Universitat Oberta de Catalunya (UOC)

Máster Universitario en Ciencia de Datos (*Data Science*)

TRABAJO FINAL DE MÁSTER

Área: 5 - Modelos predictivos

Predicción de la calidad del aire

Elaboración de un modelo predictivo de la calidad del aire para la ciudad de Madrid

---

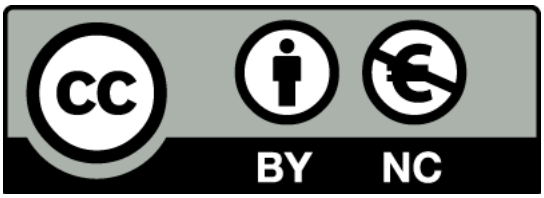
Autor: Abel Serrano Juste

Tutor: Sergio Trilles Oliver

Profesor: Albert Solé Ribalta

Barcelona, 24 de junio de 2020

Créditos y Licencia



El código fuente correspondiente a los *notebooks* publicados en la plataforma *kaggle* están licenciados doblemente con las licencias: Apache 2.0 y GPLv3. Cualquier tercero podrá acogerse a cualquiera de estas dos licencias para el uso de este código.

Cualquier otro código fuente que **no** esté publicado en la plataforma *kaggle* tiene licencia GPLv3.

FICHA DEL TRABAJO FINAL

Título del trabajo:	Predicción de la calidad del aire
Nombre del autor:	Abel Serrano Juste
Nombre del colaborador/a docente:	Sergio Trilles Oliver
Nombre del PRA:	Albert Solé Ribalta
Fecha de entrega (mm/aaaa):	06/2020
Titulación o programa:	Máster en Ciencia de Datos
Área del Trabajo Final:	Área 5: Modelos predictivos
Idioma del trabajo:	Español
Palabras clave:	*aprendizaje automático*, *modelos predictivos*, *redes neuronales*

También cosechemos unos puñados de cada árbol.
Habiendo trabajado la limpieza de la respiración
y enjuagado los gases de la atmósfera,
seremos conscientes de aquello que comemos.
Y cuando el aire no huela a monóxidos,
entonces, sólo entonces,
seremos comensales de lo plantado...

Extracto de "Ruralicemos la ciudad".
– Raúl de Tapia, Herbario Sonoro.

Agradecimientos

Desde el mismo comienzo, mi andadura en estos estudios de máster ha estado lleno de dificultades e incertidumbres. Ahora quiero aprovechar la oportunidad de agradecer a las personas que me han ayudado a hacerlo más llevadero, para que hoy pueda entregar orgullosamente el documento que pone punto y final a este año y medio de estudios de máster.

A mis ex-compañeros de trabajo durante mi etapa en la universidad: A Samer Hassan, puesto que fue él quién en primer lugar me sugirió que realizara los estudios de posgrado y me ayudó en la financiación de éste; y a Javier Arroyo, por ser tan comprensivo mientras compaginaba trabajo y estudios, y por haber sido mi "Pepito Grillo" en mis decisiones académicas, profesionales y vitales.

A mi padre, a mi madre, a mi tía y a mi abuela, por su apoyo y por haberme dado su ayuda en lo que estuvo al alcance de su mano.

A mis amigos Clara y Moisés por ser mis fieles compañeros de vida.

A mi amiga Marta por ser mi confidente más sincero y apoyarme en los momentos difíciles.

A mi amigo Paco. Por escuchar y darme nuevas ideas.

A mis compañeras y compañeros de las huertas vecinales. Sin estos pequeños oasis de vida en la ciudad difícilmente hubiese soportado tantos años el trajín artificioso en la gran ciudad.

A mis compañeros y compañeras de máster porque, aún en una universidad 100 % a distancia, han conseguido crear una pequeña comunidad en la que compartir conocimientos, alegrías y penas.

Por último, a toda la gente que, mientras compaginaba mis viajes con los estudios, me dio un lugar donde descansar, un trayecto, algo de comer o un sitio donde estudiar.

Resumen

Hoy en día muchas ciudades del mundo disponen de estaciones ambientales donde se mide la cantidad de contaminantes que hay en el aire. Estas mediciones son usadas para evaluar la calidad del aire que respiran los habitantes de esa ciudad y tomar medidas para mantener dichos valores dentro de unos rangos saludables.

En este trabajo se elaboran, de manera iterativa, diferentes prototipos usando técnicas de aprendizaje profundo (*deep learning*) para la predicción de la calidad del aire de la ciudad de Madrid. Al final de cada iteración, se presenta un resumen de los resultados obtenidos de cada prototipo, analizando cómo influyen las diferentes variables y valores de los hiperparámetros.

El modelo final es una red neuronal compleja, capaz de realizar predicciones de las 1, 8, 16 y 24 horas siguientes, de manera simultánea para las veinticuatro estaciones de calidad del aire que hay en dicha ciudad. Para ello, este modelo hace uso de datos históricos de la calidad del aire, así como de datos meteorológicos y otros datos auxiliares adicionales.

Palabras clave: calidad del aire, modelos predictivos, redes neuronales, aprendizaje automático, predicción de series temporales.

Abstract

Today, many cities in the world have environmental stations to measure the amount of pollution contained in the air. These measures are used to evaluate the quality of the air breathed in by the inhabitants of the city and to act accordingly to keep those values within a healthy range.

This work develops, in an iterative manner, different prototypes using deep learning techniques to predict the air quality for the city of Madrid. After each iteration, a summary is presented with all the results obtained by each prototype, analyzing the influence of the different variables and different values for the hyperparameters.

The final model is a complex neuronal network, capable of making predictions for the next 1, 8, 16 and 24 hours, simultaneously over the twenty-four air quality stations available in that city. To this effect, this model makes use of historical data, as well as other auxiliary data such as meteorological data.

Keywords: air quality, predictive models, neural networks, machine learning, time series forecasting.

Índice general

Índice de figuras

Índice de cuadros

Capítulo 1

Introducción

1.1. Contexto y justificación del Trabajo

De todos los riesgos para la salud del ser humano procedente de fuentes medio ambientales, la contaminación en el aire es el que está relacionado con el mayor número de muertes y de enfermedades crónicas (Collaborators [2017]). La contaminación atmosférica incide en la aparición y agravamiento de enfermedades respiratorias, así como enfermedades vasculares y cánceres (WHO2018, EEA2005, Chovin and Roussel [1970]). De hecho, en España fallecen 22 veces más personas a causa de la contaminación atmosférica que debido a accidentes de tráfico en carretera[1].

La ciudad de Madrid tiene un grave problema de contaminación del aire, el cual ha sido sancionado en sucesivas ocasiones durante los últimos años por la Unión Europea. Dicho problema ha llevado a la planificación y ejecución de importantes medidas por parte del Ayuntamiento.

La predicción de los niveles de contaminación en tiempo real puede servir para tomar mejores decisiones a la hora de tomar medidas o puede servir de información a los grupos más afectados por los efectos nocivos de la contaminación.

Además, un modelo predictivo podría usarse como evaluador de diferentes escenarios hipotéticos en los que se aplicasen diferentes medidas para controlar la contaminación en el aire.

Sin perjuicio de todo lo anterior, este Trabajo de Fin de Máster (TFM) viene motivado por una preocupación personal. Como ciudadano nacido en la ciudad de Madrid, he padecido y padezco los efectos de la polución en el aire. También, como persona preocupada por la sobreexplotación del medio ambiente, me preocupa saber cuáles son las consecuencias sobre el medio ambientes de la contaminación producida en la ciudad.

De entre todos los contaminantes que se miden en las estaciones de la calidad del aire, en este

[1] Se estiman alrededor de 24.000 muertes prematuras en España según la Agencia Europea de Medioambiente (EEA2019), mientras que se registraron 1.098 muertos por accidentes de tráfico en 2019

trabajo se ha escogido estudiar los relativos al dióxido de nitrógeno (NO_2). Este compuesto es nocivo para el cuerpo humano, más tóxico incluso que el CO (Chovin and Roussel [1970]), y está dentro de los gases considerados de efecto invernadero. Además, se trata de un contaminante volátil y el más relacionado con el tráfico rodado, por lo que se puede usar como estimador de la contaminación debida al tráfico.

1.2. Objetivos del Trabajo

Objetivos Principales

- Realizar un modelo predictivo de la calidad del aire para la ciudad de Madrid usando el histórico de los años anteriores.
- Optimizar el modelo anterior e incluir nuevas variables en este modelo para conseguir un modelo de alta precisión.
- Realizar predicciones a espaciadas en el tiempo a 8 horas, 12 horas y 24 horas.

Objetivos Secundarios

- Entender la problemática de la contaminación del aire para la ciudad de Madrid.
- Estudiar, limpiar y procesar los datos que se vayan a usar.
- Diseñar y desarrollar varios posibles modelos mediante diferentes combinaciones de parámetros, datos de entrada y diversas técnicas de aprendizaje automático. Evaluar y comparar dichos modelos.
- Investigar cómo afectan diferentes variables como la meteorología o el día de la semana en la contaminación del aire
- Difundir los resultados obtenidos y contactar con aquellas instituciones relevantes para que valoren su uso y la integración del modelo con sus servicios.

1.3. Enfoque y método seguido

Para este trabajo usaremos la "Metodología Fundamental para la Ciencia de Datos", desarrollada por IBM (Rollins [2015]).

Esta metodología consta de los siguientes diez pasos:

Etapa 1: Comprensión del negocio. En nuestro caso, revisión del estado del arte, asistencia a talleres y lectura sobre la polución atmosférica y, más específicamente, sobre la problemática de la calidad del aire en la ciudad de Madrid.

Etapa 2: Enfoque analítico. Expresar el problema bajo el contexto de las técnicas estadísticas y de aprendizaje automático.

Etapa 3: Requisitos de datos. Establecer qué datos hacen falta y qué forma deben tener.

Etapa 4: Recopilación de datos. Descarga e inspección de los datos desde los diferentes portales de datos abiertos (*open data*).

Etapa 5: Comprensión de datos. Estudio de los datos disponibles: análisis de su utilidad y caracterización mediante técnicas estadísticas.

Etapa 6: Preparación de datos. Filtrado de los datos requeridos y preparación de los datos para que sirvan de entrada para el modelo o modelos de aprendizaje automático a implementar.

Etapa 7: Modelado. Diseño de diferentes modelos predictivos para la calidad del aire.

Etapa 8: Evaluación. Evaluación de los modelos contra un conjunto de test totalmente disjunto de los datos usados durante el entrenamiento del modelo. Medición del desempeño del modelo mediante métricas usadas habitualmente para la evaluación de los modelos de regresión de series temporales: RMSE y MAPE. Contraste entre las predicciones realizadas por el modelo y las realmente observados. Comparación de los resultados obtenidos con otros sistemas de predicción de la calidad del aire dentro del Estado del Arte actual [2].

Etapa 9: Implementación. Implementación de los modelos diseñados previamente.

Etapa 10: Retroalimentación. Comparar las predicciones realizadas con los datos de calidad el aire que fueron finalmente. Usar dicho análisis para ajustar los diferentes parámetros y datos de entrada del modelo para mejorar su precisión y utilidad.

1.3.1. Reproducibilidad

Con el fin de hacer "ciencia de calidad" y que nuestros resultados sean útiles en la mayor medida posible para la sociedad, aplicaremos las pautas metodológicas necesarias para hacer que nuestro trabajo sea reproducible (Rodriguez-Sanchez et al. [2016]).

El trabajo completo consta de esta memoria, de los datos usados por el modelo, del código del modelo, de las librerías o utilidades externas utilizadas y el control de versiones del código. Para conseguir la reproducibilidad, cada una de estos elementos deben ser capaces de ser inspeccionados por cualquiera y en cualquier momento. Para ello, se han tomado las siguientes medidas:

[2] En el capítulo 2 se hace la correspondiente revisión del Estado del Arte.

1. **La memoria**, este documento, donde se explica el proceso seguido y los resultados obtenidos, tiene licencia abierta *Creative Commons*.

2. **Los datos** usados se pondrán disponibles en un repositorio de datos en la nube, en caso de que los derechos de propiedad sobre esos datos lo permitan.

3. **El código** de los diferentes modelos, así como de los *scripts* de preparación y análisis de los datos y de todas las utilidades asociadas serán subidos a algún repositorio en línea. Todo el código estará disponible con una licencia de código abierta.

Todos los enlaces y cuestiones referentes a los elementos anteriores están debidamente recogidos en el anexo A de este documento titulado Reproducibilidad.

1.4. Planificación del Trabajo

1. Trabajar en la definición y alcance del TFM (PEC1). [19 Febrero - 01 Marzo]

2. Lectura del estado del arte. Formación sobre el tema en cuestión: lectura de artículos y de informes, asistencia a talleres, contacto con instituciones implicadas, inspección de los datos disponibles (PEC2). [02 Marzo - 22 Marzo]

3. Descarga y procesamiento de datos (PEC3). [23 Marzo - 23 Mayo]

4. Diseño e implementación del modelo predictivo. Prueba con diferentes técnicas. Evaluación de los modelos resultantes y elección del modelo con mejor desempeño (PEC3). [23 Marzo - 23 Mayo]

5. Redacción de la memoria (PEC4). [24 Mayo - 10 Junio]

6. Defensa del trabajo (PEC5). [11 Junio - 24 Junio]

En el anexo C se encuentra el diagrama de Gantt correspondiente a esta planificación [3].

[3] Debido a la situación excepcional originada por la pandemia del COVID-19, el plazo para la entrega de la PEC3 se aplazó dos semanas más (hasta el 9 de Junio), con el consiguiente retraso de los sucesivas entregas; quedando la entrega de la memoria (PEC4) para el 24 de junio y la presentación (PEC5) para el 30 de junio.

Capítulo 2

Estado del Arte

2.1. Modelos de predicción del aire

El modelo de referencia que tenemos en España es el sistema CALIOPE (Baldasano et al. [2011, 2012]) que está alojado en el Barcelona Supercomputing Center[1]. Este sistema hace estimaciones de la calidad del aire para la Península Ibérica Española y para toda Europa. El enfoque tomado es recrear el escenario real mediante la simulación de las condiciones atmosféricas, la emisión de contaminantes y los efectos químicos asociados. Para ello, usa una combinación de modelos de simulación: un modelo específico que hace una predicción de las emisiones de los contaminantes (HERMES), un modelo para el pronóstico de las condiciones meteorológicas (WRF-ARW), un modelo de transporte químico para modelar la dispersión y transformación de los contaminantes (CMAQ v4.5), y un modelo para la contaminación proveniente de polvo natural (BSC-DREAM8b).

El modelo resultante da pronósticos de la calidad del aire para las próximas 48 horas y además se evalúa y se corrige a sí mismo en *quasi* tiempo real. Las previsiones se pueden observar en forma de mapa en la página web, junto con otros mapas relacionados como la previsión de las emisiones o la evaluación de la calidad del aire para los puntos que corresponden a la red de estaciones de calidad del aire que hay en la Península Ibérica y en toda Europa. También, se dispone de una app móvil para los sistemas Android e iOS donde se puede consultar la predicción de cada contaminante en cada estación de España para cada hora dentro de un rango de las próximas 24h (ver figura 2.1). Además, este modelo se ha usado para evaluar el potencial impacto de medidas tomadas con el fin de mitigar la contaminación en el aire, como, por ejemplo, la evaluación de una zona de bajas emisiones para la ciudad de Barcelona (Benavides et al. [2020]).

No obstante, en Rybarczyk and Zalakeviciute [2018] hicieron una revisión sistemática del

[1]Página web del proyecto: http://www.bsc.es/caliope/es

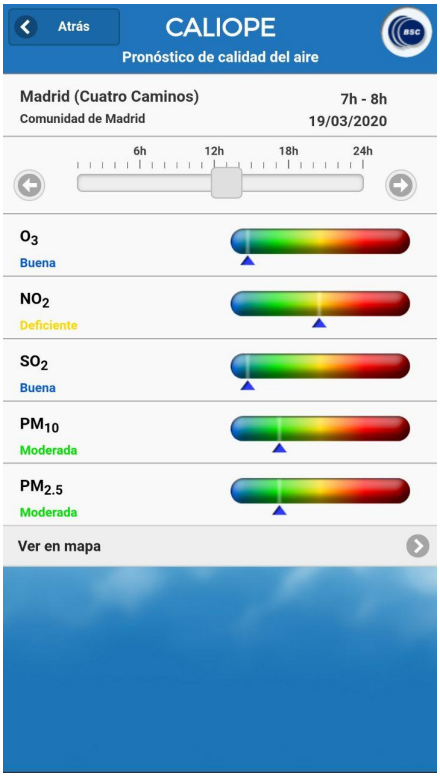


Figura 2.1: Captura de la aplicación de CALIOPE para Android mostrando la previsión a 13 horas vista para una estación seleccionada

estado del arte en cuanto a uso de técnicas de aprendizaje automático para modelar la calidad del aire. En él exponen, basados en estudios anteriores, que los modelos deterministas que intentan simular los fenómenos relacionados con la contaminación en el aire (conocidos como CTMs) –como el sistema CALIOPE–, tienen dificultades para capturar las relaciones no lineales entre la concentración de contaminantes, por un lado, y sus fuentes de emisión y dispersión, por otro. Es por esta razón, que los modelos que usan métodos de aprendizaje automático obtienen mejores resultados cuando se trata de hacer pronósticos en la calidad del aire.

En esta revisión distinguen diferentes tipos de modelos usados según son de estimación o de previsión. Según sea el contaminante que se quiera pronosticar o el objetivo de la predicción, determinan qué métodos de aprendizaje automáticos son los más adecuados para usar a partir de la literatura revisada.

En las conclusiones del estudio, confirman que estos modelos que usan técnicas estadísticas o de aprendizaje automático obtienen buenos resultados para la predicción de la calidad del aire, esto es especialmente notorio en los contaminantes de "partículas pequeñas": Particular Matter (Material Particulado en castellano) (PM). Además, funcionan aún mejor en condiciones meteorológicas extremas –la mayoría de estos modelos usan también datos meteorológicos–. No obstante, indican como un aspecto a mejorar, que no funcionan demasiado bien cuando se trata de predecir picos eventuales de alta polución.

Como muestra particular de lo anterior para el caso de Madrid, E. Pardo y N. Malpica

construyeron un modelo usando técnicas de Aprendizaje Automático (*Machine learning*) para predecir los valores de contaminación en puntos específicos, en particular, en aquellos puntos donde están situadas las estaciones desde las cuales se registran y publican los datos de calidad del aire (Pardo and Malpica [2017]).

De este modo, el modelo resultante necesita menos recursos que el costoso sistema CALIOPE y, además, obtiene unos mejores resultados en cuanto a la precisión del modelo en los puntos específicos sobre los cuales se aplicó el modelo. El estudio está limitado al contaminante NO_2, el cual es el más relacionado directamente con el tráfico rodado.

Sin embargo, el trabajo anterior se centra únicamente en la previsión de unas pocas estaciones sobre las que se ejecutó el entrenamiento y la predicción, y no sobre la ciudad completa de Madrid. Además, en la sección de trabajo futuro de esta publicación, los autores proponen posibles mejoras del modelo como añadir datos de tráfico o hacer uso de técnicas avanzadas para corregir y afinar mejor el modelo como "*dropout, batch normalization or data augmentation*".

2.2. El modelo LSTM

A continuación, veremos una serie de ejemplos e investigaciones previas que hacen un uso con muy buenos resultados del modelo Long Short-Term Memory Networks (LSTM) –o variaciones de éste– para la predicción de la calidad del aire, y que justifican su uso en trabajos como el presente.

Las redes neuronales recurrentes (RNN) son redes neuronales que disponen de mecanismos de propagación hacia atrás los cuales les permiten aprender elementos relevantes en series temporales. Sin embargo, estas redes tienen dificultades para extraer las dependencias relevantes de largo plazo (*long-term dependencies*) en series temporales largas, debido a que se pierden rápidamente por la desaparición del gradiente (*vanishing gradient problem*). Las redes neuronales LSTM son un tipo especial de redes neuronales recurrentes (RNN) que tienen una memoria la cual les permite recordar valores de entrada antiguos, lo que las hace más eficientes a la hora de aprender dependencias de largo alcance frente a las redes neuronales recurrentes estándar (Graves [2012]). Así pues, estas redes LSTM funcionan bien para el aprendizaje de series temporales de larga duración o estacionalidad.

El trabajo mencionado en la sección anterior de E. Pardo y N. Malpica (Pardo and Malpica [2017]) hace uso de estas redes LSTM para construir el modelo de predicción para la ciudad de Madrid que obtuvo mejores resultados que el sistema CALIOPE. Similarmente, otros trabajos de predicción de la calidad del aire han usado este tipo de modelo.

Li et al. [2017] hicieron una investigación usando una red LSTM Extendida para predecir los niveles de $PM_{2.5}$. El modelo es evaluado contra otros cinco diferentes modelos de aprendizaje

automático y el modelo LSTM Extendido fue el que mejor rendimiento obtuvo. Posteriormente, otras variantes del LSTM han obtenido muy buenos resultados para predecir niveles de $PM_{2.5}$, confirmando su utilidad en este área (Huang and Kuo [2018] Qi et al. [2019]).

En una investigación reciente en el contexto de Delhi (India), se usó LSTM para predecir los niveles de cada uno de los principales contaminantes en el aire (p.ej. O_3, $PM_{2.5}$, NO_x y CO) (Krishan et al. [2019]). En esta investigación el modelo usado obtuvo los mejores resultados para todas las predicciones comparados con los obtenidos por investigaciones previas. Además de la práctica habitual de incorporar datos meteorológicos al modelo, en este caso se incorporaron también datos de tráfico, los cuales se demostró que son un factor importante especialmente para la predicción de los niveles de NO_x.

Finalmente, cabe mencionar que las redes LSTM se han usado con éxito para predecir cuestiones relacionadas con el tráfico, como la velocidad del tráfico (Ma et al. [2015]).

Capítulo 3

Estudio de los datos

3.1. Introducción

Este capítulo está dedicado al estudio específico de los datos que se usan en el trabajo presente.

En primer lugar, se establecerán una serie de requisitos que servirán como criterio a la hora de elegir unos u otros conjuntos de datos. A continuación, se pasará a analizar las diferentes fuentes de datos disponibles. Después, se hará una selección debidamente justificada de qué fuentes de datos se usarán en este trabajo y, dentro de estas fuentes, qué magnitudes y puntos de medida serán los elegidos para los datos de entrada al modelo. En cuarto lugar, se desarrolla un análisis estadístico de los datos que se van a usar y se exponen las conclusiones extraídas de dicho análisis. Finalmente, se comenta la incidencia de datos inválidos o inexistentes y cuál ha sido la estrategia elegida para el tratamiento de estos casos.

3.2. Requisitos de los datos

Dado que el modelo que queremos tiene que aprender las características y los patrones en el tiempo que definen la contaminación en el aire, éste necesitará un gran conjunto de histórico de datos para usar como datos de entrenamiento y test.

Además, los datos deben ser fiables. Es decir, los datos deben estar validados y las estaciones de toma de esos datos calibradas correctamente para asegurarnos que los datos de referencia son válidos.

