



UNIVERSIDAD
DE LA REPÚBLICA
URUGUAY



FACULTAD DE
INGENIERÍA

Redes neuronales para modelado y predicción de polución en entornos urbanos

Informe de Proyecto de Grado presentado por

Josefina Cardozo Gonçalves

en cumplimiento parcial de los requerimientos para la graduación de la carrera
de Ingeniería en Computación de Facultad de Ingeniería de la Universidad de
la República

Supervisor

Sergio Nesmachnow

Montevideo, 13 de abril de 2026



Redes neuronales para modelado y predicción de polución en entornos urbanos por Josefina Cardozo Gonçalves tiene licencia [CC Atribución - No Comercial - Sin Derivadas 4.0](https://creativecommons.org/licenses/by-nc-nd/4.0/).

Agradecimientos

Agradezco a mi tutor, Sergio Nesmachnow, por su dedicación, su guía y sus aportes durante todo el desarrollo del proyecto de grado, que fueron fundamentales para poder llevarlo adelante.

Agradezco a los docentes de la carrera que, a lo largo de los años, contribuyeron a mi formación y me dieron las herramientas para llegar hasta aquí.

A mis padres, a mi hermana y a Lara por su apoyo incondicional, su paciencia y el acompañamiento durante todo el proceso y en el transcurso de la carrera.

Finalmente, agradezco a mi familia y amigos por el apoyo y motivación constante.

Resumen

La contaminación atmosférica es uno de los principales factores de riesgo ambiental para la salud humana. Las partículas finas ($PM_{2,5}$), el dióxido de nitrógeno (NO_2) y el ozono (O_3) se encuentran entre los contaminantes responsables de la mayoría de las muertes prematuras asociadas a la contaminación del aire. Por lo tanto, la predicción de sus concentraciones es de gran importancia para la salud humana. En este proyecto de grado se propone un enfoque basado en aprendizaje profundo para la predicción de contaminantes atmosféricos utilizando datos multivariantes. Se desarrollan modelos codificador–decodificador basados en redes Long Short-Term Memory (LSTM) para capturar dependencias temporales, y un modelo espaciotemporal que integra redes convolucionales sobre grafos (GCN) con LSTM para modelar relaciones espaciales entre estaciones de monitoreo. El enfoque se evalúa utilizando datos abiertos de la red de monitoreo de calidad del aire de Madrid, que incluyen observaciones horarias de $PM_{2,5}$, NO_2 y O_3 , junto con otros contaminantes y variables meteorológicas. El desempeño de los modelos se evalúa mediante Mean Absolute Error (MAE) y Root Mean Squared Error (RMSE), y se compara con un modelo base de perceptrón multicapa. Los resultados mostraron que los modelos propuestos superaron al modelo base utilizado como referencia. En el horizonte de una hora, LSTM logró reducciones del MAE de hasta 58%. En el horizonte de 24 horas, GCN–LSTM obtuvo el mejor desempeño global con errores hasta un 50% menores respecto al modelo base.

Índice de figuras

5.1. Distribución geográfica de las estaciones de monitoreo del Ayuntamiento de Madrid (datos de datos.madrid.es).	18
5.2. Histogramas de los contaminantes objetivo.	22
5.3. Patrones semanales de los contaminantes objetivo.	24
6.1. Arquitectura del modelo perceptrón multicapa.	30
6.2. Arquitectura del modelo codificador–decodificador LSTM.	31
6.3. Estructura espacial del grafo empleado en el modelo GCN–LSTM, con conexiones entre estaciones (datos de datos.madrid.es).	33
6.4. Arquitectura del modelo GCN–LSTM codificador–decodificador.	35
7.1. Esquema de generación de las ventanas temporales deslizantes.	38
7.2. Convergencia de la función de pérdida del modelo MLP en la estación urbana de tráfico Escuelas Aguirre	44
7.3. Predicciones a una hora del modelo MLP en la estación urbana de tráfico Escuelas Aguirre (— : concentraciones reales, — : concentraciones predichas)	46
7.4. Predicciones a 24 horas del modelo MLP en la estación urbana de tráfico Escuelas Aguirre (— : concentraciones reales, — : concentraciones predichas)	47
7.5. Convergencia de la función de pérdida del modelo LSTM en la estación urbana de tráfico Escuelas Aguirre	48
7.6. Predicciones a una hora del modelo LSTM codificador–decodificador en la estación urbana de tráfico Escuelas Aguirre (— : concentraciones reales, — : concentraciones predichas)	49
7.7. Predicciones a 24 horas del modelo LSTM codificador–decodificador en la estación urbana de tráfico Escuelas Aguirre (— : concentraciones reales, — : concentraciones predichas)	50
7.8. Convergencia de la función de pérdida del modelo GCN–LSTM en la estación urbana de tráfico Escuelas Aguirre para $PM_{2,5}$	50
7.9. Predicciones a una hora del modelo GCN–LSTM codificador–decodificador en la estación urbana de tráfico Escuelas Aguirre (— : concentraciones reales, — : concentraciones predichas)	52

7.10. Predicciones a 24 horas del modelo GCN–LSTM codificador–decodificador en la estación urbana de tráfico Escuelas Aguirre (— : concentraciones reales, — : concentraciones predichas)	52
B.1. Predicciones a una hora del modelo LSTM en la estación suburbana Casa de Campo (— : concentraciones reales, — : concentraciones predichas)	68
B.2. Predicciones a 24 horas del modelo LSTM en la estación suburbana Casa de Campo (— : concentraciones reales, — : concentraciones predichas)	69
B.3. Predicciones a una hora del modelo GCN–LSTM en la estación suburbana Casa de Campo (— : concentraciones reales, — : concentraciones predichas)	70
B.4. Predicciones a 24 horas del modelo GCN–LSTM en la estación suburbana Casa de Campo (— : concentraciones reales, — : concentraciones predichas)	71
B.5. Predicciones a una hora del modelo LSTM en la estación urbana fondo Ensanche de Vallecas (— : concentraciones reales, — : concentraciones predichas)	72
B.6. Predicciones a 24 horas del modelo LSTM en la estación urbana fondo Ensanche de Vallecas (— : concentraciones reales, — : concentraciones predichas)	72
B.7. Predicciones a una y 24 horas del modelo LSTM en la estación urbana fondo Sanchinarro (— : concentraciones reales, — : concentraciones predichas)	73
B.8. Predicciones a una hora del modelo GCN–LSTM en la estación urbana fondo Ensanche de Vallecas (— : concentraciones reales, — : concentraciones predichas)	74
B.9. Predicciones a 24 horas del modelo GCN–LSTM en la estación urbana fondo Ensanche de Vallecas (— : concentraciones reales, — : concentraciones predichas)	74
B.10. Predicciones a una y 24 horas del modelo GCN–LSTM en la estación urbana fondo Sanchinarro (— : concentraciones reales, — : concentraciones predichas)	75

