Universitat Oberta de Catalunya

uoc.edu

Estudio comparativo de modelos de predicción estocásticos y heurísticos aplicados a la estimación de la calidad del aire

Autor: Nadia Nathaly Sánchez Pozo

Máster Universitario en Ciencia de Datos

Área 5

Tutor: Sergio Trilles Oliver

Cotutor: Diego Peluffo Ordóñez

Profesor: Albert Solé Ribalta

julio de 2020

Copyright © 2020 Nadia Sánchez Pozo.

Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-Compartir Igual 3.0 España de Creative Commons

FICHA DEL TRABAJO FINAL

Título del trabajo:	*Estudio comparativo de modelos de predicción estocásticos y heurísticos aplicados a la estimación de la calidad del aire*
Nombre del autor:	*Nadia Sánchez Pozo*
Nombre del consultor/a:	*Sergio Trilles Oliver*
Nombre del Cotutor:	*Diego Peluffo Ordóñez*
Nombre del PRA:	*Albert Solé Ribalta*
Fecha de entrega (mm/aaaa):	06/2020
Titulación:	*Máster Universitario en Ciencia de Datos*
Área del Trabajo Final:	*Área 5*
Idioma del trabajo:	*Español*
Palabras clave	*Predicción, calidad del aire, contaminación*

Resumen del Trabajo (máximo 250 palabras): *Con la finalidad, contexto de aplicación, metodología, resultados i conclusiones del trabajo.*

Este trabajo presenta un análisis comparativo de modelos predictivos, aplicados a la estimación de calidad del aire. Actualmente, entre las inquietudes mundiales, se encuentra la preocupación por la contaminación del aire, por ello, en ciudades como Londres existen sistemas de monitoreo de contaminantes atmosféricos.

El notable deterioro de la calidad del aire en Londres es un problema cada vez más grave, considerando que existe una relación directa con problemas de salud respiratoria y cardiaca lo cual ha sido ya causa de muerte en dicha ciudad.

El objetivo de este estudio es analizar diferentes modelos predictivos, para comparar y determinar cuál de ellos nos permite realizar una mejor predicción de la calidad del aire de Londres.

Para ello, se hará uso de un conjunto de datos abiertos recuperado del portal London Datastore. Estos datos históricos son resultados de las mediciones del sistema de monitoreo de contaminantes de la ciudad. Dichos datos serán utilizados para entrenar los algoritmos ARIMA, SVM, Redes Neuronales y Facebook Prophet.

A partir de los modelos generados, se determinará cuál de ellos tiene mayor exactitud a la hora de predecir la concentración de contaminantes atmosféricos.

Abstract (in English, 250 words or less):

This work presents a comparative analysis of predictive models, applied to the estimation of air quality, currently among the world concerns is the concern about air pollution, therefore, in cities like London there are air pollution monitoring systems.

The notable deterioration of air quality in London is an increasingly serious problem, considering that there is a direct relationship with respiratory and cardiac health problems being already the cause of death in that city.

The objective of this study is to analyze and compare different predictive models, for determining which of them allows us to do a better prediction of London's air quality.

To do so, an open data set recovered from the London Datastore portal is used, which are historical data corresponding to measurements of the city's pollutant monitoring system. These data are used to train the ARIMA, SVM, Neural Networks and Facebook Prophet algorithms.

From the generated models generated, the ones reaching greater accuracy when predicting the concentration of air pollutants are to be determined.

Índice

Lista de tablas

Lista de figuras

1. Introducción

1.1. Contexto y justificación del Trabajo

Hoy en día existen varias agencias ambientales a nivel mundial que desarrollan sus propias políticas y han establecido estándares e indicadores de calidad del aire con respecto a los niveles permitidos de contaminantes atmosféricos.

En 1993, se estableció la red de monitoreo ambiental de Londres, esta red cuenta con sistemas de monitoreo de contaminantes en 30 de las áreas suburbanas de la ciudad [1]. En Londres, alrededor de 9.500 personas mueren prematuramente cada año debido a la exposición a aire contaminado, los principales contaminantes causantes de muerte son las partículas finas denominadas PM2.5 y dióxido de nitrógeno (NO2). La organización Mundial de la Salud señaló que el impacto de la contaminación es particularmente grave en los niños ya que no solo afecta el desarrollo de sus pulmones sino de sus habilidades cognitivas.

Actualmente, existen varios indicadores de calidad del aire que manifiestan los efectos de la contaminación en la salud de las personas. Entre los más importantes se encuentran: monóxido de carbono (CO), dióxido de azufre (SO2) y dióxido de nitrógeno (NO2) [2]. Cuando el nivel de concentración de un indicador excede un umbral de seguridad de calidad del aire establecido, puede afectar la salud de los seres humanos.

Los resultados de la medición de la red de sensores son observaciones igualmente espaciadas y ordenadas en el tiempo, lo que resulta en una serie de concentraciones de contaminantes[2]. Estas observaciones son almacenadas en repositorios de libre acceso, lo cual beneficia las tareas de investigación ya que dichos datos pueden ser analizados mediante técnicas de machine learning, obteniendo modelos basados en datos.

1.1.2. Justificación

El aumento de la contaminación del aire es un tema de preocupación mundial debido a la relación que existe con la salud, es por esto por lo que siempre se están investigando nuevas formas para reducir los niveles de contaminación.

El aprovechamiento de la tecnología brinda la posibilidad de contar con una extensa base de datos de mediciones de contaminantes atmosféricos de Londres los cuales se encuentran en el portal abierto "LONDON DATASTORE".

Las investigaciones basadas en datos han beneficiado al desarrollo de algoritmos eficientes aportando nuevos conocimientos a la ciencia y tecnología. Por tal razón el objetivo de este estudio es comparar la eficiencia de modelos de predicción estocásticos y heurísticos para pronosticar la concentración de contaminantes atmosféricos y así estimar la calidad del aire de Londres con mayor exactitud.

1.1.3. Motivación personal

La calidad del aire es un tema de alto interés, puesto que el aire es esencialmente importante para la vida del planeta, es por esta razón, que se ha decidido realizar un estudio comparativo de modelos predictivos para estimar la calidad del aire.

Considerando que se trabajará con datos reales recuperados del portal London Datastore, se considera este estudio como una aportación a la ciudad de Londres y al medio ambiente, además de ser un proyecto con el cual se pondrá en práctica el conocimiento adquirido durante el Máter en Ciencia de Datos.

1.2. Objetivos del Trabajo

1.2.1. Objetivo general

Realizar un estudio comparativo de modelos de predicción (Arima, SVM, redes neuronales y Facebook Prophet), mediante el análisis de datos históricos de contaminantes atmosféricos de Londres, para evaluar la eficiencia de los modelos de predicción aplicados a la estimación de calidad del aire.

1.2.2. Objetivos específicos

- Determinar las variables de entrada, para entrenar los diferentes modelos de predicción estocásticos y heurísticos.
- Realizar un pre-procesado del conjunto de datos para garantizar la calidad del dato.
- Definir las métricas de evaluación de los modelos predictivos, para evaluar la eficiencia de estos.

1.3. Metodología

1.3.1. Estrategia de investigación

La estrategia que mejor se adapta a este estudio es la de investigación de análisis de datos cuantitativos. Esta estrategia se basa principalmente en un análisis estadístico el cual permitirá encontrar patrones entre los datos además de realizar un análisis exploratorio de estos ya que se utilizan tablas y gráficos para representar de manera visual los datos.

Los datos para analizar en este trabajo serán recuperados del portal libre London Datastore. En cuanto al desarrollo del proyecto se hará uso del lenguaje de programación Python.

1.3.2. Metodología del trabajo

La metodología aplicada para esté trabajo será iterativa ya que para alcanzar los objetivos de este proyecto se definen diversas fases, las cuales contemplan el desarrollo de los objetivos planteados en el apartado anterior.

En la primera fase se realizará un estudio de los modelos: Arima, SVM, redes neuronales y Facebook Prophet, además de realizar una comparativa teórica de estos. También se determinarán las variables de entrada para entrenar los modelos predictivos.

En la segunda fase, se aplicarán dichos modelos a un conjunto de datos encontrado en London Average Air Quality Levels , los datos de salida que se encuentran en el directorio de datos constan de dos archivos csv: Monthly-averages.csv, time-of-day-per-month.csv que datan de 01/01/2008 a 31/07/2019. Estos datos serán pre-procesados para garantizar la calidad del dato.

Luego se definirá métricas de evaluación de los modelos predictivos con el objetivo de evaluar la eficiencia de estos, para medir y comparar las capacidades predictivas de los algoritmos.

Al finalizar las anteriores fases se obtendrán conclusiones sobre el desarrollo del proyecto. Una vez concluido el proyecto se compartirán los datos y el código de investigación en un repositorio GitHub.

1. Planificación del trabajo

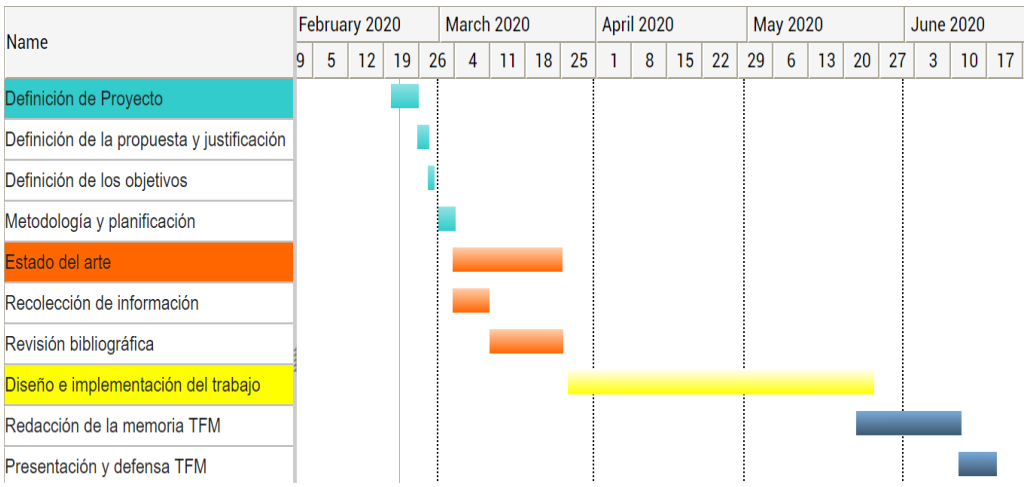
La planificación para el desarrollo de este proyecto está orientada a cumplir con las fechas de entrega establecidas en el calendario de PEC's proporcionado por la UOC. A continuación, en la Tabla 1. Planificación, se detalla las fechas de entrega.

Tabla 1. Planificación de trabajo de fin de máster

PEC	Descripción	Fecha Entrega	Duración Días
PEC 1	Definición y planificación del trabajo final	01/03/2020	11
PEC 2	Estado del arte o análisis de mercado del proyecto	22/03/2020	20
PEC 3	Diseño e implementación del trabajo	23/05/2020	61
PEC 4	Redacción de la memoria	10/06/2020	17
PEC 5	Presentación y defensa del proyecto	16/06/2020	6

En el diagrama de Gantt (Figura 1) se muestran las diferentes etapas de planificación del trabajo de fin de máster.

Figura 1: Diagrama de Gantt de la planificación de TFM



Fuente: elaboración propia

1.5. Breve descripción de los otros capítulos de la memoria

Para el desarrollo de este trabajo se han planteado diferentes fases, las cuales se distribuyen en los siguientes capítulos:

- Capítulo 2. Estado del arte

  En este capítulo se presenta una introducción a los aspectos relacionados con la estimación de la calidad del aire.

- En la sección 2.1, se presenta una revisión bibliográfica de algunos trabajos relevantes afines a la temática, específicamente, contaminantes a predecir, variables explicativas empleadas, algoritmos de aprendizaje automático y validación de modelos utilizados. En la sección 2.2, se explican los modelos de predicción. En la sección 2.3, se realiza una comparativa teórica de los modelos de predicción. Capítulo 3. Diseño e implementación del trabajo

Este capítulo se divide en tres partes: 3.1 procesamiento de datos, 3.2 aplicación de los modelos y 3.3 análisis comparativo de resultados

- Capítulo 4. Conclusiones

  En este capítulo se exponen las conclusiones obtenidas a lo largo del desarrollo del trabajo.

- Capítulo 5. Glosario

- Capítulo 6. Bibliografía

  Listado de la bibliografía consultada para la elaboración del trabajo.

- Capítulo 7. Anexos

  En este capítulo se adjunta el anexo “reproducibilidad”, donde se comparte la (URL) del repositorio GitHub donde se han guardado los datos y el código desarrollado en el proyecto.

2. Estado del arte

2.1. Revisión de trabajos anteriores

La inmisión o calidad del aire puede definirse como la cantidad de contaminante que llega a un receptor, más o menos alejado de la fuente de emisión. La calidad del aire se determina especialmente por la distribución geográfica de las fuentes de emisión de contaminantes, cuando se tiene pocos contaminantes se dice que la calidad del aire es buena [3].

En las últimas décadas, la Unión Europea ha adoptado una amplia gama de medidas ambientales para mejorar la calidad de vida de los ciudadanos, con el objetivo de controlar la contaminación del aire. En [3], se establece que los principales contaminantes atmosféricos son las partículas en suspensión, el dióxido de nitrógeno y el ozono troposférico. Por esta razón, se han definido valores límite de partículas en suspensión y los óxidos de nitrógeno (Directiva 2008/50/CE). En este trabajo, se compara las concentraciones de NOx, NO2, O3, CO, PM10 y PM2.5 medidas por los centros de monitoreo de Londres.

A continuación se presenta la revisión de algunos trabajos relevantes: en [4], se diseñan modelos basados en el uso del perceptrón multicapa con el objetivo de obtener predicciones, en tiempo real, de los niveles de ozono de Bilbao y Kostaldea. El [5], se estudian las variables meteorológicas que afectarían las concentraciones de contaminantes del aire, para entrenar algoritmos de optimización avanzada. En [6], se predice la concentración de ozono y PM10 mediante la implementación de tres tipologías de redes neuronales: Feed-Forward Neural Networks (FFNN), Pruned Neural Networks y Lazy learning, en donde se concluye que FFNN permite obtener mejores resultados a pesar de que este modelo puede presentar problemas de sobreajuste. En [7], se extraen diferentes patrones de calidad del aire en forma de reglas de asociación mediante el algoritmo CTSPD o Continuous Target Sequence Pattern Discovery el cual se divide en dos fases: fase de generación de secuencias frecuentes y fase de generación de reglas.

En [8], se desarrolla una revisión sistemática de diferentes enfoques de aprendizaje automático para el modelado de la calidad del aire exterior, en este trabajo se define una lista de contaminantes estudiados de acuerdo con la ubicación geográfica:

- Asia el mayor número de estudios se centran en predecir partículas PM2.5 seguido del NO2/NOx, ozono O3 PM10, SO2, CO, Nano PM,
- Europa se predice el ozono O3 PM10, NO2/NOX, SO2, CO y el índice general de calidad del aire (AQI),
- Norte América PM2.5, NO2/Nox, AQI, O3, PM10/BC,
- Sur América PM2.5,
- y África O3 y PM10, CO.

Entre los principales algoritmos de aprendizaje automático utilizados, se encuentran: Random Forest, algoritmos basados en redes neuronales artificiales, Support Vector Machine (SVM) y Support Vector Regression (SVR). Por su parte, para la evaluación de los modelos, en algunos artículos, se ha calculado el porcentaje de predicción, el error absoluto medio (MAE), el error cuadrático medio (RMSE) y el coeficiente de determinación (R2).

En [9], se realiza la implementación de un predictor de PM10 utilizando el método de redes neuronales para el cual se seleccionan como variables de entrada dos estacionales, y siete meteorológicas.

A continuación, se presenta una tabla resumen de los diferentes trabajos revisados.

Tabla 2. Revisión de trabajos anteriores

Referencia	Contaminantes a predecir	Variables	Modelos	Validación del modelo
[4]	O3 ($\mu g/m^3$) Ozono	Temperatura, humedad relativa, presión, radiación, gradiente térmico, velocidad del viento, dirección del viento, ozono, dióxido de nitrógeno, número de vehículos (vehic./10 min), porcentaje de ocupación, velocidad, sen(2πh/24), cos(2πh/24), sen(2πd/7), cos(2πd/7)	Redes neuronales, perceptrón multicapa, con una única capa intermedia.	-Coeficiente de correlación. -Error cuadrático medio normalizado (NMSE). -Factor de dos (FA2). -Sesgo fraccional (FB, Fractional Bias). -Varianza fraccional (FV, Fractional Variance).
[5]	Ozono, PM2.5 y dióxido de azufre	Temperatura del aire, velocidad del viento y dirección, humedad relativa,	MTL (Multi-task learning) aprendizaje multitarea regularizado.	-Error cuadrático medio (RMSE)

		radiación solar entrante y cobertura de nubes	Regresión lineal	
[6]	Ozono, PM10 ($\mu g/m^3$)	Radiación solar, temperatura y lluvia, presión atmosférica y clases de estabilidad, humedad y la velocidad del viento	Redes neuronales	-Correlación verdadera / pronosticada -Error medio absoluto MAE
[7]	CO (ppm), Ozono (ppb), NO2 (ppb), PM2.5($\mu g/m^3$), PM10($\mu g/m^3$),	Temperatura, humedad y velocidad del viento,	Algoritmo de minería de datos de series temporales CTSPD o Continuous Target Sequence Pattern Discovery	-Precisión
[9]	PM10	Dirección predominante del viento (DV), velocidad promedio del viento predominante (VV), velocidad promedio (V, teniendo en cuenta todas las direcciones), velocidad máxima	Redes neuronales	-Error cuadrático medio (RMSE) -Error medio absoluto MAE

		promedio (Vmax, teniendo en cuenta toda dirección posible), temperatura ambiente promedio (T), presión promedio (P), humedad promedio (H), mes del año (M) y día de la semana (DS)		
[10]	PM2.5	Concentración media PM2.5 por hora.	Transferred bi-directional long short-term memory (TL-BLSTM)	-Error cuadrático medio (RMSE) -Error medio absoluto MAE -Error porcentual absoluto medio, (MAPE)
[11]	Ozono O3	NO, NO2, PM10, SO2, CO, Temperatura, Humedad relativa, Velocidad del viento	Regresión múltiple y redes neuronales prealimentadas.	-Error de sesgo medio (MBE) -Error absoluto medio (MAE) -Error cuadrático medio (RMSE)

De acuerdo a la literatura revisada, se puede considerar que los criterios de evaluación más populares son el Error absoluto medio (MAE) y el Error cuadrático medio (RMSE) y los principales contaminantes del aire exterior en las ciudades son el ozono (O3), partículas (PM), dióxido de azufre (SO2), monóxido de carbono (CO), óxidos de nitrógeno (NOx). Dentro de los esquemas de predicción, las redes neuronales son los modelos predominantes en la mayoría de los trabajos revisados.