Como queremos dar una predicción de las próximas horas, la periodicidad mínima de los datos de entrada debe ser horaria.

Por último, necesitaremos filtrar el contaminante que queremos medir y también deberemos tener en cuenta la ubicación donde se han tomado las medidas de esos contaminantes.

3.3. Fuentes de los datos

3.3.1. Portal de datos abiertos del Ayuntamiento de Madrid

Los datos de calidad del aire que usaremos para nuestro modelo provienen de las estaciones de control de calidad del aire, los cuales están disponibles en el portal de datos abiertos[1] (*open data*) del ayuntamiento de Madrid.

Estas estaciones disponen de sensores específicos de alta precisión, los cuales son calibrados regularmente y verificados cada año por un equipo técnico especializado. En la figura 3.1 podemos ver uno de estos sensores.

Figura 3.1: Monitor HORIBA APCA-370, uno de los sensores utilizados en las estaciones de medio ambiente de Madrid. En este caso, se trata de un medidor de los niveles de Ozono.

En dicho portal de datos abiertos podemos encontrar: 1) datos respecto a la calidad del aire: histórico y en tiempo real; 2) datos meteorológicos: histórico y tiempo real; 3) datos de tráfico: histórico, tiempo real, mapas de intensidad y velocidad media.

El histórico de datos de calidad del aire disponible va desde 2001 hasta el año presente, y contiene los datos de todos los días desglosados por hora, contaminante y estación.

En cuanto a los datos históricos de meteorología, disponemos de datos por horas o por días de cada punto de medida con datos como: la velocidad y dirección del viento, la temperatura, la humedad, la presión, la radiación solar y las precipitaciones.

También disponemos de datos históricos de tráfico, procedentes de multitud de puntos de medida con datos como la cantidad de vehículos detectados (intensidad) o la velocidad media de los vehículos. Cabe mencionar que estos datos de tráfico son más difíciles manejar que otros conjuntos de datos, puesto que hay más de cuatro mil puntos de medida para la ciudad de Madrid, siendo además algunos de ellos móviles.

[1] El portal de datos abiertos del Ayuntamiento de Madrid se encuentra en la siguiente dirección web: https://datos.madrid.es/

3.3.2. AEMET

La Agencia Estatal de Meteorología (AEMET) también dispone de un portal de datos abiertos[2]. Para acceder a los datos de dicho portal hace falta solicitar una "clave API" (*API key*). El acceso se permite hacer tanto desde el navegador con una interfaz de usuario, como de manera programática a través de una API. En el caso de usar la interfaz de usuario, el resultado es un texto en formato JSON correspondiente a los datos solicitados.

Los datos históricos diarios que provee la AEMET comprenden valores de temperatura, precipitación, velocidad del viento, insolación, presión, entre otros. No obstante, al pedir el histórico de los datos registrados no se obtiene el histórico por horas sino con periodicidad de días. Solamente se puede obtener una periodicidad horaria si es de las últimas 24 horas (conocido como "último observado"), o mediante una solicitud explícita a la AEMET y pago de los costos que la agencia establezca para dicha solicitud.

También dispone de otros servicios como obtener la previsión meteorológica para el día actual (hoy) y para los próximos días, o consultar el archivo de previsiones pasadas.

3.3.3. Otras fuentes de datos

Otra fuente de datos que está cobrando cada vez más relevancia es la procedente de individuos particulares que instalan sensores de bajo coste en sus domicilios y de forma voluntaria ponen a disposición esos datos para uso público[3].

Un ejemplo para el caso de los datos meteorológicos es el portal Weather Underground, donde los usuarios con centrales meteorológicas domésticas muestran la meteorología registrada en su pequeña central. También existen sensores de bajo coste para registrar datos de la calidad del aire, especialmente para los contaminantes de Partículas Pequeñas (PM), aunque tienen todavía un precio elevado para la mayoría de ciudadanos.

Aunque existe ya algún estudio realizado a partir de los datos de este tipo de sensores, la poca madurez de las iniciativas, la escasez de un histórico amplio de datos y la fiabilidad aún por determinar de los datos obtenidos, hace que se haya decidido descartar esta fuente de datos para este trabajo.

3.4. Selección de los datos

De entre las fuentes de datos anteriores, se han seleccionado los conjuntos de datos que resultaran más adecuados para este trabajo.

[2]El portal de datos abiertos de la AEMET se encuentra en la dirección web: `https://opendata.aemet.es/`

[3]Se trata de un ejemplo del fenómeno conocido como *ciencia ciudadana*. Más información en el artículo de Wikipedia.

En primer lugar, se han seleccionado los datos de calidad del aire proporcionados por el Ayuntamiento de Madrid[4] relativos a una única estación de medición de la calidad del aire. Por simplificación, para los primeros experimentos se seleccionó la estación de calidad del aire con código 28079038, localizada en la glorieta de Cuatro Caminos. Dado que la estación anterior no dispone de ninguna estación cercana de observación meteorológica, a partir de la inclusión de datos meteorológicos los experimentos se realizaron sobre otra estación con código 28079039, localizada en el Barrio del Pilar. No obstante, todos los experimentos de los prototipos se podrían repetir fácilmente al resto de estaciones del ayuntamiento en caso de que se quisiera obtener una vista más general.

De entre todos los datos de contaminantes, este trabajo se ciñe al NO_2, por las razones ya adelantadas en la introducción de este documento (sección 1.1).

Resulta interesante tener en cuenta que la actividad en la ciudad de Madrid –y, por tanto, la emisión de contaminantes asociada a ésta– es poco estable durante el año: Experimenta una gran bajada durante los meses de verano o las vacaciones de semana santa, durante la semana hay mucha mayor actividad en los días laborables frente a los fines de semana y los días no laborables, y se suele experimentar una subida en periodos fríos. Además, los valores medidos no son iguales en todos los años. Durante los años 2010 y 2011 hubo una disminución debido a los efectos de la crisis económica, y desde entonces la tendencia ha sido a la alza[5].

Dado que se va a trabajar con datos horarios, para el resto de datos se necesitan también datos con frecuencia, al menos, horaria. Los datos meteorológicos a los cuales se han podido tener acceso desde este trabajo a través del portal de la AEMET solo tienen frecuencia diaria, por lo que no servirían.

Así pues, para los datos meteorológicos se usaron de nuevo los disponibles en el portal de datos abiertos del Ayuntamiento de Madrid, aunque, por desgracia, solo están disponibles desde 2019[6]. Para obtener el registro de estos datos lo más acorde posible a los datos de calidad del aire medidos, se ha escogido la estación meteorológica de Peñagrande (código de la estación: 28079108), que dista alrededor de unos 700 metros de la estación de calidad del aire del Barrio del Pilar (código de la estación: 28079039). En la figura 3.2 se puede ver un mapa donde están representadas las dos estaciones.

Siguiendo los trabajos anteriores de Pardo and Malpica [2017], Li et al. [2017], Zhang et al. [2020], de entre los datos meteorológicos disponibles se han escogido los correspondientes a: la

[4]Los datos de calidad del aire usados están disponibles en el enlace al conjunto de datos 1 dentro del anexo A.1.

[5]Se puede leer más información sobre la tendencia de los valores de contaminantes en los últimos años, así como una caracterización más detallada de éstos para la ciudad de Madrid, en los informes anuales de la organización "Ecologistas en Acción", como por ejemplo el Informe de la calidad del aire en la ciudad de Madrid durante 2019

[6]Los datos meteorológicos usados están disponibles en el enlace al conjunto de datos 2 dentro del anexo A.1.

Figura 3.2: Mapa donde se muestran la distancia entre la estación de meteorología escogida (punto morado) y la estación de calidad del aire escogida (punto verde)

velocidad del viento, la dirección del viento, la temperatura, la humedad relativa, la precipitación y la presión atmosférica.

Estos datos meteorológicos son todos numéricos, exceptuando la dirección del viento, que viene expresada en grados centígrados indicando la orientación cardinal del viento. Para discretizar este atributo y también reducir su dimensionalidad[7], hice una conversión consistente en dividir los 360 grados en 8 franjas iguales de 45° cada una, correspondientes a los puntos cardinales: Norte, Noreste, Este, Sureste, Sur, Suroeste, Oeste y Noroeste.

3.5. Análisis estadístico de los datos

En esta sección se muestra un resumen de las conclusiones de los análisis exploratorios que se han efectuado sobre los datos usados en este proyecto.

3.5.1. Datos de calidad del aire

En la tabla 3.1 se puede ver un ejemplo de la distribución de los datos de NO_2 para el intervalo comprendido entre febrero y mayo de 2019. Nótese que la magnitud usada para medir las concentraciones de NO_2 son los $\mu g/m^3$.

[7]Más información sobre discretización y reducción de dimensionalidad de variables en Sangüesa i Solé, 2010b, Capítulos 3 y 4.

	NO2
count	2880,00
mean	35,65
std	31,20
min	1,00
25 %	13,00
50 %	25,00
75 %	49,00
max	211,00

Cuadro 3.1: Distribución de los datos de NO_2 usados

Se puede observar que se trata de datos con una gran variabilidad y que pueden ir desde 1 hasta más de 200 µg/m^3. Haciendo un diagrama de cajas se ve, de hecho, que el conjunto de datos tiene muchos datos extremos por arriba, muy alejados de la media aritmética, y que la diferencia entre la media y la mediana es más pronunciada en los valores por encima de la media. No pasa así en cuanto a valores pequeños, es decir, alejados de la media por abajo.

En la gráfica de la figura 3.3 se puede constatar más claramente esta gran variabilidad. Además, en la gráfica se observa que se producen muchos vaivenes de subida y bajada en poco tiempo, y se observan grandes picos de subida. En el siguiente apartado se estudiará más a fondo las características de esta serie temporal como la tendencia o la estacionalidad.

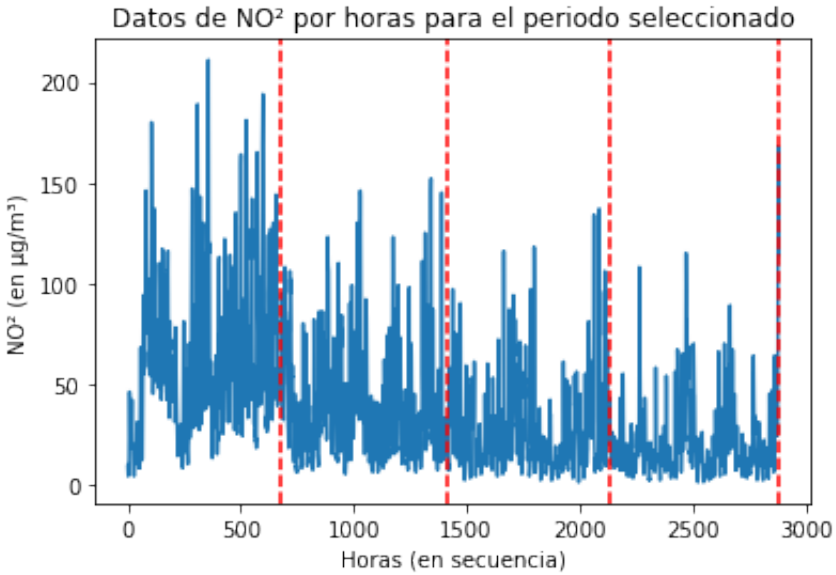


Figura 3.3: Gráfica mostrando la evolución de los datos NO_2 entre los periodos de febrero y mayo de 2019. Las barras verticales discontinuas separan los distintos meses de febrero, marzo, mayo y junio.

El análisis estadístico y visual de los datos de calidad del aire se ha realizado en un *notebook* de Python disponible en el siguiente enlace 1 dentro del anexo A.2.

3.5.1.1. Análisis de la serie temporal

En el apartado anterior se mostró un análisis descriptivo de los datos y, a continuación, se muestra un análisis de la serie temporal, basado en la teoría de Casas Roma [2020].

Antes de todo, se evalúa si la serie temporal es, en efecto no estacionaria[8], tal y como parece. Para ello, se usa el test de Dickey-Fuller aumentado [9]. Los resultados del test dicen que con muy alta seguridad, los datos tienen una raíz unitaria y, por tanto, no son estacionarios.

Con este resultado se puede afirmar que no se va a poder descomponer perfectamente la serie temporal en características estacionarias. No obstante, a continuación se describe y realiza igualmente un estudio de ésta, ya que resulta de utilidad para entender mejor los datos.

En primer lugar, se analiza la tendencia general. Previamente en la figura 3.3 se podía apreciar cierta tendencia a la baja en los datos. Para estudiar esta tendencia de manera más analítica, se ajusta un modelo de regresión lineal a los datos, el cual produce una recta con pendiente negativa, lo cual confirma efectivamente la tendencia descendiente de los datos en los meses seleccionados. Posiblemente esto es debido a que la tendencia en el uso del transporte rodado es mayor en los meses fríos que en los meses más primaverales.

La siguiente característica interesante consiste en averiguar si hay cierta estacionalidad en los datos y cuál es su periodicidad. Para ello, se realiza un análisis mediante coeficientes de autocorrelación entre los datos horarios, buscando correlación entre todas las posibles horas que hay en un mes. Este análisis reporta que tenemos cierta estacionalidad diaria (cada 24 horas); y, con menos significación, estacionalidad semanal (cada 168 horas)[10].

Por último, se puede ver cómo quedan los residuos una vez quitado el patrón de estacionalidad de los datos definidos cada 24 horas. El resultado, mostrado en la figura 3.4, es una serie temporal muy parecida a la original, con todavía mucho ruido y muchísima variabilidad. Así pues, se puede concluir que la serie temporal es muy difícil de predecir y que sería inviable realizar los pronósticos usando técnicas de la estadística clásica.

Este análisis de la serie temporal ha sido realizado en un *notebook* de Python disponible en el enlace 2 dentro del anexo A.2.

[8] La definición sobre los procesos estacionarios puede encontrarse en el artículo de Wikipedia: https://es.wikipedia.org/wiki/Proceso_estacionario

[9] Para la realización de este test usé la implementación de la librería de Python *statsmodels*.

[10] No obstante, también se observó una alta correlación para las horas inmediatamente anteriores (1, 2 ó 3 horas), este es un resultado esperable dada la periodicidad y naturaleza de los datos, pero no aporta mucho en lo que a la estacionalidad de los datos se refiere. También se obtuvo una correlación relativamente alta con los dos días anteriores (48 horas) y con las 12 horas anteriores.

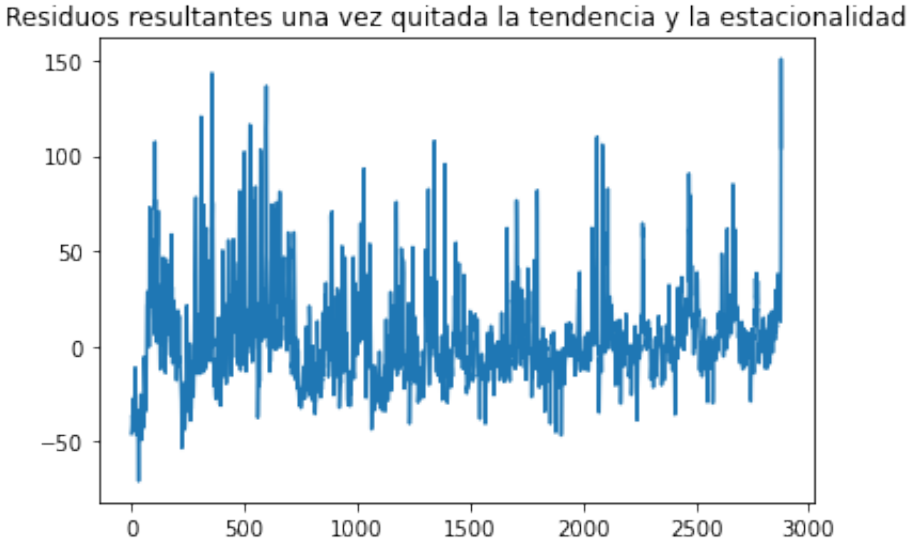


Figura 3.4: Gráfica mostrando como quedan los datos tras quitarle la tendencia y la estacionalidad cada 24 horas.

3.5.2. Datos de meteorología

En la tabla 3.2 se puede ver un ejemplo de la distribución de los datos meteorológicos que se han utilizado para el intervalo comprendido entre febrero y mayo de 2019.

magnitud	hora	vel_viento	dir_viento	temp	humedad	presion	precp
count	2880,00	2880,00	2880,00	2880,00	2880,00	2880,00	2880,00
mean	12,50	1,06	174,85	10,78	42,75	943,54	0,03
std	6,92	0,72	87,83	6,23	23,26	6,00	0,26
min	1,00	0,00	1,00	-5,20	1,00	800,00	0,00
25 %	6,75	0,53	90,00	6,10	25,00	940,00	0,00
50 %	12,50	0,93	199,00	10,20	41,00	944,00	0,00
75 %	18,25	1,42	254,00	14,90	59,25	948,00	0,00
max	24,00	4,72	359,00	30,80	100,00	953,00	5,00

Cuadro 3.2: Distribución de los datos meteorológicos usados

A partir del análisis estadístico de los datos realizado se pueden sacar las siguientes conclusiones:

1. El viento suele ser estable y no se observa ninguna tendencia particular. Aunque a veces aparecen rachas de mucho viento.

2. El viento coge todas las direcciones cardinales posibles y parece haber cierta tendencia o patrón temporal en la evolución de la dirección del viento.

3. La mayoría de las temperaturas oscilan entre los 5 y 15 ºC, cambiando según avanza el día y da paso a la noche. Si bien, hay valores bastante extremos, más frío: -5 ºC y más caliente: 30 ºC.

4. No se observan muchas precipitaciones y en general la temporada fue más seca que húmeda.

5. Se observa, como cabría esperar, que según va avanzando la primavera hay un aumento de las temperaturas y un ambiente más seco (menor humedad)

6. La presión es relativamente constante y se mantiene en un intervalo entre los 940 y los 950 mb. Tenemos un único valor atípico que desciende de forma brusca hasta los 800 mb, dicho valor requeriría de un estudio adicional puesto que posiblemente se trate de un error o medida mal tomada.

El análisis estadístico y visual de los datos meteorológicos se ha realizado en un *notebook* de Python disponible en en el siguiente enlace 3 dentro del anexo A.2.

3.6. Tratamiento de datos no válidos o perdidos

En los conjuntos de datos seleccionados existen algunos datos no verificados, que –como indican los documentos de interpretación para estos conjuntos de datos– no son válidos y, por tanto, no se deben usar en investigaciones o análisis. Afortunadamente, dentro de los datos utilizados para este trabajo hay muy pocos datos de este tipo (menos del 1 % del total). En algunas ocasiones también ocurre, aunque de nuevo con muy poca frecuencia, que en contadas fechas hay datos perdidos o inexistentes en ciertas estaciones de medida.

Existen varias estrategias para lidiar con este tipo de datos[11]. En este caso, resulta primordial mantener la frecuencia de los datos, es decir, que siempre haya un dato cada hora y, por tanto, no es buena solución simplemente borrar esos datos. Tampoco es buena solución dejar el dato como valor nulo puesto que muchas de las operaciones usadas no permiten que haya valores nulos entre los datos. Sumado a lo anterior, debemos considerar que se trata de pocos datos y que además, con mucha frecuencia, se trata de datos parecidos a los anteriores. Por tanto, se ha decidido tomar la estrategia de rellenar dichos datos perdidos con el último valor anterior que hubiera válido.

Finalmente, aparte de estos datos no verificados o perdidos se realizó un análisis para investigar la existencia de algún dato con valores atípicos. Estos datos atípicos son datos con valores no esperados o fuera de la escala que les corresponde, como por ejemplo un valor negativo o cero

[11] Para el estudio de algunas de ellas véase [Han et al., 2011, capítulo 3: Data Preprocessing, págs 88-89]

donde no debería o un único dato muy distanciado del resto de valores de la misma variable. En general, no se han encontrado muchos valores de este tipo, aunque sí han aparecido algunos casos donde, a pesar de estar marcar los datos como "verificados", la humedad relativa era inferior a 0 y la temperatura marcaba -55 ºC; o el caso aislado de la presión atmosférica de un valor muy inferior al resto, comentado anteriormente en la sección 3.5.2. También en estos casos se ha seguido la misma estrategia anterior: Rellenar estos datos con el último valor válido registrado.

Capítulo 4

Diseño e implementación del modelo predictivo

4.1. Introducción

Este Trabajo de Fin de Máster (TFM) consiste en la elaboración de un modelo predictivo que, dado un histórico de la concentración de un contaminante (NO_2), prediga el valor de este contaminante para la siguiente hora u horas. Se trata pues de un problema del pronóstico de una serie temporal univariante[1]. Para este tipo de problemas existe una amplia gama de estrategias que se pueden aplicar usando diferentes técnicas de aprendizaje automático (Bontempi et al. [2013]).

Como previamente se mostró en el análisis del Estado del Arte (Capítulo 2), los modelos de aprendizaje automático han presentado buenos resultados en la predicción de series temporales, especialmente en el caso del pronóstico de la calidad del aire. Entre ellas, destacan en particular las redes neuronales usando implementaciones como las redes recurrentes y, más en concreto, las redes LSTM.

Estas redes aprenden a base de estudiar la evolución de los datos organizados en forma de secuencias y son capaces de predecir los siguientes valores de las secuencias. En nuestro caso, las secuencias estarán compuestas por las medidas registradas en el histórico de las horas antes del contaminante a predecir. Además, se pueden añadir otras variables adicionales que puedan aportar información valiosa y mejorar la capacidad de predicción del modelo. De este modo, todos los modelos que se han desarrollado siguen el esquema general mostrado en la figura del anexo B.1. La implementación de estos modelos se ha hecho usando *keras*[2], la API más popular

[1] Se denomina univariante a las predicciones temporales que se hacen sobre un solo valor.

[2] Keras es una librería de alto nivel para hacer experimentos de aprendizaje profundo o *deep learning* en inglés (Chollet et al. [2015])

para la implementación de modelos de aprendizaje profundo.

Como es práctica habitual cuando se trabaja con redes neuronales, se ha usado un conjunto de entrenamiento, otro de validación y, finalmente, otro de prueba o *test*. El conjunto de entrenamiento sirve como entrada de muestra para que la red neuronal "aprenda". El conjunto de validación sirve para evaluar qué tal se comporta la red y cambiar la configuración de sus parámetros en aras de mejorar su desempeño. Finalmente, el conjunto de test se usa posteriormente para evaluar la calidad del modelo.

Como trabajo previo al entrenamiento y evaluación de los diferentes modelos, se han tenido que realizar las pertinentes labores de carga, selección, limpieza y conversión de los datos a un formato adecuado, las cuales fueron comentadas previamente en la sección 3.4 de este documento.

Para convertir los datos históricos a un problema de aprendizaje supervisado, se ha aplicado la técnica conocida como "Método de la ventana deslizante", descrita con detalle en el recurso [Casas Roma, 2020, páginas 23-25]. En resumen, este proceso consiste en crear secuencias de cierta longitud w para cada instante de tiempo t y usar los $w-1$ primeros valores de la secuencia, $H_1, ...H_{w-1}$, como variables conocidas (X) y el último valor, H_w, como el valor objetivo a predecir (y).