Índice de cuadros

4.1. Comparación de trabajos relacionados sobre la predicción de la contaminación del aire, en cuanto a los contaminantes considerados, los datos adicionales, las estaciones utilizadas ($\#e$), el período de recolección de datos (<i>período</i>), el horizonte de predicción (<i>ventana</i>) y los modelos empleados.	16
5.1. Clasificación de las estaciones según su entorno.	19
5.2. Estadísticos descriptivos de los contaminantes seleccionados.	20
5.3. Estadísticos descriptivos por estación ($\#e$) para los contaminantes medidos.	21
5.4. Rangos físicos plausibles para cada variable según literatura y normativa.	26
7.1. Espacio de hiperparámetros considerado.	41
7.2. Valores de hiperparámetros considerados para el modelo LSTM.	41
7.3. Hiperparámetros seleccionados en los horizontes de predicción.	42
7.4. Valores de hiperparámetros considerados para el modelo GCN–LSTM.	42
7.5. Hiperparámetros seleccionados en los horizontes de predicción.	43
7.6. MAE y RMSE del modelo MLP para horizontes de una y 24 horas en la estación Escuelas Aguirre ($\mu\text{g}/\text{m}^3$).	45
7.7. MAE y RMSE del modelo LSTM codificador–decodificador para horizontes de 1 h y 24 h en la estación Escuelas Aguirre ($\mu\text{g}/\text{m}^3$).	48
7.8. MAE y RMSE del modelo GCN–LSTM codificador–decodificador para horizontes de 1 h y 24 h en la estación Escuelas Aguirre ($\mu\text{g}/\text{m}^3$).	51
7.9. Resultados validación cruzada a una hora en la estación Escuelas Aguirre ($\mu\text{g}/\text{m}^3$)	53
7.10. Resultados de la validación cruzada a 24 horas en la estación Escuelas Aguirre ($\mu\text{g}/\text{m}^3$)	54
A.1. Resultados de la búsqueda aleatoria de hiperparámetros para el modelo LSTM en la estación Escuelas Aguirre en el horizonte de una hora.	62

A.2. Resultados de la búsqueda aleatoria de hiperparámetros para el modelo LSTM en la estación Escuelas Aguirre en el horizonte de 24 horas.	63
A.3. Resultados de la búsqueda aleatoria de hiperparámetros para el modelo GCN-LSTM en el horizonte de una hora.	64
A.4. Resultados de la búsqueda aleatoria de hiperparámetros para el modelo GCN-LSTM en el horizonte de 24 horas.	65
C.1. MAE y RMSE para $PM_{2,5}$ en todas las estaciones para horizontes de una y 24 horas ($\mu g/m^3$).	77
C.2. MAE y RMSE para NO_2 en todas las estaciones para horizontes de una y 24 horas ($\mu g/m^3$).	78
C.3. MAE y RMSE para O_3 en todas las estaciones para horizontes de una y 24 horas ($\mu g/m^3$).	79

Índice general

1. Introducción	1
2. Fundamentos teóricos	3
2.1. Aprendizaje automático	3
2.1.1. Clasificación de los algoritmos de aprendizaje automático	3
2.1.1.1. Aprendizaje supervisado	3
2.1.1.2. Aprendizaje no supervisado	4
2.1.2. Redes neuronales artificiales	4
2.1.2.1. Arquitectura de las redes neuronales	4
2.2. Modelado secuencial	5
2.2.1. Modelos secuencia a secuencia	5
2.2.2. Arquitectura codificador–decodificador	5
2.3. Modelado temporal	6
2.3.1. Series temporales y dependencias temporales	6
2.3.2. Redes neuronales recurrentes	6
2.3.2.1. Redes LSTM	7
2.4. Modelado espacial	8
2.4.1. Redes neuronales convolucionales	8
2.4.2. Redes neuronales gráficas	8
2.4.2.1. Redes convolucionales sobre grafos	9
3. Metodología	11
3.1. Estrategia de desarrollo	11
3.2. Criterios metodológicos	12
4. Estado del arte	13
5. Presentación y análisis de los datos	17
5.1. Selección de los datos	17
5.2. Preprocesamiento de los datos	24
5.2.1. Integración de los datos	24
5.2.2. Detección y manejo de valores atípicos	25
5.2.3. Valores faltantes	26
5.3. Transformación de los datos	27

6. Modelos predictivos	29
6.1. Modelos de redes neuronales	29
6.1.1. Perceptrón multicapa	29
6.1.2. LSTM codificador–decodificador	30
6.1.3. GCN-LSTM codificador–decodificador	31
7. Experimentación	37
7.1. Entorno de desarrollo y ejecución	37
7.2. Configuración experimental	37
7.2.1. Esquema experimental	38
7.3. Selección de hiperparámetros	39
7.3.1. Estrategia de búsqueda	39
7.3.2. Espacio de hiperparámetros	40
7.3.3. Resultados de la selección de hiperparámetros	41
7.3.3.1. Modelo LSTM	41
7.3.3.2. Modelo GCN–LSTM	42
7.4. Esquema de evaluación	43
7.5. Análisis de resultados	44
7.5.1. Modelo Perceptrón multicapa	44
7.5.2. Modelo LSTM codificador–decodificador	47
7.5.3. Modelo GCN–LSTM codificador–decodificador	49
7.5.4. Validación cruzada temporal	53
8. Consideraciones finales	55
8.1. Conclusiones	55
8.2. Trabajo futuro	56
Referencias bibliográficas	59
A. Selección de hiperparámetros	61
A.1. Modelo LSTM	61
A.2. Modelo GCN–LSTM	64
B. Resultados gráficos	67
B.1. Estaciones suburbanas	67
B.1.1. LSTM	67
B.1.2. GCN–LSTM	70
B.2. Estaciones urbanas fondo	72
B.2.1. LSTM	72
B.2.2. GCN–LSTM	74
C. MAE y RMSE por estación	77
C.1. PM _{2,5}	77
C.2. NO ₂	78
C.3. O ₃	79