Los trabajos revisados abordan el problema de la estimación de la calidad del aire usando herramientas computarizadas, principalmente, basadas en inteligencia artificial y aprendizaje automático

Pese a los desarrollos logrados, se aprecia que el diseño de un sistema que alcance un buen equilibrio entre coste computacional, interpretabilidad, y precisión en la predicción es aún un problema abierto.

2.2. Modelos de predicción

2.2.1. Modelo autorregresivo integrado de media móvil (Arima)

ARIMA es uno de los métodos lineales más típicos para las predicciones de series de tiempo[12]. El modelo utiliza las variaciones y regresiones existentes entre los datos para determinar los patrones intrínsecos en la serie y, a partir de ellos, puede generar un pronóstico de estos. Este modelo se caracteriza por ser de bajo costo computacional y además su rendimiento depende de pocas configuraciones de parámetros. Los errores se deben estimar período a período una vez que el modelo se ajusta a los datos.

ARIMA consta de las siguientes componentes:

- Autorregresiva (AR): asume que el valor de la serie en un determinado instante se corresponde con la combinación lineal de la función en instantes anteriores (hasta un número máximo determinado de ellos, llamado “p”), a lo que se añade un componente de error aleatorio. Es decir, la información presente de un evento está relacionada con los valores pasados [13].

- Integración (I): se aplicarán sucesivas diferenciaciones en los casos en que las series muestren evidencia de no-estacionalidad[13].

- Promedio Móvil (MA): asume que el valor observado en un instante se corresponde con un término de error aleatorio a lo que le adiciona una combinación lineal de errores aleatorios previos (hasta un número máximo de ellos, llamado “q”). [13]

ARIMA se presenta como un modelo ARIMA (p, d, q).

Donde **p** representa el número de términos autorregresivos, **d** representa el número de diferencias no estacionales necesarias para estacionariedad, y **q** representa el número de pronósticos rezagados errores en la ecuación de predicción [14].

2.2.2. Facebook Prophet

Prophet es un software de código abierto lanzado por el equipo de Core Data Science de Facebook. Prophet es un procedimiento para pronosticar datos de series temporales basados en un modelo aditivo donde las tendencias no lineales se ajustan a la estacionalidad anual, semanal y diaria. Prophet es robusto ante los datos faltantes y cambios de tendencia, además maneja bien los valores atípicos.

Prophet es un modelo regresivo aditivo con cuatro componentes:

- Tendencia: detecta automáticamente cambios en la misma seleccionando los distintos quiebres de tendencia dentro del conjunto de datos y así arma la función (definida por partes) de tendencia lineal o de crecimiento logístico (que alcanza nivel de saturación) [13].
- Estacionalidad anual: la modela utilizando series de Fourier.
- Estacionalidad semanal: la modela con variables de tipo dummy.
- Fechas importantes, feriados, etc: el usuario las puede definir de antemano si significan un quiebre a tener en cuenta por el modelo [13].

Se combinan en la siguiente ecuación:

$$y(t) = g(t) + s(t) + h(t) + \epsilon_t$$

$g(t)$: curva de crecimiento lineal o logística por partes para modelar cambios no periódicos en series de tiempo.

$s(t)$: cambios periódicos (por ejemplo, estacionalidad semanal / anual)

$h(t)$: efectos de vacaciones (proporcionados por el usuario) con horarios irregulares.

ϵ_t: el término de error explica cualquier cambio inusual que el modelo no tenga en cuenta.

Prophet utiliza el tiempo como regresor tratando de ajustar varias funciones lineales y no lineales del tiempo como componentes.

El modelado de la estacionalidad como componente aditivo es el mismo enfoque adoptado por el suavizado exponencial en la técnica Holt-Winters. En efecto, se está enmarcando el problema de pronóstico como un ejercicio de ajuste de curvas en lugar de mirar explícitamente la dependencia basada en el tiempo de cada observación dentro de una serie de tiempo [13].

2.2.3. Máquinas de vectores de soporte SVM

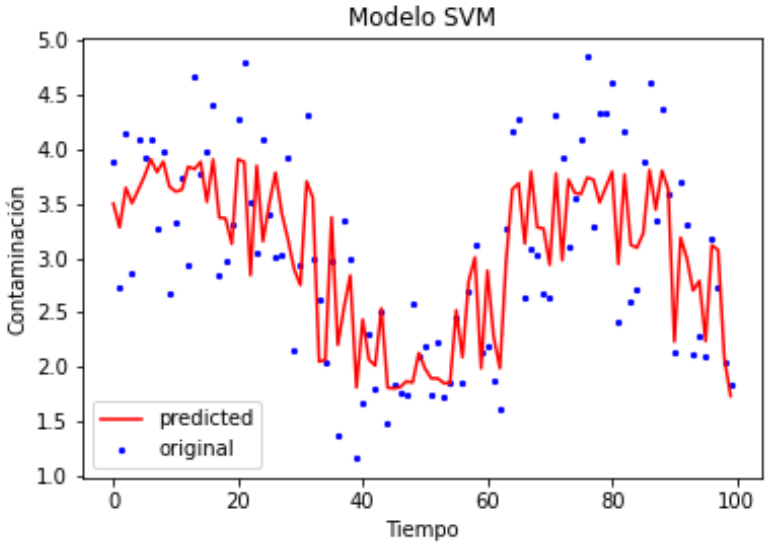
Máquinas de vectores de soporte SVM por sus siglas en inglés (Support Vector Machines), pueden ser usadas tanto para regresión (ver Figura 2) como para clasificación. Las máquinas de soporte vectorial son un conjunto de algoritmos de aprendizaje supervisado.

Las SVM construyen un hiperplano o conjuntos de hiperplanos en un espacio de dimensión muy alta.

Su funcionamiento consiste en correlacionar datos en un espacio de características de grandes dimensiones de forma que los puntos de datos se pueden categorizar, incluso si los datos no se pueden separar linealmente de otro modo. Se detecta un separador entre las categorías y los datos se transforman de forma que el separador se puede extraer como un hiperplano. Luego, las características de los

nuevos datos se pueden utilizar para predecir el grupo al que pertenece el nuevo registro [15].

Figura 2: Modelo SVM regresión



Fuente: elaboración propia

2.2.4. Redes Neuronales

Las redes neuronales son una de las técnicas heurísticas más utilizadas para abordar temas de predicción como por ejemplo la predicción de la calidad del aire, especialmente mediante la configuración de los modelos Perceptrón Multicapa (MLP- de sus siglas en inglés: Multi-Layer Perceptron) y Funciones de Base Radial (RBFN- de sus siglas en inglés: Radial Based Functions Network) [16].

Un perceptrón multicapa (MLP en sus siglas en inglés) es una clase de red neuronal artificial de alimentación directa (ANN de sus siglas en inglés: Artificial neural network). Un MLP consta de al menos tres capas de nodos: una capa de entrada, una capa oculta y una capa de salida. Excepto por los nodos de entrada, cada nodo es una neurona que usa una función de activación no lineal, ver Figura 3.

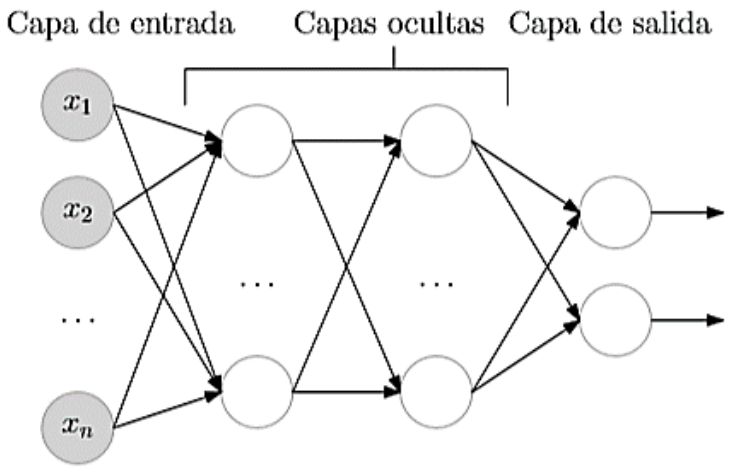
Las redes de base radial son un tipo de redes neuronales artificiales que calculan la salida de la función en función de la distancia a un punto denominado centro. Son redes de tipo multicapa que tienen conexiones hacia delante y que solo tienen una capa

oculta. Mientras que las neuronas ocultas poseen carácter local, las neuronas de salida realizan una combinación lineal de las activaciones de las neuronas ocultas.

En las redes neuronales algunos parámetros deben ser declarados al comienzo del proceso de entrenamiento como por ejemplo número de capa oculta y nodos ocultos, tasas de aprendizaje y función de activación.

Las unidades de procesamiento o neuronas de un ANN consisten en tres componentes principales: Los pesos sinápticos que conectan los nodos, la función de suma dentro del nodo y la función de transferencia Los pesos sinápticos son conocidos por su fuerza que corresponde a la importancia de la información que proviene de cada neurona. Lo que significa que la información está codificada en estos pesos de fuerza[17].

Figura 3: Explicación topológica de un perceptrón multicapa



Fuente: [18]

2.3. Análisis comparativo de los modelos

ARIMA es un método estadístico clásico utilizado para realizar el análisis de series de tiempo, pero en las últimas décadas las redes neuronales han sido métodos alternativos a los problemas de predicción con mayor índice de aceptación.

El modelo estadístico ARIMA y redes neuronales han sido de utilidad cuando la serie estudiada presenta características no lineales. Hoy en día las redes neuronales son preferidas para trabajar con series temporales, ya que, se las considera más

robustas. De acuerdo a los trabajos revisados al trabajar con redes neuronales es necesario realizar una buena configuración de la red con el objetivo de alcanzar óptimos resultados.

La ventaja de trabajar con modelos ARIMA es que son fáciles de interpretar, pero estos modelos son menos robustos que las redes neuronales y las máquinas de vectores de soporte SVM.

SVM tiene en cuenta el riesgo empírico, esto permite óptimo local, se deriva completamente sobre la base de varias técnicas matemáticas simples (la derivada parcial y los multiplicadores de Lagrange).

Facebook Prophet, ajusta los datos utilizando armónicos de funciones trigonométricas, Prophet no requiere mucho conocimiento previo o experiencia en pronosticar datos de series de tiempo ya que automáticamente encuentra tendencias estacionales debajo de los datos. En algunos casos dependiendo de los ajustes de parámetros se puede obtener mejores resultados al emplear modelos ARIMA en comparación a Facebook Prophet. Asimismo, en el caso de comparar el rendimiento de redes neuronales artificiales y máquinas de vectores de soporte, la diferencia de precisión se basa en el ajuste de parámetros de los diferentes modelos.

3. Diseño e implementación del trabajo

En el presente capítulo se describen, el proceso al que se someten los datos para implementar los modelos de predicción de calidad del aire de Londres y los resultados obtenidos en esta investigación.

3.1. Procesamiento de datos

3.1.1. Datos

Los datos han sido recuperados de London Average Air Quality Levels ,

- *air-quality-london-time-of-day.csv*
- *air-quality-london-monthly-averages.csv*

En estos conjuntos de datos se encuentra la lectura promedio en carretera y de fondo para dióxido de nitrógeno, óxido nítrico, óxidos de nitrógeno, ozono, material particulado (PM10 y PM2.5) y dióxido de azufre.

Los datos han sido almacenados en Google Drive para su posterior procesamiento por parte de la herramienta Google Colab que permite hacer uso del lenguaje de programación Python mediante el uso de Jupyter Notebooks.

Fichero: air-quality-london-time-of-day.csv

Tamaño: 3336 filas, 16 columnas

Detalle de los primeros registros del conjunto de datos (Figura 4).

Figura 4: Detalle de los 10 primeros registros dataset time-of-day

	Date	Time	RNO	RNO2	RNOx	RO3	RPM10	RPM2.5	RSO2	BNO	BNO2	BNOx	BO3	BPM10	BPM2.5	BSO2
0	Jan-08	00:00	NaN	42.3	NaN	32.6	23.0	15.5	3.4	NaN	34.2	NaN	41.8	19.2	.	3.1
1	Jan-08	01:00	NaN	33.8	NaN	35.3	21.2	13.5	2.6	NaN	29.0	NaN	45.3	18.9	.	3.1
2	Jan-08	02:00	NaN	28.8	NaN	43.2	19.6	12.7	2.1	NaN	25.4	NaN	46.6	17.7	.	3.3
3	Jan-08	03:00	NaN	27.3	NaN	42.4	18.3	11.9	2.0	NaN	23.4	NaN	46.6	16.4	.	3.3
4	Jan-08	04:00	NaN	29.4	NaN	40.1	18.1	12.0	2.2	NaN	24.2	NaN	45.2	16.0	.	2.8
5	Jan-08	05:00	NaN	38.3	NaN	33.8	18.7	12.2	2.2	NaN	29.4	NaN	40.3	15.5	.	2.4
6	Jan-08	06:00	NaN	53.7	NaN	26.7	21.5	14.5	3.4	NaN	39.6	NaN	34.1	16.7	.	2.8
7	Jan-08	07:00	NaN	66.2	NaN	22.4	23.4	15.0	4.2	NaN	49.3	NaN	29.3	17.1	.	3.0
8	Jan-08	08:00	NaN	69.5	NaN	21.5	25.3	14.9	4.7	NaN	52.9	NaN	28.1	17.7	.	3.5
9	Jan-08	09:00	NaN	67.1	NaN	23.7	26.6	14.8	5.2	NaN	50.5	NaN	31.4	19.1	.	3.8

Fuente: elaboración propia

A simple vista se observa que, en el conjunto de datos existen valores nulos, los cuales serán tratados posteriormente.

Estadistica de los registros del fichero time-of-day (Figura 5)

Figura 5: Estadística del dataset time of day

```
(data.describe())
```

	RNO	RNO2	RNOx	RO3	RPM10	RPM2.5	RSO2	BNO	BNO2	BNOx	BO3	BPM10	BSO2
count	2760.000000	3336.000000	2760.000000	3336.000000	3336.000000	3336.000000	3336.000000	2760.000000	3336.000000	2760.000000	3336.000000	3336.000000	3336.000000
mean	75.619275	55.206894	136.866884	27.315288	25.007464	15.608124	3.408903	21.402971	34.538489	55.532283	37.381115	19.246972	3.304107
std	41.557217	13.571274	52.126009	10.495312	6.111342	5.206202	1.956399	17.098273	10.890930	26.464547	15.148828	4.998690	1.118884
min	6.300000	21.000000	31.700000	6.400000	11.800000	5.900000	-2.400000	1.400000	11.500000	13.500000	8.700000	10.200000	0.700000
25%	43.875000	46.100000	97.975000	19.500000	20.700000	11.900000	2.275000	9.100000	25.900000	35.900000	26.000000	15.800000	2.500000
50%	69.150000	55.400000	133.550000	26.100000	24.100000	14.300000	3.200000	16.000000	32.800000	48.000000	35.100000	18.100000	3.300000
75%	101.425000	64.600000	169.200000	33.900000	28.300000	18.100000	4.400000	28.225000	41.700000	69.225000	46.825000	21.500000	4.000000
max	257.300000	95.600000	336.300000	66.300000	52.100000	36.500000	14.000000	120.400000	74.300000	193.700000	96.100000	43.400000	9.000000

Fuente: elaboración propia

Fichero: air-quality-london-monthly-averages.csv

Tamaño: 139 filas, 14 columnas

Detalle de los primeros registros del conjunto de datos (Figura 6).