La elección de la anchura de la ventana, w, puede ser crítica para el buen funcionamiento del modelo. En mi caso he elegido 24h, puesto que es el periodo de estacionalidad más razonable que encontré durante el análisis de series temporales previo (explicado en el apartado 3.5.1.1 de este documento).

La métrica principal usada para evaluar la calidad del modelo es la conocida como Round Mean Square Error (RMSE). Esta métrica es la que habitualmente se ha usado por trabajos similares para evaluar la bondad de sus resultados (Pardo and Malpica [2017], Baldasano et al. [2011], Li et al. [2017]). Aunque de manera alternativa también se ha incluido en los análisis la métrica Mean Absolute Percentage Error (MAPE), la cual mide en términos relativos el error entre el valor predicho y el valor observado, y es menos sensible a grandes diferencias entre las predicciones y los valores reales que la métrica RMSE. Nótese que para ambas métricas, los resultados son mejores cuanto menores sean sus valores.

Las siguientes secciones describen, de forma pormenorizada, el diseño y desarrollo de cada uno de los modelos de predicción que se han desarrollado y optimizado en el marco de este Trabajo de Fin de Máster.

En los primeras secciones se realizan predicciones con los diferentes modelos a una distancia de una hora, es decir, para el siguiente elemento de la secuencia. Dentro de la tipología de predicción de series temporales, estas predicciones son conocidas como *One-step forecast*. A partir de la sección 4.6 se realizan también predicciones a 8, 16 y 24 horas de distancia desde

el último valor de la secuencia. Éstas últimas corresponden a problemas de tipo *Multi-step forecast*.

Finalmente, en el anexo B se encuentran los diagramas correspondientes a los diseños de todos los modelos y prototipos realizados en este capítulo.

4.2. Modelos “base” de comparación

Antes de nada, para poder evaluar el rendimiento de los subsiguientes modelos se necesita una base de referencia sobre la que compararlos. Para este fin, se implementaron dos modelos: un primero sencillo e ingenuo, que simplemente se basa en usar el instante anterior como valor de predicción para el instante siguiente; y un segundo usando la técnica de las Máquinas Vector Soporte –en inglés Support Vector Machines (SVM)–. Estas SVM se pueden adaptar para resolver problemas de regresión, en cuyo caso son conocidas como SVR.

4.2.1. *Persistence Model Forecast* (PMF)

En primer lugar, implementé un modelo ingenuo como primer modelo base conocido como *“Persistence Model Forecast (PMF)”* o, traducido al castellano, Modelo Predictivo de Persistencia. El modelo simplemente coge el valor anterior registrado y lo usa como valor de predicción para el instante siguiente.

Su implementación se puede resumir en el código mostrado a continuación:

```
history = [x for x in train]
predictions = list()
for i in range(len(test)):
    # coger como prediccion el valor inmediatamente anterior
    predictions.append(history[-1])
    # guardar al final el siguiente valor
    history.append(test[i])
```

Listado 4.1: Código Persistence Model Forecast

Así pues, las gráficas de predicción contra datos reales registrados por este modelo tienen la forma mostrada en la figura 4.1.

Tras evaluar este modelo simple con los datos de contaminación de septiembre de 2019, siendo el conjunto de test del 20 % y el entrenamiento de un 80 %, el modelo obtiene una tasa de error RMSE de $\mathbf{19.205\,\mu g/m^3}$.

Usando los datos del intervalo de cuatro meses comprendido entre febrero y mayo de 2019, siendo siendo el conjunto de test del 10 % y el entrenamiento de un 90 %, el modelo obtiene un RMSE de $\mathbf{12.64\,\mu g/m^3}$.

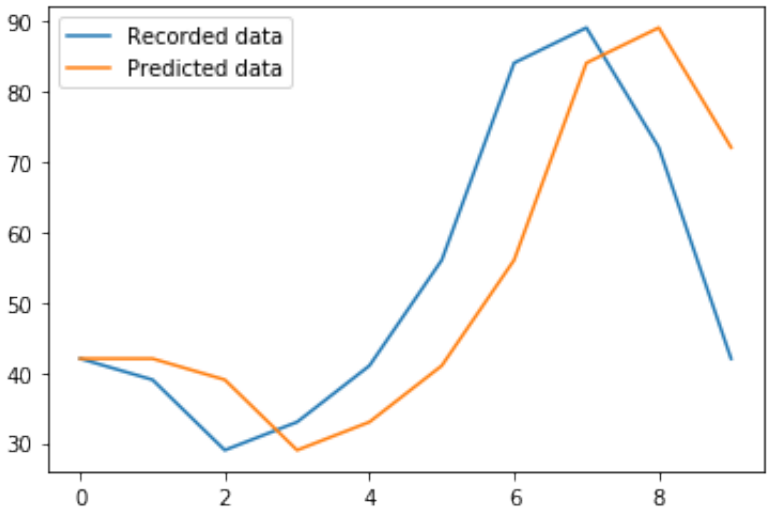


Figura 4.1: Extracto de las predicciones del modelo predictivo de persistencia (naranja) vs datos reales (azul)

4.2.2. Modelo *Support Vector Regression* (SVR)

El segundo modelo base aplica la técnica de los Vectores Soporte de Regresión (SVR por sus siglas en inglés). Su implementación hace uso de la librería *sklearn*, que ya incluye entre sus modelos disponibles el SVR, y se dio unos valores estándar a sus parámetros.

Usando los mismos conjuntos de entrenamiento y test usados para el modelo anterior se obtiene: Un RMSE de $\mathbf{29.31\,\mu g/m^3}$ con los datos de Septiembre de 2019 y un RMSE de $\mathbf{11.727\,\mu g/m^3}$ con los datos de febrero a mayo de 2019.

El código relativo a los dos modelos base anteriores se puede encontrar en el enlace 4 dentro del anexo A.2.

4.3. Prototipo 0: Predicción usando el histórico de datos

El primer prototipo de predicción que usa redes neuronales ha sido titulado como "prototipo 0". Este prototipo consiste básicamente en un modelo que predice el valor de contaminación de NO_2 que habrá en la siguiente hora, únicamente a partir de los valores registrados de las 23 horas anteriores para una única estación dada.

Se trata, sobre todo, de un modelo con un diseño simple pero potente, que sirve como primera aproximación al problema en cuestión de predicción de la calidad del aire y a realizar todas las tareas que comprende un proyecto de minería de datos[3]. En este primer prototipo, la

[3]Las diferentes etapas y tareas involucradas en un proyecto de minería de datos están explicadas con mayor profundidad en Sangüesa i Solé [2010a].

entrada consiste únicamente en datos históricos y la salida es la predicción del modelo para la siguiente hora. El diseño a alto nivel de este modelo se puede ver en el diagrama del anexo B.2.

Para la primera iteración de este prototipo, se usaron únicamente los datos horarios de NO_2 relativos al mes de septiembre de 2019. En la segunda y siguientes iteraciones, los experimentos usaron datos de cuatro meses: febrero, marzo, abril y mayo de 2019.

A continuación, se detallan con más detalle estas iteraciones, el proceso seguido y los hallazgos comprendidos en cada una de ellas.

4.3.1. Treinta días de entrenamiento

En esta primera iteración se usaron únicamente los datos horarios pertenecientes a los treinta días relativos al mes de septiembre de 2019. Por tanto, se trata de 720 datos horarios medidos, los cuales forman –una vez partidos en tres conjuntos disjuntos de entrenamiento, validación y test– un total de 648 secuencias con una longitud de 24 valores cada una.

El tamaño del conjunto de test es del 20 % del total de los datos, el de validación de un 20 % dentro del subtotal restante, quedando el 64 % restante para el conjunto de entrenamiento.

Esta es la primera iteración y el primer acercamiento al problema usando redes neuronales profundas. Debido a esto, esta iteración abarcó muchos días dedicados a escribir y refactorizar código, experimentar y observar cómo afectan los diferentes hiperparámetros al comportamiento de la red y al resultado final. Los hiperparámetros susceptibles de estos experimentos son: el ratio de aprendizaje (*learning rate*), el número de neuronas para cada capa de la red LSTM (*units*), el tamaño del lote (*batch size)*, el número de etapas de entrenamiento (*epochs*) y la función de optimización (*optimizer*). Dentro de esta iteración también se hicieron pruebas consistentes en cambiar otros factores relativos a la definición de la red, como por ejemplo el número de capas de la red LSTM, el uso de una red LSTM *stateful* o la longitud de la secuencia.

Después de estos ciclos de experimentación-observación, durante estos experimentos se obtuvieron diversos resultados de RMSE, variando desde los **22** hasta los **28** o incluso **30 µg/m³**. Las configuraciones que obtuvieron los mejores resultados consistían en redes tupidas con muchas neuronas, pero que, al mismo tiempo, tardaban un tiempo en ejecutarse desproporcionado en comparación a su mejoría.

El código relativo a estas primeras pruebas se puede encontrar en el enlace 5 dentro del anexo A.2.

4.3.1.1. Optimización de hiperparámetros con Talos

Dado el enorme espectro de posibles combinaciones de valores para los parámetros, probar todas las combinaciones y evaluar su resultado de manera manual es una tarea harto tediosa.

Para solventar este inconveniente se hizo uso de la librería externa "Talos"[4], la cual automatiza de manera sencilla la evaluación de las distintas combinaciones de parámetros. Por ejemplo, una posible configuración de hiperparámetros a optimizar sería la mostrada en el listado 4.2. Sobre esta configuración, Talos evaluará todas las combinaciones posibles y facilitará su análisis posterior [5].

```
params = {
    'units_first_layer': [50, 100, 1000],
    'units_second_layer': [100, 1000],
    'optimizer': ['rmsprop', 'adam'],
    'batch_size': [1, 24, 120, 216],
    'epochs': [ 10],
}
```

Listado 4.2: Hiperparámetros a optimizar con Talos

A través del análisis de estos experimentos, se pudo constatar que el optimizador Adam daba mejores resultados que el optimizador RMSProp. La configuración con los mejores resultados de entre todas las combinaciones probadas, usando 10 etapas de entrenamiento (*epochs*), es la mostrada en el listado 4.3.

```
params = {
    'units_first_layer': 50,
    'units_second_layer': 100,
    'optimizer': 'adam',
    'batch_size': 1
}
```

Listado 4.3: Mejor configuración obtenida con Talos

El resultado de RMSE para el prototipo 0 con esta configuración de parámetros para los 30 días de datos es de $\mathbf{23.250\,\mu g/m^3}$.

En la gráfica de la figura 4.2, se muestran las predicciones hechas por el modelo frente a los datos reales. En ella se puede observar que el modelo sigue bastante bien la silueta media habitual, pero funciona mal cuando aparecen picos altos de contaminación.

El código relativo a los experimentos con Talos se puede encontrar en el enlace 6y el relativo al prototipo 0 usando la mejor configuración de parámetros mencionada anteriormente está en el enlace 7, ambos dentro del anexo A.2.

[4]Talos es una librería de software libre publicada por la organización Autonomio. (2019). Disponible en: https://github.com/autonomio/talos.

[5]Como nota informativa, cabe añadir que para estos experimentos se normalizó los datos de entrada, lo cual no se ha hecho en otras iteraciones del modelo.

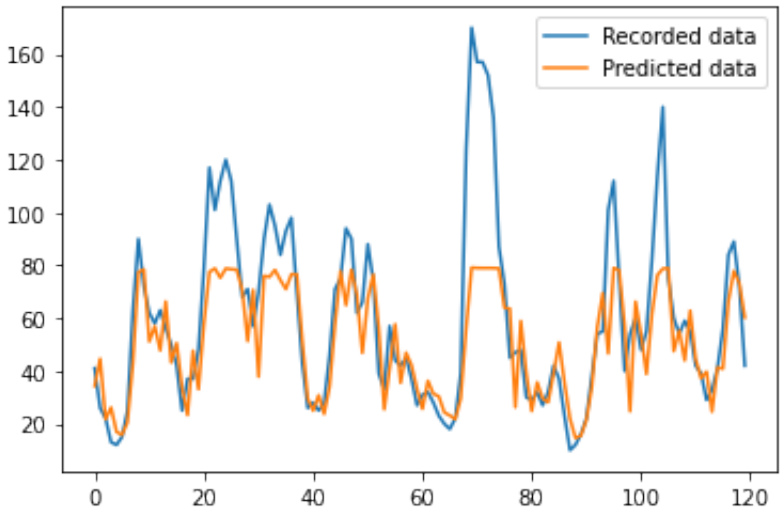


Figura 4.2: Predicciones del modelo prototipo 0 (naranja) vs datos reales (azul) para el conjunto de test - Iteración 1

4.3.1.2. LSTM stateful

Una variante de la red LSTM es usar la opción *"stateful"*, la cual hace que el modelo no aprenda con cada secuencia proporcionada por separado (modalidad *"stateless"*), sino que mantiene la memoria entre secuencias. De este modo, el modelo es capaz de recordar y reconocer patrones pasados durante mucho más tiempo que el involucrado en la longitud de la secuencia.

Sin embargo, los inconvenientes de usar este tipo de modelo son varios:

1. Los datos (secuencias en nuestro caso) deben estar ordenados cronológicamente todo el tiempo, lo cual implica que no se puedan mezclar aleatoriamente en ningún momento

2. En keras, el entrenamiento se hace a lo largo de varias etapas incrementales, llamadas *epochs*. Cuando se usa la modalidad stateful, el proceso de entrenamiento por varias etapas debe ser implementado por el propio usuario mediante un bucle. Esto hace que el análisis posterior del entrenamiento también deba ser implementado por el mismo usuario.

3. No siempre la capacidad de recordar datos antiguos es necesaria. Si, por ejemplo, tenemos un periodo de correlación cada 24 horas y las secuencias introducidas tienen ya 24 valores de longitud, el hecho de recordar sucesos anteriores puede hacer que la red sea más "torpe" a la hora de aprender los patrones que ocurren cada 24 horas.

Después de probar varias posibles configuraciones usando la modalidad *stateful* implementada en este trabajo, se obtuvo un resultado de $\mathbf{25.707\,\mu g/m^3}$ después de 100 etapas de entrenamiento. Este resultado es peor que el obtenido mediante la modalidad *stateless* con los mismos parámetros.

Se podría estudiar la influencia del sobreentrenamiento y, posiblemente, reducir dicho valor de error usando menos etapas de entrenamiento. Sin embargo, dados los inconvenientes anteriores y debido a que el desempeño obtenido no es muy superior a los obtenidos previamente, se decidió descartar esta modalidad en lo sucesivo. Además, como se pudo ver en la figura 4.2, donde el modelo realmente tiene dificultades para predecir con exactitud es en los picos inesperados de contaminación, los cuales no están sujetos a ciertos patrones o ciclos de largo recorrido.

El código relativo a este modelo *stateful* se puede encontrar en el enlace 8.

4.3.1.3. Resultados preliminares

Hasta ahora el mejor resultado obtenido, $\mathbf{23.250\,\mu g/m^3}$, es mejor que el obtenido por el modelo base SVR, pero todavía peor que el obtenido por el modelo ingenuo PMF. Lo cual indica que aún hay mucho margen de mejora.

4.3.2. Cuatro meses de entrenamiento

Los experimentos realizados en las primera iteración de experimentos con este prototipo, sirvieron para aprender cómo afectan cada uno de los valores de los diferentes hiperparámetros que usa el modelo. A continuación, se realizó una segunda tanda de iteraciones usando un conjunto de datos mayor, de modo que los resultados obtenidos son más representativos y la red dispone de más variedad sobre la que aprender.

Para ello, se seleccionó el rango de meses comprendidos entre febrero y mayo de 2019. Esta elección está motivada por tratarse de unos meses que no están muy influenciados por periodos vacacionales largos como pueden las vacaciones estivales y navideñas, cuyo acontecimiento afecta de manera determinante a los valores de calidad del aire en el caso de la ciudad de Madrid. En total, estos cuatros meses representan 2880 datos horarios.

Las proporciones de los diferentes conjuntos de entrada son: un 10 % del total de los datos para el conjunto de test, por un lado, y el otro 90 % restante dividido entre el conjunto de entrenamiento y el conjunto de validación. Del anterior 90 %, un 90 % de aquél se destina al conjunto de entrenamiento (quedando un 81 % del total), y el otro 10 % del restante al conjunto de validación (un 9 % del total). Al descomponerse en secuencias tenemos: 265 secuencias para el conjunto de test (equivalente a unos 11 días), 2310 secuencias para el conjunto de entrenamiento (aproximadamente 3 meses y una semana) y 236 secuencias para el conjunto de validación (equivalente a entre 9 y 10 días). Estos serán los datos de entrada y evaluación del modelo.

4.3.2.1. Optimización de hiperparámetros

Para la elección de los hiperparámetros, se comenzó con la mejor configuración obtenida en la iteración anterior (listado 4.3). Sin embargo, al estudiar las gráficas de entrenamiento, enseguida se observó que el modelo nunca llegaba a sobre entrenarse con los datos de entrenamiento (*overfitting*). Es decir, que no era capaz de aprender correctamente todo el espectro de posibles secuencias contenido en estos 3 meses de datos. Esto significa que el modelo necesita de mayor complejidad para poder ajustarse mejor a las posibles secuencias ([Chollet, 2017, Capítulo 5]).

Así pues, se aumentó la complejidad de la red añadiendo más neuronas a las capas de la LSTM. Después de probar con diferentes valores, se encontró el óptimo usando 128 neuronas en ambas capas. Similarmente, se descubrió que el modelo también mejoraba notablemente al aumentar el *batch size* a 16 y también se hicieron pruebas con diferentes combinaciones de *learning rate*. De este modo, la mejor combinación de parámetros conseguida es la mostrada a continuación:

```
params = {
    'units_first_layer': 128,
    'units_second_layer': 128,
    'optimizer': optimizers.Adam,
    'batch_size': 16,
    'epochs': 100,
    'lr': 0.001
}
```

Listado 4.4: "Mejores hiperparámetros para el prototipo 0 con cuatro meses de entrenamiento"

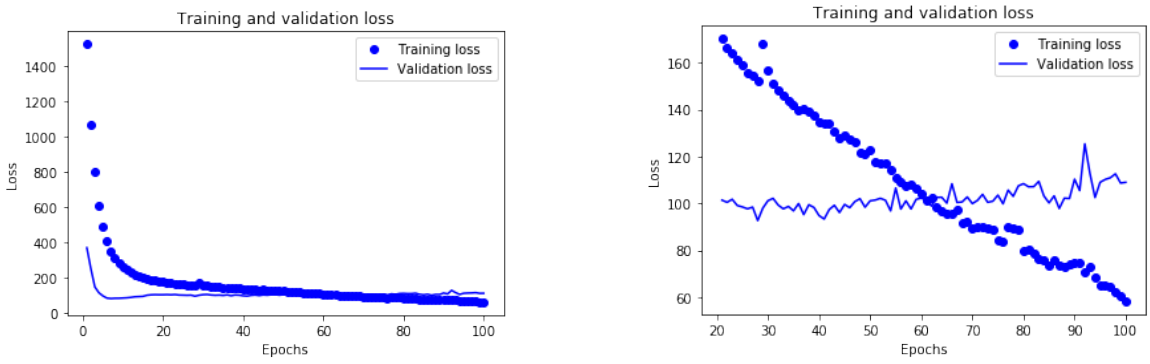


Figura 4.3: Gráficas de la evolución del error durante el entrenamiento del modelo. En la izquierda se muestran todas las etapas y a la derecha la evolución a partir de la etapa 20.

En la figura 4.3 se puede observar la evolución del error que tiene el modelo en cada *epoch*, usando los nuevos parámetros definidos en el listado 4.4. Como se puede observar, los mejores valores se dan lugar entorno a las *epochs* 60 y 70, donde se cruzan las líneas de validación y de

entrenamiento (punto conocido como *robust fit*). Después de esto, se produce el fenómeno de *overfitting*: el modelo va mejorando sus resultados para el conjunto de entrenamiento, pero no así para el conjunto de validación.

4.3.2.2. Resultados preliminares

Para obtener el resultado final, el proceso de entrenamiento se para en el punto anteriomente mencionado de *robust fit*. En este punto, el modelo ya deja de mejorar sus resultados con respecto al conjunto de validación. De este modo, los resultados obtenidos para las métricas son los siguientes: un RMSE de $\mathbf{10.46\,\mu g/m^3}$ y un MAPE de $\mathbf{28.68\,\%}$. Como se puede ver, estos resultados son mucho mejores que los obtenidos mediante los datos de un solo mes de entrenamiento, y son significativamente superiores a los obtenidos por los dos modelos base, que habían obtenido un RMSE de $\mathbf{12.64\,\mu g/m^3}$ y $\mathbf{11.727\,\mu g/m^3}$ para los casos del PMF (sección 4.2.1) y del SVR (sección 4.2.2) respectivamente.

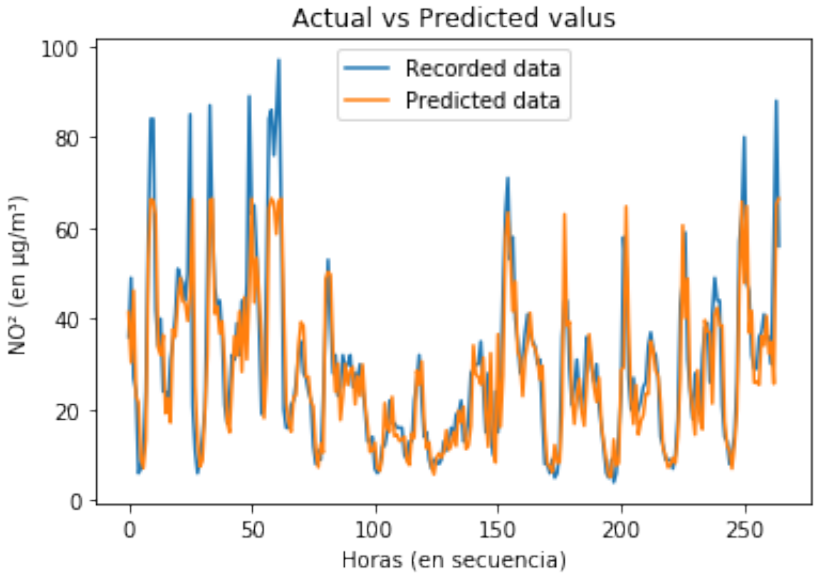


Figura 4.4: Predicciones del modelo prototipo 0 (naranja) vs datos reales (azul) para el conjunto de test usando datos de cuatro meses - Mejor iteración

En la gráfica de la figura 4.4 podemos observar como todavía donde más falla el modelo es en las predicciones de picos altos de polución.

El *notebook* con el código correspondiente a la mejor iteración para este prototipo 0 con 4 meses está en el enlace 9.

4.3.3. Resultados prototipo 0

Durante el desarrollo de este primer prototipo, se realizaron muchas tareas de análisis y experimentación hasta conseguir la mejor configuración para los hiperparámetros.