Capítulo 1

Introducción

La contaminación del aire es una de las principales amenazas ambientales para la salud humana a nivel mundial, según la Organización Mundial de la Salud (World Health Organization, 2021). El material particulado fino ($PM_{2,5}$), el dióxido de nitrógeno (NO_2) y el ozono (O_3) se encuentran entre los contaminantes más peligrosos, ya que provocan importantes problemas de salud y daños ambientales en las grandes ciudades. Por lo tanto, los pronósticos a corto plazo de sus concentraciones pueden resultar de gran utilidad para apoyar la toma de decisiones en políticas ambientales. En 2024, la Agencia Europea de Medio Ambiente informó que en 2022 estos contaminantes fueron responsables de aproximadamente 239.000 muertes prematuras debidas al $PM_{2,5}$, 70.000 debidas al O_3 y 48.000 al NO_2 en toda la Unión Europea (European Environment Agency, 2024).

Los modelos tradicionales lineales o estadísticos proporcionan referencias útiles para el análisis de series temporales, entre ellos el modelo de regresión logística (Aditya et al. 2018), ARIMA (Marinov et al. 2022) y SARIMA (Deepan y Saravanan, 2024). No obstante, estos métodos presentan limitaciones debido a la complejidad de las dependencias no lineales y espaciotemporales presentes en los datos ambientales. En la última década, el aprendizaje profundo ha sido utilizado ampliamente para desarrollar modelos capaces de capturar patrones no lineales complejos en datos ambientales.

En esta línea de trabajo, este proyecto de grado presenta un enfoque basado en redes neuronales profundas para la predicción de contaminantes atmosféricos. Se proponen dos arquitecturas codificador–decodificador. La primera se basa en redes neuronales recurrentes Long Short-Term Memory (LSTM) diseñadas para capturar dependencias temporales. La segunda corresponde a una extensión híbrida espaciotemporal que integra redes convolucionales sobre grafos (GCN) con LSTM para modelar las relaciones espaciales entre las estaciones de monitoreo. Como referencia de desempeño se utilizó un perceptrón multicapa (MLP). El enfoque propuesto se aplicó a un caso de estudio basado en datos abiertos provenientes de la red de monitoreo de calidad del aire de Madrid, España (Tou-touh et al. 2020). El conjunto de datos incluye observaciones horarias de $PM_{2,5}$,

NO₂ y O₃, y variables meteorológicas.

Los resultados obtenidos mostraron que el uso de información temporal y espacial permite mejorar las predicciones de los modelos de calidad del aire. En términos de Mean Absolute Error (MAE), en el horizonte de una hora LSTM obtuvo los mejores resultados, mostrando mejoras respecto al perceptrón multicapa de hasta 58 %. En el horizonte de 24 horas, GCN-LSTM obtuvo el mejor desempeño global al incorporar información espacial entre estaciones de monitoreo, logrando reducciones del error de hasta 50 % respecto al perceptrón multicapa y mejoras de hasta 10 % respecto a LSTM.

El proyecto de grado contribuye al campo de la predicción de calidad del aire mediante el desarrollo y evaluación de un enfoque basado en aprendizaje profundo. En particular, se propone una metodología estructurada y reproducible para modelar series temporales multivariantes utilizando arquitecturas codificador-decodificador basadas en redes LSTM y su extensión espaciotemporal mediante redes convolucionales sobre grafos.

El documento está organizado de la siguiente manera. En el Capítulo 2 se desarrollan los fundamentos teóricos utilizados en el proyecto de grado. En el Capítulo 3 se presenta la metodología utilizada en el proyecto. En el Capítulo 4 se presenta una revisión de los trabajos relacionados con el problema abordado. El Capítulo 5 describe el conjunto de datos utilizado, así como su análisis y preparación. En el Capítulo 6 se presentan los modelos predictivos propuestos, incluyendo las arquitecturas desarrolladas. El Capítulo 7 detalla el diseño experimental, la metodología de evaluación y el análisis de los resultados obtenidos. Por último, en el Capítulo 8 se presentan las conclusiones del proyecto y se discuten posibles líneas de trabajo futuro para su extensión.

Capítulo 2

Fundamentos teóricos

En este capítulo se abordan los principales conceptos para comprender las temáticas tratadas en el proyecto de grado.

2.1. Aprendizaje automático

El aprendizaje automático (Goodfellow et al. 2016) es una rama de la inteligencia artificial en la cual los sistemas tienen la capacidad de identificar patrones en datos de manera autónoma. A partir de conjuntos de datos, los sistemas aprenden modelos que permiten realizar predicciones o tomar decisiones sin ser explícitamente programados para cada caso particular.

En esta sección se describen las principales categorías de los modelos de aprendizaje automático.

2.1.1. Clasificación de los algoritmos de aprendizaje automático

Los algoritmos de aprendizaje automático pueden clasificarse según el tipo de información disponible durante el proceso de entrenamiento.

2.1.1.1. Aprendizaje supervisado

El aprendizaje supervisado consiste en aquellos algoritmos que aprenden a asociar una entrada con una salida a través de un conjunto de datos etiquetados. En este caso, se conoce la clase o el valor de la función objetivo de cada entrada, y el objetivo del modelo es minimizar el error al predecir nuevas observaciones desconocidas. Entre las aplicaciones más comunes de este tipo de aprendizaje se encuentran los problemas de regresión y el uso de redes neuronales.

2.1.1.2. Aprendizaje no supervisado

En el aprendizaje no supervisado los algoritmos aprenden a partir de un conjunto de datos sin etiquetas, con el objetivo de identificar patrones, estructuras o relaciones subyacentes entre las instancias. Este tipo de aprendizaje se asocia a tareas como la estimación de la densidad de los datos, la reducción de dimensionalidad, la eliminación de ruido o el agrupamiento de instancias con características similares. Entre las técnicas más utilizadas se encuentran los métodos de clustering y el análisis de componentes principales.

