores y detectores, y la automatización de ciertas tareas para optimizar la operación del telescopio durante la noche. Un ejemplo de estos componentes es el sistema *FOVIA*, para el guiado y seguimiento de objetos estelares (véase Figura 1.4).

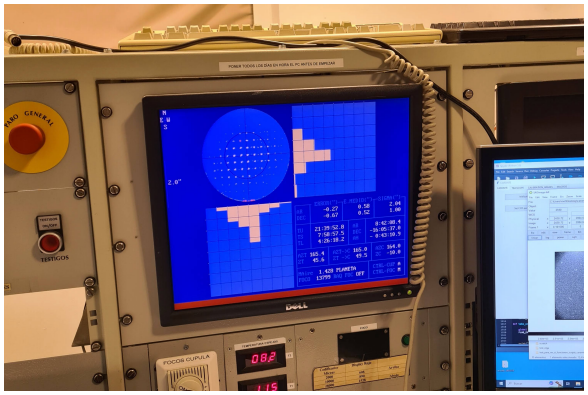

Figura 1.4: Sistema de seguimiento FOVIA en una pantalla del IAC80.

1.2.2. Proyecto de control TTNN

El proyecto de control TTNN[6] tiene como finalidad dotar de una mayor robustez y versatilidad al sistema de control de los telescopios IAC80 y Carlos Sánchez. Se pretende que el nuevo entorno permita el control remoto y eficiente de ambos telescopios, de manera que se facilite su operación automática.

[6]Control TTNN

Arquitectura de software

La refactorización del software se ha diseñado respectando los principios *SOLID*[7] y bajo el enfoque de una arquitectura mixta, formada por:

- *Arquitectura hexagonal*[8]. Es una arquitectura limpia que separa claramente la lógica de negocio de los detalles de implementación y tecnología. Está formada por un núcleo central que contiene la lógica de negocio y está rodeado por puertos (interfaces) que definen las formas en que el núcleo se comunica con el exterior. Los adaptadores proporcionan las implementaciones concretas de estos puertos y conectan el núcleo con los diferentes sistemas externos. Esta separación permite que el núcleo se mantenga independiente de los detalles de implementación, lo que facilita las pruebas, el mantenimiento y la escalabilidad del sistema.
- *Arquitectura enfocada a eventos (publicador/suscriptor)*[9]. Es un patrón de diseño en el que los componentes del sistema se comunican entre sí mediante eventos que son enviados a través de un canal centralizado. En esta arquitectura, los componentes del sistema pueden actuar como publicadores, emitiendo eventos que son enviados a través del canal centralizado, o como suscriptores, que se registran para recibir eventos específicos del canal. Cuando un evento es enviado a través del canal, todos los suscriptores registrados para ese evento reciben una copia del mismo.

Servidor

Para la implementación del servidor, se ha empleado ROS (*Robot Operating System*)[10], un framework de desarrollo de software basado en una arquitectura de comunicación distribuida. Los nodos ROS son programas que se comunican a través de esta arquitectura, siendo el principal de ellos el nodo *Core*, que proporciona el registro y búsqueda de nombres. Los nodos se comunican a través de mensajes que se enrutan a través de un sistema de transporte con arquitectura de publicador/suscriptor y a través de servicios que permiten las interacciones petición/respuesta. El servidor ofrece distintas funcionalidades, como un modelo de llamada a procedimiento remoto y asíncrono, las bags, que permiten almacenar y reproducir los mensajes de ROS a diferentes velocidades y los paquetes, que son la unidad fundamental de agrupar los nodos que tienen un significado común.

Jerarquía

Se estableció una estructura jerárquica de paquetes para representar los diferentes componentes del telescopio. Esta estructura incluye los siguientes paquetes:

- *device*. Representa los elementos más simples del sistema, que inyectan información en el flujo de datos del sistema

[7] Los principios *SOLID* son un conjunto de cinco principios de diseño de software que buscan promover la creación de sistemas más fáciles de entender, mantener y evolucionar. Cada letra en SOLID representa uno de los cinco principios: S (Responsabilidad Única), O (Abierto/Cerrado), L (Sustitución de Liskov), I (Segregación de Interfaces) y D (Inversión de Dependencias).

[8] Arquitectura hexagonal

[9] Arquitectura enfocada en eventos

[10] ROS (Robot Operating System)

- *coordination*. Representa los nodos que agrupan la funcionalidad de más alto nivel, como la coordinación de ejes o el sincronismo de seguimiento cúpula/telescopio
- *management*. Contiene aquellos paquetes relacionados con la gestión del sistema, así como todos aquellos nodos/procesos que no estén relacionados directa o indirectamente con la gestión del hardware o la operativa.

1.3. Justificación del trabajo

El punto de rocío es la temperatura a la cual el aire debe ser enfriado para que el vapor de agua en él se condense en forma de rocío. Es un indicador importante de la humedad del aire y se utiliza comúnmente en diversas aplicaciones, como el control de la humedad en procesos industriales, la planificación agrícola y la predicción del clima. Por ejemplo, si la temperatura del aire es de 30°C y el punto de rocío es de 20°C, entonces el aire debe ser enfriado a 20°C para que el vapor de agua en él se condense en forma de rocío. Si la temperatura del aire sigue disminuyendo por debajo de ese punto, el exceso de vapor de agua se condensará en forma de líquido.

La temperatura de rocío es una variable crucial en el ámbito de la observación astronómica, ya que la formación de rocío en los telescopios puede dañar los componentes y afectar negativamente la calidad de la imagen observada, provocando desenfoque y distorsión. En un telescopio astronómico, el espejo principal es un componente crítico que requiere un cuidado especial. Si se expone al rocío, éste puede erosionar su superficie y dañar la reflectividad, además de generar manchas y corrosión.

Actualmente, el personal encargado de la operación de los telescopios nocturnos del IAC solo dispone de información sobre el punto de rocío en el momento actual, lo que limita su capacidad de planificación. Si se prevé que la temperatura alcanzará el punto de rocío, se pueden tomar medidas para reducir el tiempo de observación o posponerla hasta que la temperatura suba. Esto permite mejorar la eficiencia en la planificación y gestión de la observación y asegurar la calidad de la imagen.

Por tanto, y teniendo en cuenta lo anterior, el presente proyecto tiene como objeto el desarrollo de un modelo de predicción del punto de rocío para su implementación en el proyecto de actualización y mejora del Sistema de Control de TTNN. El modelo se basa en variables ambientales tales como la temperatura, la humedad y la presión atmosférica con el fin de pronosticar el punto de rocío con una precisión aceptable para los investigadores del IAC. La implementación de este modelo en el sistema del telescopio permitirá una planificación más eficiente de las operaciones de observación, ya que el personal del telescopio podrá anticipar el momento en que el punto de rocío alcanzará niveles no compatibles con la observación. Además, ayudará a prevenir daños en el espejo del telescopio debido a la condensación del rocío, y a su vez, permitirá reducir los costos de mantenimiento.

1.4. Objetivos

El objetivo general de este trabajo es desarrollar un modelo de predicción del punto de rocío empleando técnicas clásicas y avanzadas de aprendizaje automático con el fin de mejorar la eficiencia en la planificación, gestión y mantenimiento de la observación astronómica en un telescopio.

1.4.1. Objetivos específicos

- Análisis del estado del arte en la predicción del punto de rocío y en la aplicación del aprendizaje automático en el ámbito de la climatología y astrofísica.
- Estudio de métodos y técnicas para el análisis de series temporales.
- Recopilar los datos meteorológicos necesarios para la creación del modelo de predicción del punto de rocío.
- Realizar un preprocesamiento de los datos obtenidos que incluye el análisis y visualización de los datos, la detección y corrección de anomalías que puedan afectar a la predicción y la imputación de posibles valores faltantes en las series temporales a utilizar. También habrá que identificar las características que más relación tienen con el punto de rocío para utilizarlas en su predicción.
- Desarrollo y evaluación de modelos de pronóstico para la predicción de la temperatura del punto de rocío utilizando los datos recopilados.
- Integrar el modelo de predicción del punto de rocío en el sistema de control del telescopio.

1.5. Estructura del resto del documento

En el Capítulo 2, se aborda el estado del arte de la predicción de la temperatura del punto de rocío. Se realiza una descripción detallada de las técnicas de modelado de series temporales más relevantes en el mercado. Además, se revisan diversos documentos relacionados con la predicción en los campos de la astrofísica y la meteorología.

El Capítulo 3 se dedica a describir la metodología y las herramientas utilizadas en el proyecto de predicción de la temperatura del punto de rocío. Se explica cómo se llevó a cabo el desarrollo del trabajo y se detallan las herramientas y técnicas empleadas.

El Capítulo 4 se centra en el proceso de desarrollo del proyecto e implementación del código en de las etapas mencionadas en el capítulo 3. Se describen los pasos seguidos y las decisiones tomadas durante el desarrollo.

En el Capítulo 5 se presentan los resultados obtenidos en el conjunto de testeo y una comparativa entre las predicciones obtenidas antes y después del preprocesamiento de los datos. Además, se explica el proceso de implementación en el sistema de control TTNN.

En el Capítulo 6 se expone el presupuesto estimado para el desarrollo del proyecto. Se detallan los recursos necesarios y se realiza un análisis de los costos asociados.

Finalmente en los capítulos 7 y 8, se presentan, en español y en inglés, las conclusiones derivadas del trabajo realizado y se establecen las líneas futuras a seguir para la continuación del desarrollo del proyecto. Se analizan los resultados obtenidos y se reflexiona sobre su relevancia y posibles mejoras.

Capítulo 2

Estado del arte

2.1. Aprendizaje automático en astrofísica

La astrofísica es una rama de la ciencia que estudia el universo utilizando observaciones y análisis de datos de objetos celestes como estrellas, galaxias y agujeros negros. En la actualidad, con el continuo avance de la tecnología, la instrumentación utilizada en los telescopios y su capacidad para recopilar datos ha mejorado significativamente, lo que ha llevado a un aumento exponencial en la cantidad de datos disponibles y de artículos publicados en este campo (véase Figura 2.1). Alberto Conti, miembro de la dirección de *Ball Aerospace*[1] y científico del James Webb Space Telescope[2], respecto a la finalización de la construcción del *Square Kilometre Array* (SKA)[3]: "*We expect it will produce more data than we have on the entire Internet now - and that's in a single year*"[3].

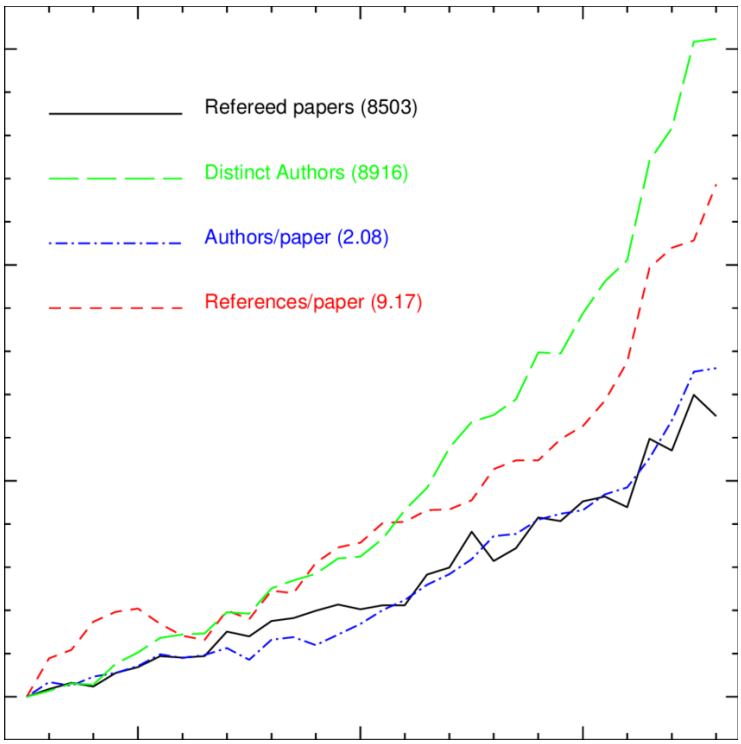

Figura 2.1: Número de publicaciones en astrófica desde 1975 hasta 2006 [52]

[1]Ball Aerospace & Technologies
[2]James Webb Space Telescope
[3]Square Kilometre Array (SKA)

El aprendizaje automático se ha convertido en una herramienta prometedora para abordar estos desafíos en la astrofísica; permite a los astrónomos desarrollar algoritmos y modelos que pueden procesar grandes cantidades de datos de manera eficiente y rápida, identificando patrones y tendencias que pueden ser difíciles de detectar con métodos tradicionales. En consecuencia, es posible obtener información valiosa de los datos observacionales y mejorar la precisión de las mediciones, lo que puede conducir a nuevos descubrimientos y avances en nuestro entendimiento del universo.

Un ejemplo de ello se puede encontrar en [14], donde se analiza un modelo novedoso que utiliza técnicas de deep learning, específicamente redes neuronales convolucionales de densidad mixta (CMDNNs)[4], para determinar la pendiente interna del perfil de densidad de los halos de materia oscura en galaxias. El objetivo es explicar uno de los principales problemas de un modelo cosmológico conocido como ΛCDM [11] (Lambda Cold Dark Matter, en inglés). En concreto hablamos del problema *cusp/core*, que se refiere a las diferencias en los perfiles de densidad de la materia oscura en las galaxias observadas (*cores*, de pendiente más llana) y las predicciones del modelo ΛCDM (*cusps*, de pendiente más pronunciada) - véase Figura 2.2. Los resultados obtenidos en el estudio son bastante satisfactorios, obteniendo una precisión que permite la distinción entre *cusps* y *cores* en un 95 % de los casos. El autor concluye por tanto que, "*la inferencia mediante el uso de redes neuronales entrenadas con datos simulacionales proporciona un método innovador y complementario a los ya existentes para la determinación de los perfiles de densidad de materia oscura en las galaxias*".

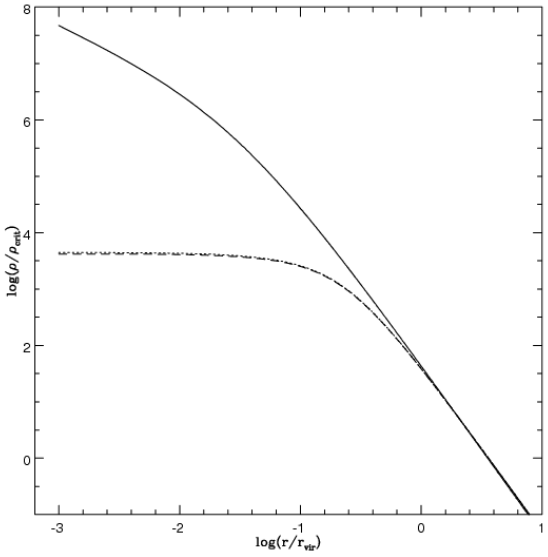

Figura 2.2: Core vs Cusp [10]

Por otro lado, el mantenimiento y la instrumentación de los telescopios también pueden beneficiarse de los mismos. La inteligencia artificial aplicada puede ser utilizada para mejorar la calibración de los instrumentos de los telescopios [30], corregir errores sistemáticos [4] y optimizar la configuración de los mismos [1], lo que resulta en una mayor precisión en las observaciones.

[4]Redes neuronales convolucionales de densidad mixta: Son un tipo de arquitectura de redes neuronales convolucionales ampliamente utilizadas en el campo de la visión por computador. Combinan capas de convolución, que extraen características de las imágenes de entrada y capas de densidad mixta, que aplican una operación de carácter estocástico que reduce la dimensionalidad del mapa de características.

En [16] se utilizan métodos de aprendizaje automático y una década de datos archivados del Telescopio Canadá-Francia-Hawái (CFHT)[5] para predecir la calidad de imagen (IQ) de observatorios a partir de las condiciones ambientales y los parámetros de operación del observatorio. Se logra predecir las funciones de distribución de probabilidad de la IQ con un error absoluto medio de aproximadamente 0.07 arcsec para las medianas predichas. El objetivo a largo plazo es construir modelos confiables y en tiempo real que puedan predecir los parámetros operativos óptimos del observatorio para optimizar la IQ. Se espera que en el futuro estas aproximaciones se conviertan en estándar para automatizar las operaciones y el mantenimiento de observatorios para cuando se instale el sucesor del CFHT, el Maunakea Spectroscopic Explorer[6], en la próxima década.

2.2. Técnicas de aprendizaje automático

2.2.1. Suavizado exponencial

El suavizado exponencial es una técnica utilizada en el análisis de series temporales que permite suavizar los datos observados mediante la asignación de pesos decrecientes a lo largo del tiempo. Es un método ampliamente utilizado en el pronóstico y análisis de series temporales, con el fin de identificar patrones o tendencias subyacentes en los datos.

- *Suavizado exponencial simple (SES).* Es una forma básica de suavizado exponencial que utiliza un solo factor de suavizado (α) para ponderar los valores observados en la serie de tiempo.
- *Suavizado exponencial triple o Holt Winter's Exponential Smoothing (HWES)* [39]. Utiliza tres factores de suavizado (α, β y γ) para ponderar los niveles, las tendencias y las estacionalidades en la serie de tiempo, respectivamente. Su fórmula incluye tres ecuaciones de suavizado, una para cada componente, y se actualizan iterativamente con cada nuevo valor observado en la serie de tiempo.

2.2.2. Modelos de la familia ARIMA

Los modelos ARIMA (*AutoRegressive Integrated Moving Average*) [27] son una herramienta comúnmente utilizada en el análisis y modelado de series temporales. Estos modelos ofrecen flexibilidad al poder utilizarse de forma individual o combinada según las necesidades específicas de cada análisis:

- *Autorregresión (AR).* Modelo de regresión que explica los valores futuros en base a una cierta número de valores pasados.
- *Promedio móvil (MA).* Utiliza los errores de predicción pasados para predecir el valores futuros.
- *Diferenciación (I).* Se utiliza para estabilizar la media y varianza a lo largo de la serie, convirtiéndola de esta forma en estacionaria.

[5] Telescopio Canadá-Francia-Hawái
[6] *Maunakea Spectroscopic Explorer*

Combinando los dos primeros componentes se obtiene el ARMA [6], que se suele utilizar solo sobre series estacionarias. En el caso de que se requiera diferenciación, es decir, que la serie sea no estacionaria, se utilizaría el ARIMA. Alternativamente, si la serie temporal a predecir presenta estacionalidad, es conveniente utilizar el modelo SARIMA [40]. Este modelo añade nuevos términos para tratar patrones estacionales y un parámetros de periodicidad que indica el número de entradas que conforma cada estación. Finalmente, SARIMAX [13] incluye todas las características del SARIMA y, además, ofrece la posibilidad de añadir predictores externos.

En [34] se utiliza el modelo ARIMA para modelar y predecir el comportamiento de la temperatura diaria y precipitación, utilizando datos registrados entre 1980 y 2010 en cuatro ciudades europeas. Se utilizan también otros métodos estadísticos como Box-Jenkins y Holt-Winters para modelar la serie temporal. Los resultados muestran que los modelos obtenidos son capaces de capturar la dinámica de los datos de la serie temporal y producir pronósticos precisos. Asimismo, se menciona que los modelos ARIMA son ampliamente utilizados en aplicaciones meteorológicas debido a sus buenos resultados: "*ARIMA models have become, in last decades, a major tool in numerous meteorological applications to understand the phenomena of air temperature and precipitation*".

2.2.3. Árboles de Regresión

Random forest (RF) es una técnica supervisada de aprendizaje automático que combina la salidas de múltiples árboles de decisión con el objetivo de alcanzar un resultado único.

Para la creación de los árboles, se emplea una técnica conocida como *Bagging* o *Bootstrap Aggregation* [19], que consiste en entrenar cada uno de ellos con una muestra aleatoria del conjunto de datos de entrenamiento. Esto implica que cada árbol se entrena en un subconjunto de datos diferente, lo que ayuda a reducir la correlación entre los árboles y a disminuir el riesgo de sobreajuste. Una vez que todos los árboles del modelo se han construido, las predicciones se obtienen promediando las propias de cada árbol en el caso de problemas de regresión, o realizando una votación para la clasificación en el caso de problemas de clasificación. Esto permite obtener una predicción final más robusta y precisa.

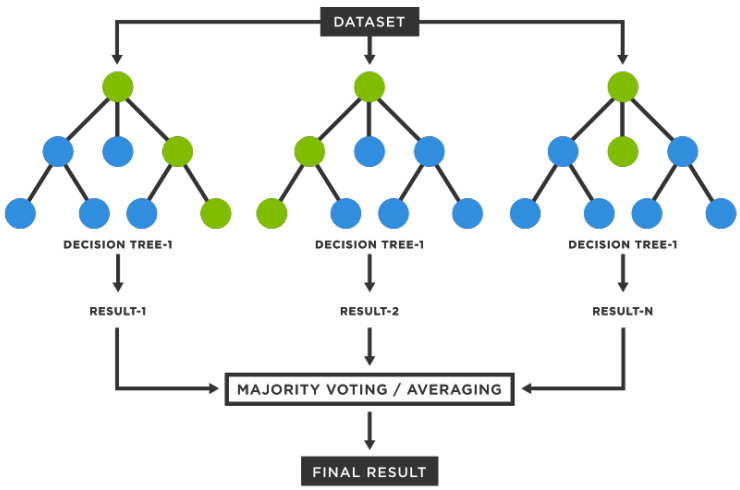

Figura 2.3: Ejemplo del proceso de un random forest [48]

En [53] se evalua la capacidad de dos métodos de soft computing, Random Forest (RF) y Multivariate Adaptive Regression Splines (MARS), para predecir la temperatura del punto de rocío a largo plazo. Se utilizan diversas variables meteorológicas, como la humedad relativa y la radiación solar, de 50 estaciones meteorológicas en Irán como entradas para los modelos RF y MARS. Los resultados muestran que ambos modelos son capaces en el contexto de esta predicción, aunque el MARS produce mejores estimaciones que su contraparte.

2.2.4. Prophet

Prophet es una algoritmo desarrollado por Facebook (perteneciente a Meta)[7] orientada a la predicción de series temporales, especialmente para aquellas que muestren varios componentes estacionales y tendencias no lineales. Está basado en un modelo aditivo generalizado (GAM, por sus siglas en inglés) que incluye términos lineales y no lineales de tendencia y estacionalidad, así como términos de cambio de tendencia y vacaciones. Utiliza un enfoque aditivo basado en descomposición de series temporales, que descompone la serie de datos en componentes de tendencia, estacionalidad y residuo. Asimismo, una de las características distintivas de Prophet es su capacidad para manejar datos faltantes y valores atípicos de manera robusta, lo que lo hace adecuado para datos temporales reales que a menudo tienen ruido y variabilidad. Una definición más detallada y precisa de este modelo puede encontrase en [45].

Un ejemplo de su implemetación en la literatura científica se puede encontrar en [44], que aborda el problema de la alta contaminación del aire en Seúl, Corea del Sur, debido a la industrialización y la falta de técnicas efectivas de predicción de la contaminación del aire. En este estudio, se utiliza el modelo de pronóstico Prophet para predecir la concentración de seis contaminantes del aire a corto y largo plazo. Los resultados muestran que se puede estimar con precisión la concentración de los contaminantes del aire hasta un año de antelación, superando a modelos similares propuestos en la literatura previa. Este estudio amplia los contaminantes del aire modelados de tres a seis, y aumenta el tiempo de predicción de 3 días a 1 año.