Usando únicamente datos históricos, con apenas cuatro meses de historial y sin hacer uso todavía de otras técnicas más avanzadas como el *Dropout* o el *batch normalization*, se obtuvieron resultados muy satisfactorios, superando a los modelos base introducidos en la sección anterior.

A partir de la inspección de las gráficas de predicción, se observa que donde el modelo comete los mayores errores es a la hora de predecir grandes picos de contaminación hacia arriba. Esto se había adelantado ya en el estudio del Estado del Arte realizado en el Capítulo 2, y es consistente con la revisión sobre modelos de aprendizaje automático aplicados a la predicción de la calidad del aire hecha por Rybarczyk and Zalakeviciute [2018].

Cabe notar que para los conjuntos de validación y test se han tomado los últimos trozos de 10 % de los datos correspondientes. Esto tiene el problema de que, de manera casual, coincida que las fechas sobre las cuales se evalúa el modelo tengan un comportamiento muy diferente a los valores habituales. Una manera de solventar este problema sería tomar estos conjuntos de forma aleatoria, o incluso usar el método del *k-fold validation*, para hacer una evaluación más precisa del modelo.

Una opción que no se ha comentado es la posibilidad de hacer la red LSTM bidireccional. Esto es, que no considere solamente la secuencia en el sentido en el cual le es transmitida (del valor más antiguo de la secuencia al más reciente), sino que también tenga en cuenta los datos en el orden inverso (del valor más reciente de la secuencia al más antiguo). Para explorar si esta sería una opción de mejora, se probó a invertir el orden dentro de las secuencias y entrenar el mismo modelo anterior con estos datos invertidos. Sin embargo, los valores obtenidos para las métricas de evaluación no fueron lo suficientemente satisfactorios, por lo que se decidió no continuar con la implementación de esta bidireccionalidad.

Todavía quedan posibilidades adicionales de mejora de este modelo, por ejemplo mediante la inclusión de otras variables de entrada auxiliares como la hora del día o el día de la semana, o mediante la inclusión de una mayor cantidad de datos.

4.4. Prototipo 1: Incluyendo variables auxiliares

Como se menciona al final de la sección anterior, una opción para mejorar el modelo podría consistir en añadir variables auxiliares que le ayuden a la red neuronal a encontrar las variaciones estacionales diarias y semanales de la secuencia. La idea subyacente es que la contaminación en el aire es previsiblemente mayor los días de diario frente a los fines de semana y, similarmente, las horas con mayor afluencia de entrada y salida del trabajo registran mayores valores de

contaminación que otras horas de menor actividad como durante la madrugada.

A la hora de realizar el diseño, se debe considerar que estas variables son una entrada auxiliar al modelo. Las redes LSTM son útiles para memorizar datos de entrada anteriores, y, por tanto, para encontrar las tendencias y patrones en las secuencias. Sin embargo, en este punto se van a añadir a añadir unas nuevas variables categóricas al modelo. Aunque dichas variables sí pertenecen a una secuencia (Lunes, Martes, Miércoles,.. hora 1, hora 2, hora 3,...), no estamos interesados en que el modelo aprenda esta secuencia, sino en que las tenga en cuenta antes de hacer sus predicciones. Por lo tanto, en lugar de incluir estas entradas auxiliares como entrada a la red LSTM previamente definida, resulta más conveniente juntarlas más adelante con la salida de la red LSTM y usarlo todo ello como entrada a una capa final de neuronas totalmente conectada (*Fully-Connected layer* en inglés). El resultado final del diseño se puede observar en el diagrama del anexo B.3.

Para implementar este diseño, han sido necesarios varios pasos de programación. El primer paso, ha consistido en modificar el *notebook* que carga, selecciona y vuelca los datos, para que añada también, en cada fila, la información de la hora y día de la semana correspondiente a la toma de cada medida. Ambos datos, la hora del día y el día de la semana se han incluido de manera numérica ordinal[6].

A continuación estos datos deben convertirse a una matriz binaria para que la red neuronal los trate como variables categóricas y no como variables numéricas[7]. Para ello, se ha usado la técnica conocida como *One-hot encoding*.

Después, se ha de partir en trozos el resultado anterior de forma que cada trozo corresponda a los datos de entrada para los conjuntos de entrenamiento, validación y test.

El cuarto y último paso ha consistido en cambiar la implementación del modelo con acuerdo al nuevo diseño. Nótese que, hasta ahora, los modelos implementados siguen una secuencia desde un único *input* hasta un único *output*. Con este nuevo diseño, sin embargo, hay dos fuentes de inputs separadas, de manera que ya no sirve el modo secuencial de keras para implementar el modelo, sino que es necesario adaptar la implementación al uso del modo funcional. Una vez en el modo funcional, la función de concatenación de capas (*layers.concatenate*) de keras permite combinar la entrada de datos auxiliares con la salida de la red LSTM. En el listado 4.5 se puede ver un extracto del pseudocódigo correspondiente a esta implementación funcional y en la figura B.1 un diagrama secuencial del modelo resultante en keras.

```
# Crear LSTM layer
lstm_input , lstm_model = crear_LSTM()
# crear auxiliar input layer
```

[6]Para los días de la semana, la transformación ha consistido en una función del tipo $Domingo \rightarrow 0, Lunes \rightarrow 1, ..., Sabado \rightarrow 6$

[7]Se puede leer más sobre las diferencias entre una y otra variable en [Sangüesa i Solé, 2010b, página 18]

```
aux_input_layer = layers.Input(shape=aux_in_shape)
# concatenar las dos anteriores
concatenated = layers.concatenate([lstm_model, aux_input_layer])
# Combinar las dos capas con una capa totalmente conectada
combined = layers.Dense(units = units_fc, activation='relu')(concatenated)
# output layer
prediction = layers.Dense(units = out_dim)(combined) # activacion linear
model = Model( inputs = [lstm_input, aux_input_layer],
               outputs = prediction, name = 'lstme')
```

Listado 4.5: Pseudocódigo para construir el modelo del prototipo 1

El código completo para esta iteración está disponible en el enlace 10.

4.4.1. Resultados preliminares

Después de entrenar el modelo con los datos de entrada y de validación, y evaluarlo contra los datos de test, el prototipo obtuvo valores de RMSE en el intervalo de **10,1-10,3** y un MAPE entorno al **25 %**, lo cual todavía no supone una gran mejora respecto al prototipo 0.

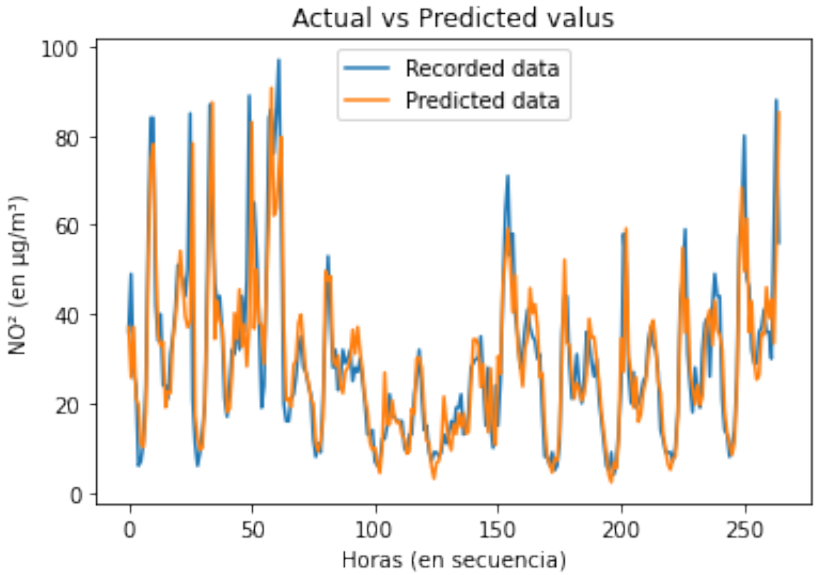


Figura 4.5: Predicciones del modelo prototipo 1 (naranja) vs datos reales (azul) para el conjunto de test - Iteración 1

En la figura 4.5 se muestra una gráfica comparando los datos registrados contra los datos predichos por el modelo. La silueta se predice francamente bien y los picos se estiman con bastante precisión, mejor que en el prototipo anterior, y casi llegando incluso a predecir saltos de gran magnitud.

Aún es constatable, no obstante, que hay algunos valores dentro de la "normalidad" de la secuencia que el modelo no ha predicho con exactitud. Al inspeccionar las gráficas de pérdidas en el entrenamiento, se puede apreciar que se está produciendo el fenómeno conocido como sobreentrenamiento u *overfitting*.

4.4.2. Segunda y tercera iteración

Una de las técnicas disponibles para mitigar los efectos del sobreentrenamiento y mejorar así la generalización del modelo, es la conocida como *Dropout*. Esta técnica consiste en quitar al azar algunas partes de una red neuronal durante el entrenamiento y se debe implementar de forma específica en redes recurrentes[8]. En este caso, se ha aplicado un Dropout de 0,01 a la red.

Ahora en la visualización de alguna de las gráficas de entrenamiento (véase, por ejemplo, la gráfica de la figura 4.6), se observa la dispersión en la precisión del modelo respecto al conjunto de entrenamiento. Esto es debido a este *Dropout* que acabamos de introducir.

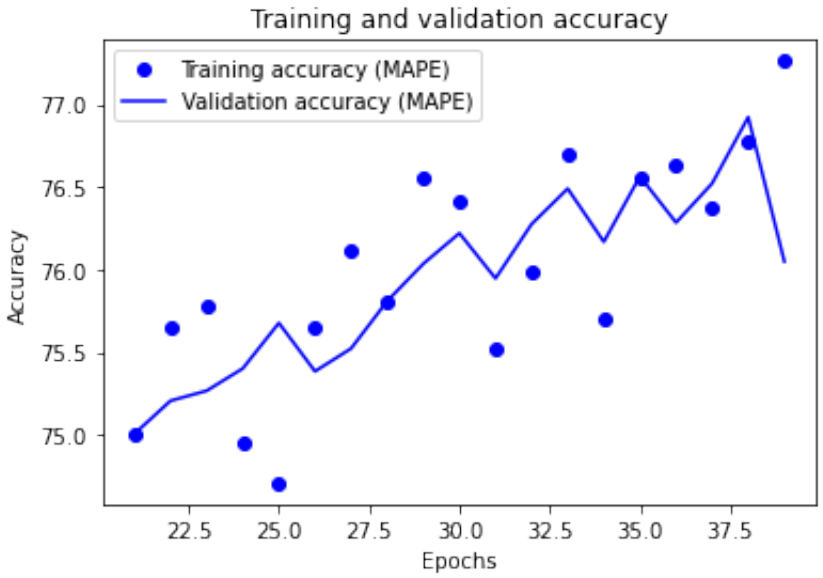


Figura 4.6: Gráfica donde se puede ver como el Dropout contrarresta el sobre aprendizaje del modelo

El código completo para esta segunda iteración está disponible en el enlace 11.

Ahora el valor obtenido de RMSE es de $\mathbf{10.05\,\mu g/m^3}$ y un MAPE de **24,85 %**.

Finalmente, se apreció que el modelo todavía pronostica valores inferiores a los reales para algunos picos hacia arriba de alta contaminación. Con el fin de mejorar estos casos, se realizaron

[8]Keras usa una implementación del Dropout para redes recurrentes basada en la investigación de Yarin Gal ([Gal, 2016, Chapter 3: Bayesian Deep Learning])

varios experimentos con distintos valores de *batch size* y se obtuvieron una ligera mejoría en los resultados con `batch_size = 32`. Por tanto, la configuración final de los hiperparámetros queda así:

```
    params = {
    'first_layer': 128,
    'second_layer': 128,
    'optimizer': optimizers.Adam,
    'batch_size': 32,
    'epochs': 100,
    'lr': 0.001,
    'dropout': 0.01,
    'rnn-dropout': 0.01,
    'units_fc_layer': 128
}
```

Listado 4.6: Configuración de Hiperparámetros para la tercera iteración del prototipo 1

En esta tercera y última iteración para este prototipo los valores para las métricas son para RMSE: **9.91 µg/m³** y para MAPE: **24.53 %**. El código completo correspondiente a esta tercera iteración está disponible en el enlace 12.

4.4.3. Resultados prototipo 1

A través de añadir dos nuevas variables auxiliares y usando la técnica del *Dropout*, el modelo ha mejorado entorno a un 5 % respecto al prototipo anterior.

Observando la gráfica de la figura 4.7, la cual muestra las predicciones hechas por el modelo frente a los datos reales, se aprecia que la silueta tiene ya una forma muy muy parecida a la silueta de los datos originales.

4.5. Prototipo 2: Incluyendo las variables meteorológicas

La meteorología afecta de dos formas en las medidas de NO_2. De forma directa, por un lado, en cómo los contaminantes se desplazan por la atmósfera o en cómo se transforman en otros compuestos químicos. De forma indirecta, por otro, ya que las condiciones meteorológicas influyen en cómo y en qué medida la población se desplaza por el medio urbano.

Efectivamente, en numerosos trabajos anteriores se ha demostrado que los datos meteorológicos mejoran en gran medida la capacidad de predicción de los modelos de previsión de los diferentes contaminantes del aire (Rybarczyk and Zalakeviciute [2018]), y más en particular para el caso del NO_2 (Pardo and Malpica [2017], Krishan et al. [2019]).

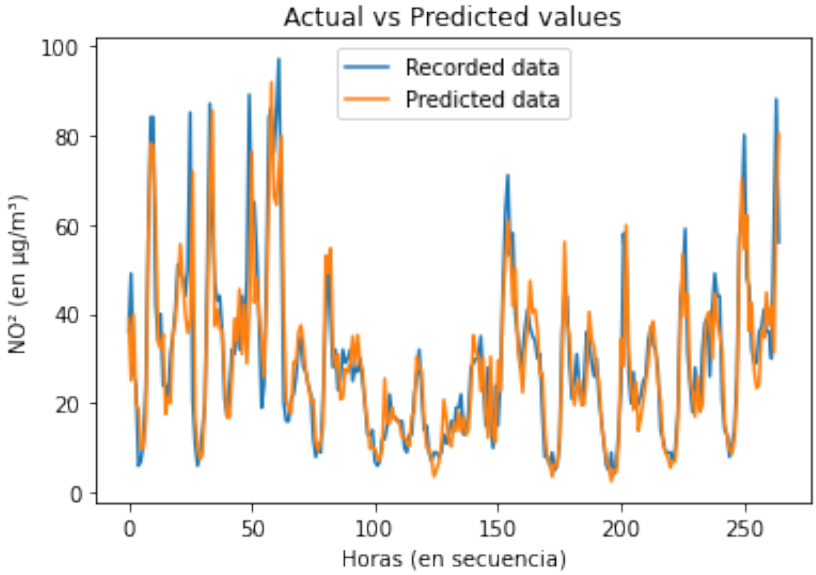


Figura 4.7: Predicciones del modelo prototipo 1 (naranja) vs datos reales (azul) para el conjunto de test - Iteración 3

El siguiente prototipo consiste en intentar mejorar el rendimiento de los resultados anteriores a base de usar datos meteorológicos como entrada adicional al modelo. Para ello, se usarán los datos procedentes del portal del Ayuntamiento de Madrid, tal y como se adelantó en la sección 3.4. De entre los datos disponibles, se han escogido las variables: velocidad del viento, dirección del viento, temperatura, humedad relativa, precipitaciones y presión atmosférica; ya anteriormente descritas en el apartado 3.5.2 de este documento. Los datos extraídos corresponden al rango de meses de febrero de 2019 a mayo de 2019.

Como se comentó también en aquel apartado 3.4, es necesario cambiar la estación de calidad del aire a predecir a otra que tenga una estación de meteorología cercana[9]. Así pues, en lo sucesivo se realizarán los experimentos de este prototipo 2 y de todos aquellos otros modelos sobre los que se compare, usando como fuente de origen de los datos a la estación del Barrio del Pilar (código 28079039).

Antes de proceder, hay que dedicar un especial detenimiento al diseño para dirimir cómo se ha de incluir esta nueva variable en el modelo de la manera más eficaz y razonable posible. Así pues, una vez revisados los diversos diseños dentro del Estado del Arte actual, y se tomó la decisión de decantarse por el usado en Li et al. [2017], bautizado por los autores como modelo

[9]Como curiosidad e información adicional, al principio y debido a un error, se hicieron los primeros experimentos usando los datos meteorológicos de la estación de Peñagrande, mientras que seguía usando los datos históricos de calidad del aire de la estación de Cuatro Caminos, ambas estaciones distan una de otra unos 3.7 km. Los resultados obtenidos para este prototipo 2 con estos datos resultaban decepcionantes, puesto que obtenía peores resultados que con el prototipo 1. Esto demuestra en qué medida es importante tener en cuenta la cercanía y coherencia entre las dos fuentes de datos.

"LSTME" (LSTM Extendido).

El motivo para esta elección es que separan las entradas correspondientes a la serie histórica de valores medidos, de las variables auxiliares que ayudan en la predicción. Como ocurriera en el prototipo anterior, la predicción de las secuencias tiene sentido hacerlas mediante una red LSTM; mientras que para las otras variables auxiliares no se quiere tener en cuenta la evolución de sus valores. Más aún, nos interesa usarlas para que la red las tenga en cuenta a la hora de hacer predicciones y mejorar así sus resultados. En el anexo B.4 se encuentran los diseños correspondientes a este prototipo: un diagrama del diseño propuesto a alto nivel, además de un diagrama secuencial del diseño generado por keras (figura B.2).

4.5.1. Primera iteración

En una primera iteración, se realizó un experimento de hasta 200 epochs, usando los mismos valores para los parámetros que había usado en el prototipo 1 y que se encuentran en el listado 4.6. En la figura 4.8 se puede observar cómo evoluciona el aprendizaje del modelo, cómo se produce finalmente el *overfitting* y que finalmente se alcanza el *robust fit* entorno a las 75 epochs. En dicho punto, el modelo deja de mejorar para el conjunto de validación y solamente mejora para el conjunto de entrenamiento.

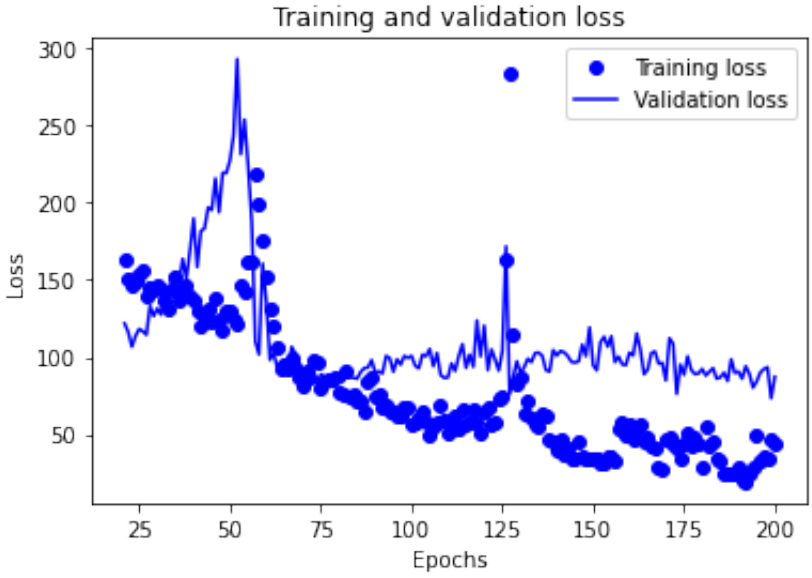


Figura 4.8: Gráfica donde se puede ver el proceso de aprendizaje del prototipo 2 comenzando desde la época 20 hasta la 200

Ejecutando de nuevo el experimento y parando el modelo cuando deja de mejorar frente al conjunto de validación, se obtiene el mejor resultado al parar el modelo a las 57 epochs.

Los valores para las métricas obtenidos son de un RMSE de **8.565 µg/m³** y un MAPE de **35.731 %**.

Después de probar varias combinaciones de valores diferentes para los hiperparámetros e incluso probando a añadir una capa oculta más antes de la salida, no se obtuvieron mejores resultados en lo referente a la medida de RMSE frente a los que ya había obtenido previamente con los parámetros mencionados anteriormente. Cabe decir, sin embargo, que en algunos casos sí se obtuvieron mejores resultados para la métrica MAPE con esas otras configuraciones, pero con peores valores de RMSE.

El código completo de esta iteración está disponible en el enlace 13.

4.5.2. Resultados prototipo 2

En la tabla 4.1 se puede ver un resumen de los mejores valores para las métricas obtenidos por los diferentes modelos con este nuevo conjunto de datos –es decir, usando los datos de la estación de calidad del aire del Barrio del Pilar–. Se puede observar como con este prototipo 2, se obtiene una mejora de un **8 %** –1,081 menos de error en la tasa de RMSE– respecto al prototipo anterior y un **35 %** –4,504 menos de error– respecto al modelo base, gracias a haber incorporado datos de meteorología.

PMF	SVR	Prototipo 0	Prototipo 1	Prototipo 2
13,069	14,650	9,715	9,646	8,565
100 %	112,1 %	74,3 %	73,8 %	65,5 %

Cuadro 4.1: Resultados de RMSE (en µg/m^3) de predicciones a 1 hora para los diferentes modelos y el porcentaje respecto al modelo base. Cuanto más bajo, mejor es la precisión del modelo

En la gráfica de la figura 4.9, la cual muestra las predicciones hechas por el modelo frente a los datos reales. De nuevo, la silueta pronosticada tiene una forma muy muy parecida a la silueta de los datos originales, llegando a "preveer" incluso el advenimiento de un gran pico de subida. Parecen pues, unos resultados difícilmente mejorables.

Finalmente, habría que ver si la incorporación de estos datos meteorológicos revertirán de manera positiva o no, a la hora de hacer previsiones a un instante de tiempo más lejano, como podría ser a las 8, 16 ó 24 horas.

4.6. Predicción a 8, 16 y 24 horas

Desde el punto de vista práctico, resulta interesante tener predicciones a una distancia mayor a la de la siguiente hora. No obstante, las predicciones a más larga distancia suelen ser menos

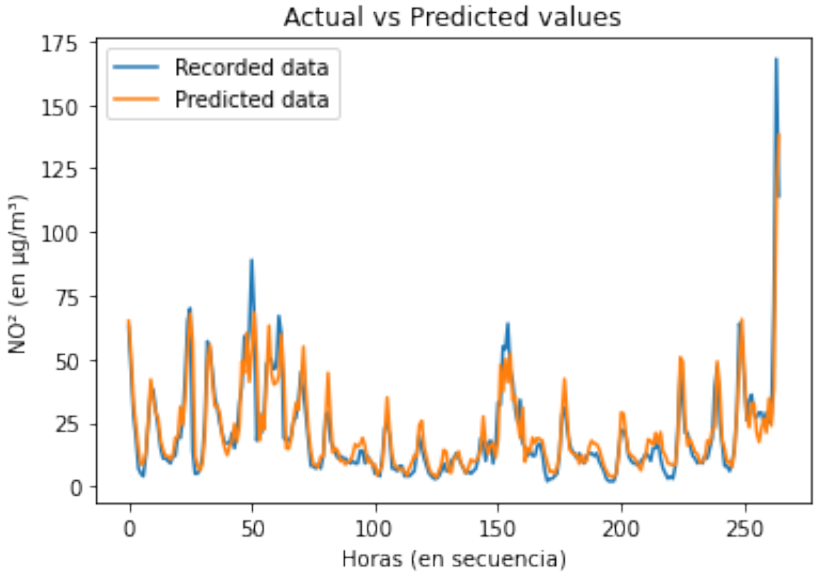


Figura 4.9: Predicciones del modelo prototipo 2 (naranja) vs datos reales (azul) para el conjunto de test - Iteración 1

precisas (Casas Roma [2020]). En esta sección se presentan las predicciones, dadas las medidas de NO_2 de las anteriores 23 horas, para dentro de ocho, dieciséis y veinticuatro horas con cada uno de los diferentes modelos anteriores.