Figura 6: Detalle de los 10 primeros registros dataset monthly-averages

Month	RNO	RNO2	RNOx	RO3	RPM10	RPM2.5	RSO2	BNO	BNO2	BNOx	BO3	BPM10	BPM2.5	BSO2
Jan-08	NaN	55.5	NaN	29.5	25.0	14.7	4.2	NaN	42.3	NaN	36.9	18.8	.	3.6
Feb-08	NaN	75.9	NaN	20.3	39.5	28.8	7.6	NaN	60.2	NaN	26.4	31.9	.	6.7
Mar-08	NaN	55.6	NaN	40.1	21.6	12.3	3.9	NaN	39.8	NaN	50.2	15.5	.	2.3
Apr-08	NaN	61.8	NaN	37.9	28.7	20.5	4.5	NaN	44.0	NaN	50.1	21.7	.	3.2
May-08	NaN	62.9	NaN	46.3	34.6	27.5	4.6	NaN	44.1	NaN	60.5	29.5	16.6	4.3
Jun-08	NaN	49.2	NaN	39.8	23.2	16.0	3.6	NaN	31.2	NaN	51.3	18.3	12.6	2.5
Jul-08	NaN	48.4	NaN	35.0	23.0	14.2	3.1	NaN	31.2	NaN	46.6	17.2	11.9	2.5
Aug-08	NaN	41.1	NaN	30.0	20.7	11.5	2.2	NaN	27.9	NaN	37.1	15.5	11.2	2.1
Sep-08	NaN	54.1	NaN	22.4	28.2	18.0	3.7	NaN	41.2	NaN	28.9	22.2	15.3	3.1
Oct-08	NaN	56.7	NaN	19.3	23.0	12.9	4.3	NaN	43.8	NaN	25.4	16.5	11.6	2.8

Fuente: elaboración propia

Estadística de los registros del fichero monthly-averages.csv (Figura 7)

Figura 7: Estadística del dataset monthly-averages

	RNO	RNO2	RNOx	RO3	RPM10	RPM2.5	RSO2	BNO	BNO2	BNOx	BO3	BPM10	BSO2
count	115.000000	139.000000	115.000000	139.000000	139.000000	139.000000	139.000000	115.000000	139.000000	115.000000	139.000000	139.000000	139.000000
mean	75.619130	55.210072	136.865217	27.314388	25.011511	15.603597	3.410072	21.397391	34.534532	55.563478	37.379137	19.241007	3.300000
std	29.784035	8.264656	33.090965	8.333237	5.210289	4.910583	1.819870	14.258906	8.611677	22.039073	11.514537	4.705236	1.005852
min	22.000000	34.900000	68.600000	10.700000	16.300000	7.900000	-1.700000	4.200000	18.400000	24.400000	13.900000	11.900000	1.100000
25%	56.150000	48.700000	114.500000	21.150000	21.450000	12.350000	2.400000	11.700000	27.800000	38.200000	29.100000	16.100000	2.650000
50%	70.600000	55.500000	129.300000	26.400000	23.800000	14.200000	3.300000	17.200000	33.700000	51.400000	36.600000	18.100000	3.200000
75%	98.800000	60.300000	159.850000	34.150000	27.900000	18.150000	4.100000	29.400000	40.850000	69.000000	46.550000	21.500000	4.000000
max	180.900000	75.900000	250.700000	46.300000	43.300000	32.600000	12.400000	79.200000	60.200000	129.200000	62.600000	36.900000	6.700000

Fuente: elaboración propia

3.1.2. Selección y tratamiento de datos

En este trabajo, los datos a tratar para realizar las predicciones son las mediciones horarias de cada elemento, es decir el conjunto de datos “air-quality-london-time-of-day.csv”.

El conjunto de datos ha sido procesado en el siguiente orden:

- Se ha unificado los campos fecha y hora en un nuevo campo con formato YY-MM-DD HH:MM: SS.
- En los valores de la concentración de PM2.5 de fondo se observa la presencia de campos con el signo “.”, por ello se considera como nulo.
- Uno de los principales problemas con los conjuntos de datos es el alto número de observaciones faltantes en los registros, en este caso se procede a eliminar las variables con un porcentaje alto de valores vacíos, en la Figura 8 se presenta la estadística de valores perdidos por variable.

Figura 8: Estadística valores perdidos

```
Date          0.000000
Time          0.000000
RNO          17.266187
RNO2          0.000000
RNOx         17.266187
RO3           0.000000
RPM10         0.000000
RPM2.5        0.000000
RSO2          0.000000
BNO          17.266187
BNO2          0.000000
BNOx         17.266187
BO3           0.000000
BPM10         0.000000
BPM2.5        2.877698
BSO2          0.000000
Datetime      0.000000
dtype: float64
```

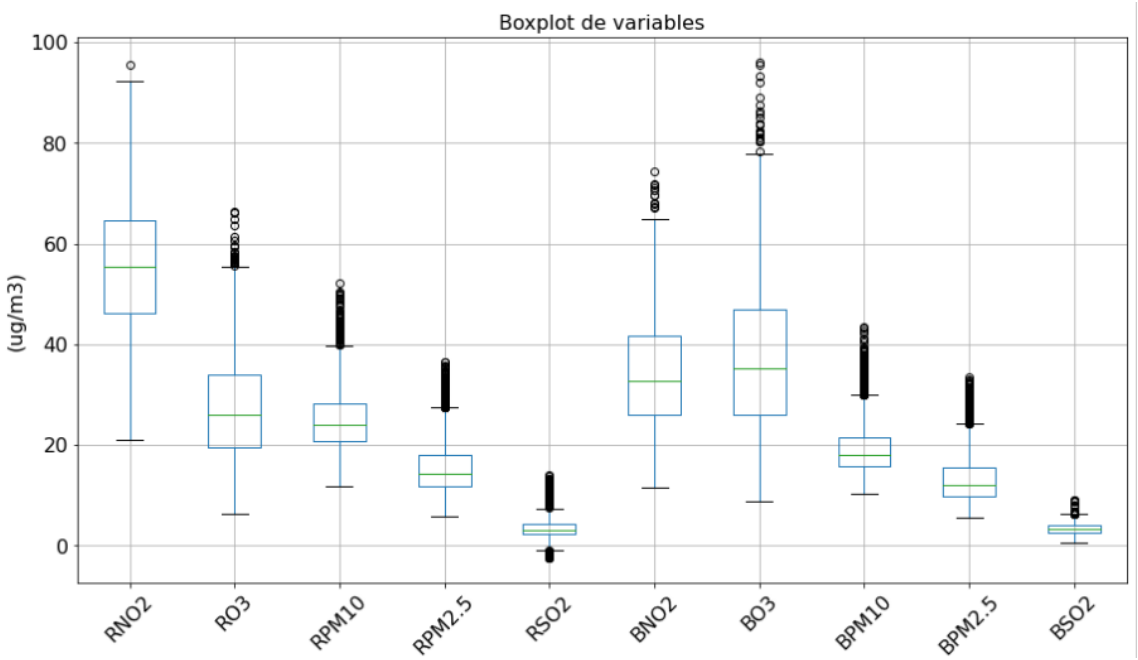
Fuente: elaboración propia

Para la eliminación de variables con porcentaje significativo de valores perdidos se establece el criterio de eliminar las variables que contengan más de 10% de valores NaN, por tanto se eliminan las variables 'R.NO', 'R.NOx', 'B.NO', y 'B.NOx'.

3.1.3. Análisis de los datos

En primer lugar, se presenta un boxplot de cada variable, con el objetivo de verificar la presencia de valores atípicos, las variables presentan valores atípicos que no se pueden considerar erróneos ya que estos pueden ser niveles muy altos de contaminación (ver representación en la Figura 9), por tanto, estos valores serán conservados.

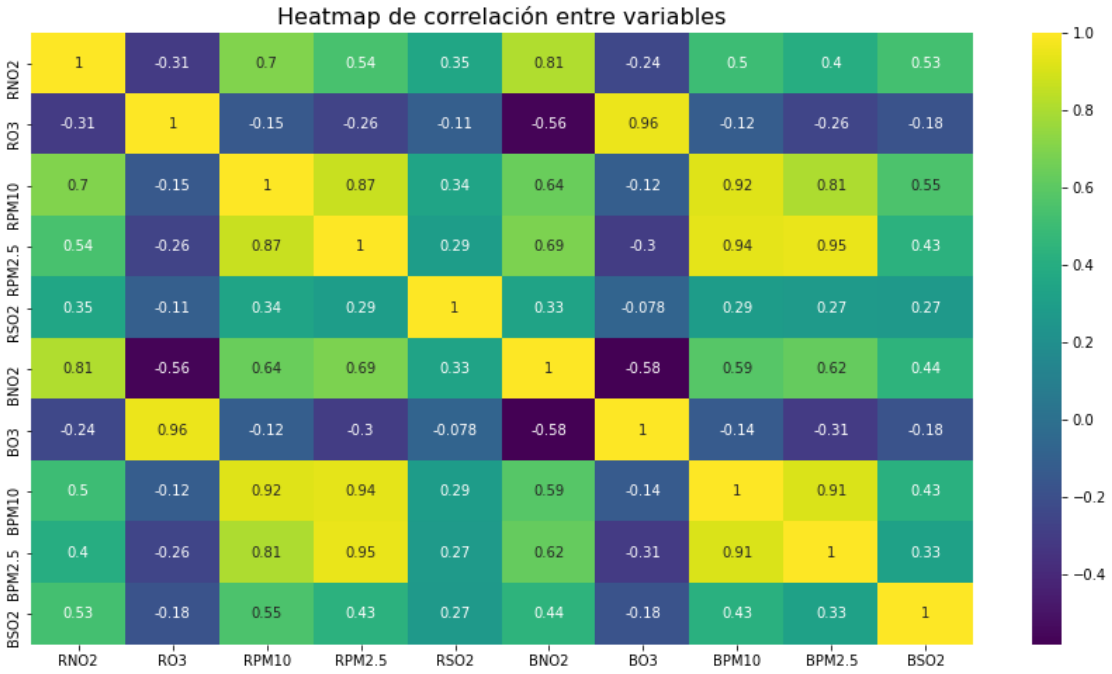
Figura 9: Boxplot de variables



Fuente: elaboración propia

A partir del análisis de valores atípicos, se realiza el análisis de correlación entre datos. En la Figura 10 se muestra la matriz de correlación obtenida por el método de Pearson.

Figura 10: Correlación datos



Fuente: elaboración propia

La correlación entre variables aporta información sobre la linealidad y la proporcionalidad existente entre variables.

Se observa que algunas variables presentan multicolinealidad, la variable RO3 (ozono) es la que menos correlación presenta con los datos, en la Figura 11 se aprecia que los coeficientes de correlación asociados al O3 son todos negativos considerando que el O3 troposférico es, por tanto, un contaminante más complejo de estimar que los contaminantes primarios como el dióxido de azufre (SO2), por ende, la presente investigación se centra en obtener las predicciones del RO3 (ozono en carretera).

Figura 11: Correlación con el RO3

```
BNO2     -0.563911
RNO2     -0.308505
RPM2.5   -0.263575
BPM2.5   -0.259265
BSO2     -0.176640
RPM10    -0.148450
BPM10    -0.122588
RSO2     -0.108712
BO3       0.955920
RO3       1.000000
Name: RO3, dtype: float64
```

Fuente: elaboración propia

3.1.4. Métricas de evaluación

Para medir la calidad del modelo y evaluar la eficiencia de este se utilizará como medidas de evaluación la Raíz del Error Cuadrático Medio RMSE (siglas en inglés Root Mean Squared Error), el Error Absoluto Medio MAE (siglas en inglés Mean Absolute Error), el Error Cuadrático Medio MSE (siglas en inglés Mean Squared Error) y el Error porcentual absoluto medio MAPE (siglas en inglés Mean absolute Percentage Error).

Para todas las fórmulas subsiguientes se considera la notación y_j como la serie original y la notación $\hat{y}_j$como la serie estimada.

La métrica **RMSE** es una de las más habituales para evaluar un modelo de regresión, ya que mide la cantidad de error que hay entre dos conjuntos de datos, en el caso del modelo de predicción proporciona la diferencia entre el valor pronosticado por el modelo y el valor real.

Fórmula RMSE:

$$RMSE = \sqrt{\frac{1}{n}\sum_{j=1}^{n}\left(y_j - \hat{y}_j\right)^2}$$ [19].

La métrica **MAE**, es la media de los errores absolutos, el error absoluto es el valor absoluto de la diferencia entre el valor pronosticado y el valor real, el valor MAE indica qué tan grande es el error que se puede esperar del pronóstico en promedio.

Formula MAE:

$$MAE = \frac{1}{n}\Sigma|y_j - \hat{y}_j| \text{ [20].}$$

La métrica **MSE**, calcula el error cuadrático medio entre el pronóstico y el valor real

Formula MSE:

$$MSE = \frac{1}{n}\sum_{j=1}^{n}(y_j - \hat{y}_j)^2 \text{ [21].}$$

La métrica **MAPE,** es el promedio de los errores porcentuales absolutos de las predicciones. Cuanto más pequeño es el MAPE, mejores son las predicciones.

Formula MAPE:

$$MAPE = \frac{1}{n}\sum_{j=1}^{n}\frac{|y_j - \hat{y}_j|}{|y_j|} \text{ [21].}$$

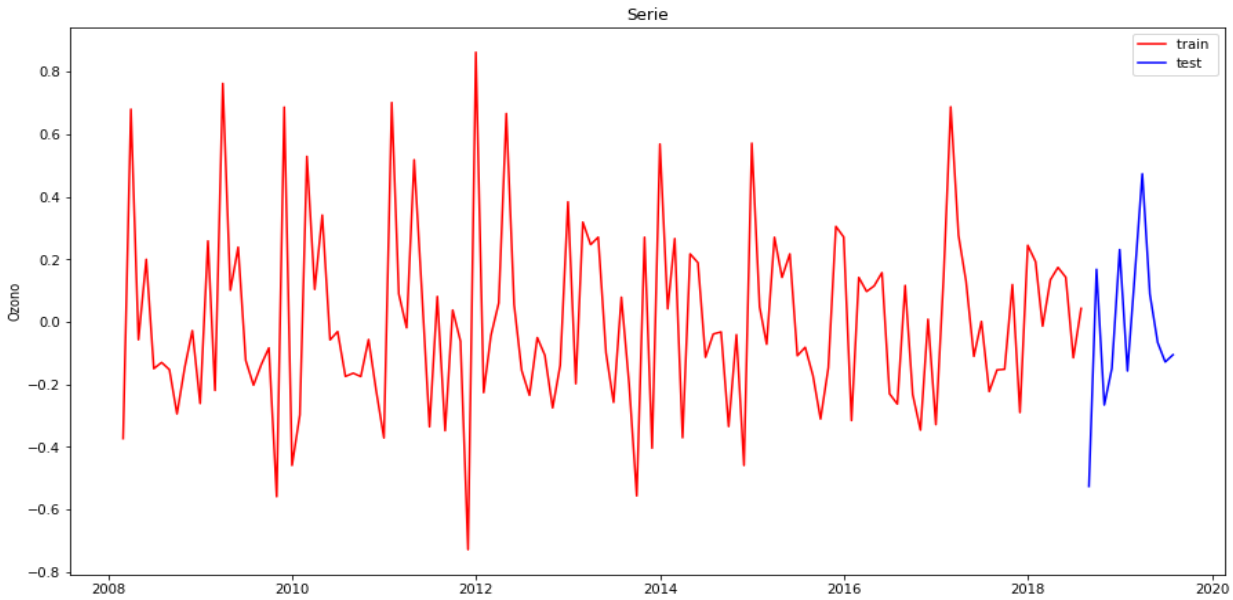
3.1.5. Selección de los datos

Para la generación de modelos predictivos es necesario tener dos conjuntos de datos diferenciados en este caso se define el conjunto train (datos de entrenamiento) empleado para generar el modelo, y el conjunto test empleado para validar la calidad del modelo.

Para este trabajo, los datos para generar el modelo serán los correspondientes a los años (2008-2018) y los empleados para estimar la calidad del modelo serán los

últimos 12 meses que corresponden de agosto 2018 a julio 2019. Los últimos 12 meses corresponde al 8 % del total de la muestra, lo cual es suficiente para realizar la evaluación de la capacidad de predicción del modelo para el último año. En la Figura 12 se presenta la distribución de los datos, conjunto train color rojo y conjunto test color azul.

Figura 12: Concentraciones Ozono Train-Test



Fuente: elaboración propia

3.2. Aplicación de los modelos

En este apartado se presenta un resumen del proceso realizado para generar los diferentes modelos, en donde se detalla el modelo empleado, datos e hiperparámetros. Los modelos se han generado en un jupyter notebook el cual es accesible por medio de la URL (ver ANEXO 1 en donde se encuentra la dirección del repositorio donde se alojan los ficheros y notebooks).

3.2.1. Modelo ARIMA

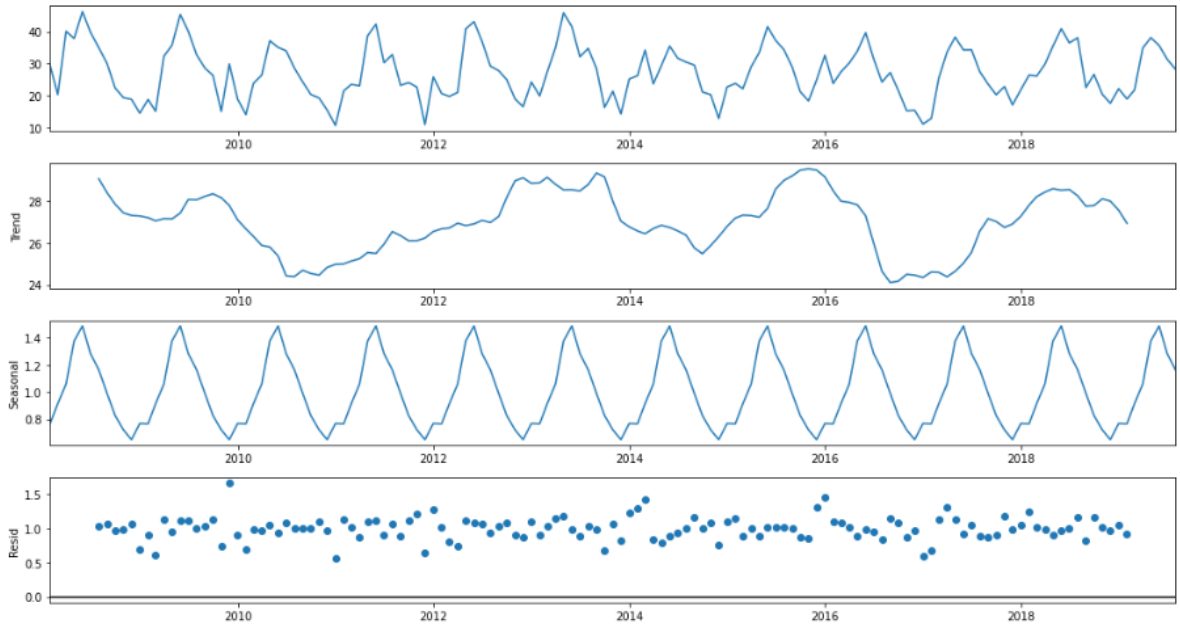
Para encontrar patrones en variaciones y regresiones estadísticas, se utiliza el modelo ARIMA (por sus siglas en inglés autoregressive integrated moving average).