2.2.5. VAR

El modelo Vector Autorregresivo (VAR) [36] es un algoritmo estadístico utilizado para analizar la relación entre múltiples series temporales. Se dice que es autorregresivo porque asume que cada variable puede ser explicada como una función lineal de los valores pasados, tanto de los propios como de los de otras variables del sistema. Se suele utilizar para analizar la dinámica temporal de las variables del sistema, identificar patrones de interacción entre las variables y hacer pronósticos a corto plazo, especialmente cuando se trata de analizar sistemas complejos con múltiples variables interrelacionadas.

2.2.6. Gradient Boosting

El Gradient Boosting es una técnica de aprendizaje automático supervisado que aplica árboles de decisión y boosting para mejorar su rendimiento. El proceso de entrenamiento comienza con un modelo de árbol de decisión simple, que se utiliza para calcular la

[7] Meta

diferencia entre la prediccion del modelo y los datos reales (residuo). El siguiente árbol se entrena en los residuos del modelo anterior, en lugar de los datos originales, con el objetivo de capturar la parte de la información que no se ha modelado correctamente y minimizar la función de pérdida (*loss function*) de los residuos del modelo anterior. Este proceso es lo que se conoce como *boosting* y es repetido de manera secuencial, de manera que el modelo final resulta en una combinación de todos los árboles de decisión, que se ponderan según su contribución a la predicción general.

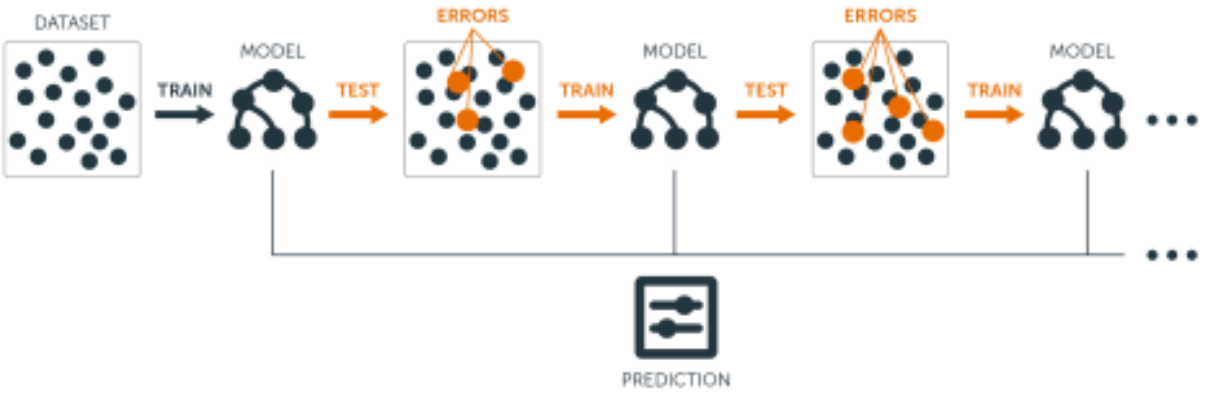

Figura 2.4: Secuencialidad del proceso de boosting [18]

La función de pérdida más comunmente utilizada es error cuadrático medio o su raíz (RMSE), pero también puede aplicarse a otras diferentes. El gradient boosting es considerado un algoritmo de descenso de gradiente, dado que utiliza un enfoque iterativo para minimizar una función de pérdida, acercándose a un mínimo local. Este algoritmo dispone de numerosos parámetros de ajuste, siendo de suma importancia escoger adecuadamente los valores adecuados para cada uno de ellos (o al menos de los de mayor relevancia). El ratio de aprendizaje es uno de ellos; como se muestra en la Figura 2.5, un ratio de aprendizaje muy bajo requeriría de numerosas iteraciones, mientras que uno alto en exceso pasaría por encima del mínimo.

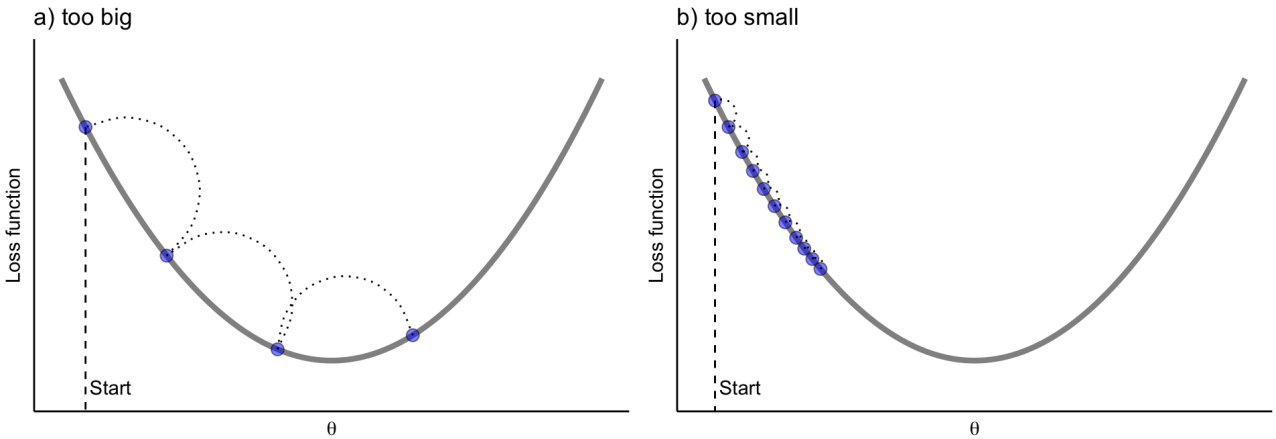

Figura 2.5: Ejemplo del ratio de aprendizaje y el descenso de gradiente [18]

Extreme Gradient Boosting

El (Extreme Gradient Boosting) (XGBoost) es un algoritmo de aprendizaje automático popular que se utiliza para resolver problemas de regresión y clasificación. A diferencia del Gradient Boosting convencional, XGBoost incorpora regularización para controlar la complejidad del modelo y evitar el sobreajuste. Además, es eficiente en términos computacionales, utiliza técnicas de manejo de datos faltantes y permite la definición de una función objetivo personalizada, lo que lo hace más robusto, rápido y flexible en muchas situaciones. Fue desarrollado por Tianqi Chen y Carlos Guestrin, de Universidad de Washintong, y su funcionamiento se describe detalladamente en el artículo [7].

En [28] se utiliza un método híbrido basado en el algoritmo de XGBoost para la predicción de series temporales de coordenadas verticales del Sistema Global de Navegación por Satélite (GNSS)[8], empleado en la monitorización de movimientos de placas tectónicas, deformaciones de presas o puentes, y mantenimiento de sistemas de coordenadas globales o regionales. El método, bautizado como *cyclic multi model-eXtreme gradient boosting* (*CMM-XGBoost*), utiliza el ajuste y predicción de un modelo de predicción para crear una nueva serie temporal de características que se utiliza para la predicción final con el modelo XGBoost. Esta técnica muestra resultados de predicción más precisos en comparación con el modelo CNN-LSTM, con reducciones del 30.23 % al 52.50 % en el error absoluto medio (MAE) y del 31.92 % al 54.33 % en la raíz del error cuadrático medio (RMSE). Se destaca que las predicciones son altamente precisas y que muestran una gran correlación respecta a las series temporales originales, lo que permite una mejor predicción del movimiento vertical de las estaciones GNSS respecto al CNN-LSTM.

2.2.7. Redes Long Short Term Memory

Las redes *Long Short-Term Memory* (LSTM) son un tipo de red neuronal recurrente (RNN) [25] que se utiliza para procesar datos secuenciales, como el lenguaje natural, la música o las series temporales. A diferencia de las RNN convencionales, que solo tienen una capa de unidades recurrentes, las redes LSTM tienen una arquitectura más compleja (véase Figura 2.6) que les permite aprender y recordar patrones a largo plazo en los datos de entrada.

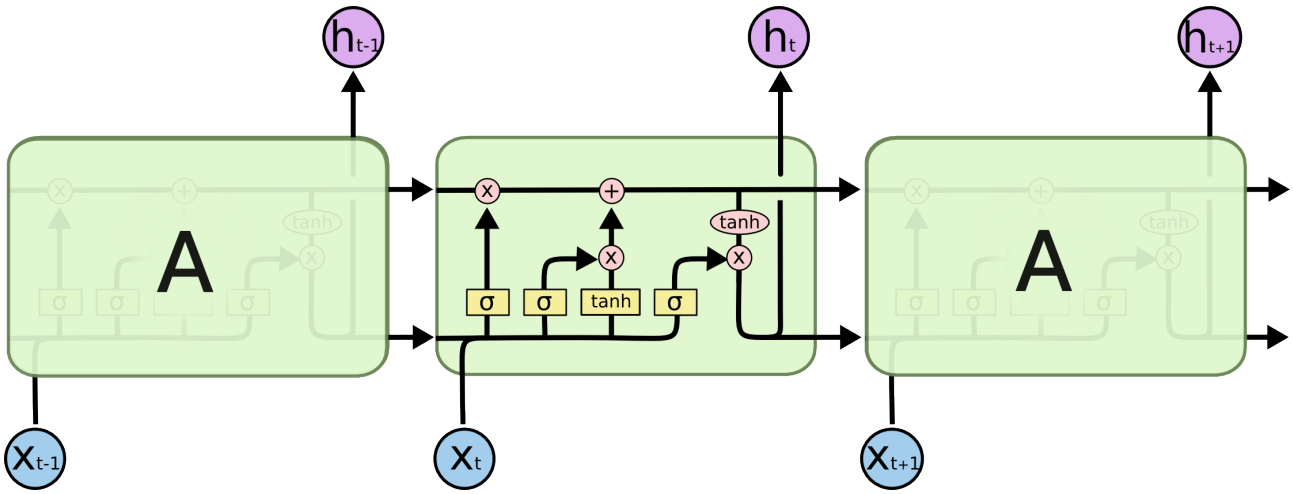

Figura 2.6: Módulos de memoria de una red LSTM [37]

[8] Sistema Global de Navegación por Satélite

Las redes LSTM están diseñadas para superar el problema del desvanecimiento de gradiente que ocurre en las RNN convencionales, que se debe a que el gradiente de error se propaga hacia atrás a través de varias capas de unidades recurrentes y se vuelve cada vez más pequeño, lo que dificulta el aprendizaje de patrones de largo plazo. En cambio, las celdas de memoria de las redes LSTM pueden mantener y actualizar la información a través del tiempo, utilizando en su interior diversas capas que controlan el flujo de información dentro y fuera de la celda de memoria.

En el estudio [40], se investigó el uso de redes LSTM en combinación con el modelo SARIMA para la predicción meteorológica en un hotel en Creta, Grecia. Se recopilaron datos de una estación meteorológica en el lugar, considerando variables como temperatura, humedad, presión atmosférica y velocidad del viento. Los resultados mostraron que el enfoque híbrido proporcionó las mejores predicciones para la temperatura y velocidad del viento, mientras que LSTM fue más efectivo en la estimación de la humedad. Estos hallazgos destacan la utilidad de las redes LSTM en la predicción meteorológica, lo que puede ser beneficioso para la toma de decisiones en la industria turística y la planificación de actividades y destinos en función del clima.

2.3. Punto de rocío

El punto de rocío es un indicador climatológico de suma importancia en un amplio abanico de ámbitos, siendo uno de ellos la gestión y monitorieo agrícola. En [51] se pretende predecir el comportamiento de variables ambientales (entre las que se encuentra el punto de rocío) en el proceso de almacenamiento del cacao. Para este propósito se empleo el modelo de Holt-Winters, aplicando un triple suavizado exponencial que tiene en cuenta tanto la estacionalidad de las variables como su tendencia. La serie temporal empleada tiene una frecuencia de muestreo mensual, por lo que la varianza y desviación típica de la serie es mucho menor que si se tuviera una frecuencia horaria. Los resultados arrojados por el modelo presentaron una precisión suficiente para predecir con antelación la presencia de hongos o moho, por lo que el autor concluye que "el modelo propuesto podría ser considerado como una excelente alternativa para el control y monitoreo del grano durante su almacenamiento".

En [2] se desarrolla un método híbrido que combina máquinas de aprendizaje extremo (ELM, por sus siglas en inglés) [29] con la aplicación de la transformada ondícula (WT)[9] para la predicción de la temperatura de rocío diaria. La efectividad del método propuesto es puesta a prueba contra varias técnicas de deep learning y máquinas de vector soporte (SVM)[10] y utilizando métricas de error como el error cuadrático medio (RMSE) o el error porcentual absoluto medio (MAPE). Se encontró la combinación de indicadores con mejor precisión fue aquella formada por el promedio de la temperatura ambiente y la humedad relativa, obteniendo de este modo un RMSE de 0.7621 y un MAPE de 6.1664 %. A la luz del método propuesto (ELM-WT) es altamente eficiente para la predicción precisa de la temperatura de rocío diaria, mejorando incluso aquellas que obtiene el ELM por sí solo.

[9] La tranformada ondícula o tranformada de wavelet es una técnica matemática que permite analizar señales en términos de funciones llamadas wavelets, que son pequeñas ondas o pulsos localizados en el tiempo y en la frecuencia. Es útil para el análisis de señales no estacionarias, por lo que es ampliamente empleada en áreas como el procesamiento de señales, análisis de datos o reconocimiento de patrones.

[10] Una máquina de vectores de soporte (SVM) es un modelo de aprendizaje automático supervisado que se utiliza para la clasificación y regresión de datos. Su es encontrar un hiperplano en un espacio N-dimensional que separe los datos de diferentes clases de manera óptima.

Por otro lado, [42] realiza un estudio sobre la capacidad de tres métodos basados en datos para modelar y estimar la temperatura de punto de rocío diaria en la estación de Tabriz, Irán. Los métodos evaluados son la programación de expresiones de genes (PEG)[11], *M5 Model Tree* (M5)[12] y Support Vector Regression (SVR). Se utilizan datos meteorológicos de temperatura promedio diaria, humedad relativa, presión de vapor real, velocidad del viento y horas de sol del período 1998 a 2016. Se definen 15 combinaciones diferentes de parámetros meteorológicos como entrada para los métodos mencionados, y se utiliza el error cuadrático medio (RMSE) y el coeficiente de determinación (R2) para analizar la precisión de los modelos propuestos. Los resultados muestran que el método M5 con cinco parámetros de entrada (temperatura, humedad relativa, presión de vapor real, velocidad del viento y horas de sol) presenta el RMSE más bajo (0.37), seguido por SVR con dos parámetros de entrada (temperatura y humedad relativa), y GEP con tres parámetros de entrada (temperatura, humedad relativa y horas de sol). El estudio concluye que el modelo M5 es el más preciso en la estimación de la DPT en comparación con los otros modelos evaluados.

En [33] se aplicó un sistema de inferencia neuro-borroso adaptativo (ANFIS, por sus siglas en inglés) para la predicción diaria del punto de rocío en dos ciudades de Irán. ANFIS es un enfoque de modelado híbrido que combina técnicas de redes neuronales para realizar el aprendizaje y adaptación y lógica borrosa para representar la incertidumbre en los datos. Se emplea un registro de 7 años de longitud para analizar la influencia de las 8 variables meteorológicas que lo conforman en la temperatura de rocío. Los resultados indican que para ambas ciudades, el presión del vapor de agua (V_p) es la variable más relevante, mientras que la humedad relativa (R_h) la que menos. Se concluye que la mejor combinación de variables se da con la temperatura mínima (T_{min}) y el V_p, obteniendo un RMSE de 0.5445 en la fase de testeo.

Por su parte, [55] evalúa la capacidad de dos modelos, regresión lineal múltiple (MLR, por sus siglas en inglés)[13] y red neuronal de retroalimentación hacia adelante con algoritmo de Levenberg-Marquardt (LM-NN)[14], para estimar la temperatura de punto de rocío a una hora en el futuro. Se utilizaron cinco datos meteorológicos: temperatura del aire (TA), humedad relativa (RH), presión (Pr), velocidad del viento (WS) y dirección del viento (WD), junto con un dato conceptual para las condiciones meteorológicas (WT), en 29 combinaciones diferentes para desarrollar los modelos. Por otro lado, para la evaluación del desempeño se los modelos se empleó el error cuadrático medio, el error absoluto medio y el coeficiente logarítmico absoluto de eficiencia de Nash-Sutcliffe (NSE) [31]. Los resultados mostraron que el punto de rocío (TD) no pudo ser pronosticado utilizando únicamente parámetros individuales. La comparación de los resultados de las 29 estrategias de entrada para MLR/LM-NN indicaron que incluir datos anteriores de TD(t-1) y TD(t-2), así como la cantidad previa de TA(t-1), tuvo el mayor impacto en el rendimiento de ambos modelos. Sin embargo, en general, los mejores resultados se obtuvieron cuando los modelos aprovecharon todos los parámetros de entrada. Según los

[11] La programación de expresiones de genes (PEG) es un algoritmo evolutivo que genera estructuras arbóreas donde los niveles de expresión de los genes se consideran características y las condiciones biológicas o temporales son etiquetas.

[12] El M5 Model Tree es un modelo de aprendizaje automático que combina la regresión y los árboles de decisión. Utiliza diferentes técnicas de poda y ajuste para mejorar la precisión y la generalización del modelo.

[13] La regresión lineal múltiple es un método estadístico que permite modelar la relación entre una variable dependiente y un conjunto variables independientes, mediante una ecuación lineal.

[14] Red neuronal de retroalimentación hacia adelante con algoritmo de Levenberg-Marquardt

parámetros estadísticos de evaluación del rendimiento, el modelo LM-NN fue el de mejor desempeño, obtenido un RMSE mínimo de 0.904.

Finalmente, una implementación utilizando *extreme gradient boosting* para la predicción del punto de rocío es analizada en [12]. En concreto, se aborda un método híbrido basado en XGBoost y el algoritmo de optimización del saltamontes (*Grasshopper Optimisation Algorithm* (*GOA*))[15], para la estimación horaria y diaria un paso a futuro de la temperatura del punto de rocío. Asimismo, se comparan los resultados obtenidos mediante dicha ténica con los modelos de XGBoost y random forest (RF). Se recolectaron datos meteorológicos, como la presión de vapor (e_a) , la temperatura máxima del aire (T_{max}), la humedad relativa máxima (RH_{max}) o la presión atmosférica (P_a), durante el período 2016-2019 a escalas de tiempo diarias y por hora de la estación Sijiqinglin en China, para entrenar, probar y validar cada modelo. Las predicciones efectuadas a nivel horario mostraron que el modelo GOA-XGBoost tuvo el mejor rendimiento y que todos los modelos mostraron una mejor precisión y estabilidad cuando la entrada era e_a, con un RMSE de 0.164 en el caso del GOA-XGBoost. Por otro lado, este modelo presentó errores más significativos cuando las entradas eran T_{max}, T_{min} y el tiempo, obteniendo un RMSE de 7.487. Se destaca que la variabilidad de los datos meteorológicos fue menor a escala horaria que a escala diaria y que las escalas horarias son más adecuadas para evaluar los efectos de las simulaciones en situaciones extremas.

2.4. Contribuciones a la literatura

El proyecto planteado tiene como objetivo el desarrollo y evaluación de técnicas punteras de predicción del punto de rocío para su uso en un telescopio astronómico. Tal como se ha señalado, hasta ahora la mayoría de los estudios revisados en la literatura se enfocan en la predicción del punto de rocío a corto plazo y en espacios temporales más amplios que en el escenario bajo estudio. A diferencia de lo encontrado, este proyecto se enfoca en el desarrollo de un modelo de predicción del punto de rocío con una anticipación de 8 horas a futuro, lo cual presenta un enfoque único repecto a lo ya existente. El planteamento de mayor parecido de entre los estudiados es el que contemplan [55] y [12], analizando el pronóstico de la temperatura del punto de rocío con una ventana de una hora a futuro.

Asimismo, la integración de este modelo de predicción del punto de rocío en el sistema del telescopio para su mantenimiento predictivo representa un objetivo disruptivo en el campo de la instrumentación astronómica, dado que dicho objetivo no es planteado en ningún otro documento de la literatura. Esto permitirá una planificación más eficiente de las operaciones de observación y mejorará la eficiencia y seguridad del telescopio.

[15] El Grasshopper Optimisation Algorithm (GOA) es una técnica metaheurística inspirada en el comportamiento social de un enjambre de saltamontes. Utiliza una serie de operadores matemáticos y de movimiento para actualizar la posición de los saltamontes virtuales en cada iteración, y una función de aptitud para evaluar la calidad de cada solución candidata.

Capítulo 3

Metodología y herramientas utilizadas

3.1. Esquema de trabajo

Este capítulo describe la metodología y las herramientas empleadas en el proyecto de predicción de la temperatura de punto de rocío. Con este objetivo en mente, se presenta un esquema de la metodología utilizada (ver Figura 3.1).

En primer lugar, se importaron las series temporales de variables meteorológicas para el período de estudio, seguido de un análisis previo para entender las características de las variables y su relación con la variable objetivo. Luego, se realizó una limpieza de outliers para eliminar los valores atípicos y se manejaron los valores faltantes utilizando técnicas de interpolación e imputación. Realizar la limpieza de outliers antes del manejo de valores nulos es conveniente para evitar la introducción de outliers en los datos imputados. Posteriormente, se seleccionaron las características relevantes para la predicción utilizando técnicas de filtro y no supervisadas.

En la etapa de predicción, se aplicaron diferentes técnicas de modelado univariable y multivariable para predecir la temperatura de punto de rocío. En la predicción multivariable, se utilizaron las variables seleccionadas previamente como regresores para mejorar la precisión de la predicción.

Finalmente, se compararon los resultados obtenidos en las etapas de predicción univariable y multivariable para determinar la eficacia de la inclusión de regresores externos en la predicción de la temperatura de punto de rocío.

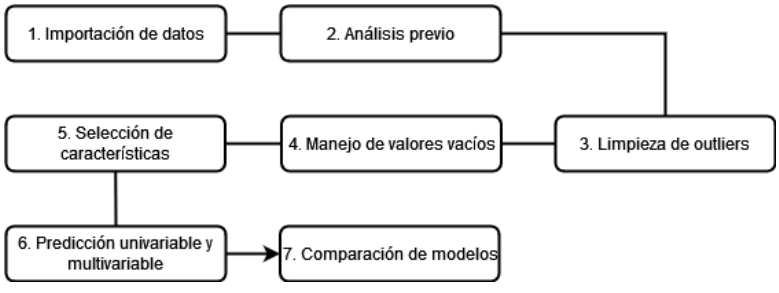

Figura 3.1: Esquema general de trabajo.

3.2. Conjuntos de datos

El IAC proporcionó diversos conjuntos de datos con los que trabajar, varios de ellos pertenecientes a la Agencia Estatal de Meteorología (AEMET). Sin embargo, ninguno de

estos conjuntos de datos está disponible para el acceso público ni se pueden compartir.

- *Meteo*. Contiene variables meteorológicas medidas en intervalos de 10 minutos, tales como velocidad del viento, dirección del viento, temperatura promedio, humedad relativa, presión y estabilidad del aire. También incluye el punto de rocío (DP) que es la característica que se persigue predecir en este trabajo. Para mayor detalle respecto a sus variables, véase Tabla A.1.