En primer lugar, se realizarán las predicciones con los diferentes modelos a las veinticuatro horas, puesto que es aquí donde la precisión se espera peor y los parámetros han de estar mejor ajustados. A continuación, se iterará añadiendo algunas mejoras para intentar conseguir mejores resultados como: la optimización de los hiperparámetros, la alineación de los datos de meteorología con el instante de predicción para el prototipo 2 o la inclusión de más datos procedentes de otros meses. Después, se hará la predicción a ocho y dieciséis horas de distancia. Finalmente, se hará un análisis de los resultados obtenidos comparándolos con trabajos anteriores.

4.6.1. Predicción a 24 horas

4.6.1.1. Primera iteración

Para tener unos valores base que hagan uso los mismos datos contra los cuales comparar los resultados obtenidos por los diferentes prototipos, se ha adaptado el modelo SVR descrito en la sección 4.2.2. Al ejecutar este modelo haciendo predicciones a 24h de distancia y con los datos de contaminación medidos desde febrero a mayo de 2019, se obtiene un RMSE de $\mathbf{19.82\,\mu g/m^3}$.

Los prototipos 0 y 1 con la predicción a veinticuatro horas y los datos de febrero a mayo de

2019, usando los mejores parámetros de los apartados anteriores, obtenían un RMSE entre **17** y **18 µg/m³**. En las figuras 4.10 y 4.11, se pueden ver las gráficas de predicción y valores reales de estos los prototipos 0 y 1 respectivamente. En ellas, se observa que los valores de predicción son demasiado constantes y que los modelos no son capaces de predecir los picos.

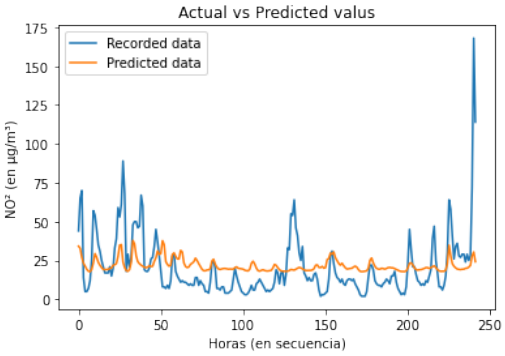


Figura 4.10: Predicciones a 24h del prototipo 0 (naranja) vs datos reales (azul) - Iteración 1

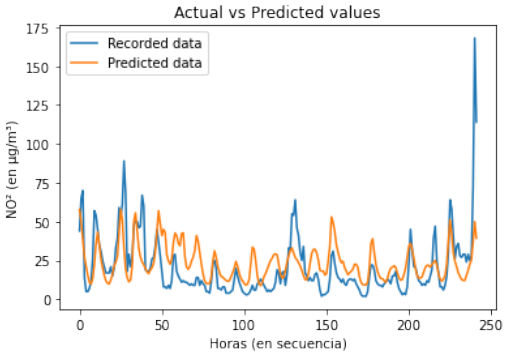


Figura 4.11: Predicciones a 24h del prototipo 1 (naranja) vs datos reales (azul) - Iteración 1

Se observó que a medida que se aumentaba el valor para el parámetro *batch_size* y *epochs*, se mejoraba la predicción de los picos de contaminación y se reducía sensiblemente el valor de RMSE. Después de experimentar con diversos valores, se obtuvieron los mejores resultados con unos valores de *batch_size* de 64 y de 32 para el prototipo 0 y 1 respectivamente.

Los mejores resultados para esta iteración para el prototipo 0 fueron de RMSE **17.63 µg/m³** y MAPE **106.75 %**. Todavía a este modelo le cuesta predecir los picos y se mantiene en unas predicciones muy conservadoras.

Para el prototipo 1 los resultados son un poco mejores, un RMSE de **16.63 µg/m³** y MAPE **68.35 %**. Especialmente al aumentar el número de *epochs* (entorno a unas 70), el modelo se "atreve" más y se esfuerza mejor en predecir los picos de contaminación.

Finalmente, el prototipo 2 obtuvo unos resultados muy similares a los del prototipo 1.

4.6.1.2. Iteración 2: Alineación de los datos auxiliares con el instante a predecir

Con el fin de intentar solventar o mejorar los resultados anteriores, se probaron algunos cambios. El primero consistente en alinear los datos auxiliares con el instante en el que se hace la predicción y el segundo en añadir más datos a los modelos.

En el apartado anterior hemos visto cómo los prototipos tienen dificultades para predecir las subidas y bajadas de los datos, las cuales no se daban antes cuando la predicción era a 1 hora vista. Los datos auxiliares añadidos al modelo en las secciones anteriores, estaban

alineados con el último valor de la secuencia a partir de la cual se hace la predicción. Es decir, si la secuencia acababa en un martes a las 8 am, el prototipo 2 tendría de entrada el día de la semana (martes), la hora (ocho) y el tiempo meteorológico para ese día a esa hora. Sin embargo, ahora las predicciones están más alejadas y el tiempo meteorológico puede ser sensiblemente diferente. De modo que el primer cambio introducido fue alinear los datos auxiliares (fecha y meteorología) con el valor que va a ser predicho. Además, se observó que el prototipo 2 también ganaba una precisión adicional al recibir como entrada secuencias con las 24 horas anteriores, en lugar de las 23 horas anteriores como veníamos haciendo anteriormente.

Haciendo estos cambios, el prototipo 2 mejoró hasta conseguir un RMSE de **13.352 µg/m³** y un MAPE de **50.04 %**. El prototipo 1 se mantuvo con unos resultados similares a los anteriores –hecho esperable ya que para los datos de fecha se trata de una simple traslación en el tiempo–. El prototipo 0 no fue re-evaluado puesto que no hace uso de ningún dato auxiliar.

Al observar de cerca las predicciones de las figuras 4.12 y 4.13, se observa que ambos modelos, pero especialmente el prototipo 2, generan una silueta muy parecida a la de los datos originales, prediciendo con bastante acierto las subidas y bajadas de contaminación. Únicamente a veces las predicciones para los picos de subida se quedan algo cortas.

Los mejores resultados de cada modelo para esta iteración están reunidos en la tabla 4.2.

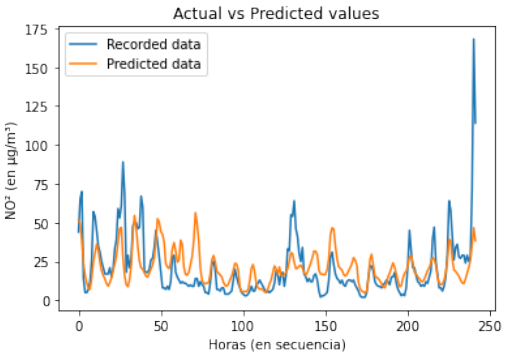


Figura 4.12: Predicciones a 24h del prototipo 1 (naranja) vs datos reales (azul) - Iteración 2

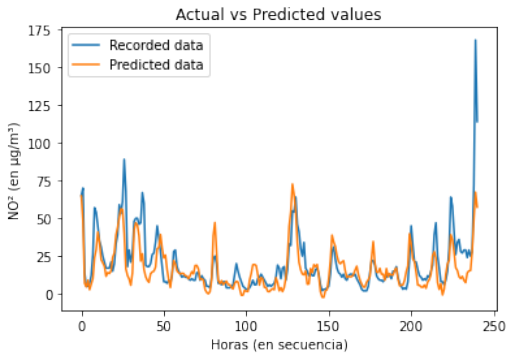


Figura 4.13: Predicciones a 24h del prototipo 2 (naranja) vs datos reales (azul) - Iteración 2

Predicción a 24 horas (4 meses)	SVR	Prototipo 0	Prototipo 1	Prototipo 2
RMSE	19,82	17,63	16,63	13,35
MAPE	169 %	107 %	80 %	50 %

Cuadro 4.2: Resultados de RMSE (en µg/m³) y de MAPE (%) para los diferentes prototipos haciendo predicciones a las 24h, usando datos de los meses febrero a mayo de 2019. A valores más bajos de estas métricas, mejor precisión del modelo

4.6.1.3. Iteración 3: Datos de un año completo

Como tercer y último intento de mejorar la predicción de estos picos de subida, se ha probado a añadir más datos de entrenamiento al modelo.

Dado que las fuentes de datos meteorológicos solo proveen datos desde 2019, se extendió el conjunto de datos de entrada a los doce meses de 2019. Esto forman tres veces más datos de los que habían sido usados hasta ahora.

En la figura 4.14 se pueden ver las gráficas de predicción correspondientes a los diferentes prototipos y en la tabla 4.3 los resultados obtenidos para las métricas de evaluación.

Predicción a 24 horas (1 año)	SVR	Prototipo 0	Prototipo 1	Prototipo 2
RMSE	18,823	20,64	20,90	15,45
MAPE	84,25 %	70 %	69 %	39,22 %

Cuadro 4.3: Resultados de RMSE (en $\mu g/m^3$) y de MAPE (%) para los diferentes prototipos haciendo predicciones a las 24h y usando datos del año completo de 2019. A valores más bajos de estas métricas, mejor precisión del modelo

Al observar los resultados obtenidos por los diferentes prototipos usando estos doce meses de 2019, notamos que son sensiblemente peores que los que se habían obteniendo con los datos de los cuatro meses primaverales en cuanto a los valores de RMSE, aunque no ocurre así con los valores de la métrica MAPE.

Con todo, la predicción de las siluetas no es del todo deficiente y los modelos detectan más o menos bien la tendencia al alza al final del año, aunque no la magnitud de los grandes picos de subida que se dan. El prototipo 2 destaca especialmente por su buena precisión al prever picos al alza (a veces incluso previendo valores más altos de los reales), y por detectar los ciclos de subida y bajada que ocurren espontáneamente.

Aparentemente estos peores resultados, pudieran deberse a que los datos de test corresponden a diciembre y a las últimas semanas de noviembre, meses caracterizados por el frío y con mucha actividad de tráfico rodado en la ciudad de Madrid. Para solventar esta casuística, se probó a mover la selección del conjunto de test a los meses de octubre y noviembre. Sin embargo, los resultados siguieron sin mejorar respecto a los obtenidos en la iteración anterior antes de hacer este último cambio.

De cualquier modo, hay que notar que las variaciones estacionales que existen con periodicidad anual no pueden ser aprendidas por el modelo cuando éste solamente posee datos de 12 meses. Queda como trabajo futuro el repetir este proceso usando datos históricos de varios años.

En el enlace 14, dentro del anexo A.2, se listan las direcciones web donde están disponibles los *notebooks* correspondientes al código de los prototipos usados para las predicciones a 24

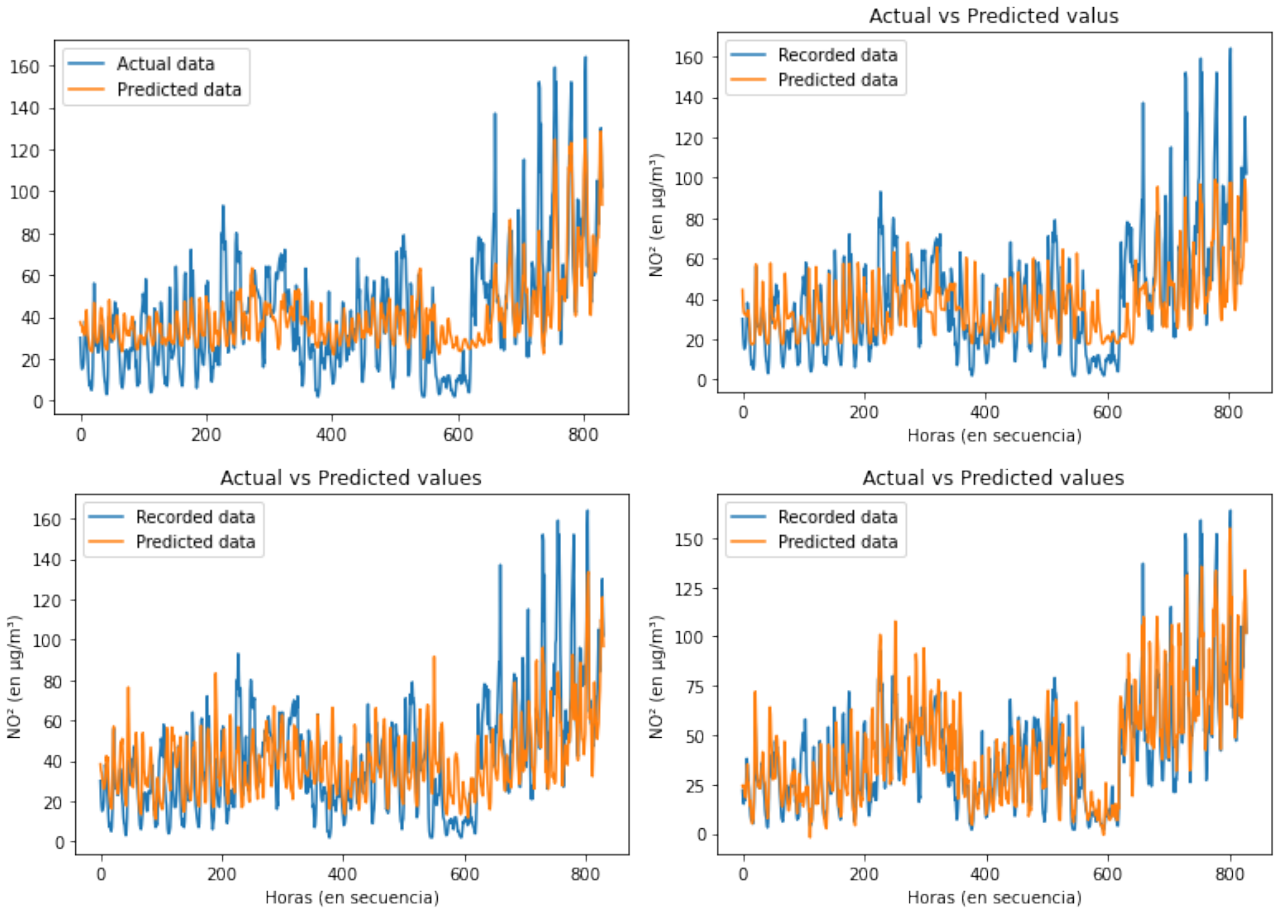


Figura 4.14: Predicciones a 24h del modelo (naranja) vs datos reales (azul) usando todos los datos de 2019 - Iteración 3. Los modelos son de izquierda a derecha y de arriba a abajo: SVR, Prototipo 0, Prototipo 1 y Prototipo 2.

horas.

4.6.2. Predicción 8 y 16 horas

Una vez optimizados los diferentes prototipos mediante diferentes estrategias, se dispusieron estos mismos modelos con los parámetros anteriores para hacer predicciones a ocho y dieciséis horas. Cabe decir, no obstante, que en este proceso se hicieron algunos cambios en los hiperparámetros a otros que obtuvieron mejores resultados. Ejemplos de estos cambios son el valor de batch_size a 16 para el prototipo 1, o la inclusión de una segunda y tercera capa interna y un aumento de las unidades de esas capas internas a 256 para el prototipo 2.

Los resultados de las métricas de evaluación para estas predicciones se pueden ver en la tabla 4.4. En ella se mantiene la tendencia de que los prototipos van mejorando a medida que tienen más datos auxiliares y también se comprueba efectivamente que con predicciones más cercanas

en el tiempo los modelos obtienen mejores resultados –lo cual ya se ha había adelantado al principio de esta sección–.

Predicción a 8 horas	SVR	Prototipo 0	Prototipo 1	Prototipo 2
RMSE	19,15	17,80	17,49	11,58
MAPE	162,15 %	106,58 %	75,30 %	40,59 %
Predicción a 16 horas				
RMSE	19,70	19,62	17,67	12,38
MAPE	170 %	110 %	108 %	41,01 %

Cuadro 4.4: Resultados de RMSE (en $\mu g/m^3$) y de MAPE (%) para los diferentes prototipos haciendo predicciones a las 8h y 16h, usando datos de los meses febrero a mayo de 2019. A valores más bajos de estas métricas, mejor precisión del modelo

El código para estos notebooks está disponible en el enlace 15 dentro del anexo A.2.

4.6.3. Resultados de las predicciones a 8, 16 y 24 horas

En esta sección, se ha pasado de hacer predicciones para el instante inmediatamente siguiente, a hacer predicciones a las ocho, dieciséis y veinticuatro horas. Los resultados, como cabría de esperar, han empeorado respecto a las predicciones que se hacían para la hora siguiente.

Con todo, el prototipo 2, que además de usar los datos medidos de contaminación usa también la fecha, hora y datos meteorológicos del instante a predecir, obtiene entre un 30 y un 40 % mejor de rendimiento comparado con el modelo base SVR propuesto para los mismos datos y haciendo las predicciones a la misma distancia.

Según Pardo and Malpica [2017], CALIOPE –el sistema de referencia para las predicciones de la contaminación en España presentado en el capítulo 2 de este documento– tiene una media de RMSE en las predicciones a 24h de $\mathbf{26\,\mu g/m^3}$. Los modelos de aprendizaje automático presentados en este trabajo tienen una mejora desde entorno un 25 % comparándolo con el modelo base, hasta cerca de un 50 % comparándolo con el prototipo 2.

En ese mismo trabajo, Pardo and Malpica [2017], los autores obtienen unos niveles de RMSE entre 10 y $\mathbf{11\,\mu g/m^3}$ para predicciones de 8, 16 y 24 horas en los valores de NO_2 en la ciudad de Madrid. Dichos resultados son ligeramente mejores a los resultados obtenidos en el presente trabajo. No obstante, cabe señalar que los autores usan el conjunto de validación –el cual se usa durante el entrenamiento–, en lugar de un conjunto separado de test, a la hora de evaluar su modelo. Esto puede influir notablemente en que se obtengan mejores resultados, dado que el modelo puede haberse especializado en obtener buenos resultados con el conjunto de validación con el cual se ha entrenado, o la configuración de los hiperparámetros esté más ajustada a ese conjunto de validación.

Además, el modelo usado en dicho estudio fue entrenado con datos registrados de un total de cinco años (2011-2016), lo cual es esperable que mejore el aprendizaje del modelo, a costa de necesitar muchos más tiempo para su entrenamiento y una mayor complejidad del modelo. En mis pruebas, los prototipos no se han beneficiado de añadir más datos procedentes del resto de meses de 2019 –tres veces más datos–; esto bien puede deberse a las grandes variaciones estacionales que tienen lugar en la ciudad de Madrid dentro del mismo año (periodos vacacionales, meses de verano, semanas de navidad, etc.), como ya se mencionó anteriormente en las secciones 3.4 y 4.6.1.3.

Así pues, queda como trabajo futuro incluir datos de varios años y añadir el número del mes como entrada auxiliar (similarmente a cómo se hace en Li et al. [2017] con la definición del modelo "LSTME"). Adicionalmente, habría que hacer los ajustes pertinentes para aumentar la complejidad del modelo para ajustarlo a este nuevo volumen de datos.

4.7. Predicción de todas las estaciones al mismo tiempo

La ciudad de Madrid dispone de 24 estaciones de calidad del aire distribuidas a lo largo de todo el municipio. Hasta ahora, hemos estado usando los datos provenientes de una sola estación al mismo tiempo y hemos entrenado un único modelo específico para esa estación.

Sin embargo, a la hora de dar un pronóstico de la situación de la calidad del aire para la ciudad, resulta mucho más interesante disponer de la predicción de todas las estaciones disponibles. Además, se ha observado por otros trabajos que existe una correlación espacial entre los datos de calidad del aire provenientes de distintas estaciones geográficamente cercanas (Li et al. [2017]), lo cual justifica incluir todas ellas en un mismo modelo.

En esta sección se explica el desarrollo de unos modelos únicos, basados en los prototipos 1 y 2 anteriores, en los cuales se realiza simultáneamente la predicción para todas las estaciones de Madrid disponibles.

Para ello, se han tenido que introducir cambios en los *scripts* de carga y selección de datos. Puesto que ahora tenemos que usar los datos de todas las estaciones de calidad del aire y asegurarnos que todas tienen la misma cantidad de datos para las mismas fechas. Se ha observado que hay algún día perdido o inexistente en los datos de alguna estación. Para estos casos, ha sido necesaria su localización y rellenado mediante los datos anteriores más cercanos que hubiese disponibles, siguiendo la misma estrategia que se realiza para los datos no verificados, previamente explicado en la sección 3.6.

Un proceso similar de selección y tratamiento de valores perdidos ha sido llevado a cabo para los datos meteorológicos. En este caso disponemos de 26 estaciones de datos pero no todas miden los mismos parámetros de variables meteorológicas y, de hecho, la gran mayoría solo miden

temperatura y humedad relativa. De todas estas estaciones, se han seleccionado únicamente aquellas que proveen de, al menos, las siguientes variables: velocidad del viento, temperatura, humedad relativa y precipitaciones. Se trata de 9 estaciones meteorológicas distribuidas por el municipio de Madrid, con 4 variables meteorológicas diferentes cada una, lo que suma un total de 36 entradas de datos meteorológicos para cada instante de tiempo del modelo.

4.7.1. Modelo único 1: Sin usar datos meteorológicos

El primer modelo está basado en el prototipo 1, en el cual se usaban los datos históricos de las anteriores 23 horas y algunas variables auxiliares relativas a la hora y día de la semana. En este caso, se ha incrementado la entrada desde una sola secuencia de 23 horas a 24 secuencias (una por estación) de 23 horas. Correspondientemente, la salida del modelo ha pasado a ser de 24 valores (una por estación).

Con este gran incremento de datos y predicciones, la complejidad del problema aumenta y fue necesario incrementar también la complejidad de la red neuronal para que ésta aprendiese. Así pues, las neuronas en las capas LSTM se incrementaron de 128 a 512 y las de la capa densa final completamente conectada a 256.

4.7.2. Modelos únicos 2 y 3: Usando datos meteorológicos

Los segundo y tercer modelo, además de los datos de las 24 estaciones y de las entradas auxiliares relativas a la fecha, incorporan el histórico de datos de meteorología.