Para la implementación del modelo ARIMA se deben tener ciertas consideraciones como la estacionariedad de los datos.

El conjunto de datos está formado por las concentraciones horarias de un día por mes, en este caso se procede a realizar un remuestreo, para obtener el promedio mensual de concentración de O3.

A continuación, se estudia la serie temporal, para lo cual es necesario realizar una descomposición estacional como se aprecia en la Figura 13.

Figura 13: Descomposición estacional de la serie de tiempo

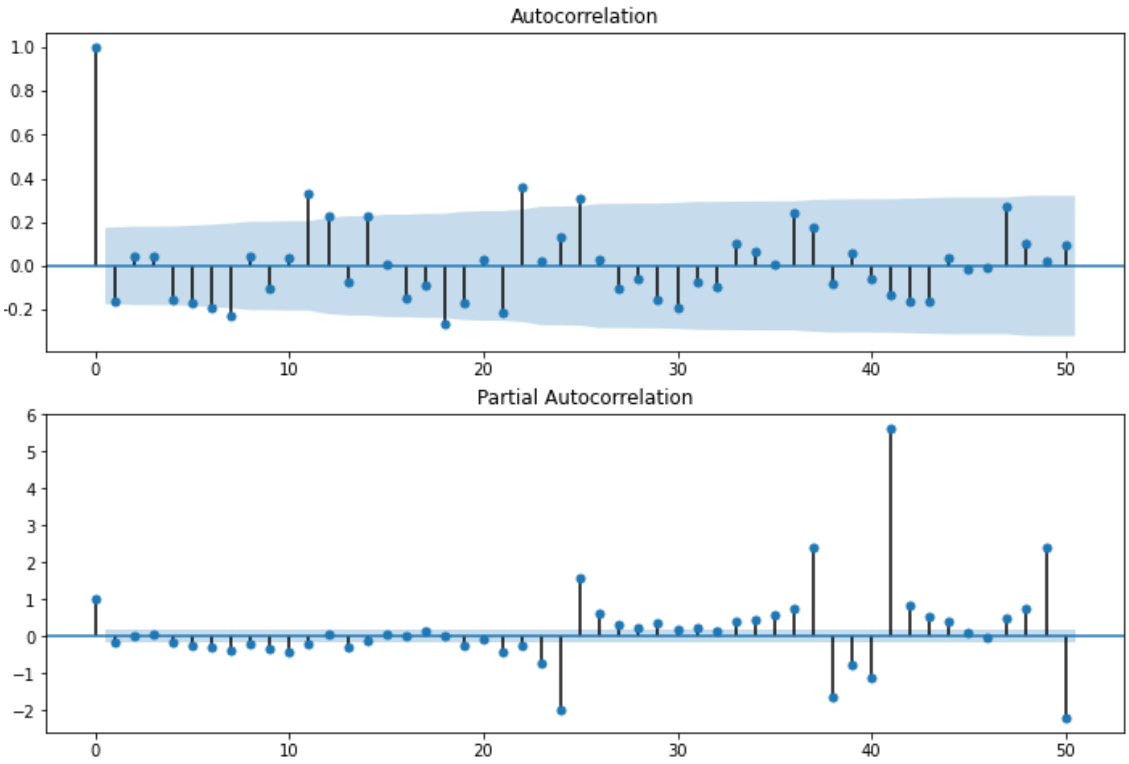


Fuente: elaboración propia

La serie temporal no es estacionaria, por tanto, es necesario transformar la serie a estacionaria. En este caso se realiza la transformación de datos aplicando una función logarítmica, luego se procede a realizar la diferencia de la serie.

Una vez que la serie es estacionaria, se obtiene las funciones de autocorrelación simple y parcial muéstrales (ver Figura 14), para determinar el proceso ARIMA(p,d,q) más adecuado.

Figura 14: Autocorrelación simple y parcial serie temporal



Fuente: elaboración propia

Aunque es posible determinar los parámetros de entrenamiento a partir de las gráficas de autocorrelación, se realizar un proceso automático para encontrar el mejor hiperparámetro utilizando “Auto Arima” de la biblioteca pmdarima. El resultado de este procedimiento señala que es necesario usar la extensión SARIMAX (acrónimo en inglés de Seasonal Autoregressive Integrated Moving Average), La implementación se llama SARIMAX en lugar de SARIMA porque la adición "X" al nombre del método significa que la implementación también admite variables exógenas.

SARIMA es una extensión del modelo ARIMA, la diferencia es que SARIMA admite series temporales con componentes estacionales o con tendencia. Para hacer esto posible, agrega 4 nuevos hiperparámetros (sp, sd, sq, s), tres de ellos son similares a los hiperparámetros ARIMA (p, d, q) pero implican retrocesos del período estacional. El último hiperparámetro agregado 's' es el período de estacionalidad, donde 12 significa datos mensuales, 4 datos trimestrales, 0 sin datos estacionales [22].

La mejor configuración encontrada para los hiperparámetros fue:

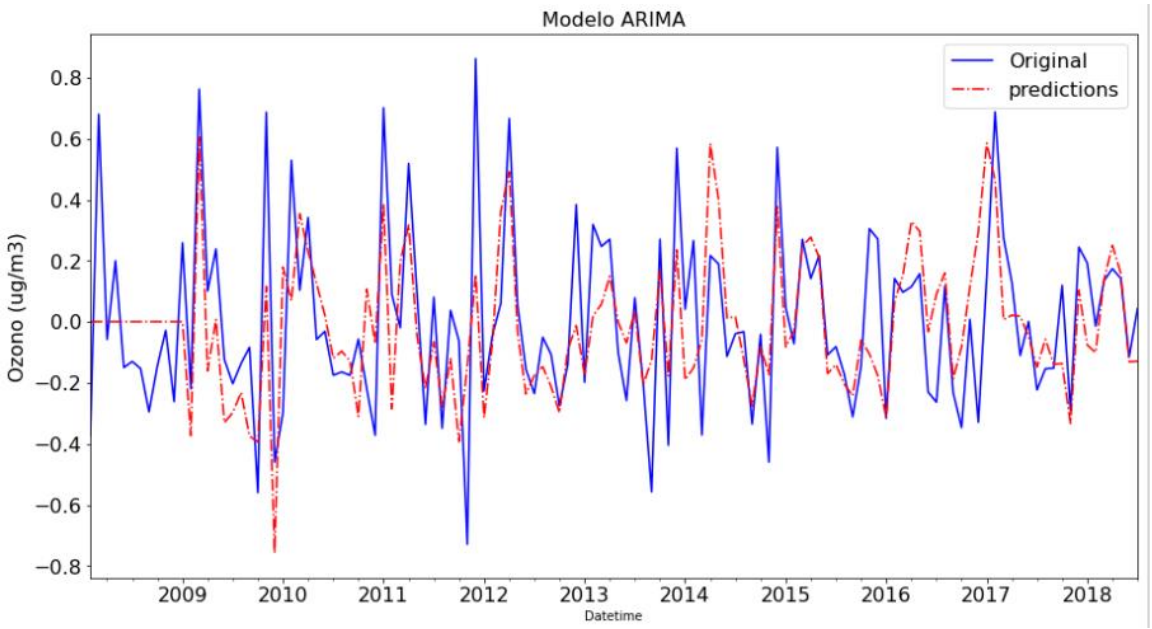
(p = 0, d = 0, q = 1), (sp = 0, sd = 1, sq = 1, s = 12)

Una vez que se define el modelo con la función SARIMAX (from statsmodels.tsa.statespace.sarimax import SARIMAX) el modelo se ajusta llamando a la función fit (), ajustar el modelo devuelve una instancia de la clase SARIMAXResults. Este objeto contiene los detalles del ajuste, como los datos y los coeficientes, así como las funciones que se pueden utilizar para hacer uso del modelo.

Predicción

Luego de entrenar el modelo puede usarse para hacer un pronóstico de la variable objeto de estudio, se representa el resultado de la predicción junto con los datos de la concentración de ozono (O3), En la Figura 15 se presenta la representación gráfica de la predicción, las concentraciones reales se presentan en color azul y las predichas en color rojo, los datos predichos siguen la tendencia de los valores reales a partir del año 2009.

Figura 15: Representación modelo ARIMA, datos reales y predicción

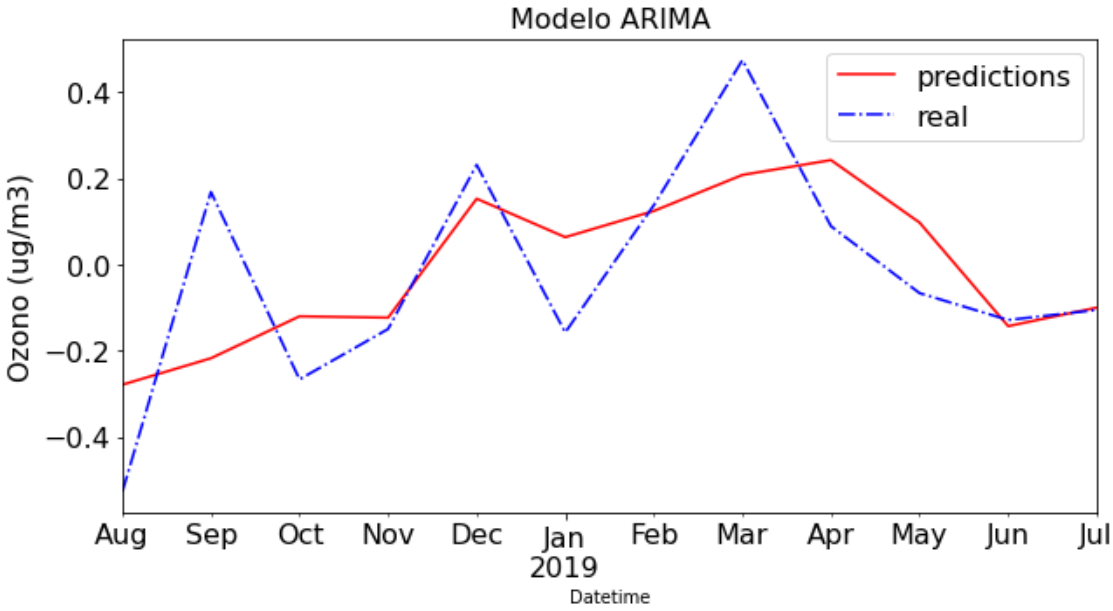


Fuente: elaboración propia

Resultados

Para comprobar la precisión del modelo predictivo, se predice los datos de test (últimos 12 meses), los resultados de la predicción, se aprecian en la Figura 16, los datos de test se muestran de color azul y la predicción de color rojo.

Figura 16: Gráfica datos test junto con la predicción



Fuente: elaboración propia

Como puede verse, las concentraciones mensuales de O3 correspondiente al año (agosto 2018- julio 2019) tienen las siguientes características:

- Presenta un valor pico alrededor del mes de marzo con un valor de 0.4.
- La concentración de O3 oscila entre -0.2 y 0.2.

La predicción del modelo frente a los valores de test tiene un comportamiento muy parecido, a pesar de que el modelo generado no ha sido capaz de reproducir los picos alrededor del mes de septiembre y el pico del mes de marzo al tratarse de un valor atípico. El modelo ha sabido predecir razonablemente el pico en la concentración alrededor del mes de diciembre.

Después de entrenar y probar el modelo las métricas obtenidas son las siguientes:

- error absoluto medio (MAE) 0.143
- error cuadrático medio (MSE) 0.033
- raíz del error cuadrático medio (RMSE) 0.184.

3.2.2. Modelo Máquinas de soporte vectorial (SVM)

Las máquinas de soporte vectorial son un algoritmo que puede ser utilizado para predicción y clasificación. Con la finalidad de obtener un algoritmo óptimo para la implementación de este modelo es necesario la optimización de una serie de parámetros uno de ellos aparece en el mapeo no lineal en el espacio de características, llamados funciones Kernel, en la tabla 3 se presenta la formulación matemática del Kernel, para la implementación de dichas funciones es necesario la instalación de la librería "sklearnkernels", para luego importar la librería KSVR, la cual permite modificar una serie de parámetros, en este caso se va a modificar la función Kernel, C y gamma.

Tabla 3. Formulación matemática del kernel

Kernel	Definición
RBF	$\kappa^{RBF}(\chi,\chi') = e^{-\sum_{i=1}^{d}\gamma(\chi_i-\chi_{i'})^{\beta}}$ $\gamma > 0, \beta \in (0,2]$
Triangular	$\kappa^{tri}(\chi,\chi') = \begin{cases} \|x-x'\| \le a \rightarrow 1-\dfrac{\|x-x'\|}{a} \\ \|x-x'\| > a \rightarrow 0 \\ a > 0 \end{cases}$
Truncated Euclidean	$\kappa^{tru}(\chi,\chi') = \dfrac{1}{d}\sum_{1}^{j}\max\left(0,\dfrac{\lvert x_i - x_i'\rvert}{y}\right)$ $\gamma > 0$

Fuente: [23]

Para la selección de parámetros existen diversas metodologías entre ellas la búsqueda en cuadricula (grid search), que consiste en la búsqueda exhaustiva de un subconjunto especificado manualmente de parámetros.

Los parámetros considerados para la búsqueda son los siguientes:

- **C**: Parámetro de regularización. controla la compensación entre la maximización de margen y la penalización del error.
- **Gamma(γ):** Coeficiente de Kernel.

Para la configuración de los hiperparámetros se realiza una búsqueda de los mejores parámetros mediante el uso de la clase GridSearchCV disponible en scikit-learn, en donde se especifica los siguientes parámetros: función Kernel, valores gamma, C, dicha clase permite evaluar el rendimiento mediante validación cruzada.

Validación cruzada, es una técnica con la que se puede identificar la existencia de problemas durante el entrenamiento, por ejemplo, el sobreajuste.

Los parámetros a modificar son las funciones Kernel, C y gamma, el resto de hiperparámetros tomaran el valor por defecto.

Resultados búsqueda parámetros

A continuación, se indica los resultados de la búsqueda de parámetros realizada por la clase GridSearchCV por función Kernel.

Parámetros Kernel triangular

```
{'C': 1.0, 'gamma': 0.01, 'kernel': 'triangle'}

KSVR(C=1.0, cache_size=200, coef0=0.0, degree=2, epsilon=0.1, gamma=0.01,
     kernel=<function triangle_kernel.<locals>.tk at 0x7fa4f38989d8>,
     max_iter=-1, shrinking=True, tol=0.001, verbose=False)

clflr -0.39334956225405526
```

Parámetros Kernel Truncated

```
{'C': 1.0, 'gamma': 0.01, 'kernel': 'tru'}

KSVR(C=1.0, cache_size=200, coef0=0.0, degree=2, epsilon=0.1, gamma=0.01,
     kernel=<function truncated_kernel.<locals>.trk at 0x7fa4f2ed28c8>,
     max_iter=-1, shrinking=True, tol=0.001, verbose=False)

clflr -0.39334956225405526
```

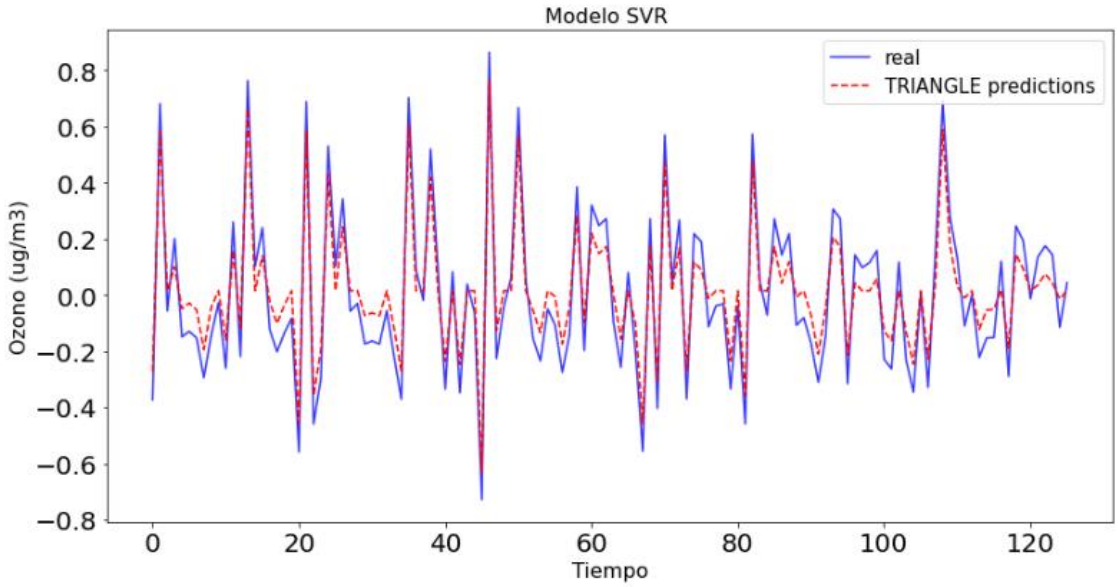
Parámetros Kernel RBF

```
{'C': 1.0, 'gamma': 1.0, 'kernel': 'rbf'}

KSVR(C=1.0, cache_size=200, coef0=0.0, degree=2, epsilon=0.1, gamma=1.0,
     kernel='rbf', max_iter=-1, shrinking=True, tol=0.001, verbose=False)

clflr -0.4224214921414295
```

Predicción.