2.1.2. Redes neuronales artificiales

Las redes neuronales artificiales (Aggarwal, 2018) son un modelo de aprendizaje automático que tiene un paralelismo con la forma de aprendizaje de los humanos. El cerebro humano tiene neuronas que se conectan entre sí para aprender. En cambio, las redes neuronales artificiales contienen unidades de cómputo denominadas neuronas, que están conectadas entre sí a través de pesos. En esta sección se presentan las diferentes arquitecturas y modelos más relevantes para el desarrollo del presente proyecto.

2.1.2.1. Arquitectura de las redes neuronales

La red neuronal artificial calcula la salida a través de las entradas y aprende ajustando los pesos, que son los parámetros intermedios del modelo, mediante un proceso llamado retropropagación. Las redes están compuestas por capas de neuronas que se conectan entre sí. La información comienza en la capa de entrada, pasa por una o varias capas ocultas y llega a la capa de salida. Las capas ocultas son las encargadas de transformar esta información utilizando diferentes funciones de activación, que permiten que el modelo aprenda relaciones más complejas que simplemente combinaciones lineales de las entradas.

El conjunto de datos de entrenamiento es el encargado de brindar la retroalimentación necesaria para ajustar los pesos y permitir refinar el modelo. Así, en varias iteraciones, puede aprender en función de cuántas veces la salida generada coincide con la etiqueta real. A partir de dicho funcionamiento, pueden clasificarse según su arquitectura, en aquellas de una sola capa y las de múltiples capas.

Red neuronal de una sola capa. La red neuronal más simple se conoce con el nombre de perceptrón y se compone de una capa de entrada no computacional, conectada a una neurona de salida a través de un conjunto de pesos. En el modelo, los cálculos se realizan en la neurona de salida, donde se multiplican los pesos por las características, luego se suman y se aplica una función de activación.

Aunque el perceptrón tiene una única capa computacional, también puede tener un término llamado sesgo, una variable adicional que permite capturar lo que no depende directamente de los valores de las entradas. Generalmente,

se tiene una neurona de valor constante asociada a la salida, en donde el peso funciona como el valor del sesgo.

Red neuronal multicapa. En esta arquitectura se tienen varias capas computacionales, donde las capas intermedias de procesamiento se conocen como capas ocultas, ya que los cálculos que se realizan en las mismas no son visibles para el usuario. Este tipo de arquitectura se conoce como feedforward, en la que cada capa utiliza los resultados de la capa inmediatamente anterior.

2.2. Modelado secuencial

En esta sección se introduce el problema del modelado secuencial, que permite solucionar problemas en los que el objetivo no es predecir un único valor, sino una secuencia completa a partir de una secuencia de entradas, en donde los valores de las longitudes pueden ser diferentes.

2.2.1. Modelos secuencia a secuencia

Los modelos secuencia a secuencia son un enfoque general para los problemas en los que la entrada y la salida se representan como secuencias de datos. Los modelos tradicionales de predicción asocian una secuencia de entrada a un único valor de salida. En este caso, los modelos secuencia a secuencia aprenden una función que relacione una secuencia de entrada $\mathbf{x} = (x_1, x_2, \dots, x_{n_x})$ a una secuencia de salida $\mathbf{y} = (y_1, y_2, \dots, y_{n_y})$, en donde ambas pueden tener longitudes variables y no necesariamente coinciden.

2.2.2. Arquitectura codificador–decodificador

La arquitectura codificador–decodificador es un esquema del modelo secuencia a secuencia. Esta arquitectura se compone de un codificador que procesa la secuencia de entrada en un vector de contexto C , y un decodificador, que genera la secuencia de salida a partir de la información que obtiene del codificador. Este tipo de arquitectura puede componerse de varios codificadores y decodificadores, que son redes neuronales independientes.

El codificador recibe la secuencia de entrada \mathbf{x} , y su estado oculto final es usado como el contexto, que resume la información importante de toda la secuencia de entrada. El vector guarda una representación aprendida que contiene información sobre patrones, dependencias y tendencias de los datos. A partir del contexto C , el decodificador genera la secuencia de salida \mathbf{y} de manera progresiva.

La arquitectura codificador–decodificador no restringe las longitudes de las secuencias de entrada y salida, por lo que pueden ser diferentes. A su vez, es una arquitectura flexible porque pueden utilizarse diferentes arquitecturas para implementar el codificador y el decodificador (Goodfellow et al. 2016).

2.3. Modelado temporal

En esta sección se introduce el problema del modelado temporal de series de tiempo y se describe la arquitectura de redes neuronales recurrentes para su resolución.

2.3.1. Series temporales y dependencias temporales

Las series temporales son un conjunto de observaciones ordenadas en el tiempo, en donde para cada punto temporal se tiene asociado un valor correspondiente. En las series temporales el orden de las observaciones es un factor crucial, ya que los valores en un instante suelen depender del comportamiento histórico de la serie. Los valores actuales no son independientes del pasado, sino que están influenciados por tendencias, ciclos y patrones anteriores.

Las series temporales se observan en intervalos de tiempo discretos y equiespaciados. A partir de la información disponible hasta un determinado instante, se pueden realizar las predicciones sobre valores futuros de la serie. El objetivo del modelado temporal es estimar el valor de la serie en un horizonte futuro a partir de las observaciones actuales y pasadas. Dada una serie temporal $\{z_t\}$, el problema de predicción se define como la estimación de un valor futuro z_{t+l} usando la información disponible hasta el tiempo t , donde l representa el horizonte de predicción tomado en cuenta (Box et al. 2015).

2.3.2. Redes neuronales recurrentes

En los modelos tradicionales las observaciones son tratadas como instancias independientes, por lo que no toman en cuenta la estructura secuencial de los datos. Para trabajar en problemas temporales, tales como el de la predicción de la polución, es necesario hacer uso de redes neuronales que tengan memoria y permitan tomar en cuenta información histórica. En este contexto, las redes neuronales recurrentes permiten capturar la manera en la que los valores de entrada en los diferentes instantes de tiempo están directamente relacionados.