Dado que tenemos una cantidad moderada de entradas de datos meteorológicos (36) en cada instante de tiempo, es buena idea que estas variables tengan también una red neuronal dedicada a aprender la influencia de éstos. Por ello, se añadieron dos capas completamente conectadas (*fully-connected layers*) a las entradas meteorológicas, antes de conectarlas con el resto del modelo. En el anexo B.5 se pueden ver los diagramas correspondientes al diseño de la arquitectura de estos dos modelos únicos, y en las figuras B.3 y B.4 los diagramas secuenciales de los modelos 2 y 3 respectivamente generados por *keras*.

En el caso del modelo 2, los datos de meteorología usados son los correspondientes al último valor de la secuencia, es decir, los existentes en el instante de tiempo en el cual se hace la predicción. En el supuesto contexto de estar dando predicciones en tiempo real, se trataría de los únicos valores meteorológicos seguros.

En el caso del modelo 3, los datos de meteorología usados son los registrados en el instante de tiempo en el cual se hace la predicción, es decir, si las predicción es a las 24 horas, no se usará la meteorología actual sino la que tendrá lugar en 24 horas. Esto es esperable que haga que el modelo obtenga mejores resultados que el modelo anterior. Sin embargo, a la hora de dar

predicciones en tiempo real no dispondremos de dichos datos meteorológicos sino de pronósticos del tiempo, los cuales tienen un grado de inexactitud asociado.

La configuración de los hiperparámetros para estos dos modelos también han sido escogidos después de diversas pruebas usando diferentes combinaciones de configuraciones.

4.7.3. Resultados de los modelos únicos

Finalmente, todos los modelos han sido entrenados y evaluados para predicciones a 1h, 8h, 16h y 24h. Los mejores resultados obtenidos en estos experimentos se muestran en la tabla 4.5.

Predicción a 1 hora	Modelo 1	Modelo 2	Modelo 3
RMSE	10,16	9,41	9,17
MAPE	38,66 %	38,69 %	42,12 %
Predicción a 8 horas			
RMSE	16,21	16,52	11,73
MAPE	75,12 %	64,65 %	41,31 %
Predicción a 16 horas			
RMSE	17,35	17,59	11,53
MAPE	96,08 %	75,79 %	46,51 %
Predicción a 24 horas			
RMSE	17,41	16,16	11,89
MAPE	97,90 %	67,16 %	47,39 %

Cuadro 4.5: Resultados de RMSE (en $\mu g/m^3$) y de MAPE (%) para los modelos únicos para las 24 estaciones de calidad del aire de Madrid. Las predicciones son a las 1h, 8h, 16h y 24h, usando datos de los meses febrero a mayo de 2019. A valores más bajos de estas métricas, mejor precisión del modelo

En general, los resultados son un poco peores con estos modelos únicos de todas las estaciones, comparados con los prototipos anteriores específicos de una estación en concreto. No obstante, siguen teniendo un resultado aceptable y de mayor precisión que el sistema CALIOPE.

En efecto, el modelo 3 que conoce la meteorología que va a ocurrir en el momento para el que se hace la predicción, obtiene unos resultados mucho más precisos que los demás, especialmente en las predicciones que más alejadas de 1 hora. Esto se hace patente al observar las gráficas de predicción. Por ejemplo, en la figura 4.15 se muestran las predicciones a 16h de cada modelo único y los datos reales medidos en una de las estaciones de calidad del aire tomada al azar.

Nótese además, que en los casos de predicciones a 8 horas o de predicciones a las 16h, el modelo 1 –que no usa datos meteorológicos– funciona mejor que el modelo 2 –el cual usa datos de meteorología pero no de la hora sobre la cual se está prediciendo, sino de la hora del último instante de la secuencia–. Posiblemente esto se deba a que la predicción a las 24h sí se ve beneficiada de usar la meteorología actual, puesto que la meteorología correspondiente a la

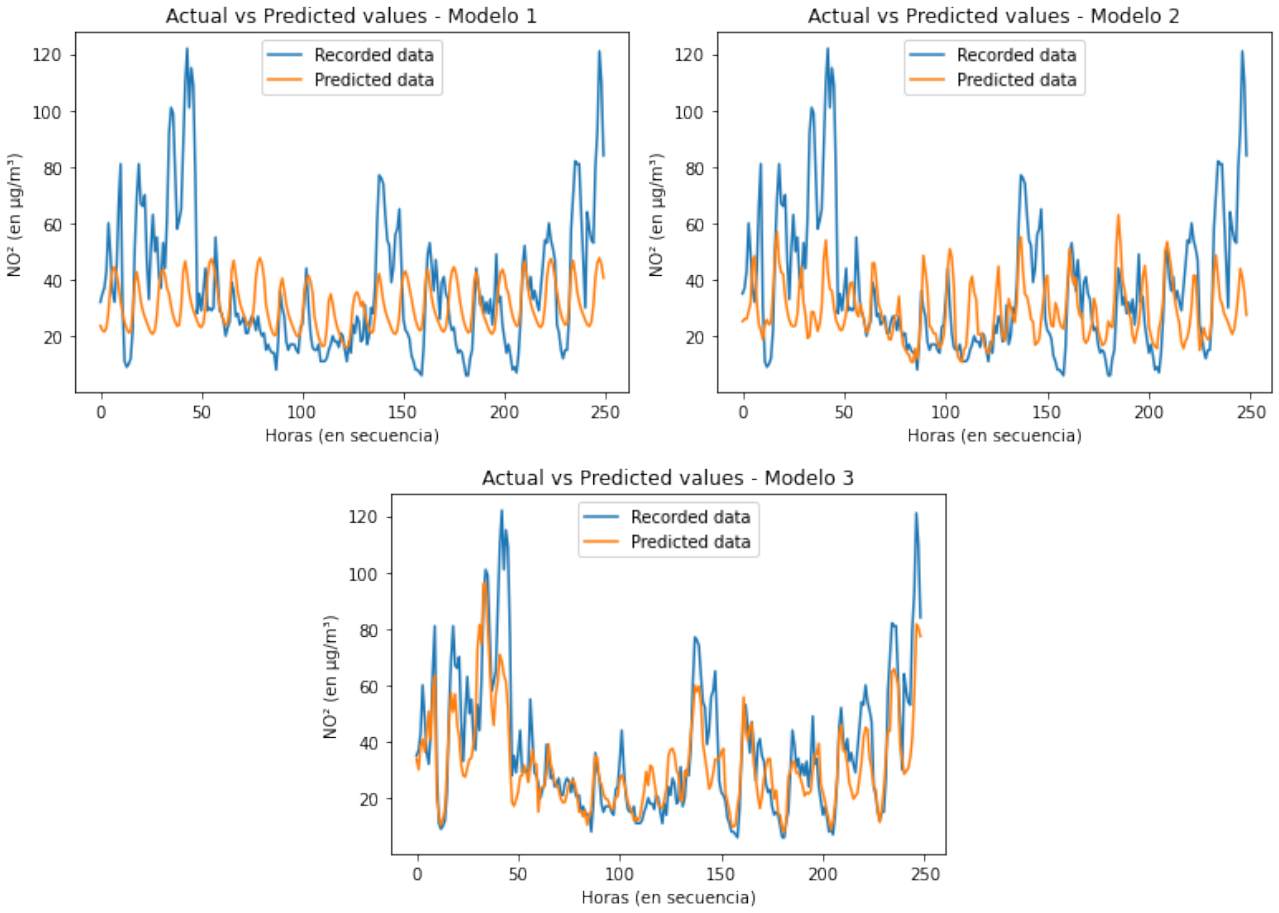


Figura 4.15: Predicciones a 16h de los modelos únicos (naranja) vs datos reales (azul), para una de las estaciones de calidad del aire, usando los datos de febrero a mayo de 2019. Los modelos son de izquierda a derecha y de arriba a abajo: Modelo 1, Modelo 2 y Modelo 3.

temperatura o la humedad relativa tiene cierta correlación cada 24h (la temperatura de hoy a las 8 am es probablemente similar a la de mañana a las 8 am).

La conclusión es que los datos meteorológicos sobre el instante en el cual se hacen las predicciones son muy valiosos a la hora de hacer las predicciones. Así pues, un modelo que quisiese hacer predicciones en tiempo real, con gran seguridad se verá beneficiado de usar pronósticos del tiempo sobre la hora en la cual se quieran hacer las predicciones.

El código para estos notebooks correspondiente a estos modelos está disponible en el enlace 16 dentro del anexo A.2.

Capítulo 5

Conclusiones y trabajo futuro

5.1. Conclusiones

Este Trabajo de Fin de Máster ha consistido en la realización de todo un proceso de investigación y desarrollo hasta conseguir un modelo único predictivo de la calidad del aire para la ciudad de Madrid. Este modelo está construido mediante el uso de redes neuronales profundas (*deep learning*), en particular, un tipo específico conocido como redes Long Short-Term Memory Networks (LSTM), las cuales son especialmente útiles para el aprendizaje y predicción de secuencias. La codificación de este modelo se ha hecho en lenguaje Python, mediante la última versión de la librería *keras*.

La implementación de este modelo se ha hecho de manera iterativa, experimentando con diferentes configuraciones de valores para los hiperparámetros cada vez; y mediante la realización de diferentes prototipos, los cuales introducen nuevos datos y variables al modelo. El trabajo ha estudiado la eficacia de los diferentes prototipos para predicciones a 1, 8, 16 y 24 horas de distancia desde el último elemento de la secuencia. Los mejores resultados son los obtenidos por los modelos que hacen uso de datos meteorológicos. En especial, para las predicciones a más de 1 hora de distancia resulta muy relevante usar los datos meteorológicos de la hora sobre la cual se está haciendo la predicción, y no sobre el último instante de tiempo desde el cual se hace la predicción.

Finalmente, se ha implementado una serie de "modelos únicos", basados en los diferentes prototipos anteriores, los cuales se entrenan y hacen predicciones de manera simultánea sobre las veinticuatro estaciones de calidad del aire que hay situadas en la ciudad de Madrid. El resultado final es un modelo único que hace predicciones simultáneas para las veinticuatro estaciones de Madrid, que hace predicciones para las 1, 8, 16 y 24 horas siguientes, y que usa la meteorología de la hora sobre la cual se está haciendo la predicción, además de tener como información adicional la hora del día y el día de la semana.

Las predicciones que da el modelo único tienen una cantidad de error reducido, menor a medida que las predicciones son más cercanas en el tiempo. Este modelo único tiene un rendimiento superior a las predicciones que da CALIOPE, el sistema de referencia para predicciones de calidad del aire en España. Efectivamente, al comparar la gráfica de valores predichos con valores observados, se puede comprobar que las siluetas siguen una tendencia notablemente similar; es sobre todo en la predicción de ciertos picos repentinos de alta polución, donde encontramos que el modelo se queda algo escaso a la hora de dar predicciones.

5.2. Trabajo futuro

Por último, cabe comentar algunos puntos que no se han podido abarcar en este Trabajo de Fin de Máster (TFM), pero que merecía la pena ser abordados en posteriores trabajos.

El primero y más directo, ya mencionado previamente en la sección 4.6.1.3, consistiría en incluir mayor cantidad de datos para el entrenamiento de la red, cubriendo todos los meses del año y de varios años (por ejemplo, Pardo and Malpica [2017] usaron datos de 6 años completos para el entrenamiento de su modelo). En este caso, además, valdría la pena añadir una variable auxiliar más correspondiente al mes del año sobre el que se están haciendo las predicciones. El principal inconveniente de incluir tantos datos es que los modelos tardarán del orden de varios días en ser entrenados, lo cual ralentiza enormemente la obtención de resultados y su posterior análisis. Es por esta razón que no ha podido ser abarcado en este trabajo.

El segundo punto tiene que ver con los valores dados a los diferentes hiperparámetros. Así pues resultaría interesante realizar una optimización automatizada de los hiperparámetros para los modelos únicos, similar a la que se llevó a cabo en la sección 4.3.1.1. Para ello se pueden usar herramientas como *Talos*[1], *Keras Tuner*[2] o *Hyperkeras*[3]. Estas librerías facilitan encontrar la mejor configuración de los hiperparámetros a través de la iteración y evaluación de las combinaciones de sus diferentes valores. De nuevo, ha resultado imposible abarcarlo en este trabajo debido a la gran inversión de tiempo necesaria para obtener resultados.

En tercer lugar, es posible mejorar la evaluación de los modelos y obtener así unos valores para las métricas más fiables. Para ello se tendrían que escoger los conjuntos de validación y test de manera aleatoria dentro del total de los datos, yendo más allá del acercamiento que se hizo en la sección 4.6.1.3. La dificultad de la puesta en práctica de esto se ve determinada por el hecho de estar usando secuencias consecutivas de los datos como entrada a los modelos, a lo que hay que añadir que no queremos mezclar datos que pertenezcan a un determinado conjunto, ya

[1] Más información en el repositorio del proyecto Talos: https://github.com/autonomio/talos

[2] Más información en la página web del proyecto Keras Tuner: https://keras-team.github.io/keras-tuner/

[3] Más información en la página web del proyecto hyperkeras: http://maxpumperla.com/hyperas/

sean de entrenamiento, test o validación, con datos relativos a los otros conjuntos. Además, una vez realizado el desordenamiento de estos datos sería más difícil la interpretación de las gráficas de predicción y no sería posible una hipotética implementación de la red LSTM como *stateful*, descrita en la sección 4.3.1.2). La evaluación del modelo podría mejorarse también usando la técnica del *k-fold cross validation*, la cual requiere de mucho más tiempo de cómputo, pero daría una idea de cómo se comporta el modelo para cualquier franja temporal del año.

En cuarto lugar, no se ha abarcado la inclusión de datos de tráfico para estudiar su efecto en la capacidad predictiva de los modelos. Ciertamente, se trata de datos muy relacionados a la concentración de NO_2 (Briggs et al. [2000]) y ha sido utilizado con éxito en estudios anteriores (Zhang et al. [2020], Krishan et al. [2019]); de modo que la inclusión de éstos es de esperar que mejore de manera significativa el poder predictivo del modelo. Sin embargo, similarmente a lo que ocurre con los datos meteorológicos a la hora de hacer predicciones en tiempo real, no se dispone de más que de los datos del instante en el cual se hace la predicción o el histórico de datos registrados en horas anteriores. Tal y como vimos en la sección 4.7.3, previsiblemente el usar datos de tráfico que no correspondan con los del instante en el cual se está haciendo la predicción, no reportaría resultados mejores para predicciones más allá de una hora. Lo cual es de mayor preocupación para el caso de los datos de tráfico, puesto que desafortunadamente no disponemos de pronósticos del tráfico tan concretos como los que sí existen para los datos de meteorología. No obstante, en este caso se podría experimentar usando los datos históricos de tráfico que hubo 24 horas antes a cuando se hace la predicción y estudiar si se consigue una mejora significativa mediante el uso de estos datos históricos de tráfico.

Por quinto y último lugar, existe la posibilidad de normalizar los datos de entrada de las medidas de los contaminantes antes de ser introducidos en el modelo. Si bien es cierto que, como práctica habitual, se suele realizar una normalización indiscriminada a todos los datos de entrada antes de introducirlos en las redes neuronales; en unas primeras pruebas iniciales (sección 4.3.1.1) no se mostró mejoría aparente de los resultados y se dejó apartada la subsiguiente exploración de esta técnica sobre los datos de calidad del aire. No obstante, sí se ha realizado esta transformación para los datos de meteorología, dado que aquí si encontramos datos mezclados con magnitudes muy dispares (velocidad del viento, temperatura, humedad, etc.).

Glosario

AEMET Agencia Estatal de Meteorología. 13, 14

API Application Programming Interface (Interfaz de Programación de Aplicaciones en castellano). 13, 21

CTM Chemical Transport Models. 8

LSTM Long Short-Term Memory Networks. 9, 10, 21, 49

MAPE Mean Absolute Percentage Error. 5, 22

PM Particular Matter (Material Particulado en castellano). 8, 9

PMF Persistence Model Forecast. 23, 28, 30

RMSE Round Mean Square Error. 5, 22, 25

SVM Support Vector Machines. 23

SVR Support Vector Regression. 28, 30

TFM Trabajo de Fin de Máster. 3, 6, 21, 22, 49, 50

Bibliografía

J.M. Baldasano, M.T. Pay, O. Jorba, S. Gassó, and P. Jiménez-Guerrero. An annual assessment of air quality with the caliope modeling system over spain. *Science of The Total Environment*, 409(11):2163 – 2178, 2011. ISSN 0048-9697. doi: https://doi.org/10.1016/j.scitotenv.2011.01.041. URL http://www.sciencedirect.com/science/article/pii/S0048969711000787.

José María Baldasano, Oriol Jorba, Santiago Gassó, María Teresa Pay, and Gustavo Arevalo. Caliope: sistema de pronóstico operacional de calidad del aire para europa y españa. 2012. URL https://repositorio.aemet.es/bitstream/20.500.11765/5767/1/Baldasano_etal.pdf.

Jaime Benavides, Albert Soret, Marc Guevara, Carlos Pérez-García Pando, Michelle Snyder, Fulvio Amato, Xavier Querol, and Oriol Jorba. Potential impact of a low emission zone on street-level air quality in barcelona city using caliope-urban model. In Clemens Mensink, Wanmin Gong, and Amir Hakami, editors, *Air Pollution Modeling and its Application XXVI*, pages 171–176, Cham, 2020. Springer International Publishing. ISBN 978-3-030-22055-6.

Gianluca Bontempi, Souhaib Ben Taieb, and Yann-Aël Le Borgne. *Machine Learning Strategies for Time Series Forecasting*, pages 62–77. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013. ISBN 978-3-642-36318-4. doi: 10.1007/978-3-642-36318-4_3. URL https://doi.org/10.1007/978-3-642-36318-4_3.

David J Briggs, Cornelis de Hoogh, John Gulliver, John Wills, Paul Elliott, Simon Kingham, and Kirsty Smallbone. A regression-based method for mapping traffic-related air pollution: application and testing in four contrasting urban environments. *Science of the Total Environment*, 253(1-3):151–167, 2000. ISSN 0048-9697. doi: https://doi.org/10.1016/S0048-9697(00)00429-0.

Jordi Casas Roma. *Introducción al análisis de series temporales*. Universitat Oberta de Catalunya, 02 2020.

François Chollet. *Deep Learning with Python*. Manning, December 2017.

François Chollet et al. Keras. https://keras.io, 2015.

Paul Chovin and André Roussel. *La Polución Atmosférica*. oikos-tau ediciones, 1970.

GBD 2016 Risk Factors Collaborators. Global, regional, and national comparative risk assessment of 84 behavioural, environmental and occupational, and metabolic risks or clusters of risks, 1990–2016: a systematic analysis for the global burden of disease study 2016. *The Lancet*, 390(10100):1345 – 1422, 2017. ISSN 0140-6736. doi: https://doi.org/10.1016/S0140-6736(17)32366-8. URL http://www.sciencedirect.com/science/article/pii/S0140673617323668.

EEA2005. Environment and health. Technical report, European Environment Agency, 2005. URL https://www.eea.europa.eu/publications/eea_report_2005_10/at_download/file.

EEA2019. Air quality in Europe — 2019 report. Technical report, European Environment Agency, october 2019.

Yarin Gal. *Uncertainty in Deep Learning*. PhD thesis, University of Cambridge, 2016.

Alex Graves. Long short-term memory. In *Supervised sequence labelling with recurrent neural networks*, pages 37–45. Springer, 2012.

Jiawei Han, Micheline Kamber, and Jian Pei. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 3rd edition, 2011. ISBN 0123814790.

Chiou-Jye Huang and Ping-Huan Kuo. A deep cnn-lstm model for particulate matter (pm2.5) forecasting in smart cities. *Sensors*, 18(7), 2018. ISSN 1424-8220. doi: 10.3390/s18072220. URL https://www.mdpi.com/1424-8220/18/7/2220.

Informe de la calidad del aire en la ciudad de Madrid durante 2019. La calidad del aire en la ciudad de Madrid durante 2019. Technical report, Ecologistas en Acción, enero 2020. URL https://www.ecologistasenaccion.org/133360/.

Mrigank Krishan, Srinidhi Jha, Jew Das, Avantika Singh, Manish Kumar Goyal, and Chandrra Sekar. Air quality modelling using long short-term memory (lstm) over nct-delhi, india. *Air Quality, Atmosphere & Health*, 12(8):899–908, 2019. doi: 10.1007/s11869-019-00696-7.

Xiang Li, Ling Peng, Xiaojing Yao, Shaolong Cui, Yuan Hu, Chengzeng You, and Tianhe Chi. Long short-term memory neural network for air pollutant concentration predictions: Method development and evaluation. *Environmental Pollution*, 231:997 – 1004, 2017. ISSN 0269-7491.

doi: https://doi.org/10.1016/j.envpol.2017.08.114. URL http://www.sciencedirect.com/science/article/pii/S0269749117307534.

Xiaolei Ma, Zhimin Tao, Yinhai Wang, Haiyang Yu, and Yunpeng Wang. Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. *Transportation Research Part C: Emerging Technologies*, 54:187 – 197, 2015. ISSN 0968-090X. doi: https://doi.org/10.1016/j.trc.2015.03.014. URL http://www.sciencedirect.com/science/article/pii/S0968090X15000935.

Esteban Pardo and Norberto Malpica. Air quality forecasting in madrid using long short-term memory networks. In José Manuel Ferrández Vicente, José Ramón Álvarez-Sánchez, Félix de la Paz López, Javier Toledo Moreo, and Hojjat Adeli, editors, *Biomedical Applications Based on Natural and Artificial Computing*, pages 232–239, Cham, 2017. Springer International Publishing. ISBN 978-3-319-59773-7.

Yanlin Qi, Qi Li, Hamed Karimian, and Di Liu. A hybrid model for spatiotemporal forecasting of PM2.5 based on graph convolutional neural network and long short-term memory. *SCIENCE OF THE TOTAL ENVIRONMENT*, 664:1–10, MAY 10 2019. ISSN 0048-9697. doi: {10.1016/j.scitotenv.2019.01.333}.

Francisco Rodriguez-Sanchez, Antonio Pérez-Luque, Ignasi Bartomeus, and Sara Varela. Ciencia reproducible: qué, por qué, cómo. *Revista Ecosistemas*, 25(2), 2016. ISSN 1697-2473. URL https://www.revistaecosistemas.net/index.php/ecosistemas/article/view/1178.

John B. Rollins. *Metodología Fundamental para la Ciencia de Datos*. IBM Analytics, 2015.

Yves Rybarczyk and Rasa Zalakeviciute. Machine learning approaches for outdoor air quality modelling: A systematic review. *Applied Sciences*, 8(12), 2018. ISSN 2076-3417. doi: 10.3390/app8122570. URL https://www.mdpi.com/2076-3417/8/12/2570.

Ramón Sangüesa i Solé. El proceso de descubrimiento de conocimiento a partir de datos. In *Data mining*. Universitat Oberta de Catalunya, 2010a.

Ramón Sangüesa i Solé. Preparación de datos. In *Data mining*. Universitat Oberta de Catalunya, 2010b.