Una vez obtenidos los resultados de optimización de parámetros se realizó el entrenamiento del modelo, en este caso ha sido entrenado un modelo por función Kernel. La Figura 17 muestra los resultados del modelo SVR con Kernel triangular

Figura 17: Resultado modelo SVR kernel triangular

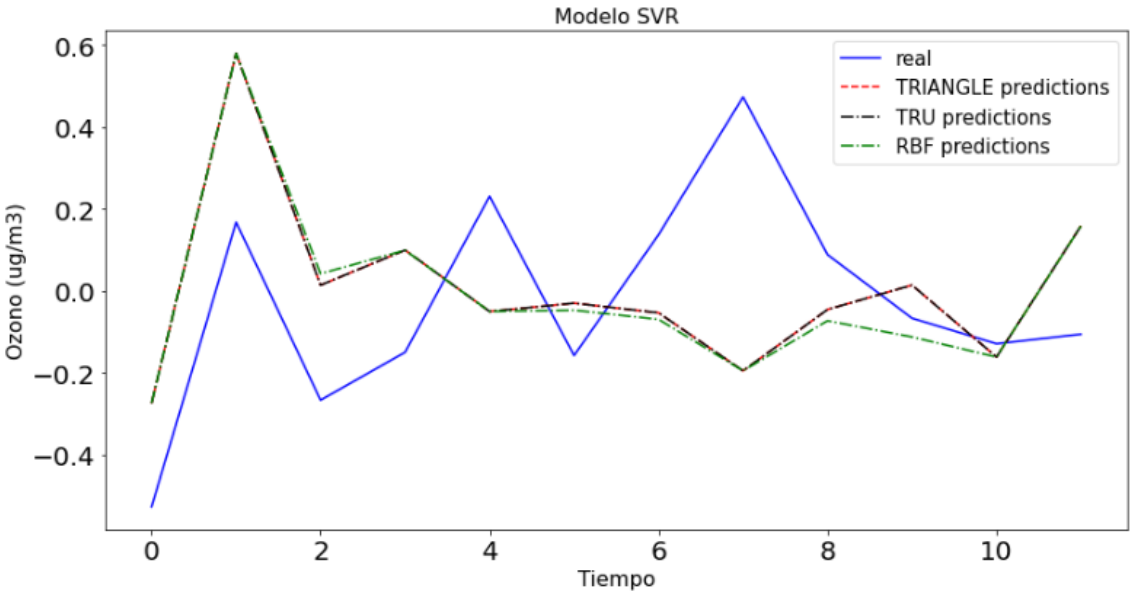


Fuente: elaboración propia

Resultados:

Para comprobar la precisión de la predicción, se realizó la predicción del conjunto test (últimos 12 meses), el resultado se presenta en la Figura 18, en donde se visualiza la predicción de los modelos generados con diferente función Kernel.

Figura 18: Resultados Predicción SVR



Fuente: elaboración propia

Los resultados obtenidos por los kernels Truncated y triangular son idénticos, como se observa en la Figura 18 existe solapamiento entre las predicciones, los tres modelos tienen comportamientos muy parecidos dado el grado de solapamiento entre los tres conjuntos predichos.

Las predicciones de los tres modelos son buenas, ya que han sido capaces de reproducir el primer pico de las mediciones reales, aunque el valor predicho se aleja del real. En la tabla 4 se encuentran los diferentes valores de las métricas calculadas en cada uno de los modelos SVR de acuerdo a la función Kernel empleada.

Tabla 4. Resultados modelos SVR con diferente función Kernel

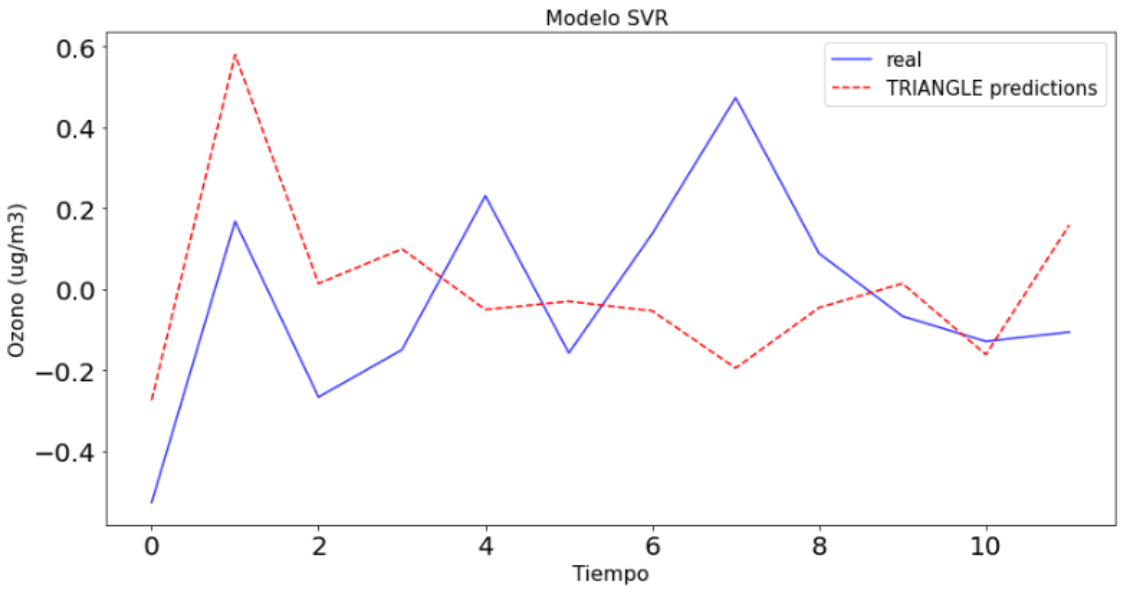
Kernel	MAE	MSE	RMSE
Triangular	0.247789	0.0873260	0.295509
Truncated	0.247789	0.0873260	0.295509
RBF	0.249328	0.0891480	0.298576

Fuente: elaboración propia

De estos resultados cabe señalar que todos los modelos muestran un valor MAE y RMSE similar. Se observa que en las métricas de calidad los kernels Triangular y Truncated presentan mejor resultado.

En la Figura 19 se muestran los valores predichos empleando el modelo con función Kernel Triangular y, como puede verse, los valores predichos están por encima de los valores reales, aparentemente se obtiene una predicción razonable.

Figura 19: Representación mejor modelo SVR



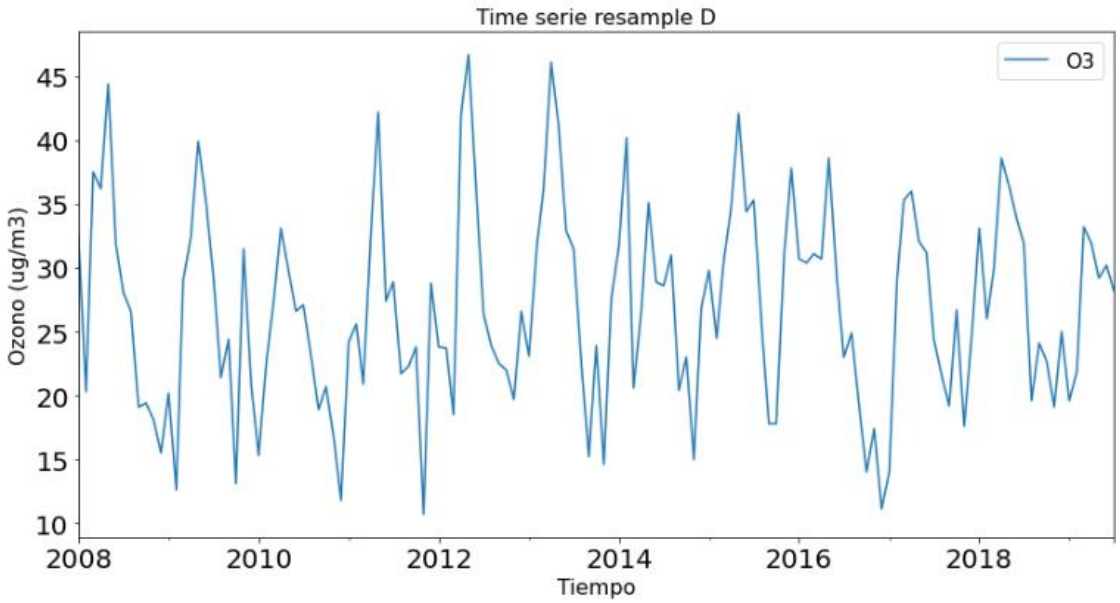
Fuente: elaboración propia

3.2.3. Modelo Facebook Prophet

Para pronosticar la tendencia futura de la serie de tiempo es necesario que los datos se encuentren muestreados por día, por tanto, para la implementación de este algoritmo es necesario aumentar la frecuencia de muestreo de datos. Para ello, se realizó la interpolación de valores faltantes mediante el uso de la función “interpolate()” de la librería pandas para objetos serie.

En la Figura 20 se aprecia la representación gráfica de la serie temporal resultante de la interpolación de datos.

Figura 20: Representación serie de tiempo para el modelo Prophet



Fuente: elaboración propia

Para entrenar el modelo Facebook Prophet, es necesario que las variables sean nombradas de la siguiente manera:

- y – Target (mediciones O3)
- ds – Datetime (Fecha)

A continuación, en la Figura 20, el detalle de los 10 primeros registros del conjunto de datos, preparado para la implementación del modelo Facebook Prophet.

Figura 21: Representación 10 primeros registros conjunto de datos Facebook Prophet

	ds	y
0	2008-01-01	32.600000
1	2008-01-02	32.203226
2	2008-01-03	31.806452
3	2008-01-04	31.409677
4	2008-01-05	31.012903
5	2008-01-06	30.616129
6	2008-01-07	30.219355
7	2008-01-08	29.822581
8	2008-01-09	29.425806
9	2008-01-10	29.029032

Fuente: elaboración propia

El algoritmo de Facebook Prophet permite añadir días festivos con el objetivo de estudiar el impacto de estos en la predicción, por tal razón se realizó la predicción en dos escenarios, con días festivos y sin días festivos.

Para la selección de hiperparámetros se hace uso de la clase GridSearch, en este caso al tener un mayor número de parámetros para entrenar el modelo, esta tarea se la realizó en un nuevo notebook "TFM_Optimización_de_parámetros_modelo_Facebook_Prophet" (ver Anexo 1) como métrica de evaluación se calcula el Porcentaje de Error Absoluto Medio (**MAPE**).

Los parámetros definidos en la búsqueda son los siguientes:

- **seasonality_mode:** Este parámetro indica cómo deberían integrarse sus componentes de estacionalidad en los pronósticos
- **changepoint_prior_scale:** indica la flexibilidad que pueden tener los puntos de cambio.
- **n_changepoints**: número de puntos de cambio, es decir cuántos puntos de cambio pueden ajustarse a los datos.

Modelo con días festivos

Para añadir los días festivos se importó la librería holidays, para importar los días festivos de la ciudad de Londres, en la Figura 22 se presenta los días festivos considerados para este caso de estudio.

Figura 22: Días festivos Londres

```
holidays_df.holiday.unique()
```

```
array(["New Year's Day", 'New Year Holiday [Scotland]',
       "St. Patrick's Day [Northern Ireland]", 'Good Friday',
       'Easter Monday [England, Wales, Northern Ireland]', 'May Day',
       'Spring Bank Holiday', 'Battle of the Boyne [Northern Ireland]',
       'Summer Bank Holiday [Scotland]',
       'Late Summer Bank Holiday [England, Wales, Northern Ireland]',
       "St. Andrew's Day [Scotland]", 'Christmas Day', 'Boxing Day',
       'Wedding of William and Catherine',
       "New Year Holiday [Scotland], New Year's Day",
       'Diamond Jubilee of Elizabeth II'], dtype=object)
```

Fuente: elaboración propia

Como resultado de la optimización de hiperparámetros, se selecciona los parámetros con menor valor MAPE, los cuales son resultado de la búsqueda de hiperparámetros definida en la clase GridSerach.

```
{'changepoint_prior_scale': 0.05,
 'n_changepoints': 200,
 'seasonality_mode': 'additive'}
```

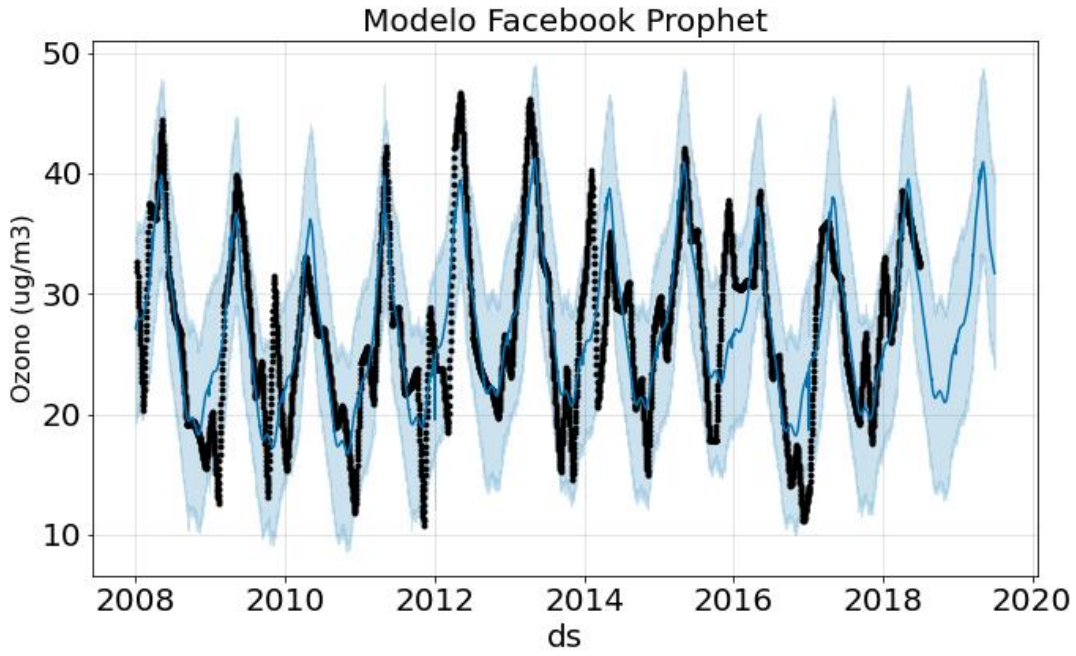
Los mejores parámetros para entrenar el modelo señalan que el modo de estacionalidad es ‘multiplicative’ lo que significa que la estacionalidad es multiplicativa, esto quiere decir que la estacionalidad no es constante y crece con la tendencia, como se observó en la Figura 20. Con seasonality_mode = ' multiplicative ', los efectos de días festivos también se modelarán como multiplicativos.

Predicción

Una vez definidos los parámetros se entrena el modelo final y se procede con la predicción de los próximos 12 meses (365 días), para lo cual se define el conjunto de

datos de base para el forecast con la función, incluida en el paquete, llamada m.make_future_dataframe(periods=365). Y finalmente, usando la función genérica predict se calcula la predicción, con el comando m.plot(forecast) se puede observar la representacion grafica resultado de la predicción (ver Figura 23)

Figura 23: Representación predicción modelo Facebook Prophet con días festivos

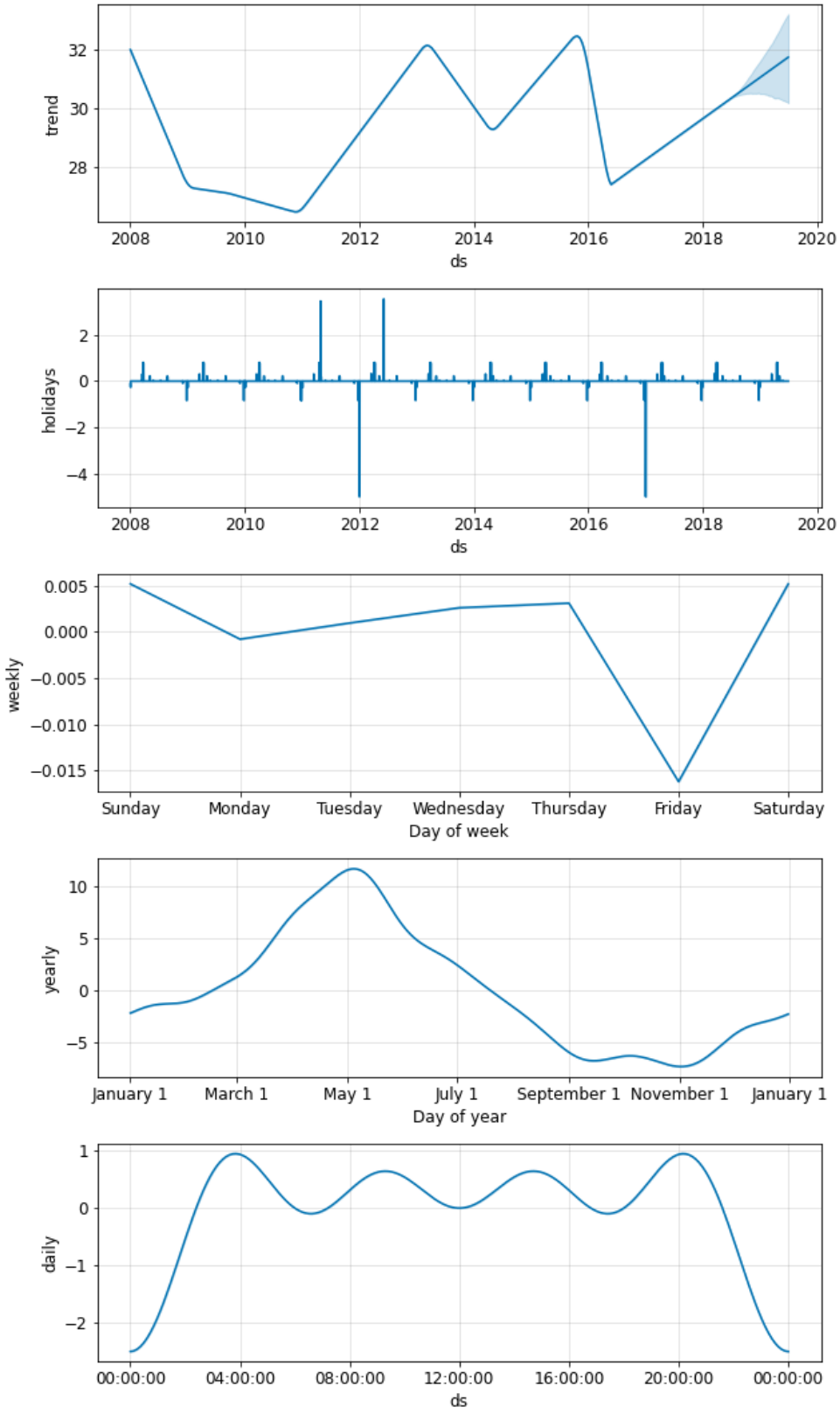


Fuente: elaboración propia

Resultados

En la Figura 24, se puede apreciar las componentes del modelo Facebook Prophet con días festivos, tendencia, la estacionalidad anual y la estacionalidad semanal de las series de tiempo. De acuerdo a la Figura 24, el día viernes los niveles de contaminación desciende notablemente y en los días sábado y domingo alcanza su mayor nivel. El mes de mayo es en donde existe mayor concentración de ozono (O3).