Para considerar una red neuronal como recurrente, a diferencia de las redes feedforward, debe contener ciclos en su arquitectura. La salida de la capa oculta en el instante de tiempo $t - 1$ se usa como entrada para la del siguiente instante t . En la ecuación (2.1) se muestra la forma matemática básica de una red neuronal recurrente, donde el estado oculto en el tiempo t , h_t , se calcula como una función no lineal f que toma como entrada una combinación lineal del vector de entrada actual x_t , ponderado por la matriz W , y del estado oculto previo h_{t-1} , ponderado por la matriz recurrente U , además del sesgo b . Esta estructura permite que la red mantenga un tipo de memoria temporal, integrando información pasada para influir en el procesamiento presente.

$$h_t = f(Wx_t + Uh_{t-1} + b) \tag{2.1}$$

Para entrenar este tipo de redes, los valores son propagados hacia atrás usando retropropagación a través del tiempo. Se realiza un recorrido hacia adelante

guardando el valor del estado oculto en cada paso y después un recorrido inverso para actualizar los pesos utilizando esta información. El problema con el entrenamiento de este tipo de redes se da principalmente debido al tamaño de la entrada y el grado de profundidad de la red. Lo que ocurre es que se multiplica varias veces la misma matriz de pesos, lo que puede ocasionar inestabilidad en el gradiente. La inestabilidad se puede dar de dos maneras: los gradientes tienden a desaparecer si los valores que se están multiplicando son pequeños o se disparan si los pesos son valores grandes. Este problema hace que las redes neuronales recurrentes sean útiles para recordar información a corto plazo pero no cuando se tienen secuencias largas como entrada.

2.3.2.1. Redes LSTM

Las redes LSTM solucionan el problema de la memoria a corto plazo ya que están diseñadas específicamente para aprender secuencias a largo plazo. En este tipo de redes, cada estado oculto tiene tres entradas que corresponden a la memoria actual, la salida del paso anterior y la entrada del paso actual, y dos salidas que son la memoria actualizada y la salida del paso actual. Se modifica la arquitectura de las redes neuronales recurrentes agregando un vector llamado estado de celda que retiene parte de la información pasada para que se mantenga a lo largo del tiempo.

En las LSTM la unidad de procesamiento básica, en vez de ser una neurona, se llama bloque de memoria y contiene varias celdas de memoria para mantener la información a lo largo del tiempo. El modelo divide el manejo del contexto en dos problemas: por un lado descartar la información que ya no es relevante, y por otro incluir la que podría serlo. Por este motivo tienen un mecanismo de puertas de entrada y salida para poder controlar la información que se retiene y se transmite. Para poder lidiar con el problema del desvanecimiento y explosión del gradiente, dentro de cada celda se tiene una neurona conectada consigo misma formando el carrusel de error constante. Esto hace que el error propagado hacia atrás durante el entrenamiento no se altere y se vuelva inestable, sino que se mantiene constante durante el tiempo.

Como se muestra en la ecuación (2.2), la arquitectura básica de una red LSTM se compone de tres puertas: puerta de entrada (i_t), olvido (f_t) y salida (o_t), para las que se utiliza la función de activación sigmoide ya que restringe los valores entre 0 y 1 permitiendo decidir cuánta información pasa. Además, se utiliza la función tanh para generar el nuevo candidato a celda (\tilde{c}_t) ya que produce valores entre -1 y 1 permitiendo tanto restar como agregar información. Esto permite que la red decida, en cada paso de tiempo, qué información incorporar, olvidar o transmitir.

$$\begin{bmatrix} i_t \\ f_t \\ o_t \\ \tilde{c}_t \end{bmatrix} = \begin{bmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{bmatrix} \left(W^{(k)} \begin{bmatrix} h_t^{(k-1)} \\ h_{t-1}^{(k)} \end{bmatrix} + b^{(k)} \right) \quad (2.2)$$

2.4. Modelado espacial

En esta sección se introduce el problema del modelado espacial, y se describen las arquitecturas de redes neuronales utilizadas para capturar este tipo de dependencias.

2.4.1. Redes neuronales convolucionales

Las redes neuronales convolucionales (Convolutional Neural Network, CNN) son un tipo de arquitectura de aprendizaje profundo diseñadas para trabajar con datos que tienen una estructura de rejilla con dependencias espaciales locales. Existen CNN unidimensionales (1D), bidimensionales (2D) y tridimensionales (3D). 1D CNN es usada principalmente para el procesamiento de datos secuenciales, 2D CNN para el reconocimiento de imágenes y texto, y 3D CNN para datos tales como videos.

Una arquitectura convolucional típica se compone de tres capas principales: la capa de convolución, la capa de agrupación y la capa totalmente conectada.

Capa de convolución. La capa de convolución hace uso de filtros para realizar operaciones que permiten extraer características importantes de la entrada. Los filtros se mueven por la entrada para obtener patrones locales, realizando el producto de la entrada y el filtro para obtener el mapa de características. La profundidad del filtro coincide con la de la entrada sobre la que se aplica.

Capa de agrupación. La capa de agrupación también aplica un filtro sobre la entrada, pero sin tener un peso asociado. Esta capa realiza operaciones de agregación dentro de una determinada región, reduciendo el tamaño de los datos e introduciendo cierta invarianza espacial. Los principales tipos de agrupación son la agrupación máxima, en la que se selecciona el valor máximo de la región y agrupación media donde se calcula el valor promedio de la región.

Capa totalmente conectada. La capa totalmente conectada se encuentra normalmente al final de la arquitectura de una red convolucional, donde cada dato de entrada está directamente conectado a todas las neuronas y funciona igual que en una red feed-forward convencional. Permite integrar toda la información que se extrajo previamente para realizar las predicciones finales. Como paso previo se aplanan los datos para obtener un vector de una sola dimensión, facilitando el procesamiento.

2.4.2. Redes neuronales gráficas

Las redes neuronales gráficas (Graph Neural Networks, GNN) son una extensión de las redes neuronales diseñadas para trabajar con datos estructurados como grafos. En las GNN cada nodo del grafo representa una entidad y las aristas describen las relaciones entre las entidades. Para poder capturar la información de los nodos y de su relación con los nodos vecinos, se requieren

modelos capaces de incorporar explícitamente la estructura del grafo durante el proceso de aprendizaje.

Las redes neuronales gráficas permiten agregar información entre los nodos conectados. A diferencia de las redes tradicionales, en las GNN la representación de un nodo se actualiza iterativamente combinando su información con la de otros nodos. Este proceso permite que la información se propague a través de la estructura del grafo, de manera que cada nodo pueda incorporar progresivamente información de nodos más lejanos a medida que se apilan capas del modelo.