WHO2018. Burden of disease from the joint effects of household and ambient air pollution for 2016. Technical report, World Health Organization, 2018. URL http://www.who.int/airpollution/data/AP_joint_effect_BoD_results_May2018.pdf.

Qi Zhang, Jacqueline CK Lam, Victor OK Li, and Yang Han. Deep-air: A hybrid cnn-lstm framework forfine-grained air pollution forecast. *arXiv preprint arXiv:2001.11957*, 2020.

Apéndice A

Reproducibilidad

A continuación se encuentran listados todos los elementos necesarios poder reproducir el presente trabajo.

A.1. Conjuntos de datos

En esta sección se listan los enlaces a los conjuntos de datos usados en este trabajo.

1. Histórico de datos de la calidad del aire de Madrid:
   https://www.kaggle.com/abeserra/calidad-del-aire-madrid

2. Histórico de datos meteorológicos Madrid:
   https://www.kaggle.com/abeserra/meterologa-madrid

A.2. Código de los notebooks

En esta sección se listan los enlaces al código de los notebooks escritos en Python para los diferentes análisis y modelos usados en este trabajo.

Todo el código fuente listado a continuación y alojado en la plataforma *kaggle* está disponible bajo las licencias Apache 2.0 y GPLv3. Cualquier tercero podrá acogerse a cualquiera de estas dos licencias para el uso de este código.

1. Análisis estadístico y visual de los datos de calidad del aire:
   https://www.kaggle.com/abeserra/seleccion-datos-1-estacion-1-contaminante?scriptVersionId=34297287

2. Análisis de la serie temporal
   https://www.kaggle.com/abeserra/analisis-serie-temporal-1-estacion-1-contaminan?scriptVersionId=34302678

3. Análisis estadístico y visual de los datos meteorológicos
   https://www.kaggle.com/abeserra/seleccion-datos-meteo-primavera-2019-1-estacion?scriptVersionId=35443346

4. Modelos base SVR y PMF:

   a) Usando datos únicamente de septiembre 2019:
   https://www.kaggle.com/abeserra/modelos-base-pmf-svr-30-dias?scriptVersionId=33889942

   b) Usando datos de febrero a mayo de 2019:
   https://www.kaggle.com/abeserra/modelos-base-pmf-svr-4-meses?scriptVersionId=33898662

5. Prototipo 0 - Primeras pruebas con 1 mes de datos:
   https://www.kaggle.com/abeserra/prototipo-0-1-prueba?scriptVersionId=32957281

6. Prototipo 0 - Optimización con Talos:
   https://www.kaggle.com/abeserra/prototipo-0-talos-experimentos

7. Prototipo 0 - Mejor configuración con 1 mes de datos:
   https://www.kaggle.com/abeserra/prototipo-0-2-1-ajuste-par-metros-refactor?scriptVersionId=34397640

8. Prototipo 0 - *LSTM Stateful*:
   https://www.kaggle.com/abeserra/prototipo-0-3-predicci-n-1h-univariante-stateful?scriptVersionId=33407268

9. Prototipo 0 - Mejor configuración con 4 meses de datos:
   https://www.kaggle.com/abeserra/prototipo-0-2-2-primavera-2019-refactor?scriptVe
   33856627

10. Prototipo 1 - Primera iteración:
    https://www.kaggle.com/abeserra/prototipo-1-1h-aux-inputs?scriptVersionId=34052669.

11. Prototipo 1 - Segunda iteración:
    https://www.kaggle.com/abeserra/prototipo-1-1h-aux-inputs?scriptVersionId=34057627.

12. Prototipo 1 - Tercera iteración:
https://www.kaggle.com/abeserra/prototipo-1-1h-aux-inputs?scriptVersionId=34060238.

13. Prototipo 2:
https://www.kaggle.com/abeserra/prototipo-2-1h-meteo-aux-inputs?scriptVersionId=34261208.

14. Predicciones a 24 horas:

    a) Modelo base SVR:
    https://www.kaggle.com/abeserra/modelos-base-pmf-svr-24-horas?scriptVersionI35321657.

    b) Prototipo 0:
    https://www.kaggle.com/abeserra/prototipo-0-24-24h-univariante

    c) Prototipo 1:
    https://www.kaggle.com/abeserra/prototipo-1-24-24h-aux-inputs

    d) Prototipo 2:
    https://www.kaggle.com/abeserra/prototipo-2-24-24h-meteo-aux-inputs

15. Predicciones a 8 y 16 horas:

    a) Modelo base SVR:
    https://www.kaggle.com/abeserra/modelos-base-pmf-svr-24-horas?scriptVersionI35321657.

    b) Prototipo 0:
    https://www.kaggle.com/abeserra/prototipo-0-x-8-16h

    c) Prototipo 1:
    https://www.kaggle.com/abeserra/prototipo-1-x-8-16h

    d) Prototipo 2:
    https://www.kaggle.com/abeserra/prototipo-2-x-8-16h

16. Modelos únicos para todas las estaciones:

    a) Modelo único 1:
    https://www.kaggle.com/abeserra/all-in-one-1-x

    b) Modelo único 2:
    https://www.kaggle.com/abeserra/all-in-one-2-x-meteo-sin-alinear

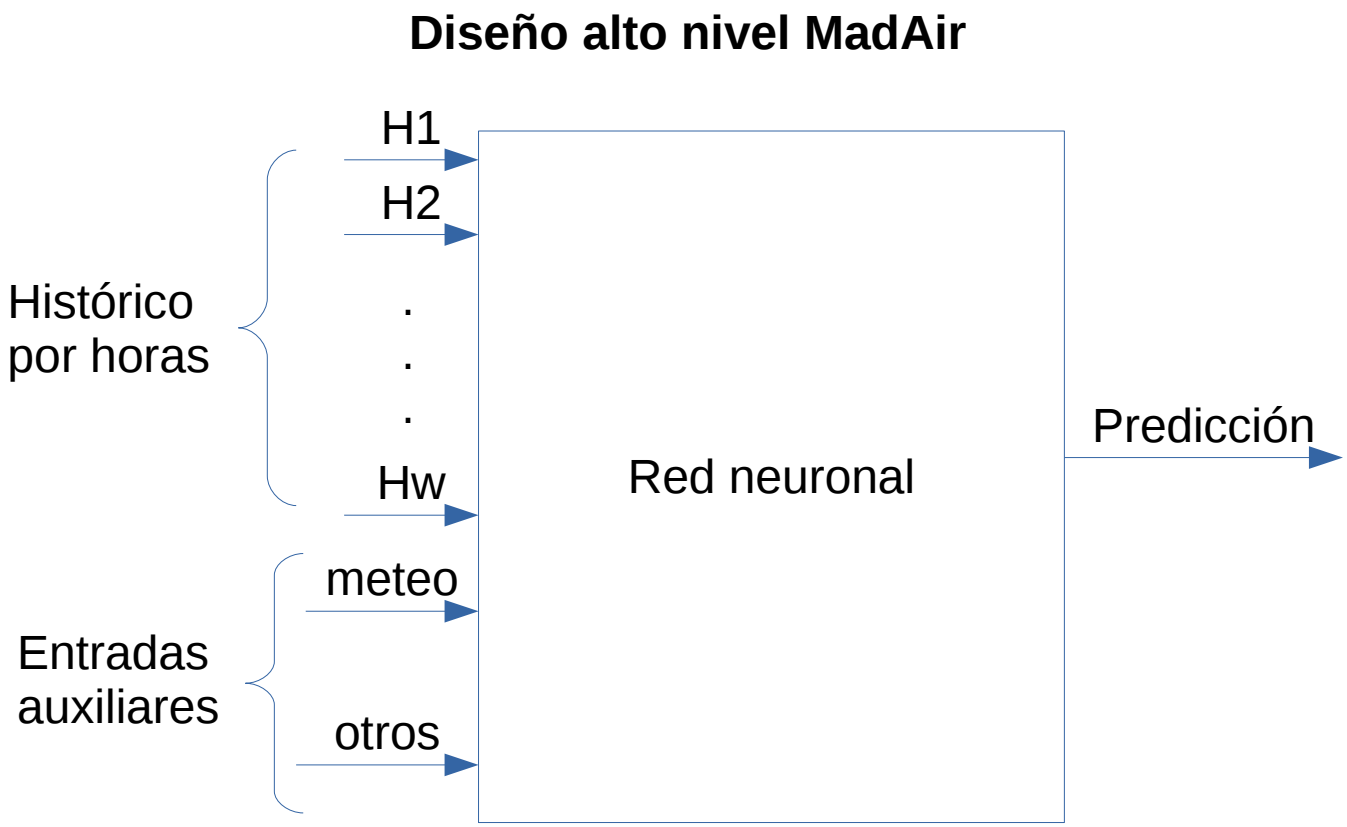
c) Modelo único 3:
https://www.kaggle.com/abeserra/all-in-one-3-x-meteo-alineada

Apéndice B

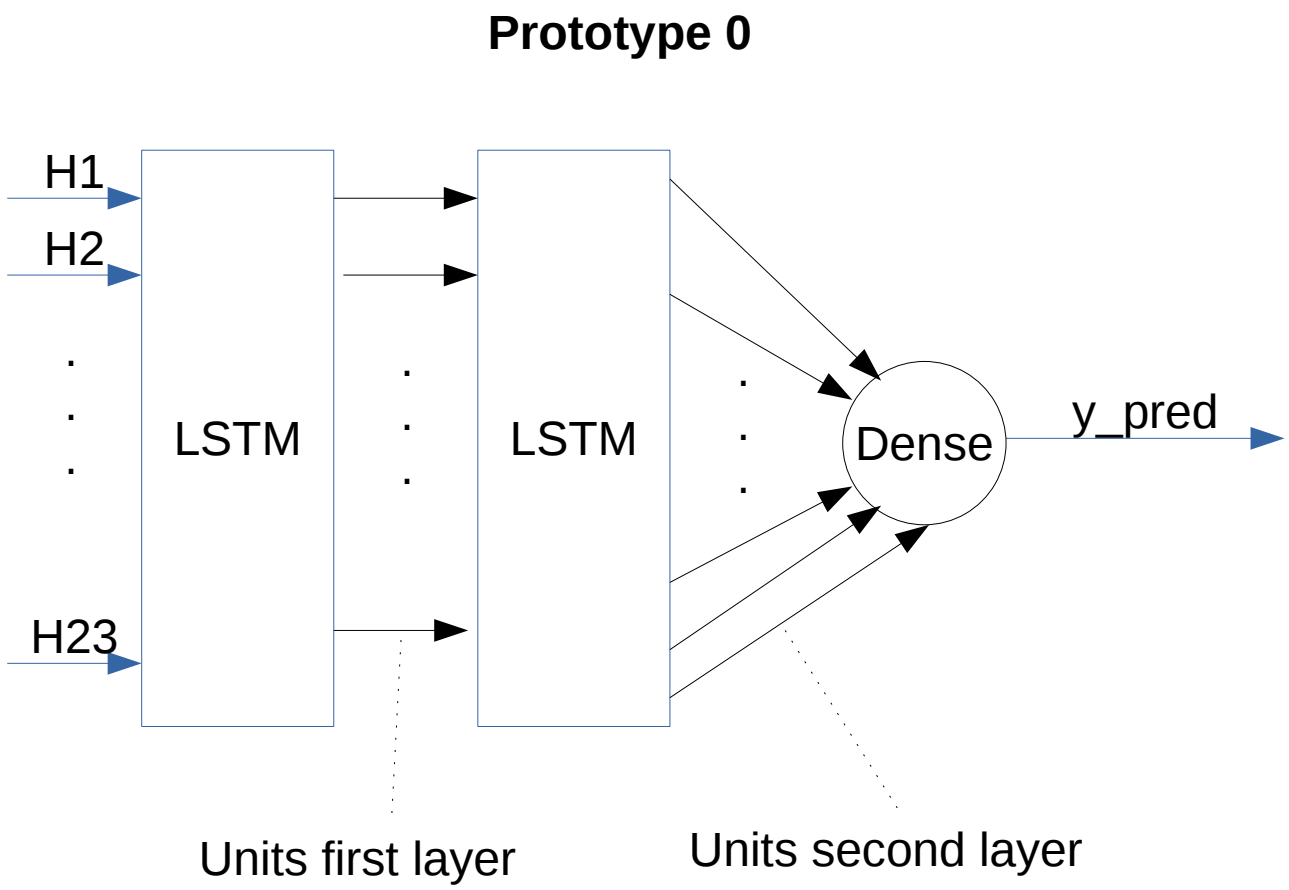
Diagramas de diseño

A continuación se encuentran los diagramas de diseño asociado a cada uno de los diferentes modelos que se han implementado en este trabajo.

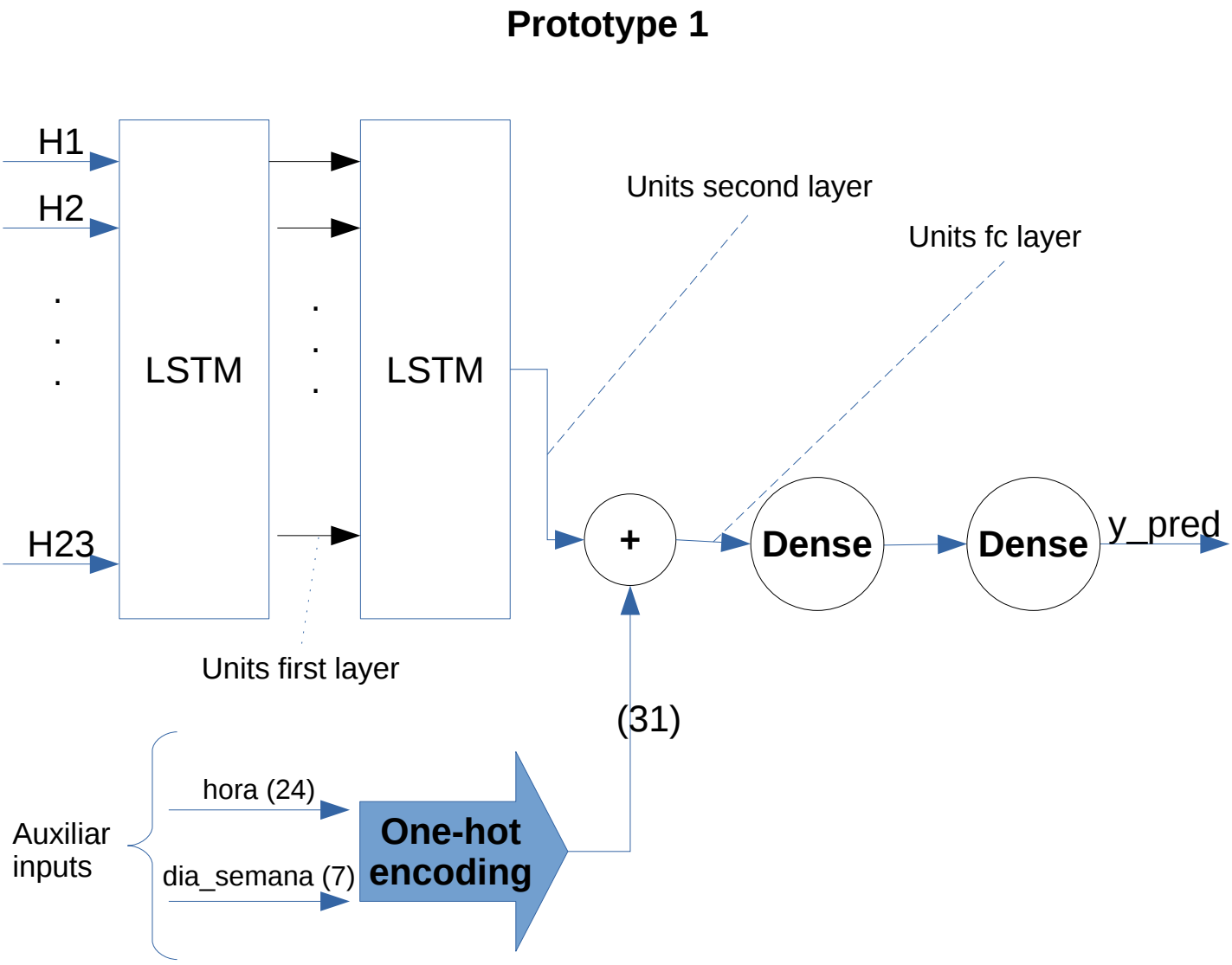
B.1. Diseño general



B.2. Diseño prototipo 0



B.3. Diseño prototipo 1



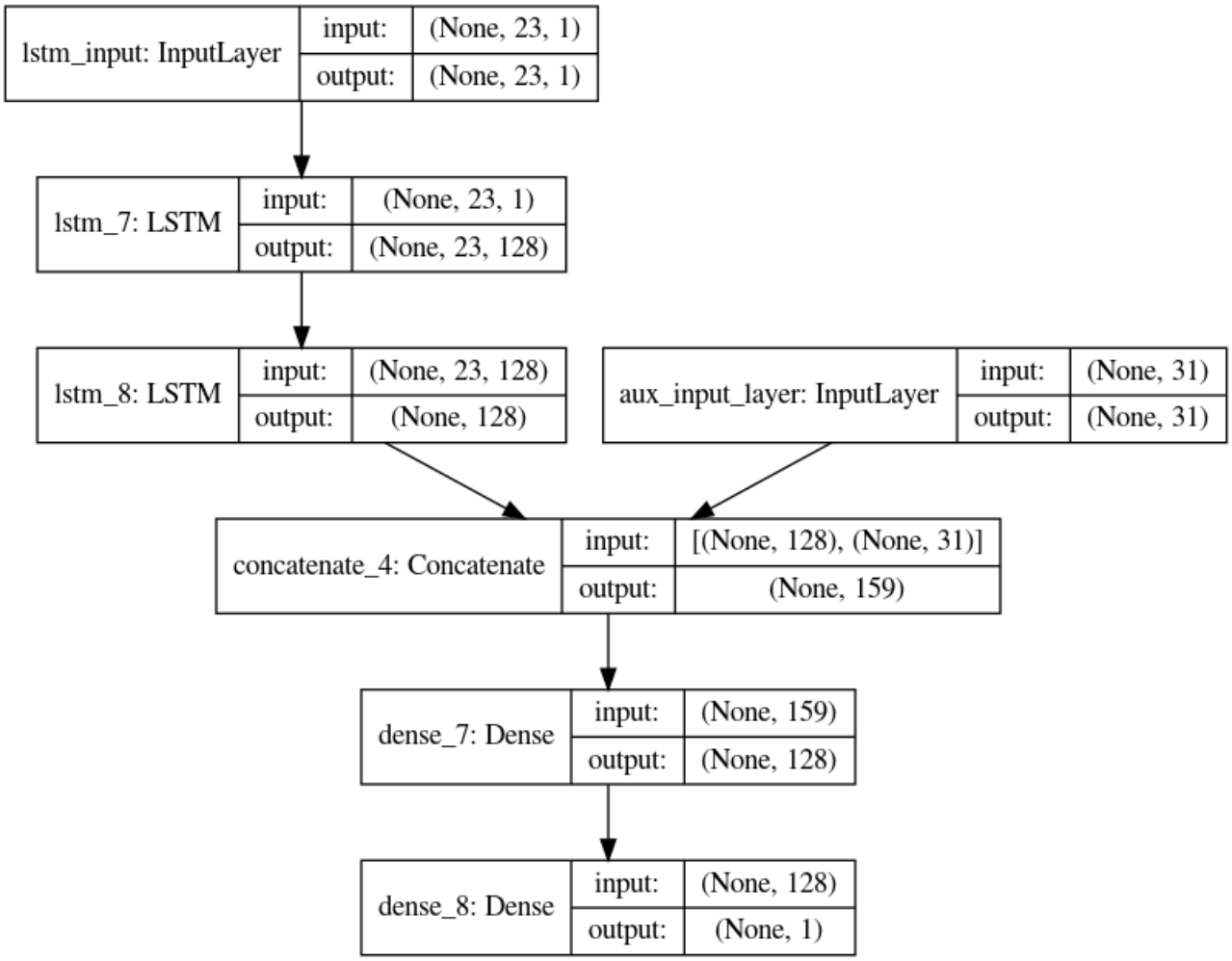
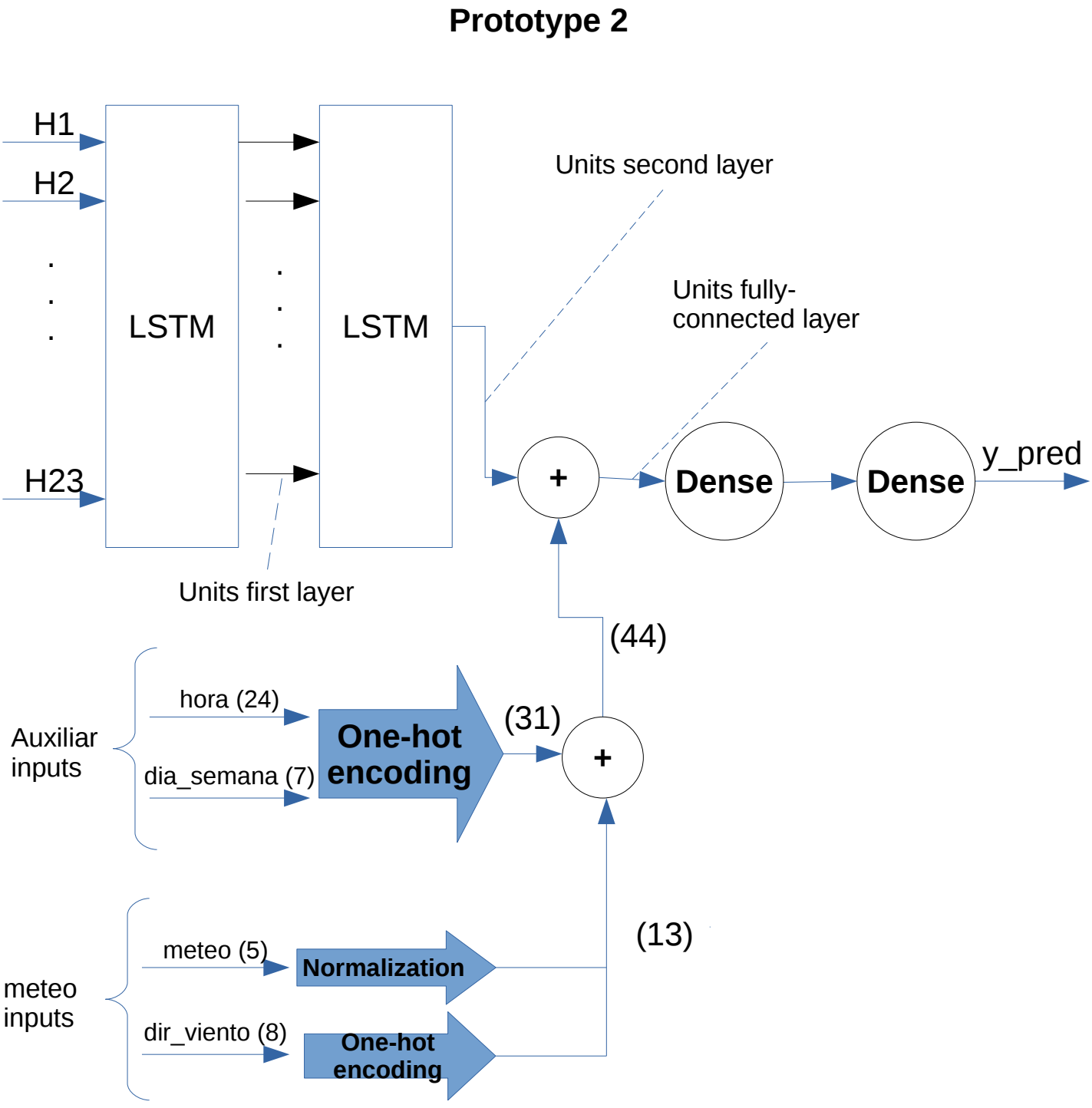