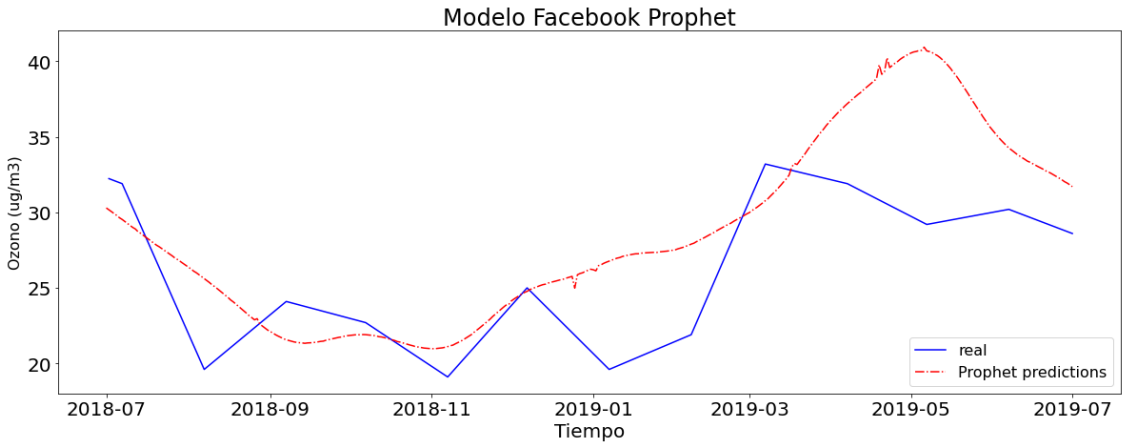
Figura 24: Componentes del modelo Facebook Prophet con días festivos



Fuente: elaboración propia

En la Figura 25, se observa el resultado de la predicción (línea punteada en rojo) y se contrasta con las mediciones reales (línea continua en azul). Claramente la predicción sigue la tendencia de la serie original, es importante señalar que los valores se encuentran alejados de la realidad, ya que no han sido capaces de predecir los valores picos.

Figura 25: Gráfica pronóstico Facebook Prophet con días festivos



Fuente: elaboración propia

Como resultado de la evaluación del conjunto de test, las métricas calculadas son las siguientes:

- error absoluto medio (MAE) 3.606.
- error cuadrático medio (MSE) 21.954.
- raíz del error cuadrático medio (RMSE) 4.685.

Modelo Facebook Prophet sin días festivos.

Para este modelo se omite los días festivos, al igual que en el apartado anterior se realizó el cálculo de los mejores parámetros mediante la implementación de una función GridSearch, seleccionando los parámetros con menor valor MAPE, para entrenar el modelo final y predecir los próximos 12 meses

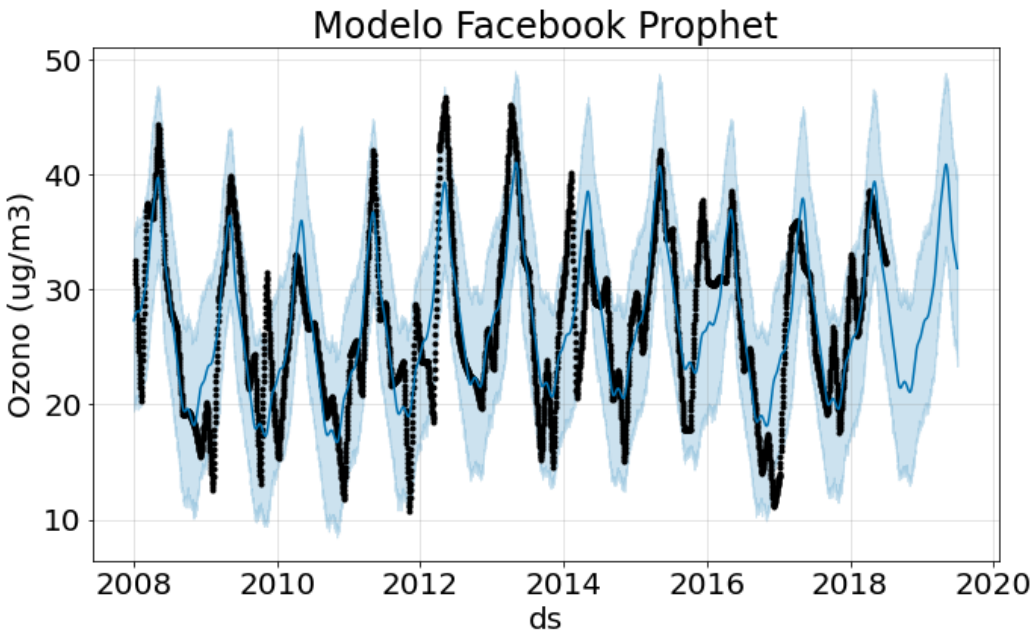
```
{'changepoint_prior_scale': 0.05,
 'n_changepoints': 200,
 'seasonality_mode': 'additive'}
```

Los parámetros en este caso se diferencian por el valor de puntos de cambio ya que en el modelo con días festivos el número de puntos de cambio es igual a 200 y en este caso ha disminuido a 150.

Predicción

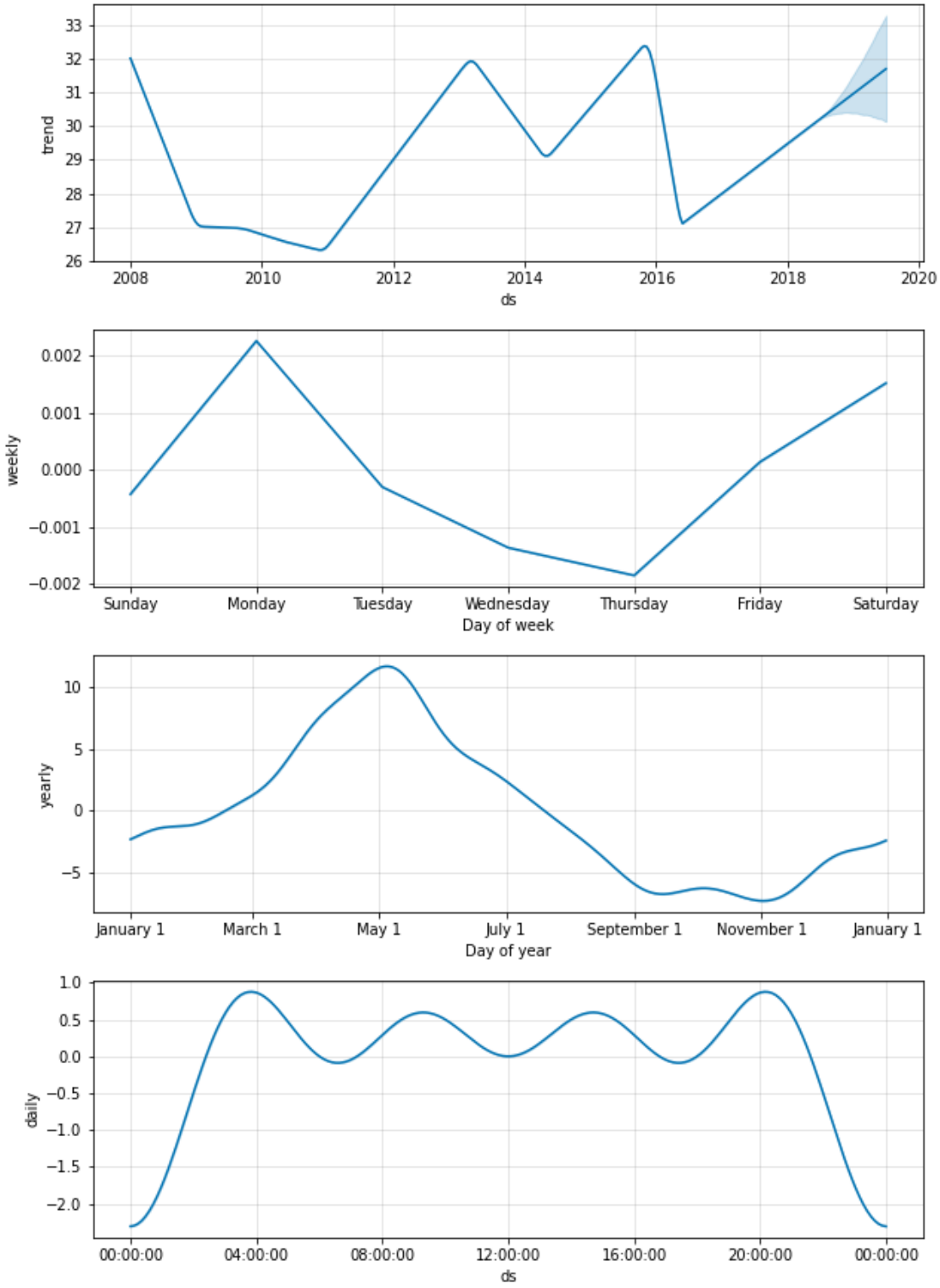
Luego de definir los parámetros, se crea el conjunto de datos de base para el forecast con la función, incluida en el paquete, llamada make_future_dataframe. Y finalmente, usando la función genérica predict se calcula la predicción, con el comando m.plot(forecast) se genera la representación gráfica resultado de la predicción (ver Figura 26)

Figura 26: Representación predicción modelo Facebook Prophet sin días festivos



Fuente: elaboración propia

Figura 27: Componentes del modelo Facebook Prophet sin días festivos



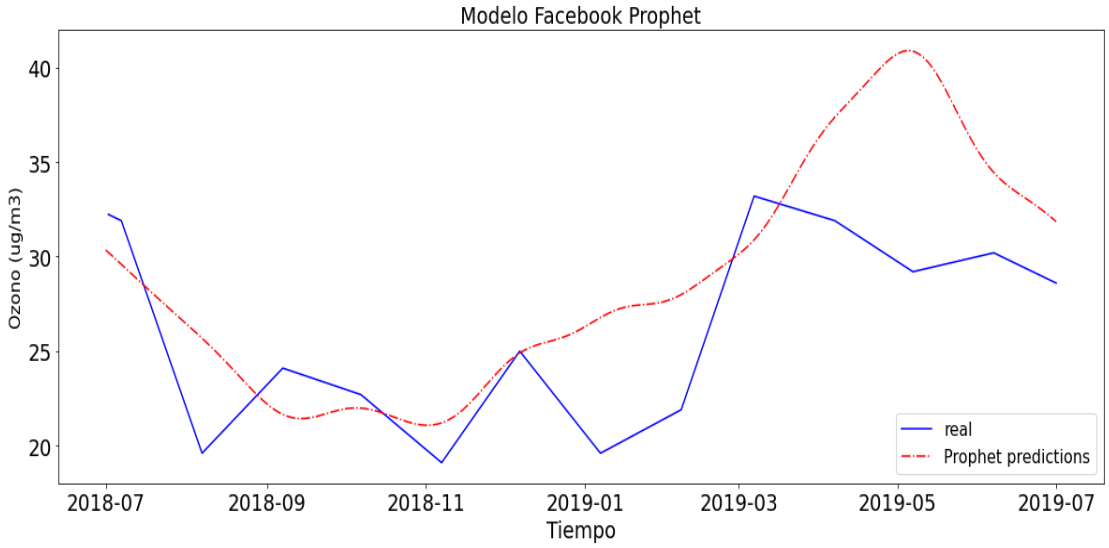
Fuente: elaboración propia

Resultados

En la Figura 27 se observan los diferentes componentes del modelo por separado: tendencia, estacionalidad anual, semanal y diaria.

En la Figura 28, se presenta el pronóstico del modelo Facebook Prophet sin días festivos junto a los valores de test (últimos 365 días) de las mediciones reales. Alrededor de 2018-11 se observa un ligero solapamiento de los valores, claramente el modelo no ha logrado predecir los valores reales ya que los valores predichos se encuentran alejados de los reales.

Figura 28: Gráfica pronóstico Facebook Prophet sin días festivos



Fuente: elaboración propia

Como resultado de la evaluación de calidad del algoritmo las métricas calculadas entre el valor real y el valor de predicción se obtienen los siguientes valores:

- error absoluto medio (MAE) 3.667.
- error cuadrático medio (MSE) 22.644.
- raíz del error cuadrático medio (RMSE) 4.758.

Luego de realizar la predicción en los dos escenarios los resultados de las métricas se presentan en la tabla 5, los valores MAE y RMSE son muy similares, el MSE del modelo con días festivos es menor, es decir que, al agregar los días festivos en el modelo, se logra una leve mejora en el rendimiento.

Tabla 5: Métricas Facebook Prophet

	MAE	MSE	RMSE
Facebook Prophet con festivos	3.606	21.954	4.685
Facebook Prophet sin festivos	3.667	22.644	4.758

Fuente: elaboración propia

3.2.4. Redes Neuronales

La red neuronal recurrente (RNN) es un tipo de red neuronal que se utiliza para procesar datos secuenciales. La memoria a corto plazo (LSTM) compensa los problemas de desaparición de gradiente, explosión de gradiente y memoria insuficiente a largo plazo de RNN. Puede hacer uso completo de la información de secuencia de tiempo de larga distancia [24]. El LSTM es un tipo especial de RNN que consta de una capa de entrada, una capa de salida y una serie de capas ocultas conectadas de forma recurrente conocidas como bloques [25].

Para la implementación del algoritmo LSTM es necesario normalizar el conjunto de datos, para lo cual se ha optado por la normalización Mínimo Máximo de la librería sklearn, en la Tabla 5 se indica la función matemática de normalizador, ya que para alimentar una red neuronal es necesario contar con un gran volumen de datos, en este caso se emplea un conjunto de datos aumentados (se realizó el proceso descrito en el modelo Facebook Prophet para el aumento de frecuencia).

Tabla 6 Formulación matemática de normalizador

Normalizador	Formula
Min Max	$Xt^i = \frac{x^i - min(x^i)}{max(x^i) - min(x^i)}$

[23]

Arquitectura red neuronal

La implementación del modelo de redes neuronales se realizó mediante el uso de la librería keras. La librería keras permite la modificación de una serie de parámetros de la red LSTM [26], a continuación, se comenta aquellos parámetros fijados, el resto de parámetros tomarán el valor por defecto.

```
import keras
from keras.layers import Dense
from keras.models import Sequential
from keras.layers import LSTM
from keras.layers import Dropout
```

```
# Construye el modelo
model = keras.Sequential()

model.add(LSTM(units = 100, return_sequences = True, input_shape = (X_train.shape[1], 1)))
model.add(Dropout(0.2))

model.add(LSTM(units = 100))
model.add(Dropout(0.2))

# salida
model.add(Dense(units = 1))
```

```
# Compilar el modelo
model.compile(optimizer = 'adam', loss = 'mean_squared_error')
```

- Número de capas: dos capas LSTM con 100 unidades, dos capas Dropout de 0.2 y una capa densa de una única neurona.
- Número de épocas de entrenamiento (epoch): 20
- Función de pérdida, define el objetivo de minimizar el error cuadrático medio entre los valores predichos y los valores de prueba durante el entrenamiento.
- Optimizador, función de optimización durante el entrenamiento, se emplea Adam.

En la Figura 29 se indica la arquitectura de la red neuronal (LSTM) definida para el modelo de predicción de Ozono. Se ha decidido añadir una capa Dropout luego de la capa LSTM con el objetivo de evitar el overfitting. La función de coste definida para la red neuronal es el MSE y como optimizador se emplea Adam, este optimizador trata de

solventar el problema con la fijación de el ratio de aprendizaje del descenso estocástico del gradiente.