2.4.2.1. Redes convolucionales sobre grafos

Las redes convolucionales sobre grafos (Graph Convolutional Networks, GCN) permiten utilizar la estructura de grafos y combinar la información de un nodo con la de sus vecinos de forma convolucional. Los grafos tienen una estructura irregular porque el número de nodos y las relaciones entre ellos pueden variar. Las redes convolucionales sobre grafos permiten modelar relaciones espaciales complejas que no pueden representarse mediante una estructura de grilla regular.

La formulación utilizada de las GCN fue introducida por Kipf y Welling (Kipf y Welling, 2017). Como se muestra en la ecuación (2.3), cada capa de una GCN actualiza la representación de cada nodo combinando su información y la de sus vecinos a través de la estructura del grafo. La matriz $\tilde{A} = A + I$ es la matriz de adyacencia original y las conexiones del nodo consigo mismo, para que cada nodo conserve su propia información y no tenga solamente la de sus vecinos. La normalización mediante \tilde{D} hace que un nodo no sea más influyente debido a la cantidad de conexiones que tiene. La transformación lineal $W^{(l)}$ permite al modelo aprender combinaciones de las características agregadas y la función de activación $\sigma(\cdot)$ permite la no linealidad.

$$H^{(l+1)} = \sigma \left(\tilde{D}^{-1/2} \tilde{A} \tilde{D}^{-1/2} H^{(l)} W^{(l)} \right) \quad (2.3)$$

Cuando se apilan varias capas de convolución y se realizan sucesivas rondas de agregación de información entre nodos vecinos, las GCN presentan un problema denominado over-smoothing. Al aumentar la profundidad de la red, las representaciones de los nodos convergen en vectores similares y se vuelven poco informativas. Cada capa de una GCN realiza una agregación que equivale a multiplicar las representaciones de los nodos por la matriz de adyacencia normalizada. Cuando esta operación se repite varias veces, las grandes diferencias desaparecen y se vuelven más uniformes, suavizando la señal. Este problema hace que el modelo no pueda distinguir entre nodos con características diferentes y no pueda captar estructuras o patrones locales importantes. Por esta razón se diseñan arquitecturas poco profundas para mitigar este efecto. (Hamilton, 2020).

Capítulo 3

Metodología

En este capítulo se presenta la descripción del método utilizado para la definición e implementación de los modelos de redes neuronales. En la Sección 3.1 se detalla la estrategia de desarrollo adoptada. En la Sección 3.2 se describen los criterios utilizados en el desarrollo del proyecto.

3.1. Estrategia de desarrollo

En el proyecto de grado se adoptó un enfoque aplicado y empírico para la predicción de la calidad del aire, donde se desarrollaron y evaluaron modelos de redes neuronales sobre datos reales. Por este motivo se definió una estrategia de desarrollo incremental, dividida en etapas para tratar progresivamente la complejidad del problema.

Para obtener datos consistentes para el entrenamiento de los modelos se realizó el tratamiento y transformación de los datos. El proceso incluyó la selección de las variables relevantes, la integración de información de estaciones, contaminantes y variables meteorológicas, la detección y el manejo de valores atípicos, el tratamiento de valores faltantes y la transformación de los datos para su uso en las etapas posteriores.

Una vez preparados los datos, se diseñaron y definieron los modelos de redes neuronales. Se comenzó con la implementación del MLP como modelo base, utilizado como referencia inicial de desempeño. Después, se implementó una red LSTM codificador–decodificador para capturar las dependencias temporales de las series. Por último, se incorporó información espacial entre las estaciones de monitoreo mediante un GCN-LSTM codificador–decodificador para capturar también las dependencias espaciales.

Para cada una de las arquitecturas se ajustaron los hiperparámetros ejecutando diferentes configuraciones experimentales. Se evaluaron varias combinaciones de hiperparámetros mediante una búsqueda aleatoria y se seleccionaron las que minimizaron la pérdida en el conjunto de validación para cada modelo.

3.2. Criterios metodológicos

Para garantizar que la metodología seguida fuera reproducible y los resultados comparables, se siguió una serie de criterios metodológicos en el desarrollo experimental. Como primer criterio, el preprocesamiento de los datos se aplicó en todos los experimentos realizados. El segundo criterio fue definir un esquema de evaluación basado en particiones temporales (hold-out y validación cruzada temporal), preservando el orden cronológico de los datos. También se estableció el uso de las mismas particiones de datos, variables de entrada y horizontes de predicción para todos los modelos, para permitir una comparación justa. Finalmente, las métricas utilizadas para la evaluación, MAE y Root Mean Square Error (RMSE), métricas estándar que permitieron analizar el error promedio y la sensibilidad a los valores extremos.

Capítulo 4

Estado del arte

En este capítulo se presentan los antecedentes relevantes para el presente proyecto y se describen las diferentes metodologías empleadas.

Inicialmente, se utilizaron métodos lineales que funcionaban bien en contextos simples, pero presentaron limitaciones al momento de predecir series temporales complejas. El estudio realizado por Aditya et al. (2018) se llevó a cabo en India y utilizó datos atmosféricos del repositorio UCI para predecir niveles de contaminación del aire ($PM_{2,5}$). El sistema tenía dos funciones: una para clasificar el aire como contaminado o no con un modelo de regresión logística, y otra para predecir valores de $PM_{2,5}$ en la ciudad, tomando en cuenta el comportamiento anterior a lo largo del tiempo utilizando un modelo autorregresivo. Los resultados tuvieron una precisión de 0.998859 en la clasificación y un Mean Squared Error (MSE) de $27,000\mu g/m^3$ en las predicciones temporales. El enfoque tiene limitaciones para las relaciones complejas, ya que fue diseñado para trabajar con datos lineales. En contraste, el objetivo del proyecto de grado es aplicar redes neuronales artificiales, que tienen capacidad de identificar patrones no lineales y mejorar la capacidad de predicción.