- *PWVMo (Precipitable Water Vapour Monitor)*. Este conjunto de datos incluye mediciones meteorológicas tomadas a intervalos regulares de 30 minutos. La variable principal es el vapor de agua precipitable (en inglés, Precipitable Water Vapour o PWV) [23] tanto en su valor bruto como corregido. Esta medida se refiere a la profundidad (en milímetros) que alcanzaría el vapor de agua contenido en un punto determinado si se condensara y precipitara sobre la atmósfera terrestre. Para mayor detalle respecto a sus variables, véase Tabla A.2.

- *INSDIA*. Se centra en la medición de la insolación solar [35], variable definida como las suma de los intervalos temporales durante los que la radiación solar directa supera cierto umbral. En el datset original se presenta esta característica en intervalos de tiempo no regulares, como una suma diaria y como un porcentaje. Para mayor detalle respecto a sus variables, véase Tabla A.4.

- *PCPDIA*. Contiene entradas diarias con información sobre la precipitación. Las variables incluyen la cantidad de precipitación acumulada en diferentes intervalos de tiempo (desde 00 hasta 24 horas, de 00 a 07, de 07 a 13, de 13 a 18 y de 18 a 24 horas), así como la cantidad máxima de precipitación en diferentes intervalos de tiempo (10, 20, 30, 60 minutos, 2, 6 y 12 horas). Además, incluye la intensidad y duración de la precipitación, así como la hora, la dirección del viento en el momento de la máxima precipitación y el tipo de precipitación que se produjo. También se indica la cantidad de precipitación en las horas anteriores a la actual. Para mayor detalle respecto a sus variables, véase Tabla A.3.

El volumen de datos proporcionado varía según el conjunto de datos. *Meteo* dispone de 10 años de registros, desde el año 2003 hasta el 2012. Por otro lado, los conjuntos de la AEMET disponen de poco más de 3 años de registros, desde el año 2009 hasta el 2012 y *PWVMo* abarca desde el año 2009 hasta el 2013. En consecuencia, para las posteriores etapas de han utilizado datos de un volumen de 3 años, desde febrero del 2009 hasta el mismo mes del año 2012.

A la hora de garantizar que los resultados que se obtengan sean aplicables a los TTNN, es de extrema importancia que las mediciones hayan sido tomadas en un lugar cercano a la ubicación de los telescopios. Esto se debe a que las variables atmosféricas pueden variar significativamente de una ubicación a otra, incluso dentro de una misma región geográfica. Al tomar las mediciones en la misma ubicación que el telescopio, se asegura que se están teniendo en cuenta las condiciones específicas que afectan al mismo y no las condiciones generales de la región. Por ello, cabe mencionar que las mediciones de los previamente descritos conjuntos de datos han sido tomadas en el Observatorio Atmosférico de Izaña. Este centro está localizado a la misma altura y a menos de kilómetro y medio de distancia del Observatorio Astronómico del Teide, ubicación de los telescopios de interés de este trabajo (véase Figura 3.2).

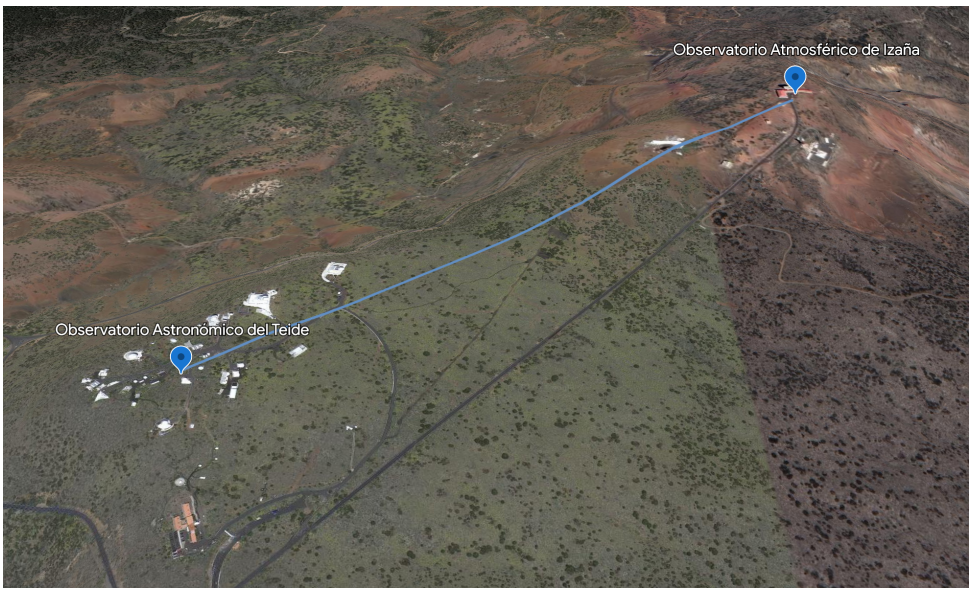

Figura 3.2: Comparación de la ubicación del Observatorio Astronómico del Teide (IAC80 y el Carlos Sánchez) y la ubicación del Observatorio Atmosférico de Izaña. Google Earth

3.3. Importación de datos

Elaborar un proceso de importación de los datos que sea reproducible con facilidad y adaptable a distintos conjuntos de datos es de suma importancia para el desarrollo de los apartado posteriores. La Figura 3.3 esquematiza los contenidos que se desarrollan en esta sección.

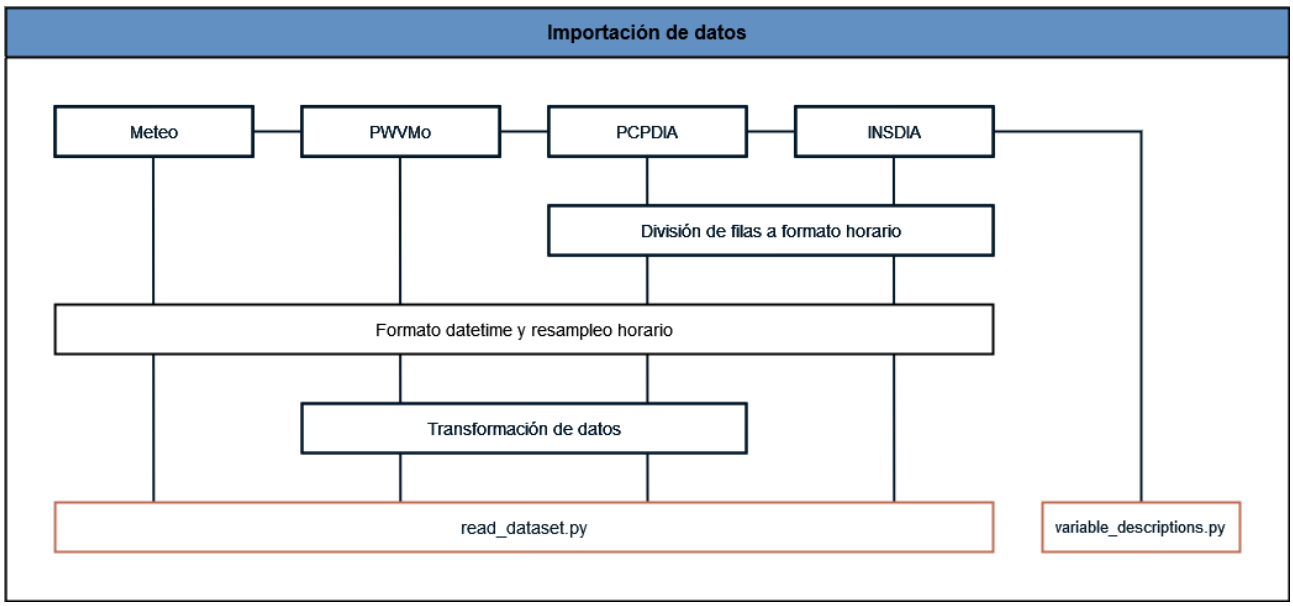

Figura 3.3: Proceso de importación de datos

Cada uno de los conjuntos de datos se encuentra alojado en archivos distintos, los cuales varían en cuanto a su formato, pudiendo ser archivos .dat o .txt. Además, se encuentran distribuidos de diferentes maneras: mientras que algunos tienen un archivo diferente para cada mes, otros están separados por años o incluso se encuentran en un solo archivo sin separación que contiene varios años de muestras. Es importante destacar que los conjuntos de datos mencionados anteriormente se presentan en distintas

frecuencias de muestreo, siendo la frecuencia más baja de un día.

Por otro lado, algunos de ellos requieren de un preprocesado para que los datos estén en el formato indicado. Es el caso de INSDIA y PCPDIA que, en vez de tener una entrada cada hora, tienen una cada día, que contiene información acerca de cada una de la horas del mismo (véase Listing E.1).

3.4. Análisis

Realizar un análisis y visualización del conjunto de datos es una tarea fundamental en cualquier proyecto de ciencia de datos. Este análisis proporciona una comprensión detallada de la estructura y características [50] del conjunto de datos, lo que a su vez ayuda a identificar problemas y patrones interesantes.

3.4.1. Componentes de las series temporales

Componente tendencial o tendencia

Se refiere a la evolución de la serie temporal a largo plazo. Su presencia no es siempre evidente en los datos, dado que, a veces, los datos pueden ser ruidosos o presentar fluctuaciones aleatorias que ocultan la tendencia subyacente.

Componente estacional o estacionalidad

Propiedad que recoge los patrones que se producen en periodos menores o iguales a un año y que se repiten en una frecuencia constante. Estas fluctuaciones se deben principalmente a dos factores: estaciones o condiciones climáticas (ej: temperatura, Figura 3.4) y convenciones artificiales establecidas por el ser humano (ej: consumo).

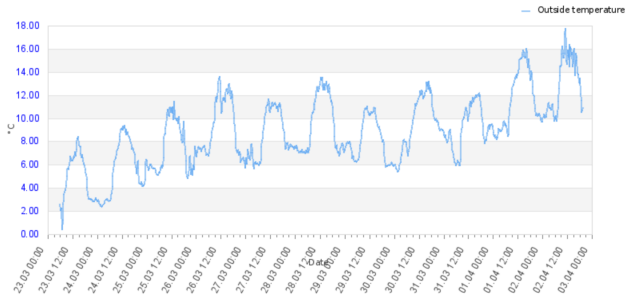

Figura 3.4: Temperatura media en el Observatorio STELLA. Archivo Stella

Componente cíclica

Refleja aquellas oscilaciones con una periodicidad superior a la anual. En estas variaciones, un periodo completo constituye un ciclo, pero un ciclo no tiene un periodo de tiempo específico. Un ejemplo de este suceso se da en índices económicos de prosperidad o recesión, por ejemplo.

Componente residual o residuo

Es el error o residuo resultante de la extracción de la tendencia y estacionalidad de la serie, es decir, aquella parte que no puede ser explicada con el modelo utilizado para su ajuste. Se puede interpretar como el ruido o variación aleatoria que no puede explicarse utilizando la información disponible. Permite detectar anomalías o comportamientos erráticos y puede proporcionar información acerca de la calidad de un modelo de análisis.

3.4.2. Estacionariedad

Una serie temporal es estacionaria si sus propiedades estadísticas (como la media o varianza) no cambian con el tiempo. Conocer la estacionariedad de una serie es de vital importancia en la predicción, dado que muchos métodos de pronóstico asumen que una serie temporal es estacionaria, o que puede hacerse estacionaria tras aplicar una diferenciación. [43]

Para evaluar esta propiedad, se emplean pruebas de raíz unitaria, que son unos métodos estadísticos utilizados para determinar si una serie de tiempo tiene una raíz unitaria, lo que indica la presencia de una tendencia determinística en los datos. Una serie de tiempo con una raíz unitaria es calificada como no estacionaria, lo que significa que su media y su varianza cambian a lo largo del tiempo. Estas pruebas evalúan la hipótesis nula de que hay una raíz unitaria en la serie de tiempo y determinan si esta hipótesis puede ser rechazada o no. Un ejemplo de ellas se encuentra en la prueba de Dickey-Fuller aumentada (ADF) o la prueba de Phillips-Perron (PP).

Por tanto, si una serie de tiempo tiene una raíz unitaria, se puede asumir que ésta presenta una tendencia estocástica y que, en consecuencia, es conveniente transformarla para eliminar dicha tendencia. Para ello, a menudo se emplea una transformación por diferencia o diferenciación, que se realiza restando a la observación actual la observación anterior, véase Figura 3.1.

$$diferencia(t) = obs_t - obs_{t-1} \tag{3.1}$$

Cicloestacionariedad

A diferencia de un proceso estacionario, un proceso cicloestacionario presenta propiedades estadísticas que varían periódicamente con el tiempo. Esto significa que las características del proceso, como su media, varianza y autocorrelación, cambian de forma repetitiva a lo largo del tiempo en función de una frecuencia fundamental o período.

La cicloestacionariedad es un concepto importante en el análisis de datos meteorológicos. En general, las variables meteorológicas, como la temperatura, la presión, la humedad y la velocidad del viento, pueden presentar patrones repetitivos a lo largo del tiempo. Por ejemplo, la temperatura en un lugar determinado puede variar a lo largo del día, alcanzando un valor máximo en el mediodía y un valor mínimo durante la noche. Sin embargo, si observamos los valores medios diarios de temperatura a lo largo de varios días, estos valores no cambian mucho. Es decir, la media de la temperatura diaria no cambia con el tiempo, sino que varía con la posición en el tiempo.

La detección de este fenómeno en los datos meteorológicos es importante para la modelización y la predicción del clima y del tiempo. Los modelos que utilizan la cicloestacionariedad pueden proporcionar una mejor descripción de los patrones cíclicos en los datos, lo que puede mejorar la precisión de los pronósticos.

3.4.3. Correlación

La correlación en series temporales es una medida estadística que indica la relación entre dos variables en función del tiempo. Se distinguen principalmente dos tipos de correlación:

- *Correlación lineal*. Se refiere a la medida en que dos variables están relacionadas de forma proporcional. Se puede calcular utilizando el coeficiente de correlación de Pearson.
- *Correlación no lineal*. Refleja la medida en que dos variables están correlacionadas de forma no proporcional, de manera que su relación no puede ser descrita por una líneal recta en un gráfico de dispersión ni por el coeficiente de correlación de Pearson.

Autocorrelación y Autocorrelación Parcial

La autocorrelación mide la relación entre una serie de tiempo y una versión retrasada de sí misma, sin tener en cuenta variables intermedias que puedan influir en su relación [41]. Formalmente el coeficiente de correlación entre dos variables de una misma serie es conocido como ACF (AutoCorrelation Function). Por otro lado, en el caso de que nos interese medir la correlación directa entre dos variables, tendríamos que utilizar la autocorrelación parcial (PACF, por sus siglas en inglés).

Este concepto crucial en el análisis de series temporales meteorológicas, dado que éstas a menudo exiben patrones de correlación que se extienden a través del tiempo. Su detección es de gran utilidad para la detección de patrones y tendencias y para la modelización de sus características. Un ejemplo de ello es la temperatura, que a menudo muestra una autocorrelación positiva en puntos cercanos o incluso en muestras recogidas sobre la misma hora de días anteriores.

Correlación Cruzada

La correlación cruzada es una técnica estadística que evalúa la relación entre dos series temporales diferentes. En vez de medir la asocación entre dos variables en una única muestra, mide la relación existente entre ellas a través del tiempo [20]. De esta manera es posible encontrar conexiones entre dos variables separadas en el tiempo y utilizar una de ellas como predictor para la otra.

3.5. Limpieza de anomalías

Una anomalía o outlier en un conjunto de datos se define como un valor atípico que se desvía significativamente de los demás valores del conjunto. Dado que las características climáticas pueden fluctuar significativamente en cortos períodos de tiempo, la presencia de outliers es común en ellos (especialmente en escenarios climatológicos anormales).

Además, la presencia de outliers puede ser causada por la acción humana, como errores de medición, mala calibración de los instrumentos o incluso fraudes intencionales. Por ejemplo, un sensor de temperatura mal calibrado puede registrar valores fuera de rango, o un registro manual incorrecto puede introducir errores en el conjunto de datos.

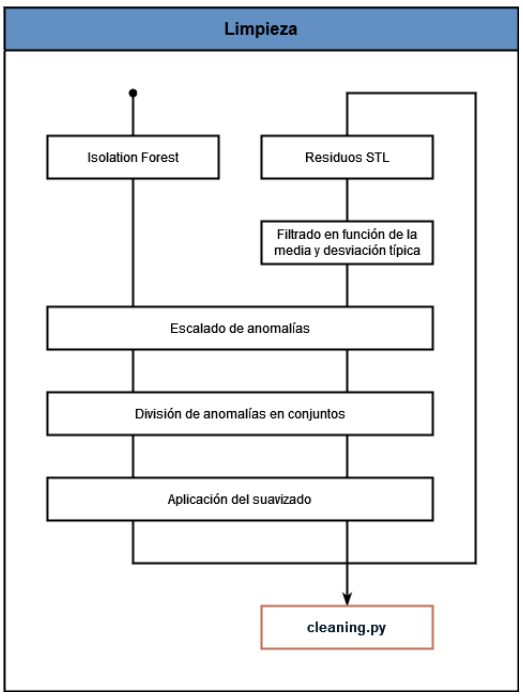

Figura 3.5: Proceso de limpieza de outliers.

En el contexto de este trabajo se emplearon dos técnicas de detección de anomalías: *Isolation Forest* y análisis de residuos STL.

3.5.1. Isolation Forest

Isolation Forest (IF) es un método no supervisado para la detección de anomalías que está basado en el funcionamiento de los árboles de decisión. Para lograr esto, el algoritmo construye un conjunto de árboles de decisión aleatorios, en los cuales se seleccionan de manera aleatoria atributos y valores para separar los datos. El proceso de construcción de árboles se repite hasta que todas las instancias estén aisladas. En este punto, se calcula la puntuación anómala de cada instancia mediante la profundidad promedio de los árboles en los cuales se encuentra.

3.5.2. Residuos del STL

La descomposición STL (Seasonal and Trend decomposition using Loess) es una técnica que se utiliza para descomponer una serie temporal en tres componentes: estacionalidad, tendencia y residuo. Para detectar anomalías utilizando este procedimiento, se compara la desviación típica de los valores de la componente residual con un umbral de referencia [5]. Si uno de estos valores supera dicho límite, podemos considerarlo anomalía. La principal ventaja de está técnica es su robustez frente a valores fuertemente atípicos o que, a pesar no tomar un valor de por sí anormal, representan una desviación inusual respecto al resto de la serie. Sin embargo, si esta desviación es parte de una tendencia formada por varios valores adyacentes, su capacidad de detección es menor.

3.6. Imputación de valores vacíos

Cuando se trabaja con series temporales, es común encontrarse con valores vacíos producidos por problemas técnicos, errores de entrada de datos o la falta de registro en determinados momentos. Si estos valores vacíos no se imputan, pueden generarse problemas al momento de predecir el comportamiento futuro de la variable, ya que la gran mayoría de los modelos no permiten su existencia. Eliminar filas con valores faltantes puede ser una solución rápida y sencilla, pero en el contexto de las series temporales, esto puede ser problemático para la predicción debido a su estructura de dependencia temporal. Por lo tanto, la imputación es mucho mejor opción para el manejo de datos faltantes en series temporales.

En este trabajo se contemplan los siguientes métodos (véase Figura 3.6):

- Imputación con ceros: Aunque este método no es adecuado para la gran mayoría de las series, resulta de gran conveniencia para las variables de PCPDIA. Esto se debe a que las precicipitaciones solo se producen en contados momentos a lo largo del año, por lo que es factible asumir que las entradas no registradas tomen valor 0.
- Interpolación: La interpolación lineal es un método utilizado para estimar un valor o conjunto de valores desconocidos entre dos puntos conocidos en una línea recta. Suele emplearse para hacer predicciones simples y aproximadas de valores dentro de un rango conocido.
- Uso de modelos de predicción: Este último procedimiento se basa en emplear la predicción de modelos predictivos sobre el propio conjunto de entrenamiento para la imputación los datos faltantes.

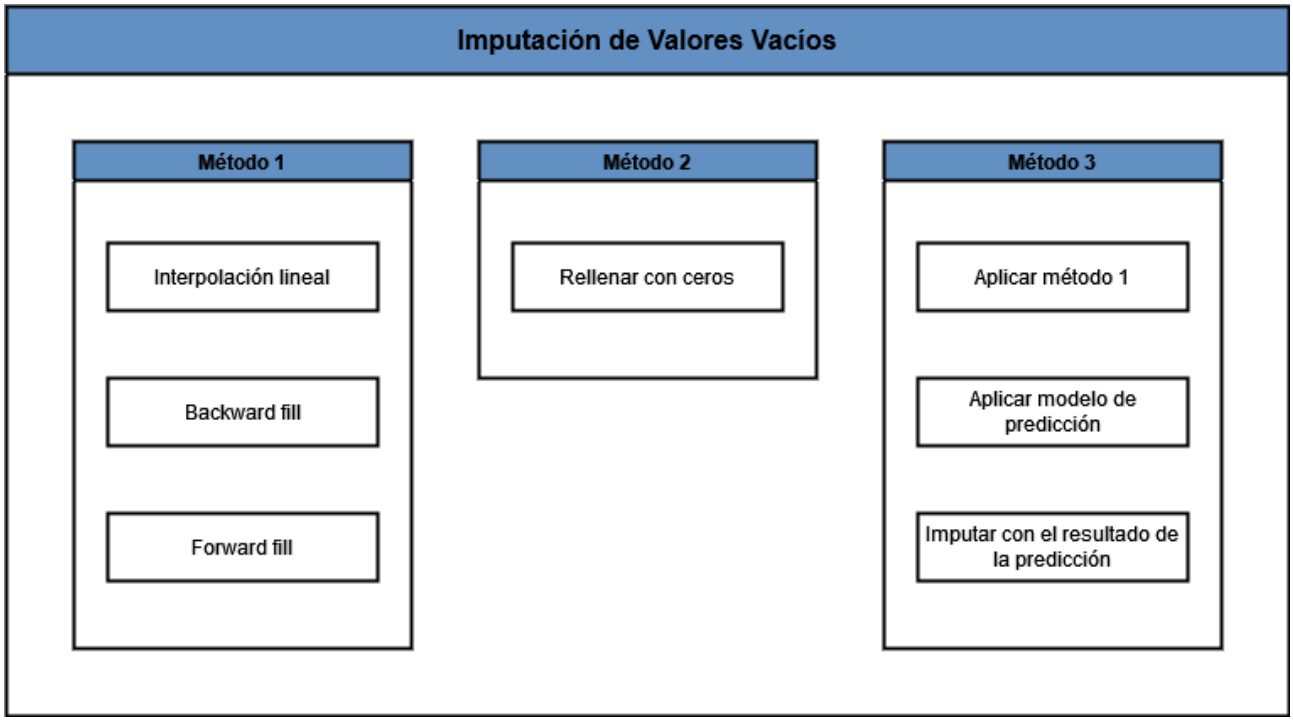

Figura 3.6: Proceso de imputación de valores vacíos.

3.7. Selección de características

La selección de características es una técnica utilizada para reducir la dimensionalidad de un conjunto de datos. El objetivo es seleccionar un subconjunto de características relevantes y significativas que permitan obtener una representación más eficiente y precisa del conjunto de datos original. No solo se persigue una reducción de complejidad, si no también mejorar la capacidad de interpretación de los modelos, dado que algunos son resistentes a regresores no relevantes o a conjuntos redundantes [26].