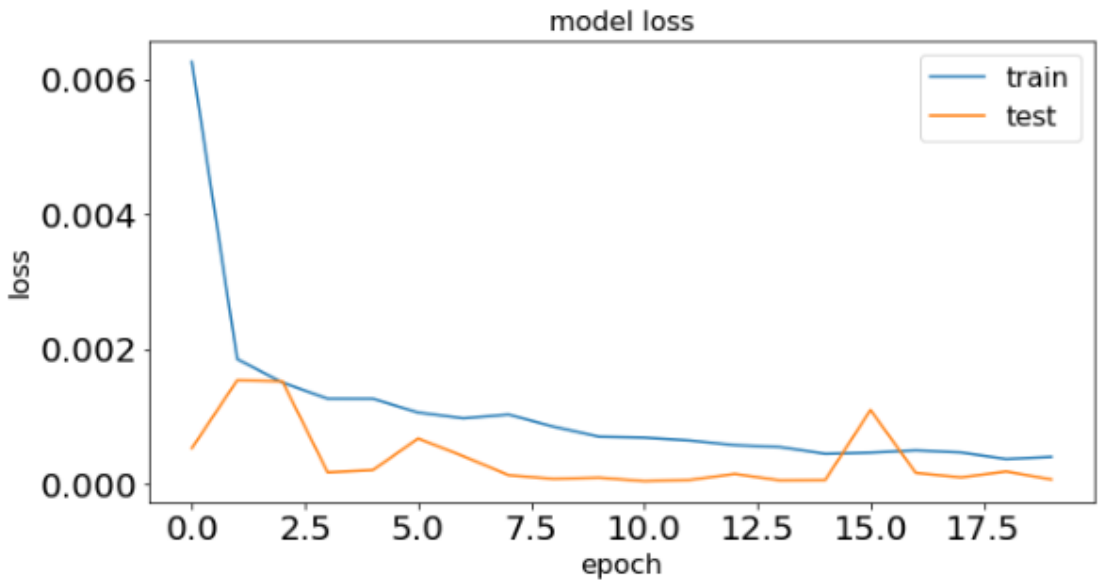
Figura 29: Arquitectura red neuronal (LSTM)

```
Model: "sequential_2"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_1 (LSTM)                (None, 36, 100)           40800
_________________________________________________________________
dropout_1 (Dropout)          (None, 36, 100)           0
_________________________________________________________________
lstm_2 (LSTM)                (None, 100)               80400
_________________________________________________________________
dropout_2 (Dropout)          (None, 100)               0
_________________________________________________________________
dense_1 (Dense)              (None, 1)                 101
=================================================================
Total params: 121,301
Trainable params: 121,301
Non-trainable params: 0
```

Fuente: elaboración propia

Luego de generar el modelo, se puede observar que la red neuronal se ha entrenado de manera satisfactoria (ver Figura 30), ya que los valores de error van disminuyendo en cada época.

Figura 30: Función de coste modelo redes neuronales



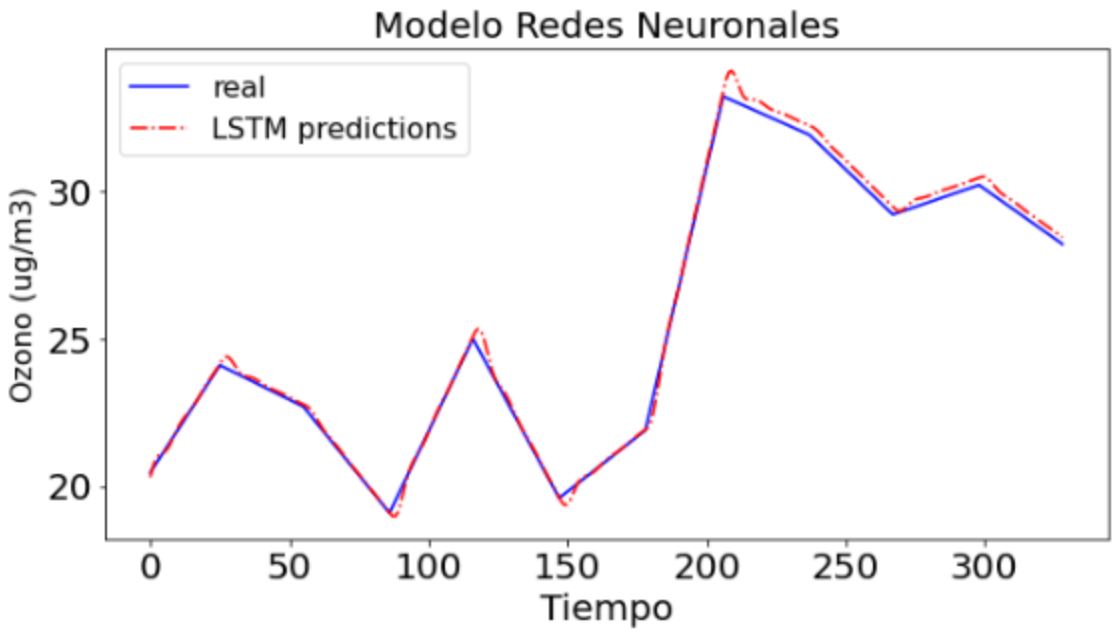
Fuente: elaboración propia

Resultados

Una vez generado el modelo se procede a realizar la evaluación con los datos de test y como resultado se obtiene que el error absoluto medio (MAE) es igual a 0.177, el error cuadrático medio (MSE) es igual a 0.057 y la raíz del error cuadrático medio (RMSE) 0.240. En la representación gráfica (Figura 31), se observa el solapamiento entre los valores de test y la predicción, visualmente se tiene una buena predicción considerando que el Error absoluto medio es igua a 0.177.

Este modelo ha detectado todos los picos existentes en el conjunto de prueba, claramente se distingue una buena predicción, los valores predichos para el primer pico sigue estando por encima del valor de test, pero los valore predichos son significativamente mejores en relación a los valores obtenidos con el modelo Facebook Prophet.

Figura 31: Representación modelo de predicción con LSTM



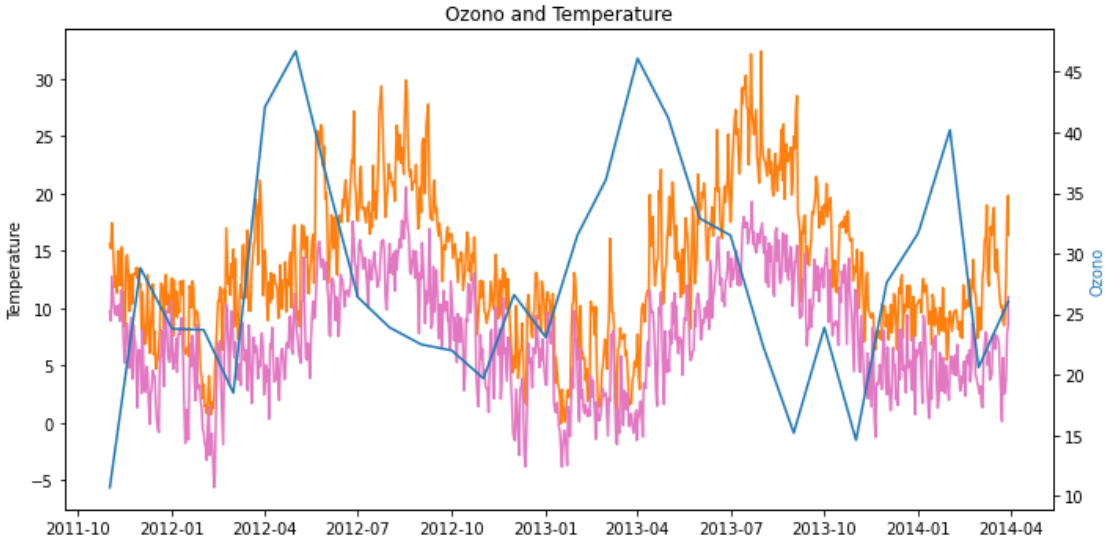
Fuente: elaboración propia

3.2.5 Modelo Experimental

Se define un modelo con otros datos, en este caso se estudia el impacto de las condiciones meteorológicas sobre la calidad del aire. Para este caso se desarrolla un análisis de datos obtenidos del conjunto de datos "weather_daily_darksky.csv" disponible en la competición Kaggle https://www.kaggle.com/rheajgurung/energy-consumption-forecast/data , el cual contiene datos de temperatura, humedad, índice UV, entre otros.

A continuación, se estudia la posible relación entre la temperatura y el ozono, comparando gráficamente la temperatura máxima (color naranja) y mínima (color rosa), junto con los niveles de ozono (Figura 32).

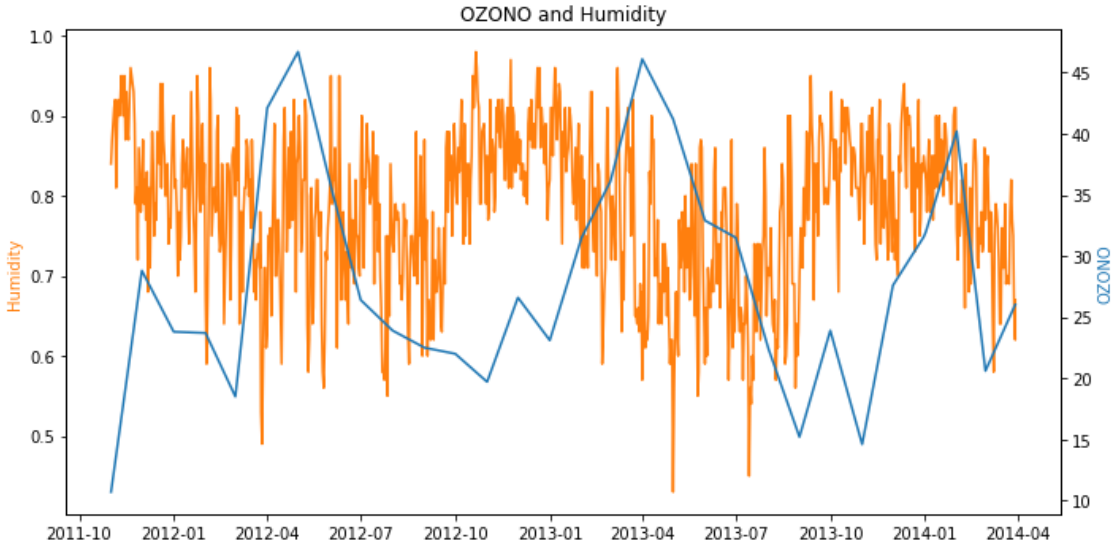
Figura 32: Ozono y temperatura



Fuente: elaboración propia

En la Figura 33 se observa que el ozono y la humedad siguen un patrón diferente, en algunos años se observa una relación inversa.

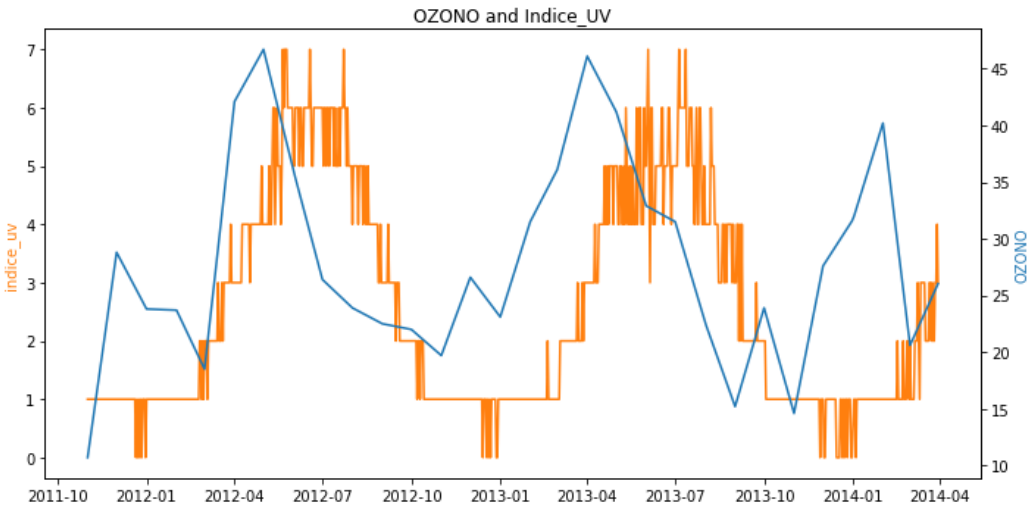
Figura 33: Ozono y humedad



Fuente: elaboración propia

El índice UV tiene una leve similitud con el ozono como se observa en la Figura 34.

Figura 34: Ozono y Indice_UV



Fuente: elaboración propia

Se calcula la correlación entre las variables y el ozono, las diferentes variables meteorológicas no presentan correlación alta con el ozono, por tanto, se selecciona la temperatura máxima, humedad y Indice_UV, ya que el conjunto de datos dispone de múltiples variables que no serán consideradas en este estudio.

Figura 35: Valores correlación variables meteorológicas - ozono

dewPoint	-0.331559
apparentTemperatureMin	-0.261835
apparentTemperatureLow	-0.259685
temperatureLow	-0.249789
temperatureMin	-0.248371
humidity	-0.234757
apparentTemperatureMax	-0.201591
windBearing	-0.200953
apparentTemperatureHigh	-0.195493
temperatureMax	-0.185636
temperatureHigh	-0.182208
pressure	-0.159963
moonPhase	-0.017532
visibility	0.035857
cloudCover	0.057904
windSpeed	0.103181
uvIndex	0.317676
O3	1.000000

Fuente: elaboración propia

Las variables **temperatura máxima, humedad** y **Indice_UV**, son discretizadas y agrupadas en una sola variable weather_cluster comprendida entre los valores 0 y 2, y dicha variable es añadida a un nuevo dataset formado por la variable objeto de interés (O3) y la variable **weather_cluster.**

El conjunto experimental de datos será utilizado para realizar la predicción bajo los parámetros del modelo redes neuronales definido en el apartado anterior, los datos son tratados de acuerdo a lo especificado en el tutorial de Jason Brownlee [27], usando retrasos de hasta 7 días, se convierte la serie temporal en un problema de aprendizaje supervisado (Figura 36).

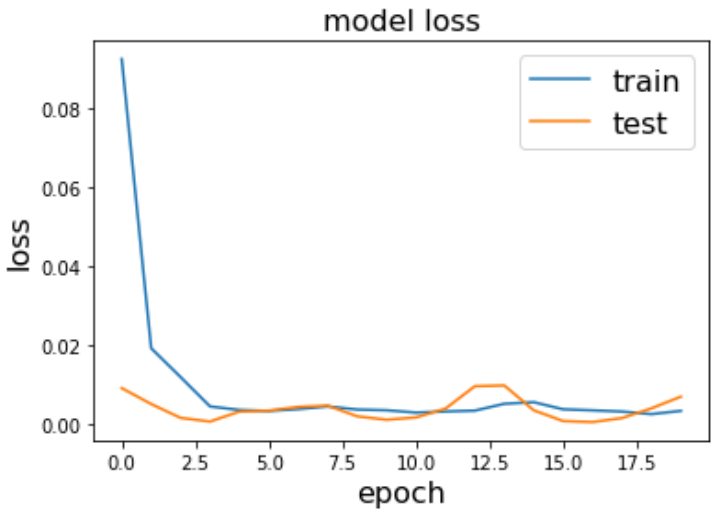
Figura 36: Detalle datos adaptados para aprendizaje supervisado

	var1(t-7)	var1(t-6)	var1(t-5)	...	var1(t-2)	var1(t-1)	var1(t)
7	10.700000	11.303333	11.906667	...	13.716666	14.320000	14.923333
8	11.303333	11.906667	12.510000	...	14.320000	14.923333	15.526667
9	11.906667	12.510000	13.113334	...	14.923333	15.526667	16.129999
10	12.510000	13.113334	13.716666	...	15.526667	16.129999	16.733334
11	13.113334	13.716666	14.320000	...	16.129999	16.733334	17.336666

Fuente: elaboración propia

Una vez entrenado el modelo experimental, se realizó la representación gráfica de la función de coste, claramente el modelo presenta una buena evolución en el entrenamiento y en la predicción ya que la función de coste tiende a 0 (Figura 37).

Figura 37: Función de coste modelo experimental



Fuente: elaboración propia

Resultados

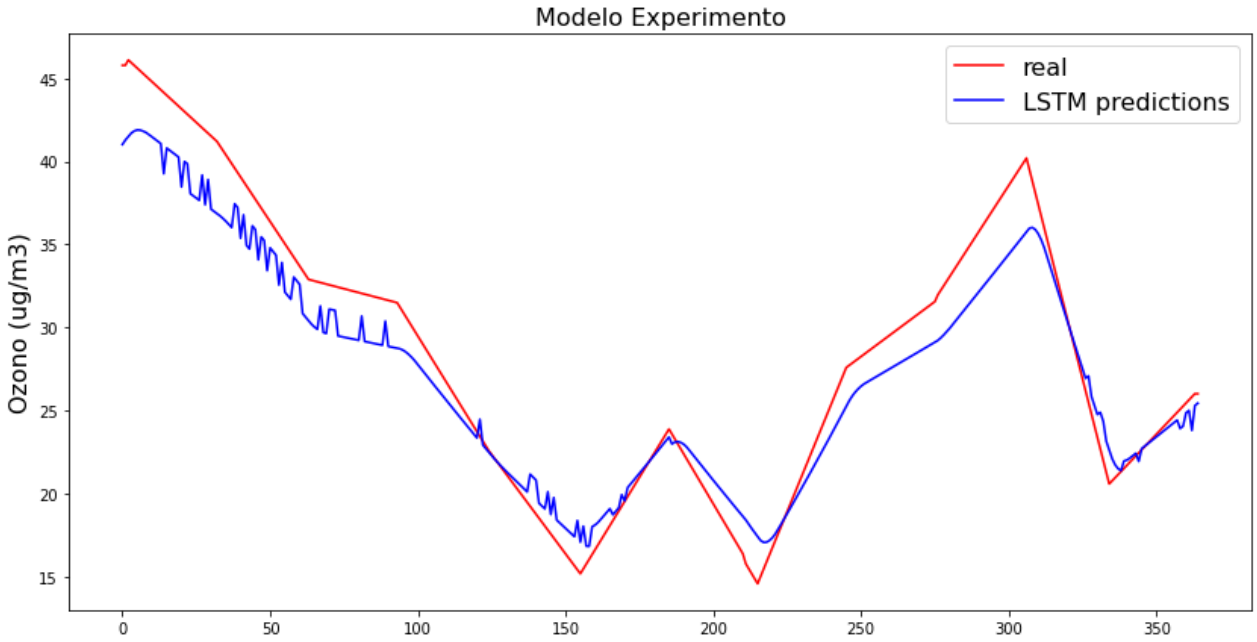
Una vez generado el modelo experimental se obtiene como resultado las siguientes métricas:

- error absoluto medio (MAE) 1.843
- el error cuadrático medio (MSE) 5.017
- raíz del error cuadrático medio (RMSE) 2.240

En la representación gráfica (Figura 38), se superponen los valores de test junto con la predicción, visualmente se tiene una buena predicción considerando que se obtiene un Error absoluto medio de 1.843.