Debido a las limitaciones de los enfoques lineales, se empezaron a desarrollar modelos basados en redes neuronales artificiales. En este sentido, Kök et al. (2017) elaboraron un modelo de aprendizaje profundo para predecir la calidad del aire en ciudades inteligentes, tomando información de Dinamarca y Rumania. Las redes neuronales utilizadas fueron LSTM con una capa de entrada, 24 capas ocultas y una de salida; una red para predecir los valores de cada gas: O_3 , NO_2 y $PM_{2,5}$. Utilizaron información sobre el tránsito, clima y datos de estacionamiento para aportar al entrenamiento del modelo. A su vez, implementaron un Support Vector Regression (SVR) como modelo de comparación de las redes neuronales. El modelo LSTM obtuvo una precisión de 0.98 en los estados críticos de polución y logró un mayor F1 en comparación al SVR. En este caso, a diferencia del objetivo del proyecto de grado, no consideraron la dimensión espacial de los datos, lo que simplificó el modelo, pero redujo la precisión.

Cordova et al. (2021) implementaron dos arquitecturas de redes neuronales: una de retropropagación hacia adelante y otra recurrente LSTM con el objetivo

de predecir el comportamiento espacio-temporal de la calidad del aire en Lima, Perú. Utilizaron datos de cinco estaciones y variables meteorológicas y entrenaron los modelos con Hold-Out, que divide al conjunto de datos en entrenamiento y prueba, y Validación Cruzada con Bloques No Solapados (BNVC), que divide a los datos en bloques de tiempo que no se solapan. Los resultados mostraron un alto desempeño en la predicción de concentración media de PM_{10} , pero su precisión se redujo en los períodos de alta contaminación. El uso de las redes neuronales es importante porque demuestra su capacidad de modelar patrones espacio-temporales y puede mejorarse utilizando más variables meteorológicas y factores relacionados con el tránsito vehicular. Aunque en este proyecto se considera la dimensión espacial al tomar en cuenta datos de diferentes estaciones, la arquitectura no lo incorpora, sino que se usa únicamente una red LSTM. En contraste, el siguiente enfoque sí incluye tanto la dependencia temporal como la espacial dentro de la implementación del modelo.

En el trabajo de Iskandaryan et al. (2023) se utilizaron los datos del portal de datos abiertos de Madrid para predecir la calidad del aire en la ciudad de Madrid, España. Se implementó un modelo avanzado Attention Temporal Graph Convolutional Network (A3T-GCN) que combina tres componentes: Attention y GRU para obtener la información temporal; y GCN para capturar las dependencias espaciales. El modelo permitió a los investigadores trabajar con datos sin una estructura euclidiana. Para entrenar el modelo se usaron datos meteorológicos, de tránsito y de la calidad del aire, de diferentes estaciones. Debido a los cambios de locación de los puntos de medición en los períodos de tiempo tomados en cuenta, los datos se agruparon en una grilla dividida en celdas. Esta solución mostró una mayor precisión en los resultados que los modelos de LSTM, GRU y Red Neuronal de Grafos Temporales (TGNN). El modelo redujo el RMSE entre 10.9% y 14.3%, el MAE entre 5.6% y 9.7% y el coeficiente de correlación de Pearson entre un 3.4% y 17%. La complejidad del modelo permite trabajar en situaciones en las que se tienen distribuidos los sensores de manera no uniforme, como ocurre en las zonas urbanas, y se quiere tomar en cuenta tanto las relaciones espaciales como las temporales. El proyecto de grado también se evaluó utilizando varias de las métricas mencionadas en este proyecto, lo que permitirá una comparación del desempeño.

Otro enfoque que comenzó a ser utilizado es el uso de modelos híbridos que combinan redes neuronales con otros modelos. En la investigación de Li et al. (2020) se tomaron en cuenta datos meteorológicos y dirección del viento de la ciudad de Beijing China, para predecir la concentración de $PM_{2.5}$ en las siguientes 24 horas. Implementó cuatro modelos: LSTM univariado, LSTM multivariado, CNN-LSTM univariado y CNN-LSTM multivariado, mostrando que el último tiene menor margen de error y tiempo de entrenamiento, convirtiéndolo en el más adecuado. El modelo obtuvo un MAE de $13,9697 \mu\text{g}/\text{m}^3$ y RMSE de $17,9306 \mu\text{g}/\text{m}^3$. La red neuronal híbrida utilizó CNN para identificar patrones en la calidad del aire y LSTM para tomar en cuenta los datos históricos. En este caso, el dataset utilizado no es accesible, lo que impide la replicación del experimento y realizar una comparación directa.

Xiao et al. (2022) construyeron un modelo híbrido utilizando datos en forma

de serie temporal de un año provenientes de China, para predecir el contenido de $\text{PM}_{2,5}$ en el aire a la siguiente hora. Utilizaron datos de la calidad del aire, humedad, presión, temperatura y cantidad de nieve. Emplearon un agrupamiento espacio-temporal para tomar en cuenta la ubicación espacial y comportamientos similares en diferentes zonas geográficas. Combinaron el modelo estadístico ARIMA para predecir el comportamiento lineal y una red neuronal artificial recurrente LSTM para trabajar sobre los residuos que no pudieron preverse. El modelo ARIMA-LSTM obtuvo un RMSE de $0,97 \mu\text{g}/\text{m}^3$, mostrando una mejora significativa respecto a un modelo LSTM simple.

Jana et al. (2024) desarrollaron tres modelos de redes neuronales de Convulsión en Grafo Espacio-Temporal (STGCN). Usaron datos de Delhi (2020–2021), caracterizados por una alta varianza, y de California (2017–2019), más estables. A su vez, tomaron en cuenta datos meteorológicos: dirección y velocidad del viento, humedad y radiación solar. El modelo contenía un grafo donde los nodos eran estaciones y las aristas se ponderaban por distancia. El objetivo fue predecir 6 contaminantes diferentes en ventanas temporales de 1, 24 y 48 horas. El modelo STGCN-A aplicaba primero capas espaciales y luego temporales; STGCN-B organizaba las capas temporales antes y después de la capa espacial; y STGCN-C que ejecutaba las capas espaciales y temporales en paralelo. STGCN-B obtuvo mejores resultados a 1 h y 48 h y STGCN-C destacó a 24 h, superando modelos como LSTM o CNN+RNN tanto en precisión como en eficiencia.