Se distinguen dos tipos de métodos de selección de características: los supervisados y los no supervisados. Esta distinción se debe al criterio de eliminación; los supervisados tienen en cuenta la variable objetivo, seleccionando aquellas características que mejoren la precisión del conjunto o su complejidad. Por otro lado, los no supervisados basan su decisión en la correlación cruzada entre los predictores o la distribución de los mismos. Para una visión esquemática del proceso, véase Figura 3.7.

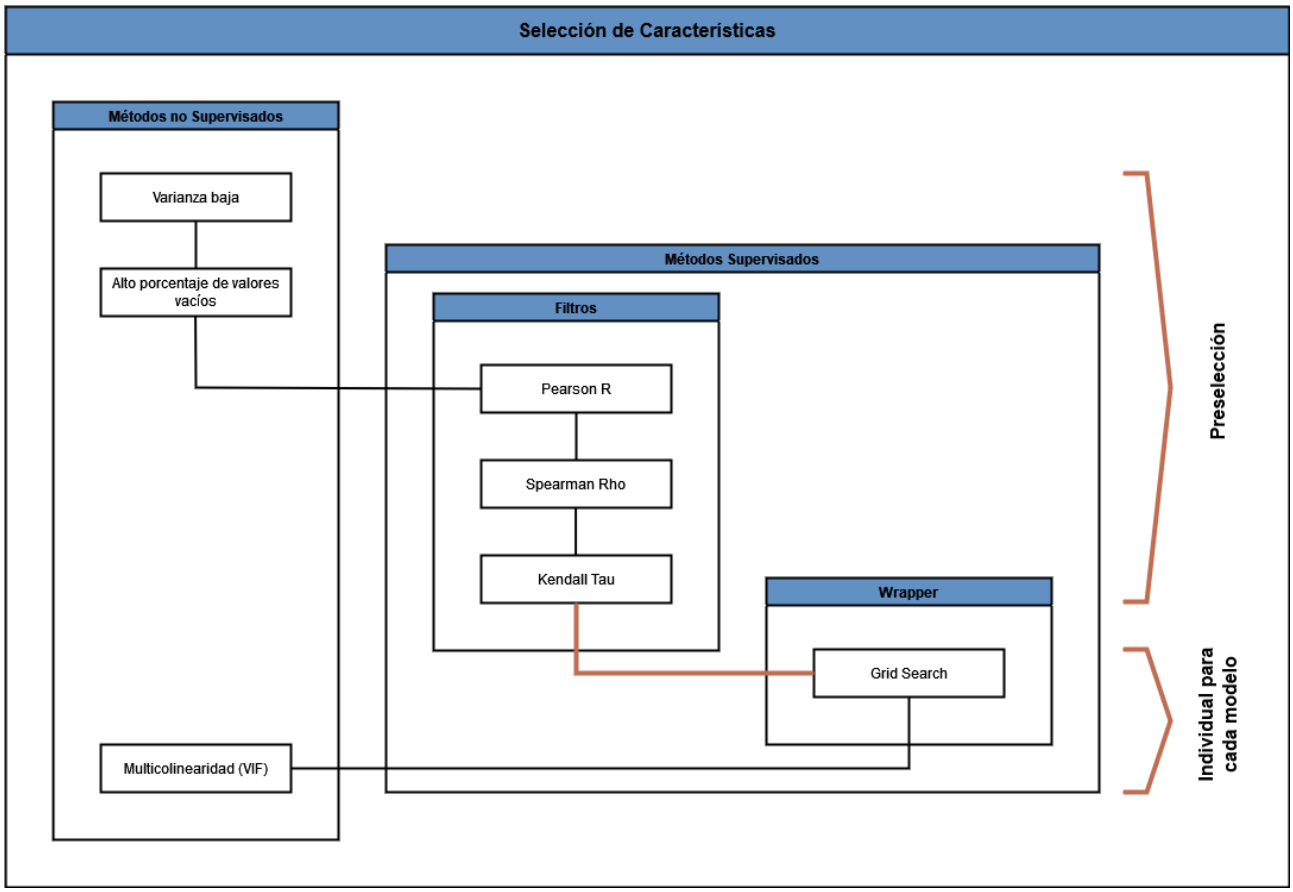

Figura 3.7: Proceso de selección de características.

3.7.1. Métodos no supervisados

- Eliminación de características de varianza cercana a cero. Esto significa que sus valores son muy similares o constantes a lo largo de las instancias del conjunto de datos, por lo que no aporta información valiosa para el modelo y puede ser eliminada sin afectar significativamente su desempeño.

- Eliminación de características con un alto porcentaje de valores nulos. Si una variable tiene un alto porcentaje de valores nulos, es probable que la aplicación

de técnicas de imputación genere incoherencias que aumenten la complejidad del modelo y, por tanto, afecte a su rendimiento.

- Análisis de la multicolinearidad. Se da cuando existe una fuerte correlación entre diferentes variables independientes, lo que puede afectar al rendimiento de modelos predictivos como los árboles de regresión. Para este propósito se empleó el factor de inflación de la varianza (VIF, por sus singlas en inglés) [46], que permite cuantificar el nivel que multicolinearidad existente en un conjunto.

3.7.2. Métodos supervisados

Métodos de filtrado

Evalúan individualmente cada uno de los predictores previo al entrenamiento de algún modelo en específico con el fin de determinar si la relación existente con la variable objetivo supera cierto umbral previamente determinado. Para dicha evaluación se pueden emplear distintos tipos de métricas; teniendo en cuenta que tanto la entrada como la salida son numéricas, se planteó el uso de las siguientes:

- Coeficiente de correlación de Pearson (*Pearson R*). Asume que las variables están normalmente distribuidas y verifica si existe una relación lineal entre ellas.
- Coeficiente de correlación de Spearman (*Spearman Rho*). Se presenta como una mejor alternativa, dado que es capaz de capturar también relaciones no lineales. Sin embargo, al contrario que el coeficiente de correlación de Pearson, compara las variables en función de su posición relativa, lo que conlleva a cierta pérdida de información.
- Coeficiente de correlación de rango de Kendall (*Kendall Tau*). Su planteamiento es similar al *Spearman Rho*, pero su formulación lo hace más resistente a outliers [38]. Dado que tratamos con unos datos que presentan anomalías frecuentes, resulta ser el más conveniente.

Métodos Wrapper

Los métodos wrapper son un enfoque de selección de características que se basa en el uso de un modelo de aprendizaje automático para evaluar la importancia de cada característica en la predicción de la serie temporal. Para ello, se utilizan diferentes subconjuntos de características para entrenar un modelo de aprendizaje automático y se evalúa el desempeño del modelo en un conjunto de datos de prueba. Este proceso se repite varias veces con diferentes subconjuntos de características hasta encontrar el subconjunto que mejor se ajuste al modelo.

Estos métodos son computacionalmente costosos, ya que requieren entrenar y evaluar varios modelos de aprendizaje automático con diferentes subconjuntos de características. Es por ello, que se reliza una selección preliminar de características con los métodos no supervisados y los métodos de filtro. En el marco de este trabajo, se ha decidido utilizar una ténica conocida como *Grid Search* (búsqueda en cuadrícula). Consiste en definir una cuadrícula de valores posibles para los hiperparámetros o regresores y evaluar el rendimiento del modelo para cada combinación de valores.

3.8. Métricas de error

3.8.1. Métricas para la predicción

- Raíz del error cuadrático medio (RMSE, por sus siglas en inglés) [12]: Raíz de la media del cuadrado de la diferencia entre el valor real y el predicho (veáse Figura 3.2). Es una métrica que penaliza con mayor intensidad los errores graves frente a los pequeños. Representa el error en la misma escala que los datos originales, lo que la hace más interpretable.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \overline{y_i})^2} \tag{3.2}$$

- Media del error absoluto (MAE, por sus sigles en inglés) [28]: Es simplemente la media del error en cada uno de los puntos (3.3). No da ningún trato especial a los outliers o errores grandes.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \overline{y_i}| \tag{3.3}$$

- Distancia de deformación dinámica del tiempo (DTW, Dynamic Time Warping): La distancia DTW es una medida de distancia entre dos series temporales que, al contario que la distancia euclídea, tiene en cuenta la posibilidad de que los patrones en ambas series se produzcan con diferentes velocidades y escalas de tiempo. Por tanto, permite cuantificar la medida en que la forma de la predicción se ajusta a la real [8]. Sin embargo, es conveniente utilizarla junto con alguna de las dos anteriores métricas, con el fin de que los resultados sean más fáciles de interpretar [15]. Aunque esta métrica no es tan conocida y ampliamente usada como las previamente mencionadas, ha sido empleada de forma exitosa en diversos artículos [32] [54]).

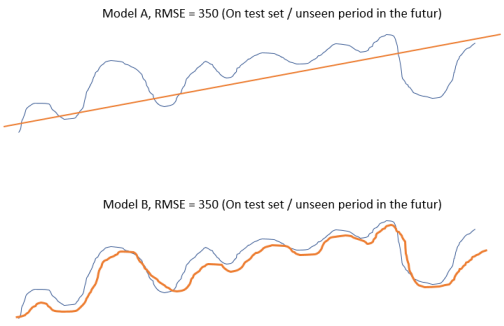

Figura 3.8: Mismo RMSE, teniendo la segunda predicción una forma mucho más parecida a la original.

3.8.2. Métricas para el intervalo de predicción

- Porcentaje de Cubrimiento (PC): Porcentaje de los puntos reales que caen dentro del intervalo del intervalo de predicción del modelo.

- Media del Error de Cubrimiento [17] (MEC): Métrica que puntúa la calidad de un intervalo de confianza en base a su tamaño y sus errores. Es especialmente útil porque penaliza aquellos modelos que, adoptando una gran amplitud de su intervalo, consiguen un alto porcentaje de cubrimiento.

$$(pu - pl) + \frac{2}{\alpha}(pl - y)1(y < pl) + \frac{2}{\alpha}(y - u)1(y > pu)$$

3.9. Predicción

La predicción univariable, ilustrada en la Figura 3.9, se centra en predecir el valor del punto de rocío en función de su propia serie temporal histórica. Se han considerado varias técnicas para este propósito, tales como el suavizado exponencial simple y triple, modelos ARIMA y SARIMA, Prophet, Gradient Boosting y Random Forest. El objetivo de la predicción univariable es comprobar si su precisión a 8 horas en el futuro es mayor que utilizando regresores externos. En caso contrario, se contempla su uso para predicciones a mayor plazo que puedan ser utilizadas como referencia, teniendo en cuenta que su precisión podría ser limitada.

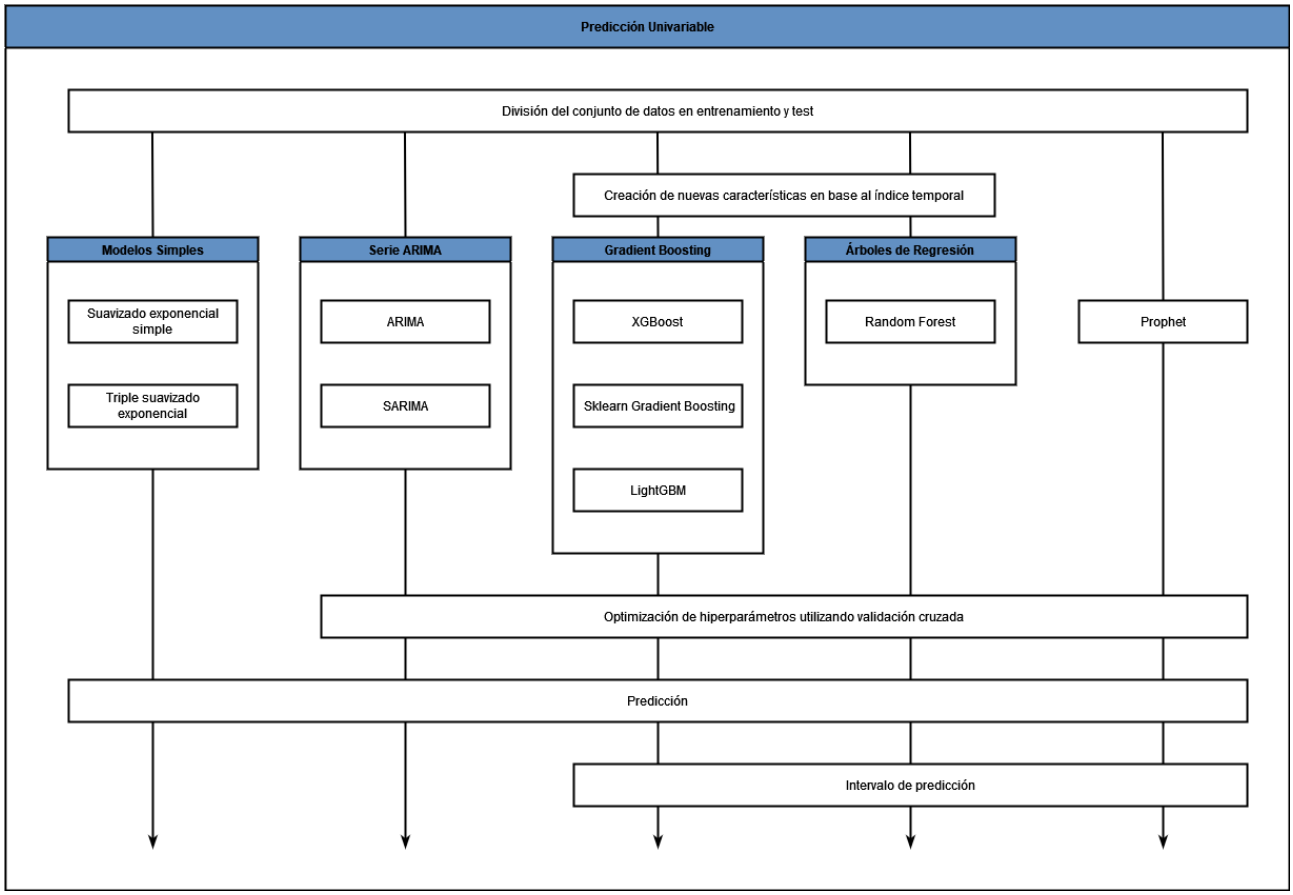

Figura 3.9: Proceso para la predicción univariable.

Por otro lado, la predicción multivariable (véase Figura 3.10) tiene como objetivo predecir con precisión el punto de rocío 8 horas a futuro en función de múltiples variables climatológicas. Las técnicas consideradas para la predicción multivariable son SARIMAX, Prophet, Random Forest y Gradient Boosting. Se pretende realizar una selección de

regresores, tras la preselección llevada a cabo en los apartados de análisis y selección de características. En este caso, se efectuó una selección individual de regresores para cada modelo utilizando validación cruzada. La validación cruzada es una técnica que permite evaluar la capacidad de generalización del modelo mediante la partición del conjunto de datos en subconjuntos de entrenamiento y prueba, permitiendo verificar si el modelo es capaz de predecir con precisión datos nuevos no vistos durante el entrenamiento.

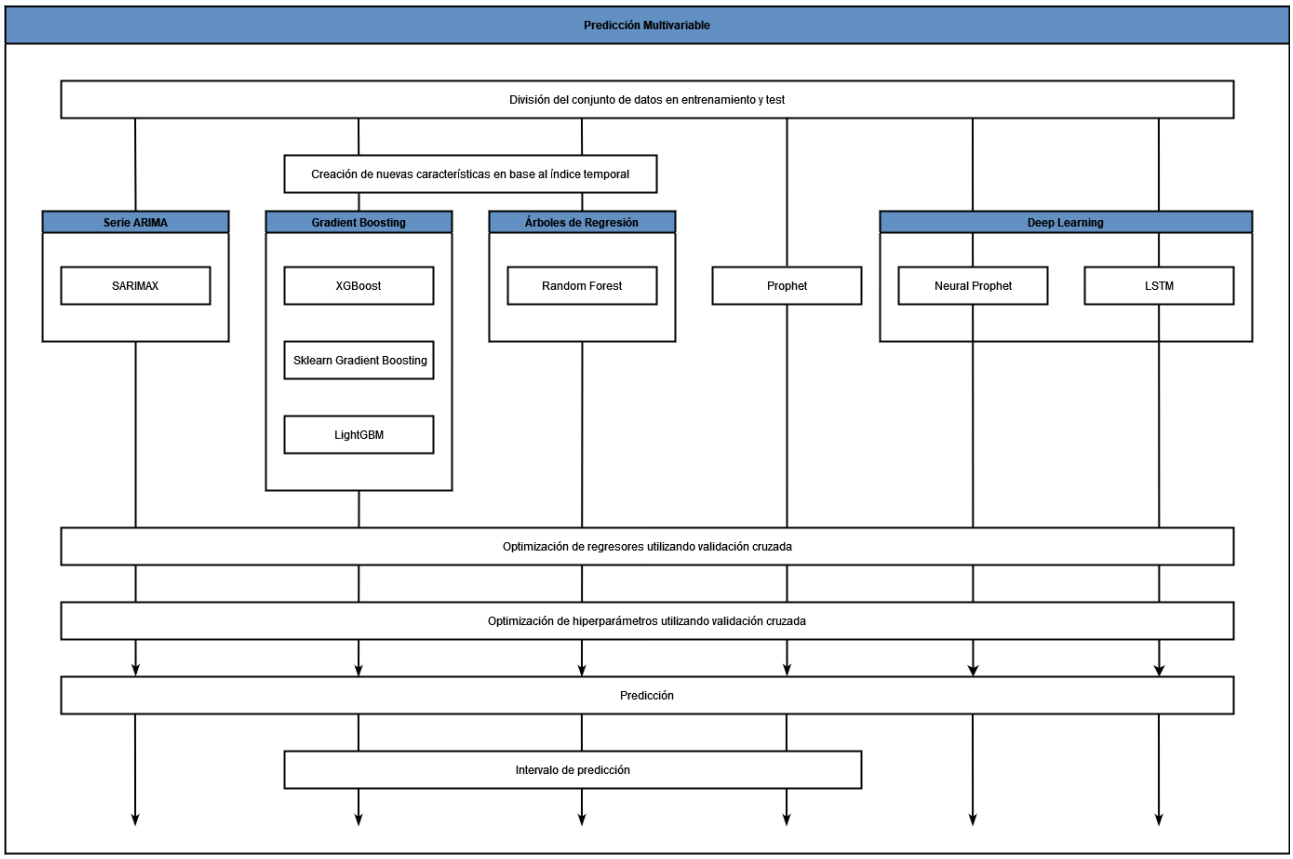

Figura 3.10: Proceso para la predicción multivariable.

Una vez realizada la selección de regresores, se efectuó la optimización de hiperparámetros utilizando también la técnica de validación cruzada. Los hiperparámetros son parámetros que se establecen antes de la fase de entrenamiento de un modelo y que no son ajustados durante el proceso de entrenamiento. Estos parámetros tienen un impacto significativo en el rendimiento del modelo, por lo que es importante encontrar los valores óptimos para cada uno de ellos. La optimización de hiperparámetros se realizó también mediante validación cruzada, de forma que se pueda obtener el mejor conjunto de hiperparámetros para cada modelo. Asimismo, dependiendo de la técnica empleada, se obtiene también un intervalo de predicción que permite acotar los valores con cierta confianza.

3.10. Herramientas

3.10.1. Entorno de desarrollo

Visual Studio Code

Para el desarrollo del código fuente de la aplicación se ha utilizado Visual Studio Code, un editor de código fuente open-source desarrollado por Microsoft. Se trata de una herramienta versátil y personalizable, que ofrece compatibilidad con diversos lenguajes de programación, control integrado de Git y un marketplace para añadir extensiones de forma sencilla, entre otras características. Una de las extensiones que se ha empleado para el desarrollo de este trabajo es la extensión de Jupyter, que permite crear y trabajar con cuadernos Jupyter[1] directamente en el entorno de desarrollo de VS Code.

Git y Github

Git[5] y GitHub[6] son herramientas esenciales en el desarrollo de software para la gestión de versiones y la colaboración en proyectos de programación. Git es un sistema de control de versiones distribuido que permite a los desarrolladores realizar un seguimiento de los cambios en los archivos de código fuente, gestionar diferentes versiones del código y retroceder a versiones anteriores si es necesario. GitHub, por su parte, es una plataforma de alojamiento remoto para repositorios Git que ofrece capacidades para colaborar en proyectos y gestionar el flujo de trabajo de control de versiones.

3.10.2. Python

Python es un lenguaje de programación de alto nivel, interpretado y de propósito general que se utiliza ampliamente en el desarrollo de software debido a su sintaxis clara y legible, así como a su amplia comunidad de desarrolladores y bibliotecas de código abierto. En el contexto de este trabajo, se ha decidido utilizar Python con el propósito de aprovechar su amplia gama de aplicaciones en el ámbito de la IA y la Ciencia de Datos. Además, se han utilizado cuadernos Jupyter, que permiten visualizar el resultado de la ejecución de código Python de forma interactiva.

3.10.3. Herramientas de visualización de datos

Matplotlib es una biblioteca de visualización de datos en Python que proporciona una amplia gama de herramientas para crear gráficos y visualizaciones de alta calidad. Proporciona una interfaz de programación de aplicaciones (API) flexible que permite a los usuarios personalizar y ajustar prácticamente todos los aspectos de un gráfico. Por otro lado, Seaborn proporciona una interfaz de alto nivel para crear gráficos estadísticos atractivos y visualmente informativos. Ofrece una amplia gama de gráficos estadísticos predefinidos, como gráficos de distribución, gráficos de barras, gráficos de regresión o matrices de correlación. Finalmente, Plotly permite crear gráficos y visualizaciones dinámicas y interactivas, de forma que los usuarios pueden interactuar con los gráficos, hacer zoom, pan, seleccionar puntos de datos, y más, directamente en el navegador web.

[1] Jupyter
[5] Git
[6] Github

3.10.4. Herramientas de gestión de datos

Para la gestión de datos se ha utilizado Pandas, una librería enfocada al análisis y manipulación de datos. Proporciona estructuras de datos flexibles y de alto rendimiento, como DataFrames y Series, ideales para trabajar con datos tabulares y realizar operaciones comunes como filtrado, agrupación, limpieza, transformación y análisis. También se empleó NumPy para el cálculo numérico y el procesamiento de matrices y arreglos multidimensionales. Es ampliamente utilizada en la comunidad científica y de análisis de datos debido a su eficiencia y capacidad para realizar operaciones matemáticas rápidas en grandes conjuntos de datos.

3.10.5. Herramientas de modelado

Scikit-Learn

Para el desarrollo de modelos, se ha utilizado scikit-learn (también conocido como sklearn) debido a su amplia capacidad en el campo del aprendizaje automático. Sklearn es una biblioteca de código abierto en Python que ofrece una amplia gama de algoritmos y herramientas para el análisis y modelado de datos.

Esta herramienta ha permitido aprovechar una variedad de algoritmos de aprendizaje automático, como regresión lineal, árboles de decisión o gradient boosting. Además, scikit-learn ofrece herramientas para la evaluación y validación de modelos, como técnicas de validación cruzada, métricas de evaluación, selección de modelos y más, lo que ha permitido medir y comparar el rendimiento de los modelos en diferentes escenarios.