El modelo generado ha sido capaz de replicar la tendencia de los datos y detectar los picos de la serie, a pesar de que los valores predichos se encuentran levemente alejados de los reales correspondiente a el error cuadrático medio de 5.017.

Figura 38: Representación datos test y predicción modelo experimental



Fuente: elaboración propia

3.3. Análisis comparativo de resultados

A continuación, se presenta una tabla (Tabla 7) con los valores de las métricas calculadas en los diferentes modelos predictivos.

Tabla 7: Métricas de evaluación obtenidas con los modelos predictivos.

Modelo	MAE	MSE	RMSE
ARIMA	0.143	0.034	0.184
SVR	0.247	0.087	0.295
Facebook Prophet	3.606	21.954	4.685
Redes Neuronales	0.166	0.041	0.203
Modelo Experimental LSTM	1.843	5.017	2.240

Fuente: elaboración propia

Como se puede observar en la Tabla 7, los modelos ARIMA, SVR y redes neuronales tienen RSME menor a 1, por tanto, son los modelos con mejor precisión.

En comparación con un modelo SVR, el modelo redes neuronales pueden capturar de manera más efectiva las correlaciones espacio-temporales, presentando un mejor rendimiento de predicción.

Los modelos Facebook Prophet, redes neuronales y modelo experimental fueron entrenados con un número mayor de datos, los resultados de predicción del modelo redes neuronales y modelo experimental son significativamente mejores con relación a la predicción de Facebook Prophet ya que las predicciones por redes neuronales se superponen los valores de prueba junto con la predicción.

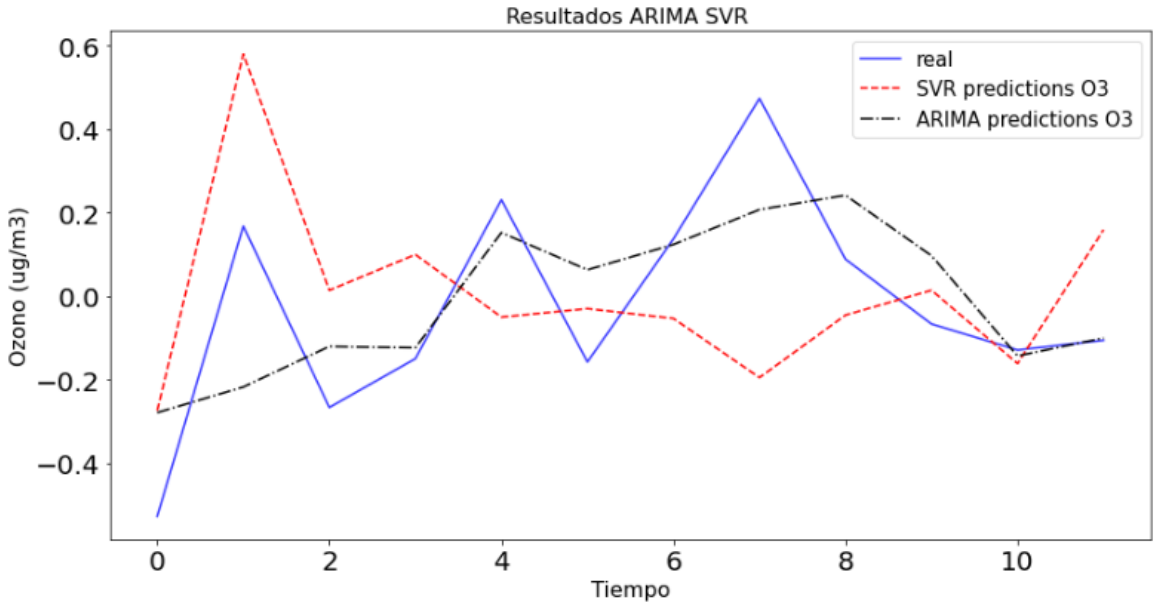
El valor alto de MSE de Facebook Prophet puede explicarse por el hecho de que el modelo no logró captar la amplitud pico a pico de estacionalidad débil. Facebook Prophet permite el estudio de las series temporales considerando los días festivos, como se pudo apreciar la calidad del modelo en este estudio tiende a disminuir al añadir días festivos.

Los modelos redes neuronales y experimental fueron modelados bajo la misma arquitectura con la diferencia que en modelo experimental se han considerado otros datos como son las variables meteorológicas, el valor RSME es superior en el modelo experimental lo que significa que el aumento de variables ha disminuido la capacidad de predicción. Los modelos generados basados en redes neuronales siguen las tendencias de los datos de prueba, logrando predecir valores más cercanos para las zonas de picos.

Las métricas del modelo experimental son significativamente mayores a las del modelo redes neuronales lo que ´puede ser causa de que los datos meteorológicos escogidos no son representativos.

Los modelos ARIMA y SVR fueron entrenados con el conjunto de datos de las concentraciones mensuales de ozono, las mejores predicciones fueron las del modelo ARIMA ya que los valores predichos por ARIMA se asemejan más a los valores reales (Figuras 39 y 40).

Figura 39: Diferencias entre ARIMA Y SVR



Fuente: elaboración propia

Figura 40: Diferencias entre los valores predichos por ARIMA Y SVR

Datetime	O3_log_diff	SVR	ARIMA
2018-08-31	-0.526295	-0.273458	-0.278268
2018-09-30	0.168150	0.580187	-0.217010
2018-10-31	-0.265907	0.014595	-0.119853
2018-11-30	-0.149229	0.100053	-0.122990
2018-12-31	0.231123	-0.049739	0.152559
2019-01-31	-0.157134	-0.029461	0.063435
2019-02-28	0.137568	-0.052971	0.124178
2019-03-31	0.473737	-0.194434	0.207641
2019-04-30	0.088861	-0.044941	0.242030
2019-05-31	-0.066187	0.014595	0.097261
2019-06-30	-0.128407	-0.160889	-0.142638
2019-07-31	-0.105184	0.159321	-0.099733

Fuente: elaboración propia

4. Conclusiones y trabajo futuro

4.1. Conclusiones

En este trabajo se han comparado diferentes modelos de predicción estocásticos y heurísticos aplicados a la estimación de la calidad del aire, ya que en la actualidad se estudian diferentes posibilidades para contrarrestar la contaminación ambiental, es por esta razón que se generan modelos de predicción con el objetivo de anticiparse a los escenarios de alta contaminación, en este caso se ha logrado comprobar que las técnicas de inteligencia artificial, específicamente, las relacionadas con predicción de series de tiempo, son de gran beneficio para la automatización de procesos de estimación de calidad del aire.

Los datos proporcionados por el fichero original fueron pre-procesados con el objetivo de garantizar la calidad del dato, se realizó un estudio exploratorio de los datos el cual permitió identificar la necesidad de interpolar los datos de acuerdo con la necesidad del modelo predictivo, esto permitió comparar los resultados de los modelos por similitud del conjunto de datos.

Los resultados de la aplicación de cinco modelos predictivos para predecir la concentración de contaminantes de aire basados en datos históricos de concentración de contaminantes, específicamente ozono (O3), señalan que los modelos con mejor capacidad de predicción fueron el modelo ARIMA y el modelo redes neuronales.

Se ha logrado obtener modelos predictivos con un error absoluto medio (MAE) menor a 1, en el caso del modelo ARIMA se ha conseguido un MAE de 0.143, en el modelo redes neuronales se obtuvo un MAE de 0.166 y en el modelo SVR se obtuvo un MAE de 0.243

El algoritmo de predicción basado en la red neuronal recurrente LSTM, incluida la construcción del modelo, el entrenamiento del modelo, la prueba del modelo y la aplicación de predicción permite determinar que los niveles de Ozono predichos por la red neuronal LSTM son consistentes con los niveles de Ozono reales.

Se generaron dos modelos mediante redes neuronales, en los cuales se obtuvo un RMSE de 0.203 para el modelo de predicción de ozono (O3), en el caso del modelo experimental en donde se emplearon datos meteorológicos el rendimiento se vio afectado ya que el RMSE fue de 2.240.

Las máquinas de soporte vectorial regresivas (SVR) presentan igual rendimiento con las funciones Kernel Trialgular y Truncated, con un valor RMSE de 0.295

Los modelos predictivos basados en redes neuronales (LSTM) permiten obtener buenos resultados, pero cabe recalcar que el tiempo y el coste computacional necesario para generar estos modelos es considerablemente mayor al tiempo y coste para generar los modelos SVR, Facebook Prophet y ARIMA.

4.2. Trabajo futuro.

La estimación de calidad del aire es un problema abierto en donde existe la posibilidad de indagar múltiples factores. A continuación, se definen las líneas de trabajo futuro que no han sido posibles explorar en este trabajo entre las cuales se encuentran:

- Realizar la evaluación de otras arquitecturas de redes neuronales mediante la combinación de hiperparámetros para mejorar la precisión del modelo predictivo.
- Realizar un estudio comparativo de modelos predictivos para estimar la calidad del aire basado en la comparación de funciones Kernel, con datos meteorológicos.
- Emplear los datos de otras fuentes de Londres, que contengan las mediciones diarias de los contaminantes, para no tener la necesidad de realizar la interpolación de datos faltantes.
- Modificar los modelos generados para realizar predicciones de una distribución menor, como por ejemplo predecir los siguientes tres días.

5. Glosario

ARIMA: Modelo de media móvil integrada autorregresiva. Acrónimo en inglés de Autoregressive Integrated Moving Average Model.

CTSPD: Descubrimiento continuo del patrón de secuencia objetivo. Acrónimo en inglés de Continuous Target Sequence Pattern Discovery.

Jupyter Notebook: Entorno de trabajo interactivo para el desarrollo de código en lenguaje Python.

LSTM: Unidades de memoria a corto y largo plazo. Acrónimo en inglés de Long Short-Term Memory Units.

MAE: Error absoluto medio, Acrónimo en inglés de Mean Absolute Error.

MAPE: Error porcentual absoluto medio. Acrónimo en inglés de Mean absolute Percentage Error.

MLP: Perceptrón multicapa. Acrónimo en inglés de multilayer perceptron.

MSE: Error cuadrático medio. Acrónimo en inglés de Mean Squared Error

NaN: No número. Acrónimo en inglés de Not a Number

PEC: Prueba de evaluación continua.

RBFN: Funciones de Base Radial. Acrónimo en inglés de Radial Based Functions Network

RMSE: Raís del error cuadrático medio. Acrónimo en inglés de Root Mean Squared Error

RNN: Red Neuronal Recurrente. Acrónimo en inglés de Recurrent Neural Network.

SVM: Máquinas de vectores de soporte**.** Acrónimo en inglés de Support Vector Machine

SVR: Regresión de vectores de soporte. Acrónimo en inglés de Support Vector Regression

TFM: Trabajo Final de Máster.

UOC: Universitat Oberta de Catalunya.

6. Bibliografía

[1] M. Gaitán, J. Cancino, and B. Eduardo, “Análisis del estado de la calidad del aire en Bogotá,” *Rev. Ing.*, vol. unknown, no. 26, pp. 81–92, 2007, doi: 10.16924/riua.v0i26.299.

[2] C. Silva, S. Alvarado, R. Montaño, and P. Pérez, “Modelamiento de la contaminación atmosférica por particulas: Comparación de cuatro procedimientos predictivos en Santiago, Chile,” pp. 113–127, 2003.

[3] C. Hood *et al.*, “Air quality simulations for London using a coupled regional-to-local modelling system,” *Atmos. Chem. Phys.*, vol. 18, no. 15, pp. 11221–11245, 2018, doi: 10.5194/acp-18-11221-2018.

[4] E. Aguirre, A. Anta, L. R, and M. Albizu, “Relevancia de las variables meteorológicas en el diseño de un modelo de predicción de los niveles de ozono, en tiempo real, basado en el uso de redes neuronales,” *Asoc. Metereológica Española*, no. 1, 2006.

[5] D. Zhu, C. Cai, T. Yang, and X. Zhou, “A Machine Learning Approach for Air Quality Prediction: Model Regularization and Optimization,” *Big Data Cogn. Comput.*, vol. 2, no. 1, p. 5, 2018, doi: 10.3390/bdcc2010005.

[6] G. Corani, “Air quality prediction in Milan: Feed-forward neural networks, pruned neural networks and lazy learning,” *Ecol. Modell.*, vol. 185, no. 2–4, pp. 513–529, 2005, doi: 10.1016/j.ecolmodel.2005.01.008.

[7] M. Yadav, S. Jain, and K. R. Seeja, “Prediction of Air Quality Using Time Series Data Mining,” in *International Conference on Innovative Computing and Communications*, 2019, pp. 13–20.

[8] Y. Rybarczyk and R. Zalakeviciute, “Machine learning approaches for outdoor air quality modelling: A systematic review,” *Appl. Sci.*, vol. 8, no. 12, 2018, doi: 10.3390/app8122570.

[9] L. L. Chiarvetto Peralta, F. A. Rey Saravia, and N. B. Brignole, “Aplicacin de Redes neuronales artificiales para la prediccin de calidad de aire,” *Asoc. Argentina Mecánica Comput.*, vol. XXVII, pp. 3607–3625, 2008.

[10] J. Ma, J. C. P. Cheng, C. Lin, Y. Tan, and J. Zhang, “Improving air quality

prediction accuracy at larger temporal resolutions using deep learning and transfer learning techniques," *Atmos. Environ.*, vol. 214, no. July, p. 116885, 2019, doi: 10.1016/j.atmosenv.2019.116885.

[11] S. I. V. Sousa, F. G. Martins, M. C. Pereira, and M. C. M. Alvim-Ferraz, "Prediction of ozone concentrations in Oporto city with statistical approaches," *Chemosphere*, vol. 64, no. 7, pp. 1141–1149, 2006, doi: 10.1016/j.chemosphere.2005.11.051.

[12] J. Ma, J. C. P. Cheng, C. Lin, Y. Tan, and J. Zhang, "Improving air quality prediction accuracy at larger temporal resolutions using deep learning and transfer learning techniques," *Atmos. Environ.*, vol. 214, no. July, p. 116885, 2019, doi: 10.1016/j.atmosenv.2019.116885.

[13] J. López, "Análisis de Series deTiempo Pronóstico de demanda de uso de aeropuertos en Argentina al 2022," 2018.

[14] S. Rani Patra, "Time Series Forecasting of Air Pollutant Concentration Levels using Machine Learning," vol. 4, no. 5, pp. 280–284, 2017.

[15] IBM, "Funcionamiento de SVM." [Online]. Available: https://www.ibm.com/support/knowledgecenter/es/SS3RA7_sub/modeler_mainhelp_client_ddita/clementine/svm_howwork.html.

[16] D. S. Hermiyanty, Wandira Ayu Bertin, "PREDICCIÓN DE SISTEMAS CAÓTICOS CON REDES NEURONALES: UN ESTUDIO COMPARATIVO DE LOS MODELOS DE PERCEPTRÓN MULTICAPA Y FUNCIONES DE BASE RADIAL," *J. Chem. Inf. Model.*, vol. 8, no. 9, pp. 1–58, 2017, doi: 10.1017/CBO9781107415324.004.

[17] S. Rani Patra, "Time Series Forecasting of Air Pollutant Concentration Levels using Machine Learning," vol. 4, no. 5, pp. 280–284, 2017.

[18] J. Casas Roma, A. Bosch Rué, L. Bagén, and Toni, *Deep learning principios y fundamentos*. .

[19] D. Z. Antanasijević, V. V. Pocajt, D. S. Povrenović, M. D. Ristić, and A. A. Perić-Grujić, "PM10 emission forecasting using artificial neural networks and genetic algorithm input variable optimization," *Sci. Total Environ.*, vol. 443, pp. 511–519, 2013, doi: 10.1016/j.scitotenv.2012.10.110.

[20] Y. Bai, Y. Li, X. Wang, J. Xie, and C. Li, “Air pollutants concentrations forecasting using back propagation neural network based on wavelet decomposition with meteorological conditions,” *Atmos. Pollut. Res.*, vol. 7, no. 3, pp. 557–566, 2016, doi: 10.1016/j.apr.2016.01.004.

[21] C. Li, N. C. Hsu, and S.-C. Tsay, “A study on the potential applications of satellite data in air quality monitoring and forecasting,” *Atmos. Environ.*, vol. 45, no. 22, pp. 3663–3675, 2011, doi: https://doi.org/10.1016/j.atmosenv.2011.04.032.

[22] J. Riofr, “Forecasting Consumer Price Index (CPI) of Ecuador : A comparative study of predictive models .”

[23] H. A. Mora Paz, “Comparativo de kernels sobre predicción de oferta de fuentes alternativas de energía,” 2019.

[24] C. Ying, “Voltages prediction algorithm based on LSTM recurrent neural network,” *Pre-proof*, p. 10, 2020.

[25] X. Li *et al.*, “Long short-term memory neural network for air pollutant concentration predictions: Method development and evaluation,” *Environ. Pollut.*, vol. 231, pp. 997–1004, 2017, doi: 10.1016/j.envpol.2017.08.114.

[26] J. Brownlee, “Time Series Prediction with LSTM Recurrent Neural Networks in Python with Keras,” 2016. [Online]. Available: https://machinelearningmastery.com/time-series-prediction-lstm-recurrent-neural-networks-python-keras/. [Accessed: 17-May-2020].

[27] J. Brownlee, “How to Convert a Time Series to a Supervised Learning Problem in Python,” 2017. [Online]. Available: https://machinelearningmastery.com/convert-time-series-supervised-learning-problem-python/. [Accessed: 18-May-2020].

Anexos

Anexo 1

https://github.com/nadia1477/TFM-UOC

Enlace repositorio GitHub en donde se encuentran los ficheros de datos originales y pre-procesados, notebook con el análisis de datos, la generación de modelos y análisis predictivo, notebook con el modelo experimental, notebook con la búsqueda de hiperparámetros para la optimización del modelo Facebook Prophet y los ficheros producto de la búsqueda de parámetros.