Figura B.1: Diagrama del modelo para el prototipo 1 generado por keras

B.4. Diseño prototipo 2



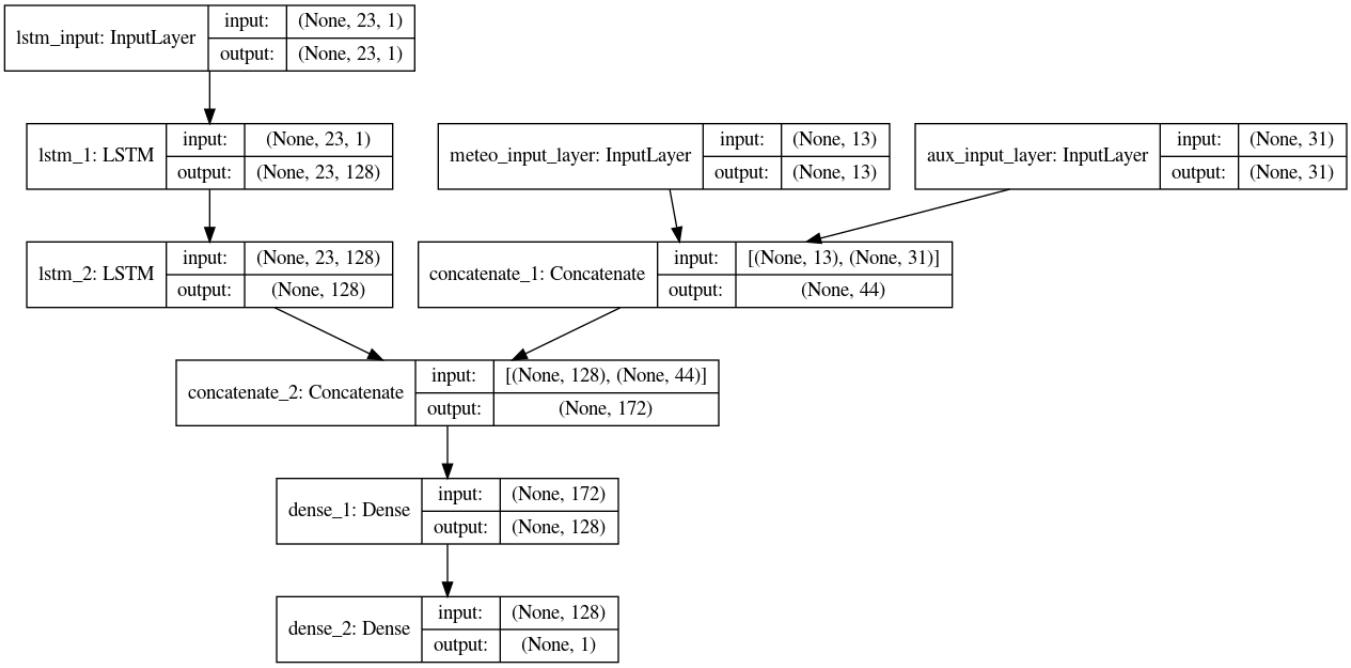
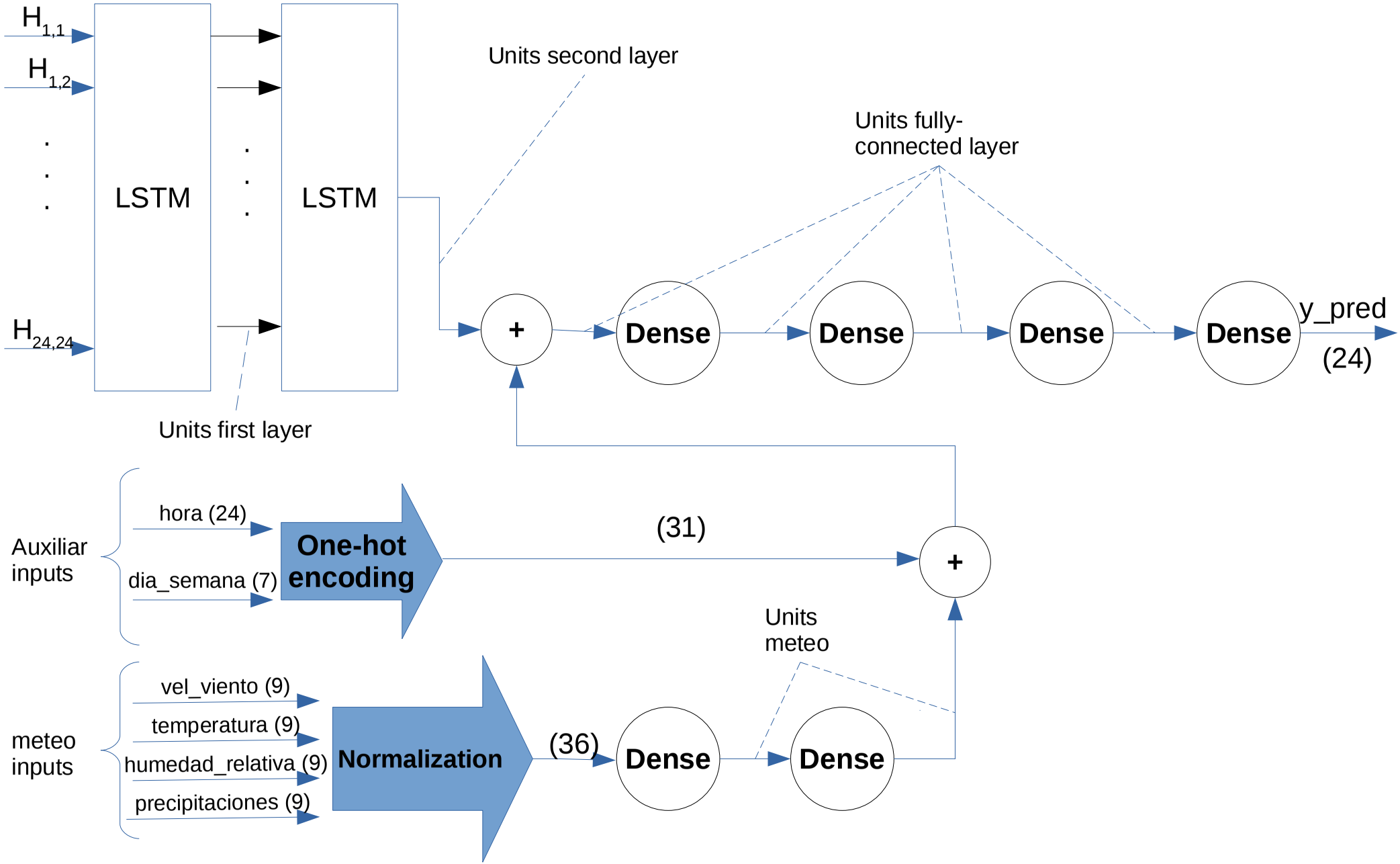


Figura B.2: Diagrama del modelo para el prototipo 2 generado por keras

B.5. Diseño modelos únicos 2 y 3



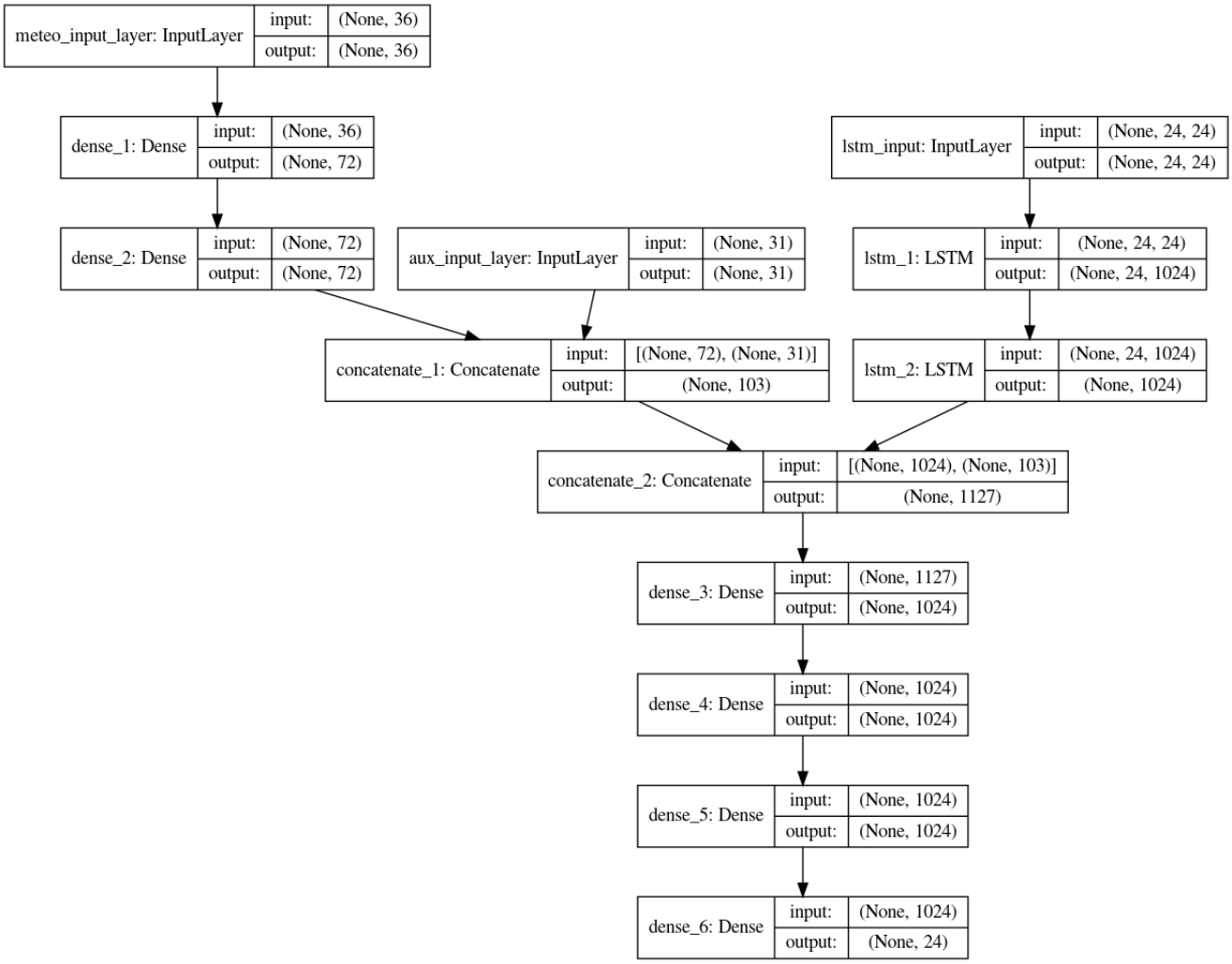


Figura B.3: Diagrama del modelo para el modelo único 2 generado por keras

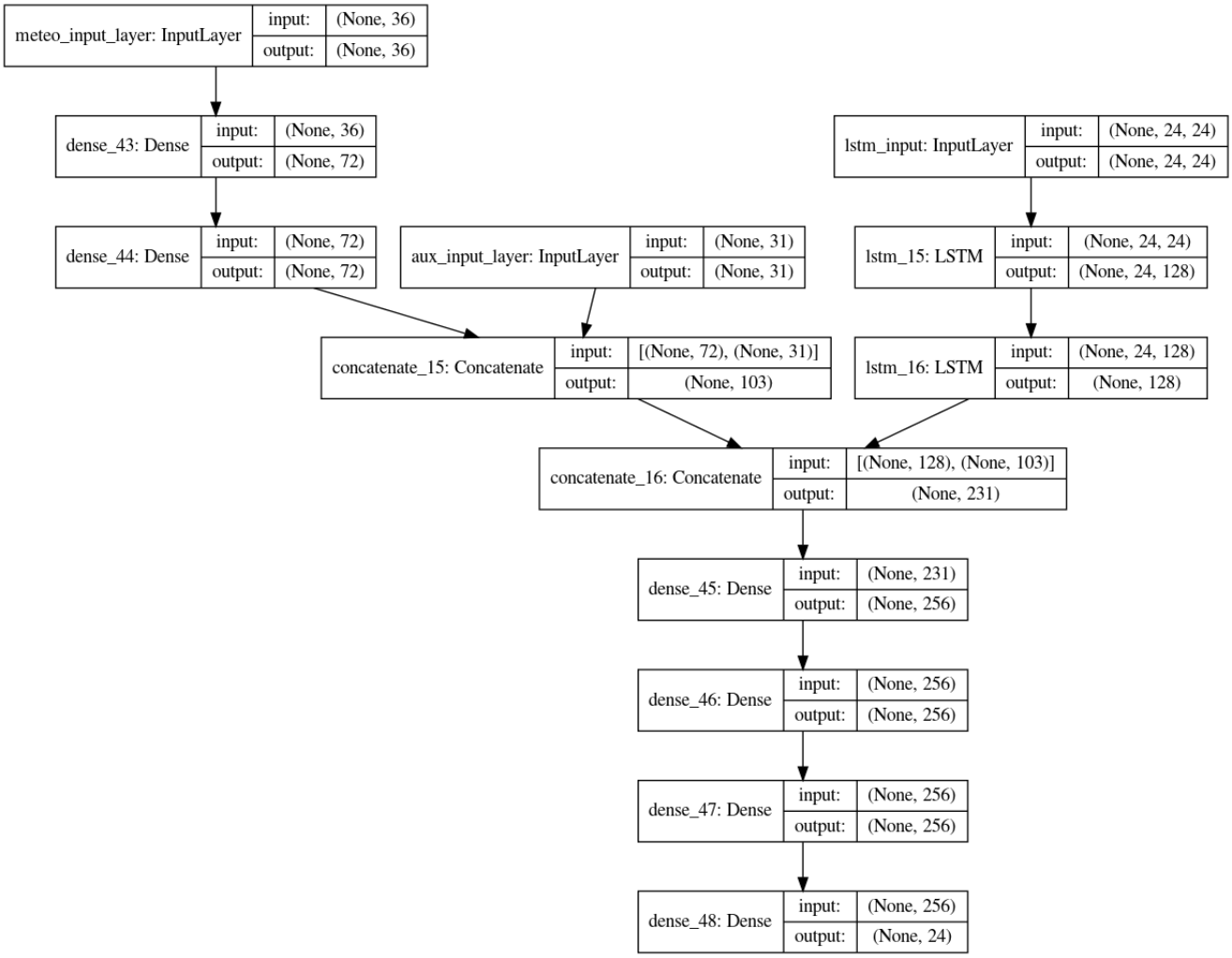


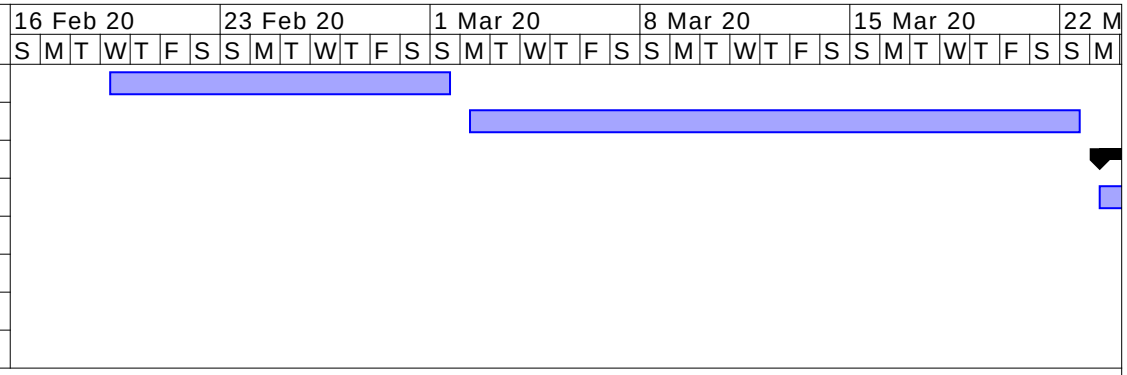
Figura B.4: Diagrama del modelo para el modelo único 3 generado por keras

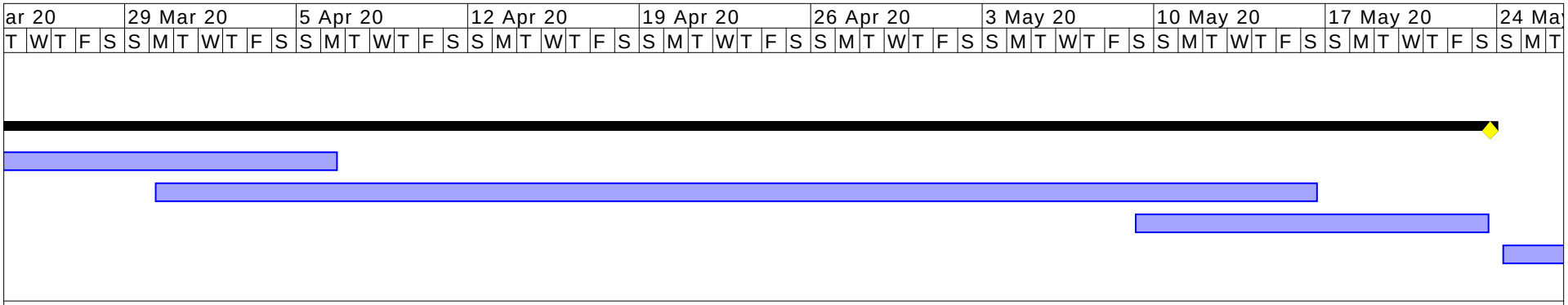
Apéndice C

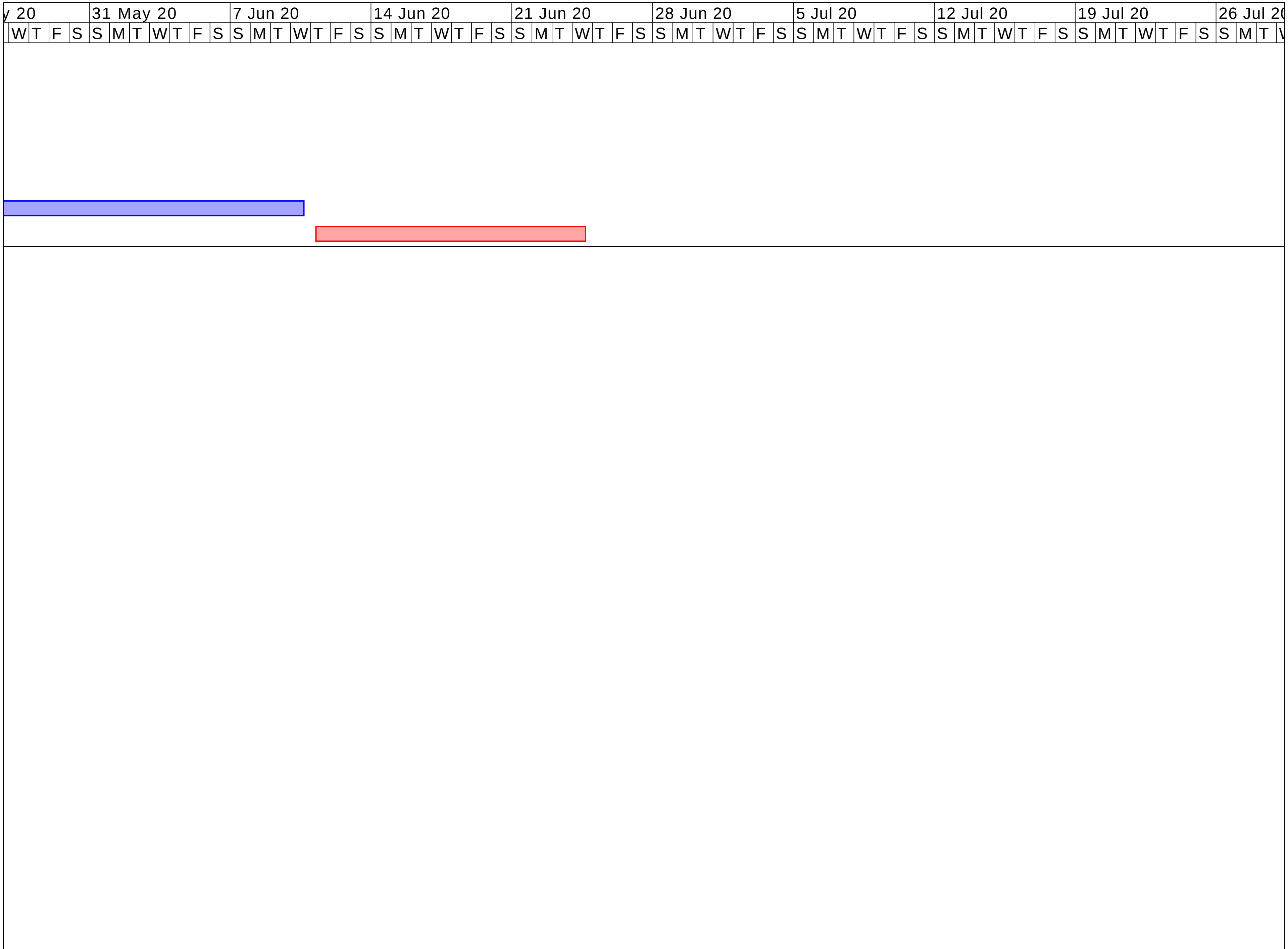
Diagramas de Gantt

A continuación se encuentran los diagramas de Gantt asociados a la planificación de este trabajo.

	Name
1	Definición y alcance del TFM
2	Lectura del estado del arte
3	**Diseño e implementación del trabajo**
4	Descarga y procesamiento de datos
5	Diseño e implementación del modelo predictivo
6	Implementación de un sistema para previsiones
7	Redacción de la memoria
8	Defensa del trabajo




ar 20
29 Mar 20
5 Apr 20
12 Apr 20
19 Apr 20
26 Apr 20
3 May 20
10 May 20
17 May 20
24 Ma
T W T F S S M T W T F S S M T W T F S S M T W T F S S M T W T F S S M T W T F S S M T W T F S S M T W T F S S M T W T F S S M T W T F S S M T
Definición y alcance del TFM - page2


31 May 20
7 Jun 20
14 Jun 20
21 Jun 20
28 Jun 20
5 Jul 20
12 Jul 20
19 Jul 20