Statsmodels

Statsmodels es una biblioteca de Python para el modelado estadístico que ofrece una amplia gama de técnicas estadísticas y herramientas para el análisis y modelado de datos en el ámbito de la econometría, la estadística y la ciencia de datos. Ha sido utilizado en este trabajo para la implementación de modelos de regresión lineal o análisis de series temporales. También ofrece gráficos de diagnóstico tales como la descomposición de series temporales o análisis de la autocorrelación, lo que ha permitido examinar cada característica de una forma más detallada.

Statsforecast

StatsForecast es una biblioteca de Python que ofrece una colección de modelos de pronóstico univariados de series temporales, incluyendo ARIMA automático, ETS o CES. Se ha empleado para reducir la carga computacional de estos modelos, por su soporte para el uso regresores externos y su capacidad de generar un intervalo de predicción.

XGBoost, LightGBM y Prophet

XGBoost es la implementación en Python del modelo *extreme gradient boosting*. LightGBM, por otra parte, es una es una implementación en Python eficiente y de alto rendimiento de un algoritmo de *gradient boosting*. Finalmente, Prophet es librería de código abierto desarrollada por Facebook que implementa el modelo Prophet.

Keras

Keras es una biblioteca de aprendizaje automático de código abierto para Python que facilita la creación y entrenamiento de modelos de redes neuronales. Proporciona una interfaz de alto nivel para la construcción y entrenamiento de modelos de redes neuronales, lo que facilita la creación de prototipos y la experimentación con diferentes arquitecturas de modelos. Además, como fue desarrollada con un enfoque orientado a la modularidad y la reutilización de código, permite a los usuarios construir modelos complejos a partir de bloques de construcción simples.

Neural Prophet

Neural Prophet [49] es una librería que está construida sobre la popular librería de machine learning PyTorch [1] y utiliza una arquitectura de red neuronal recurrente (RNN) para modelar patrones de series de tiempo. Utiliza técnicas de optimización de gradiente descendente estocástico (SGD) y una variedad de funciones de activación para entrenar y ajustar los modelos.

Entre sus principales características se encuentran la capacidad de trabajar con series de tiempo de múltiples dimensiones y con valores faltantes, de incorporar eventos externos en el modelo y de generar intervalos de confianza para las predicciones.

3.10.6. Documentación

Para generar la documentación de este proyecto se ha utilizado Mkdocs[2] y Mkgendocs[3]. Mkdocs es una herramienta para crear documentación de proyectos de software en formato HTML. Permite crear documentación de un proyecto en Python de forma fácil y rápida utilizando archivos de Markdown. Por otro lado, Mkgendocs es una herramienta que se utiliza para generar documentación a partir de los docstrings[4] de las funciones y clases en Python. Mkgendocs utiliza estos docstrings para generar una documentación clara y fácil de entender que describe el propósito y uso de cada función y clase.

Por tanto, los docstrings proporcionan una descripción clara de cada función y clase en el código, mientras que Mkgendocs se encarga de generar una documentación estructurada a partir de estos docstrings. Mkdocs luego permite organizar y presentar esta documentación en un formato HTML fácil de navegar. En el Apéndice C se describe y muestra en más detalle la página de documentación obtenida.

[1] Pytorch
[2] Mkdocs
[3] Mkgendocs
[4] Los docstrings son cadenas de texto que se encuentran al inicio de una función o clase, y se utilizan para proporcionar una descripción clara y concisa de lo que hace.

Capítulo 4

Desarrollo

4.1. Análisis

4.1.1. Estacionariedad

Se realizó el análisis de estacionariedad de las variables mediante el uso de la prueba de Dickey-Fuller. Los resultados indicaron que todas las variables del conjunto de datos son estacionarias. Esto sugiere que las series no presentan una tendencia o patrón que cambie a lo largo del tiempo, lo cual facilita la aplicación de modelos de predicción y la interpretación de los resultados obtenidos. Por lo tanto, se puede proceder con mayor confianza al aplicar modelos de predicción a las variables en cuestión.

Cabe destacar que, tras observar las series representadas en gráficos, se pudo determinar que las variables presentan una tendencia cicloestacionaria, dado que existe un patrón recurrente y cíclico en los datos. Sin embargo, es importante señalar que esta característica no afecta a la aplicación de modelos de predicción en la mayoría de los casos. Además, es común que muchas variables meteorológicas presenten esta característica debido a la naturaleza cíclica de los fenómenos meteorológicos. Por lo tanto, se puede concluir que la presencia de cicloestacionariedad no representa una limitación significativa para el análisis y la predicción de estas variables.

4.1.2. Estacionalidad

Al analizar las variables en detalle, se observa que la mayoría de ellas presenta patrones de estacionalidad anual y diaria. Es importante tener en cuenta que las predicciones que se realizan a 8 horas vista no son especialmente sensibles a la estacionalidad anual, ya que este patrón se repite en ciclos de tiempo más largos. Sin embargo, la estacionalidad diaria puede ser un factor importante a considerar al realizar las predicciones, ya que los patrones repetitivos que se presentan a lo largo del día pueden influir en las tendencias y cambios en los datos en el corto plazo.

Para realizar una descomposición de una serie de tiempo que presenta múltiples estacionalidades, es necesario aplicar el método STL varias veces, cada vez eliminando la estacionalidad ya extraída antes de aplicar el método nuevamente. En este caso, como la serie presenta una estacionalidad anual y una estacionalidad semanal, se deben realizar dos descomposiciones separadas (obteniendo de la última la tendencia y residuo), tal y como se ilustra en la Figura 4.1.

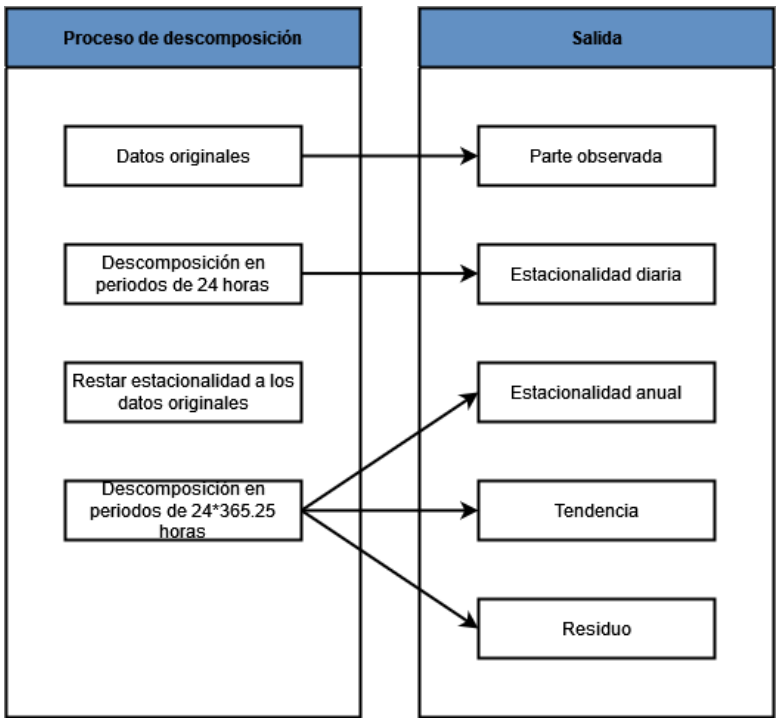

Figura 4.1: Proceso de descomposición con múltiples estacionalidades.

Se creó una función que permite obtener la descomposición en múltiples estacionalidades de una serie temporal, utilizando el proceso detallado en la Figura 4.1. Como resultado, se muestra un único gráfico que contiene una visualización diferente para componentes extraído en el proceso de descomposición de la serie (véase Figura XXX).

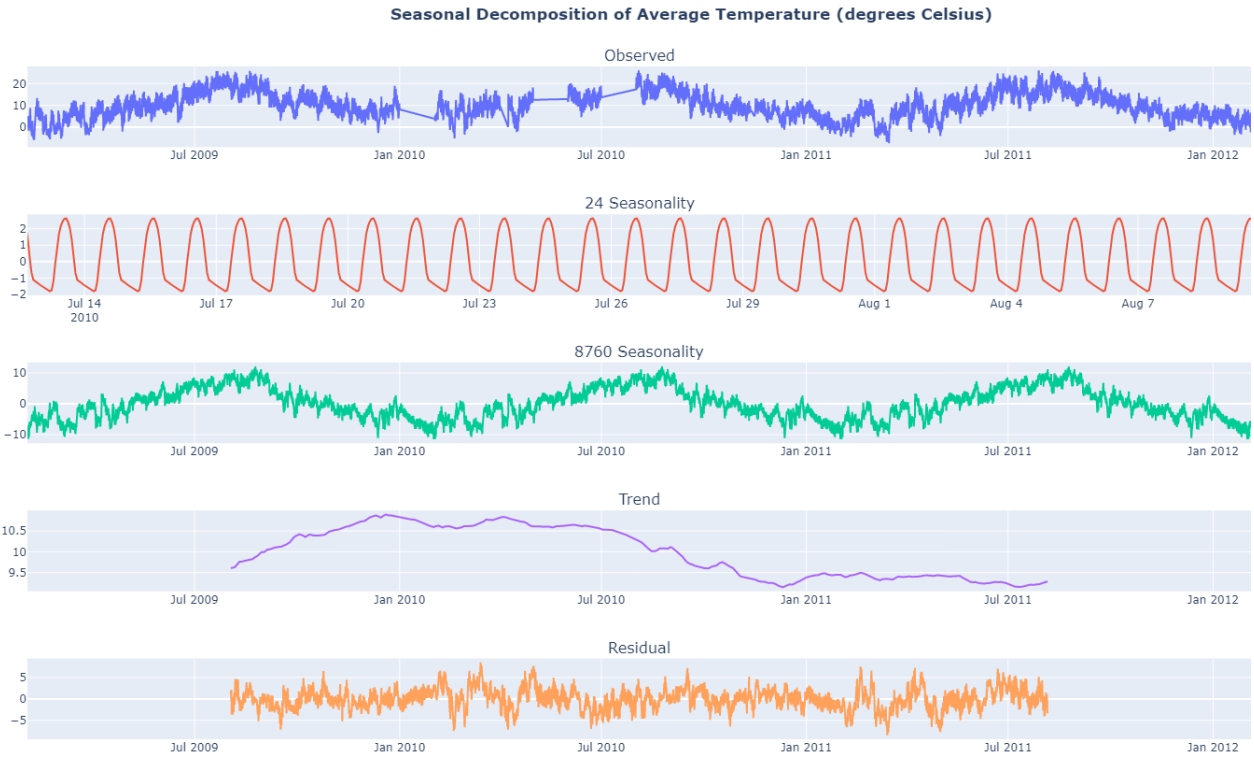

Figura 4.2: Resultado de la descomposición en múltiples estacionalidades.

4.1.3. Correlación

Antes de la selección de características, se llevó a cabo un análisis de correlación, aunque para fines prácticos, en esta memoria se presentan únicamente las visualizaciones de las variables seleccionadas para el entrenamiento. En la Figura 4.3 se muestra un mapa de calor que refleja el grado de correlación entre las partes observadas de cada variable, utilizando el coeficiente de correlación de rango de Kendall. Además, se realizó una agrupación en clústeres para cada una de las variables.

Dado que el objetivo principal del proyecto es la predicción del punto de rocío 8 horas en el futuro, se llevó a cabo un estudio de la correlación retrasando los posibles regresores externos en esta cantidad de horas. Esto permitió no solo analizar la propia correlación cruzada, sino también la relación que cada variable puede tener con el punto de rocío. Es posible consultar las figuras de estos mapas aplicados a la estacionalidad (B.1), tendencia (B.2) y residuo (B.3) en el Apéndice de la memoria.

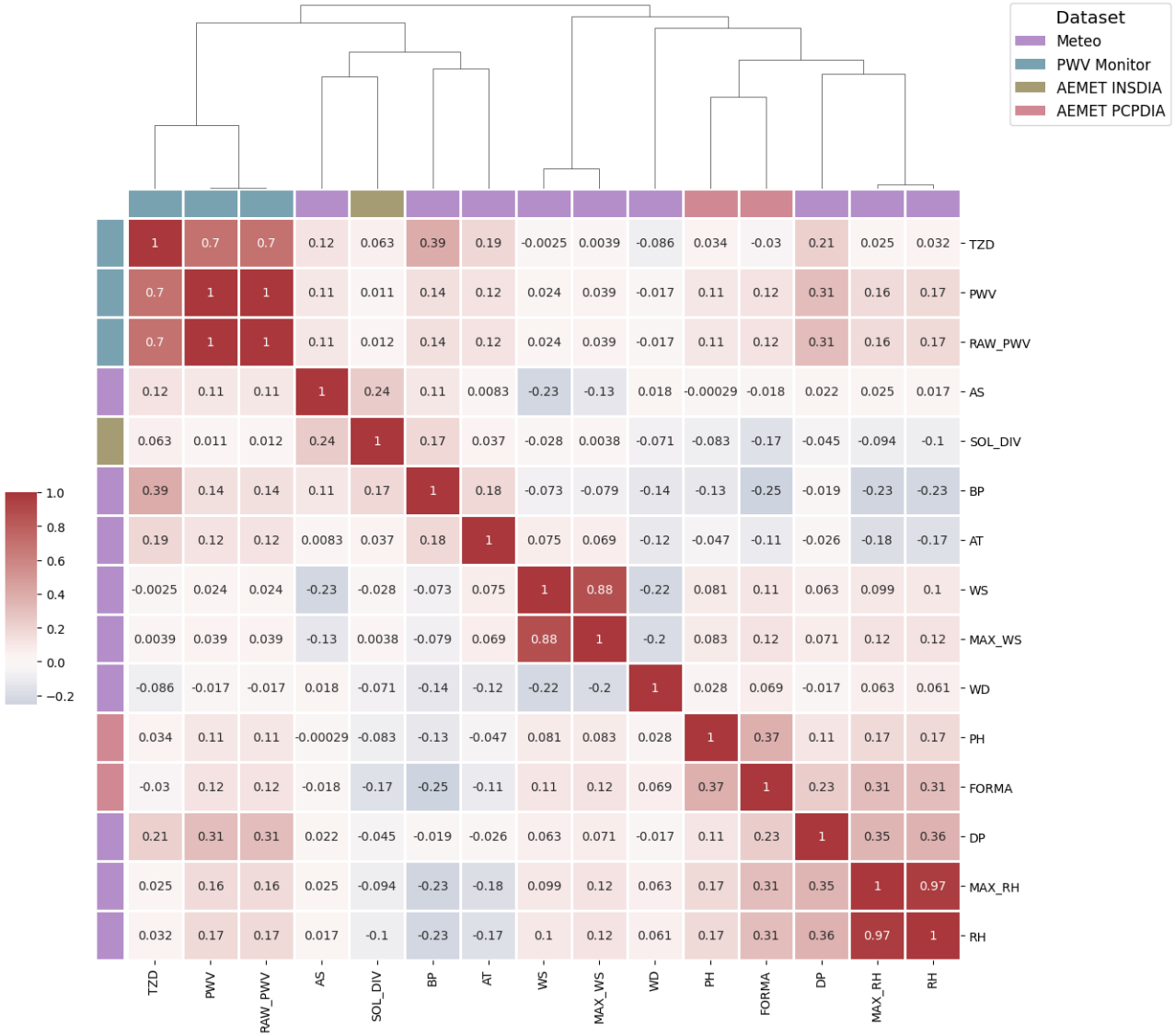

Figura 4.3: Mapa de correlación de la parte observada de variables seleccionadas. Retraso de 8 horas aplicado.

4.1.4. Autocorrelación

Durante el análisis de correlación llevado a cabo, se han observado dos patrones distintos en el comportamiento de las variables estudiadas. Por un lado, se encuentra el patrón seguido por variables como DP (ver Figura 4.4) o AT, en el que existe un ciclo diario en el que la autocorrelación disminuye significativamente a lo largo del día, para luego aumentar de nuevo conforme se acerca la medianoche. Por otro lado, se encuentra el patrón seguido por variables como WS (ver Figura 4.5), en las que la autocorrelación disminuye de forma progresiva a medida que avanza el tiempo.

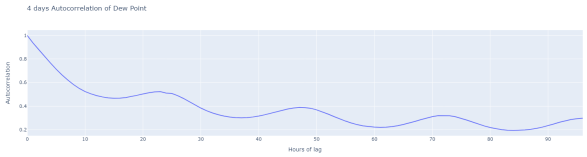

Figura 4.4: Autocorrelación en DP, vista a 4 días.

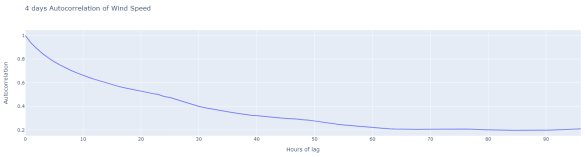

Figura 4.5: Autocorrelación en WS, vista a 4 días.

Estos patrones son importantes a la hora de la predicción, ya que pueden influir en la elección del modelo más adecuado para cada variable. En el caso de variables con una marcada estacionalidad diaria, como DP o AT, se podría considerar la utilización de modelos como ARIMA o Prophet, que permiten modelar adecuadamente la estacionalidad presente en los datos. Por otro lado, en el caso de variables con una disminución progresiva de la autocorrelación, como WS, podrían considerarse modelos de regresión que permitan modelar de forma más adecuada la tendencia presente en los datos.

4.2. Limpieza de datos

En el conjunto de datos de variables meteorológicas con el que se está trabajando, es común encontrar valores anómalos o *outliers*, en particular en la presión barométrica (véase Figuras 4.7 y 4.6). Estos valores atípicos pueden deberse a diversas razones, como errores de medición, fallos en la instrumentación o simplemente cambios abruptos en las condiciones meteorológicas. Como se ha explicado anteriormente, es importante detectar estos valores atípicos y tratarlos adecuadamente, ya que pueden afectar significativamente el análisis y la modelización de los datos.

Figura 4.6: Anomalías en la temperatura media

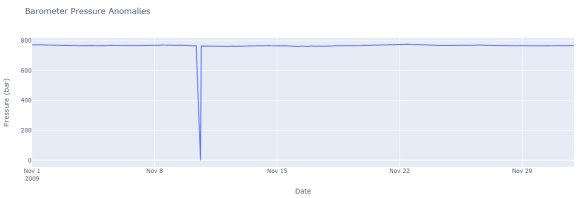

Figura 4.7: Anomalías en la presión barométrica

En el contexto del proyecto, la función `clean_dataframe` (véase Listing 4.3) se encarga de limpiar el conjunto de datos de valores anómalos. Esta función recorre cada una de las columnas y aplica un suavizado a los *outliers* detectados. Se han definido parámetros

específicos para la la detección en cada serie temporal, dependiendo del punto a partir del cual se considera que un valor de la serie es un anomalía:

- `max_scale`: Define el valor máximo sobre el que hacer la normalización. Por tanto, cuánto más bajo es, más se magnifican las anomalías.
- `limit`: Tras realizar el proceso de escalado, es el límite inferior de las anomalías a las que se quiere realizar el suavizado.
- `lower_threshold` (solo para IF): Límite inferior de valor anómalo antes de realizar el escalado.
- `deviation_factor` (solo para residuos STL): El factor por el que se va a multiplicar la desviación estándar del residuo con el fin de usarla como límite inferior antes del escalado.

Por ejemplo, en el Listing 4.1, se implementa la función `STL_anomalies`, que efectúa una detección de anomalías utilizando los residuos resultantes de la descomposición de la serie. Para ello, se calcula un límite superior basado en la mediana y la desviación estándar de los residuos, y se identifican los valores del residuo que exceden este límite. Estos valores se convierten en una serie de tiempo que indica los puntos anómalos de la serie original, normalizados y escalados según un factor máximo opcional. Por otro lado, si se establece el parámetro *verbose* como verdadero, se muestra una visualización de la serie original con los valores anómalos resaltados, véasen Figuras 4.8 y 4.9.

```
def STL_anomalies(...) -> pd.Series:
    decomposition = seasonal_decompose(df, model="additive", period=period).resid.
    to_frame()
    upper_bound = (
        decomposition.resid.median() + deviation_factor * decomposition.resid.std())
    decomposition.resid = abs(decomposition.resid).interpolate()
    anomalies = pd.Series(decomposition.resid, index=df.index)
    anomalies[decomposition.resid <= upper_bound] = 0
    anomalies = anomalies / (upper_bound)
    if max_scale is not None:
        anomalies[anomalies > 0] = (anomalies - anomalies[anomalies > 0].min()) / (
            max_scale - anomalies[anomalies > 0].min())
        anomalies[anomalies > 1] = 1
    else:
        anomalies = (anomalies - anomalies.min()) / (anomalies.max() - anomalies.min())
    if verbose:
        title = f"Anomalies in {get_name(df.name)} using STL decomposition"
        show_anomalies(df, anomalies, title)
    return anomalies
```

Listing 4.1: Detección de anomalías usando residuos STL.

Tal y como se mencionó en el capítulo anterior, en este proyecto se contemplaron los métodos de detección de anomalías por residuos STL e IF. Cada uno de ellos ofrece unos resultados diferentes y adecuados para diferentes variables.

Mientras que el método IF es particularmente eficaz para detectar valores que se salen del rango normal, los residuos STL detectan mejor aquellos valores que se desvían del patrón del resto del conjunto. Por lo tanto, en una región anómala de gran extensión, el método STL puede no ser capaz de identificar correctamente todos los *outliers* cuya desviación no sea destacable en comparación con el resto del conjunto de datos, mientras

que IF puede detectar todos los valores anómalos presentes en la región (véasen figuras 4.8 y 4.9).

Sin embargo, puede ocurrir lo contrario: valores que no son extremos en relación al resto del conjunto de datos, pero que destacan respecto a la desviación típica o al contexto de ese momento. En ese caso, como se puede observar en las figuras 4.10 y 4.10, el método STL sería más adecuado para detectarlos.

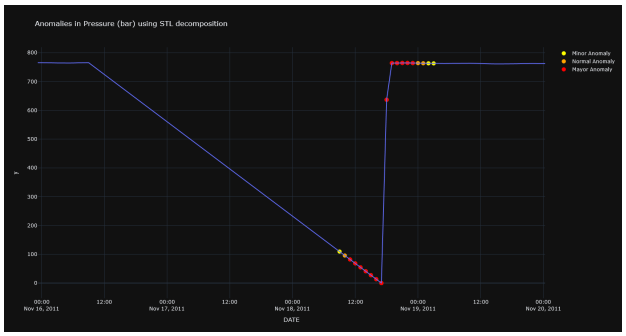

Figura 4.8: Anomalías en la BP usando STL.

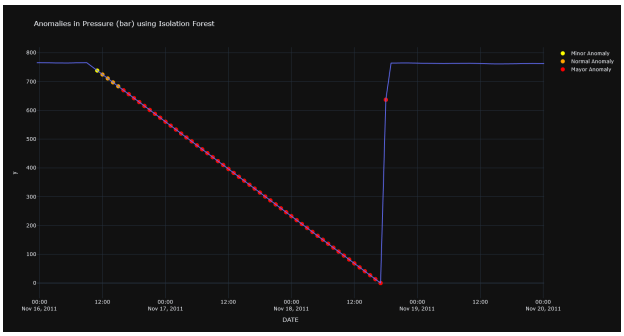

Figura 4.9: Anomalías en la BP usando IF.

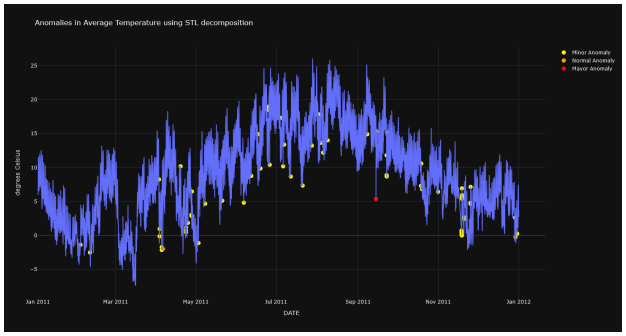

Figura 4.10: Anomalías en AT usando STL.

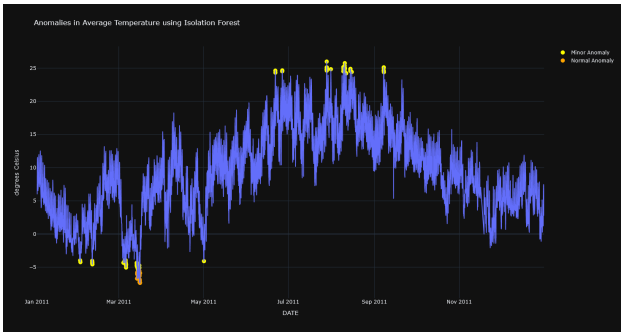

Figura 4.11: Anomalías en AT usando IF.

Después de la detección de anomalías, se aplica un proceso de suavizado a las mismas, siendo su peso mayor cuanto mayor sea la puntuación recibida. Este suavizado solo se aplica a aquellas anomalías que superen el límite definido por el parámetro `limit`. Existen dos métodos de suavizado contemplados: *bi-rolling* e interpolación. *Bi-rolling* (Listing E.4) utiliza la media de los valores ocurridos sobre la misma hora en el día anterior y posterior para calcular el valor suavizado, mientras que la interpolación (Listing E.3) utiliza la media del valor anterior y posterior más cercano que no sea anómalo.

De este modo, para cada columna, se realiza una doble limpieza. Primero se utiliza el método IF y un suavizado *bi-rolling* , y luego se utiliza el método de residuos STL y un suavizado por interpolación. Un ejemplo del resultado de este proceso se puede consultar en la Figura 4.12.

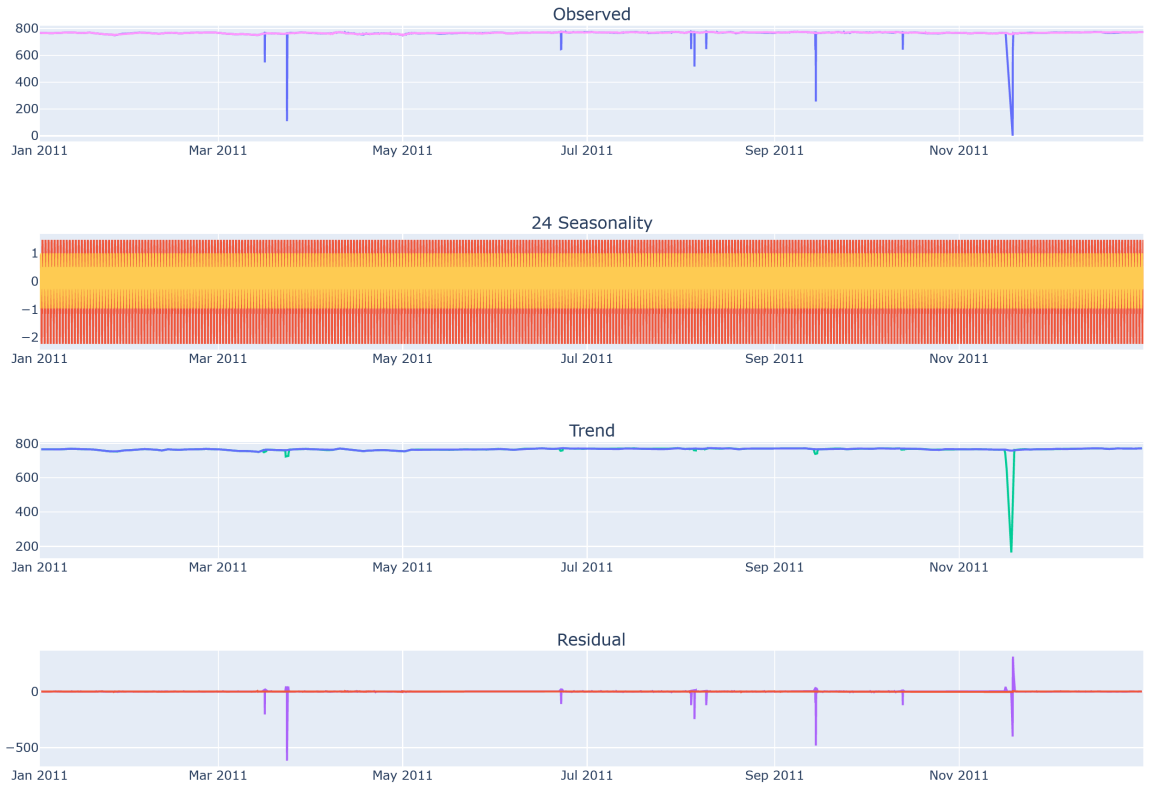

Figura 4.12: Suavizado de anomalias utilizando residuos STL e IF sobre BP.

4.3. Imputación de valores vacíos

En el contexto de series temporales de variables climatológicas, es común encontrar valores faltantes debido a problemas técnicos de medición, fallos en la transmisión de los datos, entre otros factores. La falta de observaciones puede generar sesgos en la estimación de los componentes de las variables, lo que puede llevar a conclusiones incorrectas. Además, puede limitar la cantidad de datos disponibles para entrenar modelos de predicción o incluso impedir su funcionamiento.

Como podemos observar en la figuras 4.13 y 4.14, los valores faltantes son comunes en el conjunto de datos proporcionado. Las discontinuidades llegan en algunos casos a ocupar casi un mes completo, lo que dificulta en gran medida el proceso de imputación.

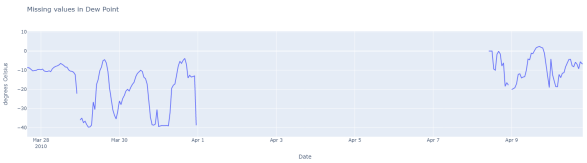

Figura 4.13: Valores faltantes en DP.

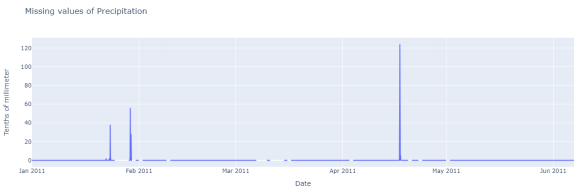

Figura 4.14: Valores faltantes en PH.

En aquellas series temporales de baja varianza, como es el caso de FORMA y PH, se ha decidido emplear una imputación de ceros, dado que éste valor representa un muy alto porcentaje de las entradas. En el resto de las columnas se decidió comparar la efectividad de la interpolación lineal respecto al rellenado de valores utilizando algoritmos

de predicción (prophet en este caso), véase Figura 4.15. Finalmente, se obtuvo que el método de interpolación conseguía mejores resultados que la interpolación utilizando prophet, por lo que se adoptó este método para el preprocesamiento del conjunto de datos.

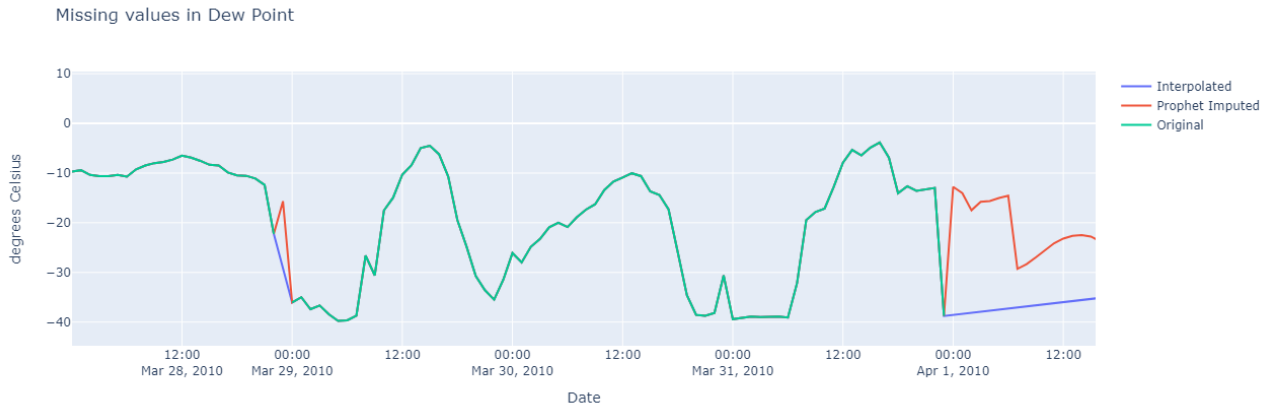

Figura 4.15: Comparación de imputación por interpolación y Prophet.

4.4. Selección de características

Como se mencionó en el capítulo anterior, se han planteado el uso de diferentes técnicas para la reducción de la dimensionalidad del conjunto. En primer lugar, se han utilizado métodos supervisados para descartar carecterísticas que, debido a su condición, no aportan información valiosa o útil. Este proceso se realiza de forma agnóstica a la variable objetivo del problema, de la manera que figura en el Listing 4.2 para la eliminación de variables de varianza baja. Gracias a ello, se detectaron 7 variables cuya varianza no superaba el límite propuesto, de las cuales la única que no se eliminó fue FORMA. Esto se debe al propio caracter de la variable, que toma valores entre 0 y 2 y presenta una gran cantidad de ceros, al ser las lluvias poco frecuentes.

```
1 def low_variance_remove(df: pd.DataFrame, threshold: int = 0.4, verbose: bool = True) ->
      pd.DataFrame:
2     sel = VarianceThreshold(threshold=threshold)
3     sel.fit_transform(df)
4     if verbose:
5         print(f'Deleted Columns = {df.columns[~sel.get_support()].tolist()}')
6     return df[df.columns[sel.get_support()].tolist()]
```

Listing 4.2: Función que elimina las variables de baja varianza.

Posteriormente, se aplicaron diversas técnicas de filtro para cuantificar la correlación entre las distintas características del conjunto de datos y la variable objetivo (DP). Al igual que en la sección de análisis, se empleó una versión atrasada en 8 horas de las variables analizadas. Para ello, se utilizaron las técnicas de filtro descritas en el capítulo anterior. Sin embargo, se tuvo en cuenta el coeficiente de correlación de rango de Kendall para la selección final, dada su mayor resistencia a los valores atípicos y su capacidad para detectar patrones no lineales.

Según los resultados obtenidos (véase Tabla B.1), se seleccionaron las variables con mayor correlación respecto a DP: DP, RH, MAX_RH, PWV, FORMA, PH y SOL_DIV. Para PH y SOL_DIV, se seleccionaron sus alternativas diarias ya que ofrecían una mayor

correlación debido a su menor variabilidad. Además, se consideró la inclusión de WS, AS y BP debido a la existencia de una relación indirecta con la variación del punto de rocío; esta relación se ha observado en diversos artículos citados en el Capítulo 2.

4.5. Predicción

En este caso, se dividió el conjunto de datos en conjuntos de entrenamiento y testeo utilizando una estrategia de remuestreo aleatorio. En particular, se seleccionaron a partir del 25 % final de los datos conjuntos de validación aleatorios de 8 horas de longitud de, mientras que las entradas previas se utilizaron como conjunto de entrenamiento. Se estableció una distancia de 24 horas entre ambos conjuntos para evitar la contaminación de información entre ellos, ya que se espera que las variables meteorológicas y el punto de rocío estén altamente correlacionados en el tiempo.

Para la selección de regresores, empleada únicamente el contexto de predicción multivariable, se contemplaron conjuntos de hasta 4 regresores, seleccionados previamente en las etapas de análisis y selección de características. La evaluación de los conjuntos se realizó utilizando el RMSE - véase la Tabla D.1 para un ejemplo.

Una vez realizada esta selección, se emplearon los 3 mejores conjuntos de regresores para proceder con la optimización de hiperparámetros, utilizando la misma técnica de validación cruzada con los conjuntos de regresores previamente seleccionados (véase Tabla D.2). Las pruebas realizadas se hicieron por tanto combinando los 3 mejores conjuntos de regresores y los valores dados para cada uno de los hiperparámetros listados (véase el Listing 4.3 para un ejemplo). Se destaca que ni *RH* ni *MAX_RH* estuvieron presentes entre las primeras opciones de los modelos, a pesar de ser las variables más correladas con el punto de rocío.

```
xgboost_params_tuning = multivariate_hyperparameter_tuning(
  df_cleaned,
  use_xgboost,
  tune_model_args={
    'n_estimators': [100, 150, 200],
    'max_depth': [3, 6],
    'learning_rate': [0.1, 0.2],
    'max_bin': [255, 300, 400],
  },
  tune_function_args={},
  tune_model_fit_args={},
  X_features=xgboost_X_tuning[:3].X_features.to_numpy(),
  fixed_X_features=True,
  end_indices=end_indices,
  gap_between=24,
)
```

Listing 4.3: Ejemplo de ejecución de la optimización de hiperparámetros.

Capítulo 5

Resultados

5.1. Comparativa de resultados antes y después del preprocesamiento

La Tabla 5.1 expone los resultados obtenidos tras las etapas de limpieza de *outliers* y manejo de datos faltantes utilizando varias muestras del conjunto de testeo. Para la ejecución de los mismos, se utilizaron modelos que permitían una frecuencia de datos no constante (dado que no se han imputado los valores faltantes); en este caso, *LigthGBM*, Redes neuronales *LSTM* y *Random Forest* (RF). De media, las predicciones sobre el conjunto imputado fueron un 4.76 % mejores que las realizadas sobre el conjunto original. Asimismo ocurre tras efectuar el suavizado de anomalías, obteniendo un error 13.24 % menor que en el conjunto original.

Modelo	**RMSE original**	**RMSE imputado**	**RMSE procesado**
LightGBM	4.608600	4.279480	**3.989652**
LSTM	5.003818	4.721905	**4.651759**
RF	4.269507	4.262023	**3.629453**

Tabla 5.1: Tabla comparativa de resultados tras el preprocesamiento.

5.2. Comparación de resultados 8 horas a futuro

En esta sección se comparan los resultados obtenidos mediante los diferentes modelos contemplados en el contexto de este proyecto. Para ello, se han empleado 10 muestras diferentes del conjunto de testeo y se ha hallado la media de todas. Todos los modelos se han ejecutado utilizando la configuración de regresores externos e hiperparámetros hallados en la fase de entrenamiento y validación.

Como se indica en la Tabla 5.2, el mejor desempeño lo obtiene el modelo de *Gradient Boosting Regressor* (GBR) de la libería SkLearn, resultando con un RMSE de 4.81º. El resto de métricas empleadas también señalan a este modelo como el más preciso y fiable, dado que muestra el menor MEC (11.26) y mayor PC (66.25). Por otro lado, nos encontramos en último lugar al modelo XGBoost, con un RMSE de 7.21º, 2.40 grados superior al obtenido con GBR. Asimismo, Random Forest es el modelo que peor desempeña en la obtención del intervalo de predicción, dado que, a pesar de no tener el menor PC, presenta el mayor MEC (12.30).

Modelo	RMSE	MAE	Distancia DTW	PC (%)	MEC
RF	4.995847	4.296664	3.741612	60.00	12.296245
XGBoost	7.214917	6.544643	6.237468	55.00	11.274243
GBR	**4.808086**	**4.134798**	**3.621553**	**66.25**	**11.262377**
LightGBM	5.796648	5.166885	4.539308	52.50	12.142811
VAR	6.623478	6.172340	6.132132	*NaN*	*NaN*
LSTM	6.365747	5.781018	5.520172	*NaN*	*NaN*
Neural Prophet	5.906514	5.470614	5.272805	*NaN*	*NaN*

Tabla 5.2: Tabla de resultados de predicción 8 horas a futuro. Los modelos que no disponen de intervalo de predicción se marcaron como *NaN*.

Teniendo en cuenta que GBR es el modelo que mejor desempeña en todos los apartados, será este el que se empleará en la implementación para predicción de la temperatura de punto de rocío en el Sistema de Control de TTNN. A continuación, en las Figuras 5.1 y 5.2, se muestran dos ejemplos de predicción de los dos mejores modelos; cabe destacar que normalmente las predicción son de menor precisión en valores muy bajos del DP.

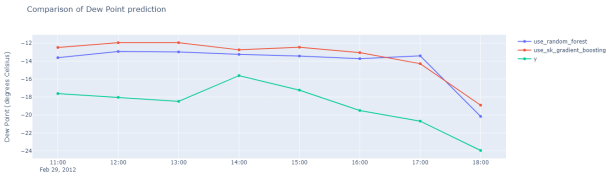

Figura 5.1: Predicción en el conjunto de testeo utilizando RF y GBR. Ninguno se ajusta muy bien a las temperaturas bajas.

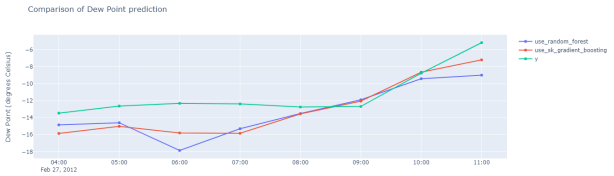

Figura 5.2: Predicción en el conjunto de testeo utilizando RF y GBR. Se observa como anticipan la pendiente.

5.3. Comparación de resultados 28 horas a futuro

En esta seccción se comparan los resultados obtenidos mediante los diferentes modelos contemplados en el contexto de la predicción 28 horas a futuro. Para ello, se ha empleado el mismo método que en la sección anterior, es decir, utilizando 10 muestras diferentes del conjunto de testeo. De la manera que figura en la Tabla 5.3, fue el modelo ARIMA el que obtuvo los mejores resultados, con un RMSE de 11.52º. Sin embargo, en cuanto al intervalo de predicción, fue el modelo XGBoost aquel que obtuvo el mejor MEC (27.95), a pesar de no tener el mejor CP.

Dada la naturaleza de los resultados, su utilidad no va más allá de una predicción meramente orientativa. Asimismo, cabe destacar que ninguno de estos modelos va a ser implementado en el transcurso de este proyecto, dado que requieren la disponibilidad en un extenso historial estrictamente previo al momento de la predicción. Actualmente el servidor del sistema de control solo tiene acceso a los valores actual de las variables meteorológicas, entre las que se encuentra el punto de rocío.

Modelo	RMSE	MAE	DTW Distance	CP (%)	MEC
SES	13.507224	11.978726	11.978726	*NaN*	*NaN*
HWES	12.368276	11.052305	10.202135	*NaN*	*NaN*
ARIMA	**11.525808**	**10.166103**	**9.849250**	**79.285714**	35.617677
Prophet	12.491616	11.464255	11.071643	54.642857	30.044771
RF	16.264563	14.694910	14.065540	26.428571	33.050096
XGBoost	15.492908	13.881340	13.331443	23.571429	**27.952266**
GBR	13.536237	11.953249	11.345498	41.071429	28.129961
LightGBM	14.527417	12.953064	12.215264	28.571429	29.687816

Tabla 5.3: Tabla de resultados de predicción 28 horas a futuro. Los modelos que no disponen de intervalo de predicción se marcaron como *NaN*.

5.4. Despliege en Control TTNN

Tras la obtención de los resultados, se realizó la implementación del modelo seleccionado en el flujo de ejecución del sistema de control de TTNN. Para ello, se tuvieron que realizar diferentes tareas orientadas a la obtención y procesamiento de los datos emitidos por la estación meteorológica y la generación y adecuación de los resultados obtenidos en la predicción, así como su publicación dentro del sistema ROS y la encapsulación del código en una clase de Python.

5.4.1. Obtención y procesamiento de los datos

La estación meteorológica emite cada pocos segundos un mensaje dentro del sistema ROS con los datos meteorológicos obtenidos en la última medición. Dado que se emplea un patrón de publicador/suscriptor, es necesario suscribirse al nodo *weatherStation/get* para que se llame a la función de *callback* proporcionada cada vez que el publicador realice una publicación (véase Listing 5.1).

```
1 rospy.Subscriber("/coordination/weatherStation/get", weatherMsg, self.
    weatherStationCallback)
```

Listing 5.1: Suscripción al nodo de la estación meteorológica.

El entrenamiento de los modelos contemplados se ha realizado utilizando datos con una frecuencia horaria, pero la estación los publica cada pocos segundos, por lo que se requiere una transformación de los datos previa a la predicción. Actualmente no se dispone de un histórico de los datos meteorológicos, pero para la predicción es necesario un mínimo de 8 horas de histórico estrictamente anterior al momento de la predicción. Se precisa, por tanto, de la creación de un histórico local al nodo en base a los datos enviados por la estación. La principal consecuencia de esta decisión es que no se podría realizar la predicción hasta pasadas 8 horas desde el arranque del sistema. Para abordar esta cuestión, se crearon un conjunto de buffers en el contexto de la clase, cuya gestión se lleva a cabo dentro del *weatherStationCallback*.

- Un buffer de observaciones (`{X_feature: 0.0 for X_feature in self.X_features}`) que almacena una suma de los valores de los indicadores climatológicos recibidos desde la estación en la última hora.

- Un buffer horario (`{X_feature: [] for X_feature in self.X_features + ["DATE"]}`) en el que se añade una nueva fila cada vez que hay un cambio de hora. Para ello, se efectúa una media de los valores almacenados en el buffer de observaciones. Con el fin de asegurar que este buffer no excede las 8 entradas necesarias para la predicción, se elimina la observación más antigua cada vez que se realiza la misma.

5.4.2. Generación y publicación de la predicción

Una vez que el buffer horario alcanza el tamaño de 8 horas requerido, comienza la generación de la predicción para las 8 horas posteriores. Para ello, se crea un *dataframe* de Pandas en base al buffer horario y se le pasa a la función `dewPointPrediction` (véase Listing 5.2) que se encarga del formateo del *dataframe*, de la generación del conjunto de entrenamiento y de llamar a la función de ejecución del modelo seleccionado.

```
def dewPointPrediction(self, observed: pd.DataFrame) -> pd.DataFrame:
        observed.index += dt.timedelta(
            hours=self.n_forecast
        )
        historical_data = import_dataset(...)
        X_train = (
            historical_data[self.X_features]
            .shift(self.n_forecast)
            .iloc[self.n_forecast:, :]
        )
        y_train = historical_data[self.y_feature].iloc[self.n_forecast:]
        X_test = observed
        return self.use_model(X_train, X_test, y_train)
```

Listing 5.2: Funcion *dewPointPrediction*

Posteriormente, tras efectuar la predicción, se adapta el conjunto de la predicción al formato seleccionado para el mensaje de salida del nodo (véanse Listings 5.3 y 5.4). Entonces, utilizando la estructura del mensaje *Predictions*, la publicación se realiza por un lado nada más generada la predicción (en el *callback*), y por otro lado en la función `mainIteration` (Listing E.5) cada vez que un nuevo nodo se suscribe a la publicación. La razón de ello es que los nuevos dispositivos que se conecten al servidor puedan recibir la predicción, y no tener que esperar a que realice la siguiente.

```
float64 forecast
float64 lower
float64 upper
```

Listing 5.3: Mensaje Prediction que contiene campos para la predicción y para el límite superior e inferior del intervalo de predicción.

```
Prediction[] predictions
NodeState nodeState
```

Listing 5.4: Mensaje PredictionsVector formado por un array del mensaje Prediction y un NodeState utilizado únicamente para propósitos internos del servidor.

De esta forma, el programa tras su puesta en funcionamiento, recogerá las observaciones de las 8 primeras horas y, tras ello, efectuará una predicción para las siguientes 8 horas en cada cambio de hora.

Capítulo 6

Presupuesto

En este capítulo figura la estimación de costes de realización del proyecto, asumiendo un coste por hora de 25€.

Tarea	Nº Horas	Coste
Estudio de tecnologías utilizadas	10	250€
Estudio del análisis de series temporales	15	375€
Análisis del estado del arte	30	750€
Recolección e Importación de los datos	35	875€
Preprocesamiento del conjunto de datos	90	2250€
Predicción univariable y multivariable	100	2500€
Comparación de resultados	30	750€
Integración en Control TTNN	20	500€
Redacción de la memoria	80	2000€
TOTAL	410	10075€

Tabla 6.1: Presupuesto

Capítulo 7

Conclusiones y líneas futuras

7.1. Conclusiones

En el presente trabajo se ha llevado a cabo un análisis del estado del arte del punto de rocío, con el objetivo de desarrollar un modelo de predicción para su implementación en el Sistema de Control de Telescopios Nocturnos del IAC. A lo largo del estudio, se han abordado diversos aspectos de gran relevancia, y a continuación, se presentan las conclusiones pertinentes respecto a cada uno de ellos.

En primer lugar, se ha logrado obtener y formatear los datos necesarios para realizar las predicciones. Sin embargo, se ha identificado una limitación importante en el dataset, que se refiere a la gran cantidad de valores faltantes. Este aspecto dificulta la precisión de las predicciones y representa un desafío a la hora de abordar el problema. Además, se ha observado que los datos están a diferentes frecuencias o incluso a frecuencias no constantes, lo cual introduce una complejidad adicional, ya que sería más conveniente contar con predicciones realizadas a una frecuencia menor.

A pesar de ello, se ha llevado a cabo un procesamiento efectivo de los datos, que ha mejorado notablemente las predicciones obtenidas. A lo largo del trabajo, se ha comparado el desempeño de distintos modelos de aprendizaje automático, tanto bajo el enfoque univariable como multivariable. Este análisis comparativo ha permitido identificar las fortalezas y debilidades de cada modelo en relación con la predicción del punto de rocío. Si bien los resultados obtenidos en relación a la precisión del modelo no son excepcionales, se consideran satisfactorios y representan un avance significativo en la implementación del sistema de control. También se ha desarrollado una documentación sobre el código, que puede proporcionar una base sólida para futuros trabajos consecutivos y permite a otros investigadores construir mejoras sobre los resultados obtenidos.

En términos de las limitaciones enfrentadas durante el desarrollo de este trabajo, se destaca el desconocimiento inicial acerca de las series temporales y su análisis. Esta limitación ha requerido un esfuerzo adicional para adquirir los conocimientos necesarios y abordar el problema de manera efectiva. Sin embargo, este desafío ha permitido una comprensión más profunda de las peculiaridades de las series temporales en el contexto de la predicción del punto de rocío. También es importante destacar que el punto de rocío presenta desafíos adicionales en comparación con otras variables climáticas, como la temperatura. A pesar de tener cierta estacionalidad, el punto de rocío muestra una alta variabilidad, pudiendo variar decenas de grados en tan solo una hora; además de ser dependiente de un gran número de factores. Estas características dificultan la predicción precisa y representan un factor a considerar al implementar el sistema de control basado

en las predicciones de esta variable.

En resumen, a lo largo de este trabajo se han abordado diferentes aspectos relacionados con la predicción del punto de rocío y su implementación en el Sistema de Control de TTNN. A pesar de las limitaciones encontradas, se ha logrado realizar un análisis exhaustivo del estado del arte, obtener y formatear los datos, comparar distintos modelos de aprendizaje automático y realizar predicciones satisfactorias. Estos resultados representan un avance significativo en la implementación del sistema de control y sientan las bases para futuras investigaciones y mejoras en la predicción del punto de rocío.

7.2. Líneas futuras

El trabajo desarrollado hasta el momento ha sentado las bases para futuras investigaciones y mejoras en la predicción del punto de rocío para su implementación en el Sistema de Control de TTNN. A continuación, se presentan las líneas futuras que pueden abordarse para continuar avanzando en este campo.

En primer lugar, se plantea adaptar la predicción a las pendientes positivas que se acercan a cero. En el sistema de control, es crucial medir con precisión cuando el punto de rocío supera los 0º, ya que esto puede tener implicaciones significativas en su funcionamiento. Por lo tanto, sería relevante investigar y desarrollar técnicas específicas para identificar y predecir con exactitud estos casos, adaptando los modelos de predicción existentes. Además, se sugiere expandir el conjunto de datos de prueba utilizado en este trabajo. Actualmente, se ha centrado en los meses de febrero y marzo, pero sería conveniente probar el modelo en otras estaciones, como el verano. Esto permitiría evaluar la capacidad de generalización de los modelos y verificar su desempeño en condiciones climáticas diferentes [47].

Otra línea futura prometedora es el uso de técnicas híbridas en la predicción del punto de rocío, como las empleadas en [2] o [12]. Estas técnicas combinan enfoques de diferentes métodos, como modelos basados en reglas, algoritmos de aprendizaje automático y técnicas de optimización, aprovechando las fortalezas de cada enfoque y mejorar la precisión de las predicciones. Por tanto, explorar y desarrollar técnicas híbridas específicas para la predicción del punto de rocío podría ayudar a brindar resultados más sólidos y confiables. Además, sería beneficioso profundizar en técnicas más complejas de deep learning, que han demostrado ser altamente eficaces en la predicción de series temporales [33] y en la extracción de características complejas de los datos.

Un aspecto importante a considerar en futuras investigaciones es probar y ajustar los modelos a mediciones generadas en una zona cercana al espejo de ambos telescopios nocturnos. Se ha observado que el punto de rocío depende en gran medida del estado del viento y de otras variables meteorológicas que varían en gran medida en base a la ubicación u orientación de los sensores [42]. Al incorporar mediciones adicionales de estas variables y considerar su impacto en la predicción del punto de rocío, se podrían obtener mejores resultados.

Por último, sería relevante comprobar si los resultados se mantienen estables al emplear un retraso superior en los modelos de predicción. De esta manera, se podría ampliar la ventana de predicción sin aumentar significativamente el error. Explorar diferentes valores de retraso y evaluar su impacto en la precisión de los modelos podría ser un paso importante para mejorar la capacidad predictiva.

Capítulo 8

Conclusions and future work

8.1. Conclusions

In the present work, an analysis of the state of the art of dew point has been carried out, with the objective of developing a prediction model for its implementation in the Nighttime Telescope Control System of the Canary Islands Institute of Astrophysics (IAC). Throughout the study, several important aspects have been addressed, and the relevant conclusions in each of them are presented below.

Firstly, the necessary data for making the predictions has been obtained and formatted successfully. However, an important limitation in the dataset has been identified, which refers to the large number of missing values. This aspect hampers the accuracy of the predictions and poses a challenge in addressing the problem. Additionally, it has been observed that the data is at different frequencies or even at non-constant frequencies, introducing additional complexity, as it would be more convenient to have predictions made at a lower frequency.

Despite this, effective data processing has been carried out, significantly improving the obtained predictions. Throughout the work, the performance of different machine learning models has been compared, both under the univariate and multivariate approaches. This comparative analysis has allowed for the identification of the strengths and weaknesses of each model regarding dew point prediction. Although the results obtained in terms of model accuracy are not exceptional, they are considered satisfactory and represent a significant advancement in the implementation of the control system. Documentation of the code has also been developed, providing a solid foundation for future consecutive work and enabling other researchers to build upon the obtained results.

In terms of the limitations faced during the development of this work, the initial unfamiliarity with time series and their analysis stands out. This limitation has required additional effort to acquire the necessary knowledge and effectively address the problem. However, this challenge has allowed for a deeper understanding of the peculiarities of time series in the context of dew point prediction. It is also important to highlight that dew point presents additional challenges compared to other weather variables, such as temperature. Despite exhibiting some seasonality, dew point shows high variability, being able to vary by tens of degrees in just one hour, and it is dependent on a large number of factors. These characteristics hinder precise prediction and represent a factor to consider when implementing the control system based on predictions of this variable.

In summary, throughout this work, various aspects related to dew point prediction and its implementation in the TTNN Control System have been addressed. Despite

the encountered limitations, a comprehensive analysis of the state of the art has been performed, data has been obtained and formatted, different machine learning models have been compared, and satisfactory predictions have been made. These results represent a significant advancement in the implementation of the control system and lay the groundwork for future research and improvements in dew point prediction.

8.2. Future work

The work developed so far has laid the foundation for future research and improvements in dew point prediction for implementation in the TTNN Control System. The following are the future lines that can be pursued to continue advancing in this field.

Firstly, it is proposed to adapt the prediction to positive slopes approaching zero. In the control system, it is crucial to accurately measure when the dew point exceeds 0º, as this can have significant implications for its operation. Therefore, it would be relevant to investigate and develop specific techniques to accurately identify and predict these cases, by adapting the existing prediction models. Additionally, it is suggested to expand the testing dataset used in this work. Currently, it has focused on the months of February and March, but it would be beneficial to test the model in other seasons, such as summer. This would allow for evaluating the generalization capability of the models and verifying their performance under different weather conditions [47].

Another promising future line is the use of hybrid techniques in dew point prediction, as employed in [2] or [12]. These techniques combine approaches from different methods, such as rule-based models, machine learning algorithms, and optimization techniques, leveraging the strengths of each approach to improve prediction accuracy. Therefore, exploring and developing specific hybrid techniques for dew point prediction could help provide more robust and reliable results. Additionally, it would be beneficial to delve into more complex deep learning techniques [33], which have shown high effectiveness in time series prediction and extracting complex data features.

An important aspect to consider in future research is to test and adjust the models to measurements generated in an area near the mirror of both nighttime telescopes. It has been observed that the dew point is heavily dependent on wind conditions and other meteorological variables that vary significantly based on the location or orientation of the sensors [42]. By incorporating additional measurements of these variables and considering their impact on dew point prediction, better results could be obtained.

Lastly, it would be relevant to check if the results remain stable when using a higher lag in the prediction models. This way, the prediction window could be extended without significantly increasing the error. Exploring different lag values and evaluating their impact on the models' accuracy could be an important step to improve predictive capability.

Apéndice A

Tablas de variables de los conjuntos de datos

A.1. Meteo

Abreviatura	Nombre	Descripción	Unidad
WS	Velocidad del Viento	Velocidad promedio del viento en los últimos 5 minutos	km/h
WD	Dirección del Viento	Dirección promedio del viento en los últimos 5 minutos	grados
AT	Temperatura Promedio	Temperatura promedio en los últimos 5 minutos	grados Celsius
MAX_WS	Velocidad Máxima del Viento	Velocidad máxima del viento en los últimos 5 minutos	km/h
RH	Humedad Relativa	Humedad relativa promedio en los últimos 5 minutos	porcentaje
MAX_RH	Humedad Relativa Máxima	Humedad relativa máxima en los últimos 5 minutos	porcentaje
BP	Presión	Presión promedio en los últimos 5 minutos	bar
AS	Estabilidad del Aire	Estabilidad del aire en los últimos 5 minutos	escala (0...7)
WG	Ráfaga de Viento	Ráfaga máxima de viento en los últimos 5 minutos	km/h
DP	Punto de Rocío	Punto de rocío promedio en los últimos 5 minutos	grados Celsius

Tabla A.1: Variables del conjunto de datos *Meteo*

A.2. PWVMo

Abreviatura	**Nombre**	**Descripción**	**Unidad**
TZD	Retardo Troposférico en el Zenit		mm
PRESS	Presión	Presión promedio en los últimos 30 minutos	hPa
TEMP_2	Temperatura	Temperatura promedio en los últimos 30 minutos	Kelvin
RAW_PWV	Vapor de Agua Precipitable Sin Procesar	Vapor de agua precipitable sin procesar promedio en los últimos 30 minutos	mm
PWV	Vapor de Agua Precipitable	Vapor de agua precipitable promedio en los últimos 30 minutos	mm

Tabla A.2: Variables del conjunto de datos *PWVMo*

A.3. PCPDIA

Abreviatura	Nombre	Descripción	Unidad
PH	Precipitación horaria	Precipitación en la última hora	Décimas de milímetro (ml)
PT	Precipitación total	Precipitación diaria total	Décimas de ml
PMAX10	Precipitación máxima en 10 minutos	-3: Sin precipitación	Décimas de ml
PMAX20	Precipitación máxima en 20 minutos	-3: Sin precipitación	Décimas de ml
PMAX30	Precipitación máxima en 30 minutos	-3: Sin precipitación	Décimas de ml
PMAX60	Precipitación máxima en 60 minutos	-3: Sin precipitación	Décimas de ml
PMAX2H	Precipitación máxima en 2 horas	-3: Sin precipitación	Décimas de ml
PMAX6H	Precipitación máxima en 6 horas	-3: Sin precipitación	Décimas de ml
PMAX12H	Precipitación máxima en 12 horas	-3: Sin precipitación	Décimas de ml
PMAX24H	Intensidad máxima de precipitación	-3: Sin precipitación	Décimas de ml
PHMAX	Hora de máxima intensidad de precipitación	-3: Sin precipitación	Hora
PWD	Dirección del viento en la máxima intensidad de precipitación	99: Viento variable; 88: Sin datos; 0: Calma	Grados
PWS	Velocidad del viento en la máxima intensidad de precipitación	Ninguna	km/h
PDUR	Duración de la precipitación	Ninguna	Minutos
PTR	Precipitación acumulada en el rango actual	Los rangos son 00-07 07-13 13-18 y 18-24	Décimas de ml por hora
PT07	Precipitación acumulada de 00 a 07	-4: Precipitación acumulada; -3: Sin precipitación	Décimas de ml
PT13	Precipitación acumulada de 07 a 13	-4: Precipitación acumulada; -3: Sin precipitación	Décimas de ml
PT18	Precipitación acumulada de 13 a 18	-4: Precipitación acumulada ; -3: Sin precipitación	Décimas de ml
PT24	Precipitación acumulada de 18 a 24	-4: Precipitación acumulada; -3: Sin precipitación	Décimas de ml
FORMA	Forma de precipitación	1: Lluvia; 2: Nieve; 3: Granizo; 4/5: Forma de precipitación no disponible	Categórica

Tabla A.3: Variables del conjunto de datos *PCPDIA*

A.4. INSDIA

Abreviatura	Nombre	Descripción	Unidad
PTJESOL	Porcentaje de insolación	Porcentaje de insolación sobre la insolación total diaria	%
SOL	Insolación en el rango horario	Insolación en el rango horario en el que se registra esta hora	Décimas de horas
TOTSOL	Insolación total diaria		Décimas de horas
SOL_DIV	Insolación por hora		Décimas de horas
SOL07	Insolación de 00 a 07		Décimas de horas
SOL13	Insolación de 07 a 13		Décimas de horas
SOL18	Insolación de 13 a 18		Décimas de horas
SOL00	Insolación de 18 a 00		Décimas de horas

Tabla A.4: Variables del conjunto de datos *INSDIA*

Apéndice B

Figuras y tablas del capítulo de desarrollo

B.1. Mapas de calor clusterizados

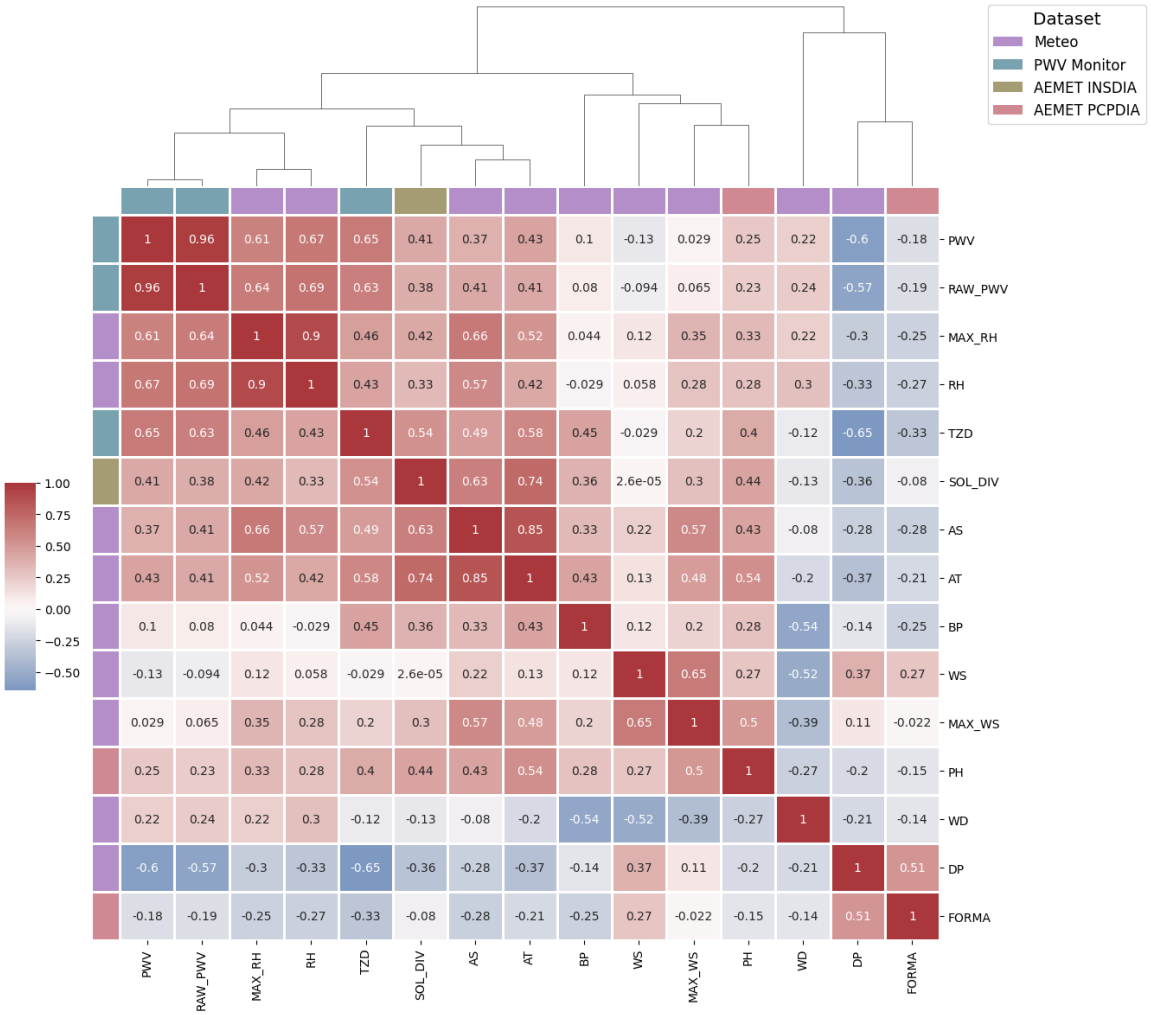

Figura B.1: Mapa de correlación de la estacionalidad de variables seleccionadas. Retraso de 8 horas aplicado.

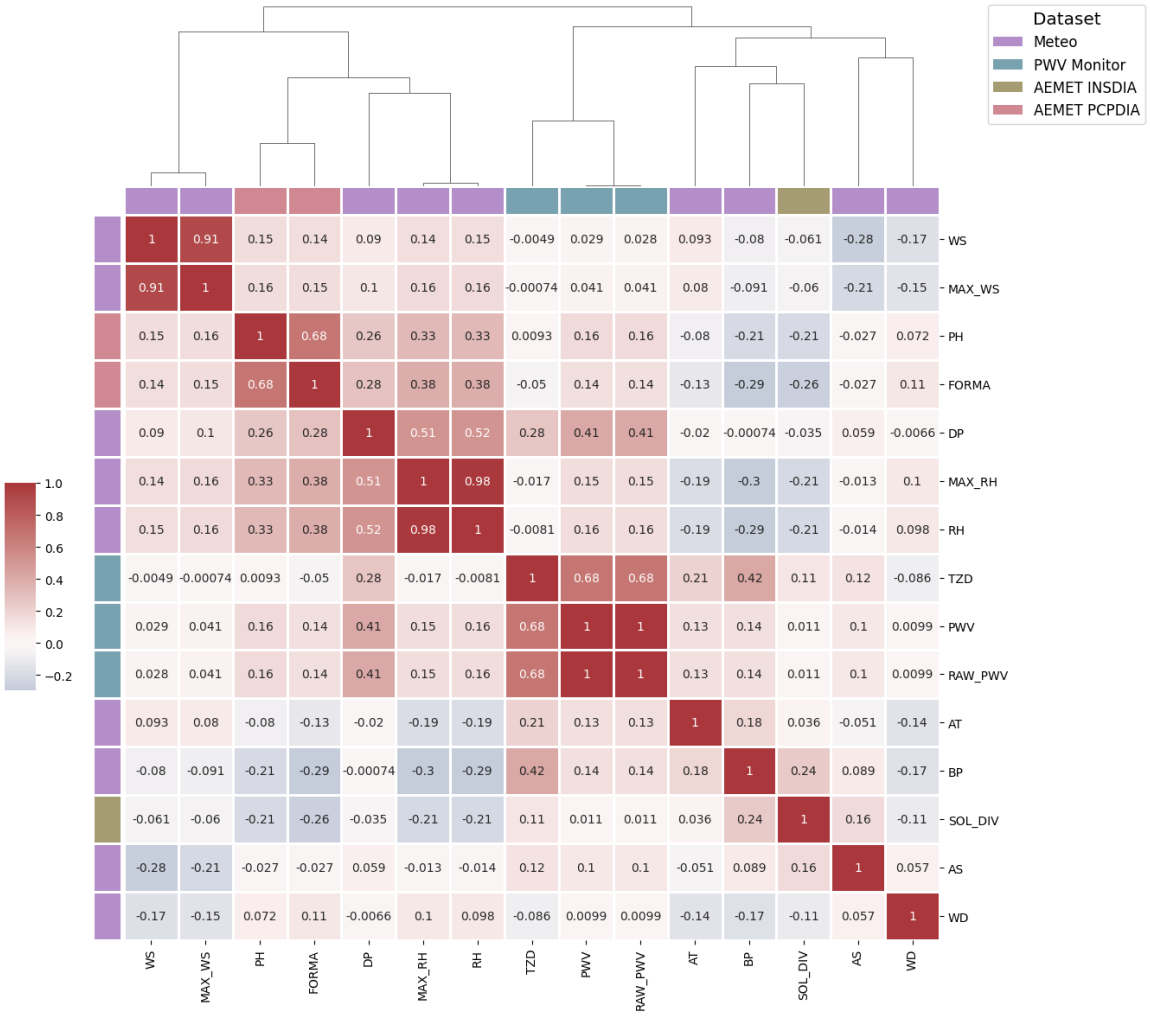

Figura B.2: Mapa de correlación de la tendencia de variables seleccionadas. Retraso de 8 horas aplicado.

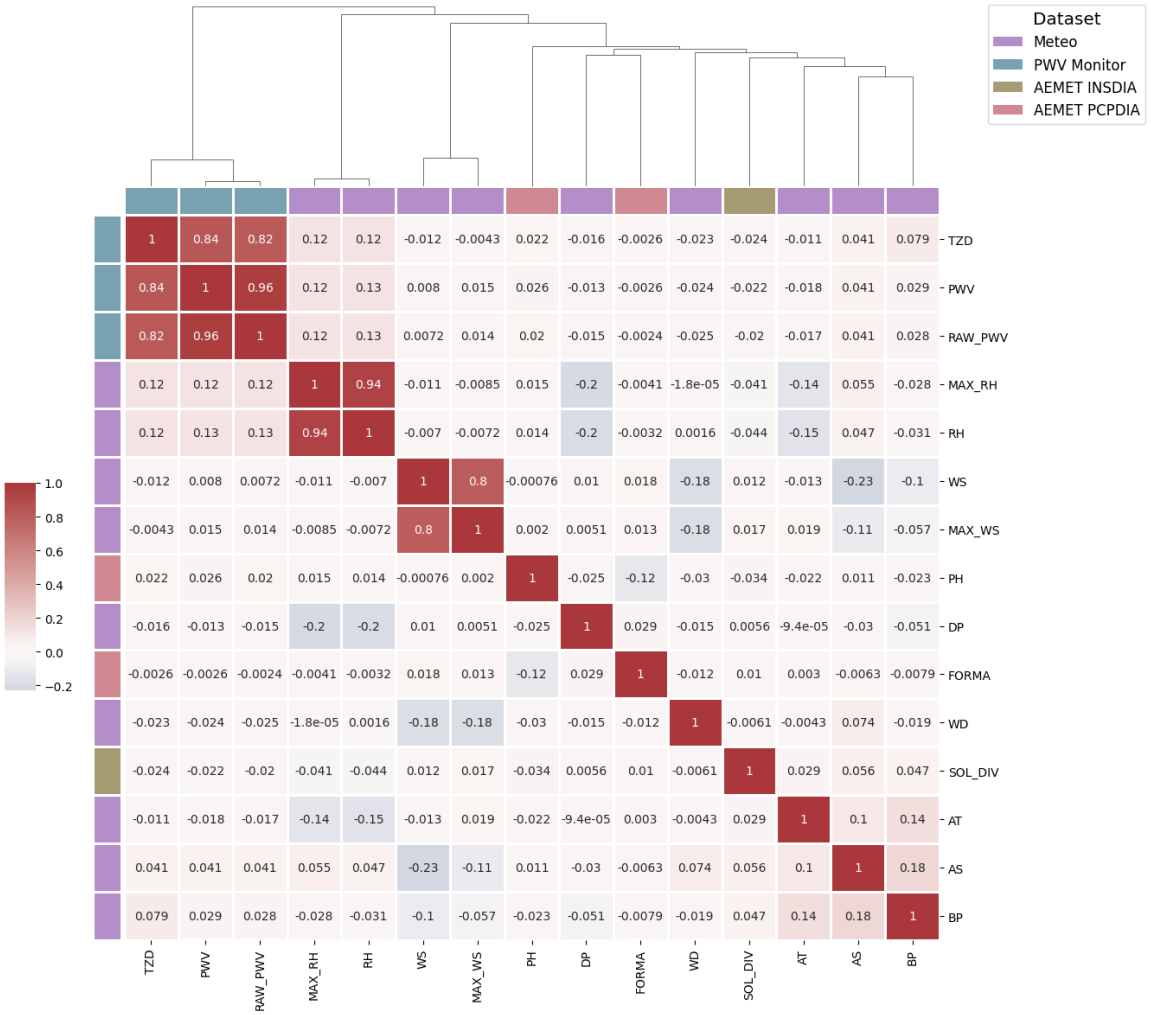

Figura B.3: Mapa de correlación del residuo de variables seleccionadas. Retraso de 8 horas aplicado.

B.2. Tabla de resultados de selección de características por filtro

Características	Pearson R	Spearman Rho	Kendall Tau
DP	0.583097	0.613219	0.456224
RH	0.447166	0.513584	0.358750
MAX_RH	0.441554	0.504338	0.351509
PWV	0.432473	0.448180	0.310540
RAW_PWV	0.431901	0.447919	0.310048

Tabla B.1: Tabla de resultados en la selección de características utilizando técnicas de filtro. Se muestran los cinco mejores.

Apéndice C

Página de documentación del código

La página de documentación utilizada [9] tiene una entrada para cada script documentado. Dentro de cada uno de ellos es posible consultar la información acerca de las funciones incluidas en dicho fichero, sus parámetros y su valor de retorno (véase Figura C.1). Asimismo, tiene integrado un buscador mediante el que se puede consultan la información requerida dentro del contenido de la página de la manera que se muestra en la Figura C.2.

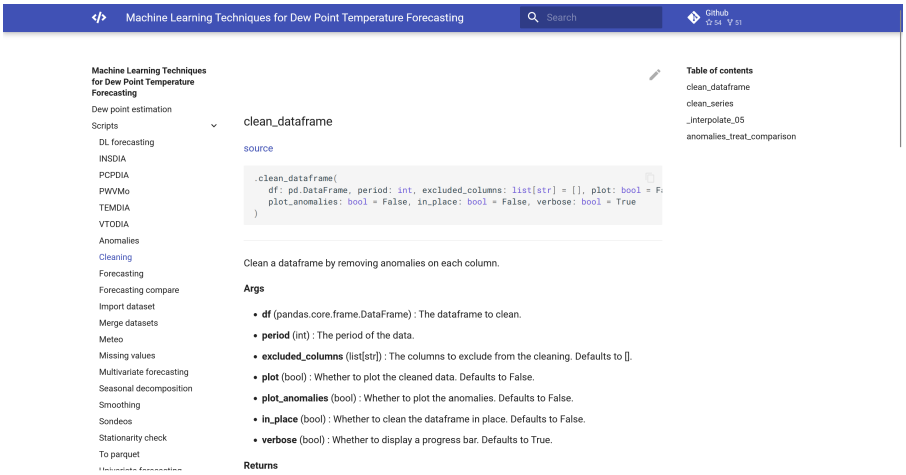

Figura C.1: Ejemplo de un script de la página de documentación.

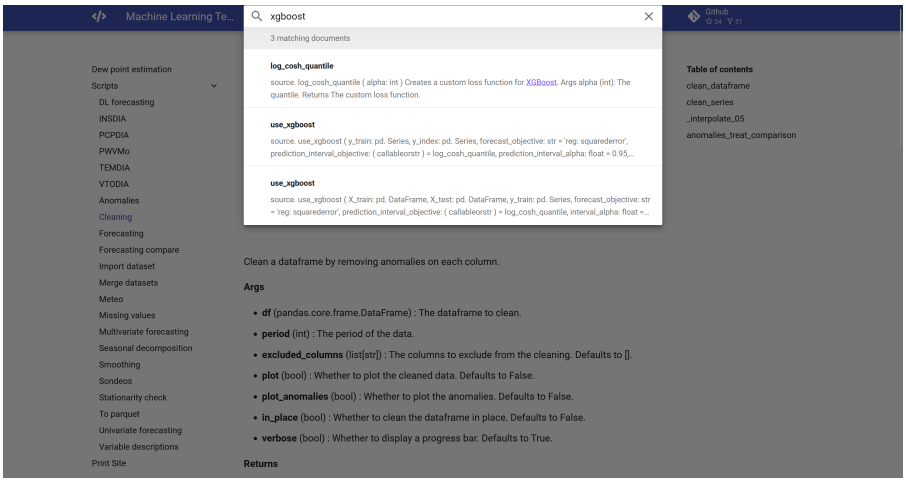

Figura C.2: Ejemplo de búsqueda en la página de documentación.

Apéndice D

Tablas de resultados de la optimización de regresores y de hiperparámetros

D.1. Tablas de resultados de selección de regresores y optimización de hiperparámetros

Índice	RMSE	X_features
0	4.512015	(DP, SOL_DIV, PWV, WS)
1	4.523510	(DP, SOL_DIV, AS, WS)
2	4.598284	(DP, RH, PWV, AS)
3	4.607802	(DP, SOL_DIV, PWV, MAX_WS)
4	4.667623	(DP, PWV, MAX_RH, AS)
5	4.676113	(DP, SOL_DIV, BP, WS)
6	4.664783	(DP, SOL_DIV, PWV, AS)
7	4.698971	(DP, PWV, AS, PH)
8	4.717039	(DP, RH, SOL_DIV, FORMA)
9	4.769260	(DP, SOL_DIV, PWV, PH)

Tabla D.1: Tabla de resultados en la selección de regresores para el modelo XGBoost ordenada por RMSE. Se muestran los diez mejores.

index	relative_score	RMSE	X_features	n_estimators	max_depth	learning_rate	max_bin
0	3.900007	4.277522	(DP, SOL_DIV, PWV, WS)	150	3	0.1	300
1	3.921801	4.269254	(DP, SOL_DIV, PWV, WS)	200	3	0.1	300
2	3.978359	4.340945	(DP, SOL_DIV, PWV, MAX_WS)	200	3	0.1	300
3	3.987145	4.368874	(DP, SOL_DIV, PWV, MAX_WS)	150	3	0.1	300
4	3.992841	4.410785	(DP, SOL_DIV, PWV, WS)	100	3	0.1	300
5	3.995299	4.405119	(DP, SOL_DIV, PWV, MAX_WS)	100	3	0.1	400
6	3.996034	4.371015	(DP, SOL_DIV, PWV, MAX_WS)	150	3	0.1	400
7	4.012114	4.423147	(DP, SOL_DIV, PWV, MAX_WS)	100	3	0.1	300
8	4.015916	4.326406	(DP, SOL_DIV, PWV, WS)	200	3	0.1	255
9	4.017572	4.373986	(DP, SOL_DIV, PWV, MAX_WS)	200	3	0.1	255

Tabla D.2: Tabla de resultados en la optimización de hiperparámetros para el modelo XGBoost. Se muestran los diez mejores.

Apéndice E

Fragmentos de código

E.1. Importación de datos

```
def split(df: pd.DataFrame) -> pd.DataFrame:
    pd.DataFrame(index=np.arange(df.shape[0] * 24))
    new_df["YEAR"] = np.repeat(df["YEAR"].values.astype(int), 24)
    new_df["MONTH"] = np.repeat(df["MONTH"].values.astype(int), 24)
    new_df["DAY"] = np.repeat(df["DAY"].values.astype(int), 24)
    new_df["HOUR"] = np.tile(np.arange(24), df.shape[0])
    new_df["TOTSOL"] = np.repeat(df["TOTSOL"].values, 24)
    new_df["PTJESOL"] = np.repeat(df["PTJESOL"].values, 24)
    new_df["SOL"] = np.repeat(
        df.filter(regex="SOL[0-9]{2}").values, [7, 6, 5, 6], axis=1
    ).reshape(-1, 1)
    new_df["SOL_DIV"] = np.repeat(
        df.filter(regex="SOL[0-9]{2}").values, [7, 6, 5, 6], axis=1
    ).reshape(-1, 1) / np.tile(
        np.repeat([7, 6, 5, 6], [7, 6, 5, 6], axis=0), df.shape[0]
    ).reshape(
        -1, 1
    )
    new_df["SOL07"] = np.repeat(df["SOL07"].values, 24)
    new_df["SOL13"] = np.repeat(df["SOL13"].values, 24)
    new_df["SOL18"] = np.repeat(df["SOL18"].values, 24)
    new_df["SOL00"] = np.repeat(df["SOL00"].values, 24)
    return new_df
```

Listing E.1: Función split en la importación de INSDIA.

E.2. Limpieza de Datos

```
def clean_dataframe(
    df: pd.DataFrame,
    period: int,
    excluded_columns: list[str] = [],
    plot: bool = False,
    plot_anomalies: bool = False,
    in_place: bool = False,
    verbose: bool = True,
) -> pd.DataFrame or None:
    df = df if in_place else df.copy()
```

```
    columns = list(set(df.columns) - set(excluded_columns))
    if verbose:
        with alive_bar(
            len(columns), bar="filling", spinner="dots_waves", title="Cleaning dataframe"
        ) as bar:
            for col in columns:
                clean_series(
                    df[col],
                    period,
                    columns_limits.get(col, default_limits),
                    plot=plot,
                    plot_anomalies=plot_anomalies,
                    in_place=True,
                )
                bar()
    else:
        for col in columns:
            clean_series(
                df[col], period, plot=plot, plot_anomalies=plot_anomalies, in_place=True
            )
    if not in_place:
        return df
```

Listing E.2: Función clean_dataframe.

```
def interpolate_op(...) -> pd.Series:
    non_anomalous = df[anomalies <= limit].copy()
    if non_anomalous.index[0] != df.index[0]:
        non_anomalous = pd.concat(
            [pd.Series([non_anomalous[-1]], index=[df.index[0]]), non_anomalous]
        )
    if non_anomalous.index[-1] != df.index[-1]:
        non_anomalous = pd.concat(
            [non_anomalous, pd.Series([non_anomalous[0]], index=[df.index[-1]])]
        )
    non_anomalous = non_anomalous.resample("10T").mean().interpolate(method="linear")
    return non_anomalous * anomalies + df * (1 - anomalies)
```

Listing E.3: Suavizado por interpolación.

```
def bi_rolling_op(
    df: pd.Series, period: int, anomalies: pd.Series, limit: float
) -> pd.Series:
    for i in range(1, len(df) - 1):
        if anomalies[i] > limit:
            prev_non_anomalous = anomalies[: (i - period)][
                anomalies[: (i - period)] <= limit
            ].index
            if prev_non_anomalous.empty:
                prev_non_anomalous = anomalies[:-1][anomalies[:-1] <= limit].index[-1]
            else:
                prev_non_anomalous = prev_non_anomalous[-1]
            next_non_anomalous = anomalies[(i + period) :][
                anomalies[(i + period) :] <= limit
            ].index
            if next_non_anomalous.empty:
                next_non_anomalous = anomalies[0:][anomalies[0:] <= limit].index[0]
            else:
                next_non_anomalous = next_non_anomalous[0]
```

```
            df[i] = ((df[prev_non_anomalous] + df[next_non_anomalous]) / 2) * anomalies[
                i
            ] + df[i] * (1 - anomalies[i])
    return df
```

Listing E.4: Suavizado por *bi-rolling*.

E.3. Despliegue en Control TTNN

```
def mainIteration(self):
    if self.lastNumbersClients != self.pub.get_num_connections() and self.is_published:
        self.pub.publish(self.currentStatus)
    self.lastNumbersClients = self.pub.get_num_connections()
    self.rate.sleep()
```

Listing E.5: Función *mainIteration* se ejecuta cada 5 minutos

Bibliografía

[1] Using machine learning to optimize the search for biosignatures. *Nature Astronomy*, 7(4):378–379, 2023.

[2] Mohsen Amirmojahedi, Kasra Mohammadi, Shahaboddin Shamshirband, Amir Seyed Danesh, Ali Mostafaeipour, and Amirrudin Kamsin. A hybrid computational intelligence method for predicting dew point temperature. *Environmental earth sciences*, 75(5):1–12, 2016.

[3] Ross Andersen. How big data is changing astronomy (again). Último acceso 20/04/2023.

[4] John A Armstrong and Lyndsay Fletcher. A machine-learning approach to correcting atmospheric seeing in solar flare observations. *Monthly notices of the Royal Astronomical Society*, 501(2):2647–2658, 2021.

[5] Aayush Bajaj. Anomaly detection in time series. Último acceso 05/04/2023.

[6] Brahim Belmahdi, Mohamed Louzazni, Mousa Marzband, and Abdelmajid El Bouardi. Global solar radiation forecasting based on hybrid model with combinations of meteorological parameters: Morocco case study. *Forecasting*, 5(1):172–195, 2023.

[7] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on knowledge discovery and data mining*, KDD '16, pages 785–794, Ithaca, 2016. ACM.

[8] M.Sc. Dave Cote. Rdr score metric for evaluating time series forecasting models, 2022. Último acceso 07/04/2023.

[9] Louis de Bruijn. Five tips for automatic python documentation.

[10] Astrophys Del Popolo. Core vs. cusp, 2009.

[11] Nathalie Deruelle and Jean-Philippe Uzan. 605The Lambda-CDM model of the hot Big Bang. In *Relativity in Modern Physics*. Oxford University Press, 08 2018.

[12] Jianhua Dong, Wenzhi Zeng, Guoqing Lei, Lifeng Wu, Haorui Chen, Jingwei Wu, Jiesheng Huang, Thomas Gaiser, and Amit Kumar Srivastava. Simulation of dew point temperature in different time scales based on grasshopper algorithm optimized extreme gradient boosting. *Journal of hydrology (Amsterdam)*, 606:127452, 2022.

[13] Ahmed M. Elshewey, Mahmoud Y. Shams, Abdelghafar M. Elhady, Samaa M. Shohieb, Abdelaziz A. Abdelhamid, Abdelhameed Ibrahim, and Zahraa Tarek. A novel wd-sarimax model for temperature forecasting using daily delhi climate dataset. *Sustainability (Basel, Switzerland)*, 15(1):757, 2023.

[14] Julen Expósito Márquez. *A probabilistic deep learning model to distinguish cusps and cores in dwarf galaxies*. 2023.

[15] Martin Gaspar, Bastian Welke, Frank Seehaus, Christof Hurschler, and Michael Schwarze. Dynamic time warping compared to established methods for validation of musculoskeletal models. *Journal of Biomechanics*, 55:156–161, 2017.

[16] Sankalp Gilda, Stark C Draper, Sébastien Fabbro, William Mahoney, Simon Prunet, Kanoa Withington, Matthew Wilson, Yuan-Sen Ting, and Andrew Sheinis. Uncertainty-aware learning for improvements in image quality of the canada–france–hawaii telescope. *Monthly Notices of the Royal Astronomical Society*, 510(1):870–902, 2021.

[17] A. Gneiting; T.; & Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):370, 2007.

[18] Bradley Boehmke & Brandon Greenwell. Hands-on machine learning with r, 2020.

[19] Bradley Boehmke & Brandon Greenwell. Hands-on machine learning with r, 2020.

[20] José Manuel Guevara Díaz. Uso correcto de la correlación cruzada en climatología: el caso de la presión atmosférica entre taití y darwin. *Terra. Nueva Etapa*, 2014.

[21] IAC. Anisotropía del fondo cósmico de microondas.

[22] IAC. Nuevo sistema de control para los telescopios nocturnos.

[23] IAC. Vapor de agua precipitable. Último acceso 02/04/2023.

[24] IBM. What is random forest? Último acceso 09/04/2023.

[25] IBM. ¿qué son las redes neuronales recurrentes?

[26] Kuhn; M.; & Johnson; K. *Applied Predictive Modeling*. 2013.

[27] Joos Korstanje. How to select a model for your time series prediction task [guide], 2023. Último acceso 07/04/2023.

[28] Zhen Li, Tieding Lu, Xiaoxing He, Jean-Philippe Montillet, and Rui Tao. An improved cyclic multi model-extreme gradient boosting (cmm-xgboost) forecasting algorithm on the gnss vertical time series. *Advances in space research*, 71(1):912–935, 2023.

[29] Javier Cantero Lorenzo. Máquinas de aprendizaje y aplicaciones.

[30] Orlando Luongo and Marco Muccino. Model-independent calibrations of gamma-ray bursts using machine learning. *Monthly notices of the Royal Astronomical Society*, 503(3):4581–4600, 2021.

[31] Richard H. McCuen, Zachary Knight, and A. Gillian Cutter. Evaluation of the nash–sutcliffe efficiency index. *Journal of Hydrologic Engineering*, 11(6):597–602, 2006.

[32] Luis Miralles-Pechuán, Ankit Kumar, and Andrés L. Suárez-Cetrulo. Forecasting covid-19 cases using dynamic time warping and incremental machine learning methods. *Expert Systems*, 2023.

[33] Kasra Mohammadi, Shahaboddin Shamshirband, Dalibor Petković, Por Lip Yee, and Zulkefli Mansor. Using anfis for selection of more relevant parameters to predict dew point temperature. *Applied thermal engineering*, 96:311–319, 2016.

[34] Małgorzata Murat, Iwona Malinowska, Magdalena Gos, and Jaromir Krzyszczak. Forecasting daily meteorological time series using arima and regression models. *International Agrophysics*, 32(2):253–264, 2018.

[35] AEMET Navarra. Insolación. Último acceso 02/04/2023.

[36] Eberly College of Science. Vector autoregressive models var(p) models.

[37] Christopher Olah. Understanding lstm networks, 2015.

[38] Michał Oleszak. Feature selection methods. Último acceso 04/04/2023.

[39] otexts. Holt-winters' seasonal method.

[40] Antonios Parasyris, George Alexandrakis, Georgios V. Kozyrakis, Katerina Spanoudaki, and Nikolaos A. Kampanis. Predicting meteorological variables on local level with sarima, lstm and hybrid techniques. *Atmosphere*, 13(6):878, 2022.

[41] Roger D. Peng. A very short course on time series analysis, 2020.

[42] SultanÑoman Qasem, Saeed Samadianfard, Hamed Sadri Nahand, Amir Mosavi, Shahaboddin Shamshirband, and Kwok-wing Chau. Estimating daily dew point temperature using machine learning algorithms. *Water (Basel)*, 11(3):582, 2019.

[43] Jordi Casas Roma. Introducción al análisis de series temporales. Último acceso 16/04/2023.

[44] Justin Shen, Davesh Valagolam, and Serena McCalla. Prophet forecasting model: A machine learning approach to predict the concentration of air pollutants (pm2.5, pm10, o3, no2, so2, co) in seoul, south korea. *PeerJ (San Francisco, CA)*, 8:e9961–e9961, 2020.

[45] Sean J. Taylor and Benjamin Letham. Forecasting at scale. *The American statistician*, 72(1):37–45, 2018.

[46] The Investopedia Team. Variance inflation factor (vif). Último acceso 03/04/2023.

[47] Lide Tian. Oxygen-18 isotopes in precipitation on the eastern tibetan plateau.

[48] Tibco. What is a random forest.

[49] Oskar Triebe, Hansika Hewamalage, Polina Pilyugina, Nikolay Laptev, Christoph Bergmeir, and Ram Rajagopal. Neuralprophet: Explainable forecasting at scale, 2021.

[50] ULPGC. Tema iv: Series temporales. Último acceso 16/04/2023.

[51] Juan Carlos Vesga Ferreira, Martha Fabiola Contreras Higuera, and José Antonio Vesga Barrera. Uso del modelo de holt-winters como estrategia para la predicción de condiciones ambientales durante el proceso de almacenamiento del cacao. *Revista EIA*, 19(38), 2022.

[52] Simon White. Fundamentalist physics: why dark energy is bad for astronomy. 2007.

[53] Guodao Zhang, Sayed M. Bateni, Changhyun Jun, Helaleh Khoshkam, Shahab S. Band, and Amir Mosavi. Feasibility of random forest and multivariate adaptive regression splines for predicting long-term mean monthly dew point temperature. *Frontiers in environmental science*, 10, 2022.

[54] Yifan Zhang and Peter J. Thorburn. A dual-head attention model for time series data imputation. *Computers and Electronics in Agriculture*, 189:106377, 2021.

[55] Mohammad Zounemat-Kermani. Hourly predictive levenberg–marquardt ann and multi linear regression models for predicting of dew point temperature. *Meteorology and atmospheric physics*, 117(3-4):181–192, 2012.