Por último, se encuentra una investigación que tomó en cuenta los datos del mismo lugar que se utilizan en el proyecto de grado y en la misma localización. Cáceres-Tello y Galán-Hernández (2024) propusieron un modelo híbrido Prophet-LSTM con el objetivo de predecir la concentración de $\text{PM}_{2,5}$ en siete distritos de Madrid. Los datos utilizados fueron del portal de datos abiertos de la calidad del aire desde el año 2019 al 2024. En la depuración de los datos eliminaron todos aquellos registros que tuviesen algún valor faltante o estuviesen incompletos, para luego utilizarlos en el entrenamiento y evaluación. El modelo consistía en dos fases: la primera descomponía la serie temporal en tendencia, estacionalidad y eventos especiales. Los residuos se utilizaban como entradas para el modelo LSTM. En la segunda se combinaban ambos resultados de Prophet y LSTM. En el estudio demostraron cómo Prophet es capaz de captar eventos especiales y estacionalidad, y LSTM logra capturar las dependencias temporales. Si bien no se reportaron valores específicos para métricas como MSE, RMSE o MAE, los autores mencionan que el modelo Prophet-LSTM obtuvo una mejora del 15 % en precisión con respecto a modelos híbridos como SARIMA-LSTM y ETS-LSTM. No obstante, en el presente proyecto se busca utilizar métodos de aprendizaje profundo, por lo que el modelo es diferente al que se realizará. A su vez, se busca tener en cuenta las dependencias espaciales entre las diferentes estaciones y utilizar datos que aporten otra información, tales como los meteorológicos que no fueron considerados.

Tabla 4.1: Comparación de trabajos relacionados sobre la predicción de la contaminación del aire, en cuanto a los contaminantes considerados, los datos adicionales, las estaciones utilizadas (*#e*), el período de recolección de datos (*período*), el horizonte de predicción (*ventana*) y los modelos empleados.

<i>Autor (año)</i>	<i>contaminantes</i>	<i>datos adicionales #e</i>	<i>período (años)</i>	<i>ventana (horas)</i>	<i>modelos</i>	
Kök et al. (2017)	O ₃ , NO ₂ , PM _{2,5}	Tránsito, clima, estacionamiento	1	3	N/A	LSTM
Aditya et al. (2018)	PM _{2,5}	Temperatura, viento, presión, rocío	1	2	N/A	LR, AR
Li et al. (2020)	PM _{2,5}	Dirección viento, temperatura, presión, nieve, lluvia	1	3	24	LSTM, CNN-LSTM
Cordova et al. (2021)	PM ₁₀	Temperatura, humedad, velocidad viento	5	2	1	MLP, LSTM
Xiao et al. (2022)	PM _{2,5}	Humedad, lluvia, temperatura, presión, nieve	1	1	1	ARIMA-LSTM clustering
Iskandaryan et al. (2023)	NO ₂	Intensidad tránsito, carga, velocidad y dirección viento, temperatura, humedad, presión barométrica, radiación solar	24	3	1–12, 12–24, 24–36, 36–48	A3T-GCN
Jana et al. (2024)	PM _{2,5} , O ₃ , PM ₁₀ , SO ₂ , CO, NO ₂	Velocidad y dirección viento, humedad, radiación solar	40	1	1, 24, 48	STGCN
Cáceres-Tello y Galán-Hernández (2024)	PM _{2,5}	Ninguno	7	6	N/A	Prophet LSTM

Capítulo 5

Presentación y análisis de los datos

En este capítulo se describe el procedimiento completo de análisis y preparación de los datos que son utilizados para la experimentación y validación de los modelos de redes neuronales desarrollados.

5.1. Selección de los datos

En esta sección se describe la fase de selección de los datos. En esta etapa se identifican y extraen las fuentes de información que se utilizarán para todo el análisis.

El modelo de predicción de redes neuronales usa datos de la ciudad de Madrid provenientes de informes horarios generados por el Sistema Integral de la Calidad del Aire del Ayuntamiento de Madrid, accesibles a través de su Portal de Datos Abiertos. Se dispone de un total de 37 estaciones (ver Figura 5.1) que miden todos o una combinación de concentraciones de ozono (O_3), dióxido de azufre (SO_2), monóxido de carbono (CO), dióxido de nitrógeno (NO_2), óxido nítrico (NO), partículas en suspensión de tamaño menor a $10\ \mu m$ (PM_{10}) y partículas en suspensión de tamaño menor a $2,5\ \mu m$ ($PM_{2,5}$), además de variables meteorológicas, incluyendo radiación ultravioleta, velocidad del viento, dirección del viento, temperatura, humedad relativa, presión barométrica, radiación solar y precipitación. Las estaciones que miden contaminantes son clasificadas por el Ayuntamiento de Madrid según su entorno (ver Tabla 5.1). La clasificación define tres tipos: urbana de tráfico, junto a vías de alta intensidad; urbana de fondo, ubicadas lejos de focos directos para representar la exposición media; y suburbana, situadas en la periferia.

Entre todas las mediciones, en este estudio se seleccionaron como variables objetivo aquellos contaminantes que tienen establecidos umbrales por el Ministerio para la Transición Ecológica y el Reto Demográfico (MITECO) y que, además, presentan una gran relevancia sanitaria según la Agencia Europea de

Tabla 5.1: Clasificación de las estaciones según su entorno.

<i>Estación</i>	<i>Tipo</i>
Plaza de España	Urbana tráfico
Escuelas Aguirre	Urbana tráfico
Ramón y Cajal	Urbana tráfico
Arturo Soria	Urbana fondo
Villaverde	Urbana fondo
Farolillo	Urbana fondo
Casa de Campo	Suburbana
Barajas Pueblo	Urbana fondo
Plaza del Carmen	Urbana fondo
Moratalaz	Urbana tráfico
Cuatro Caminos	Urbana tráfico
Barrio del Pilar	Urbana tráfico
Vallecas	Urbana fondo
Méndez Álvaro	Urbana fondo
Castellana	Urbana tráfico
Parque del Retiro	Urbana fondo
Plaza Castilla	Urbana tráfico
Ensanche de Vallecas	Urbana fondo
Urb. Embajada	Urbana fondo
Plaza Elíptica	Urbana tráfico
Sanchinarro	Urbana fondo
El Pardo	Suburbana
Juan Carlos I	Suburbana
Tres Olivos	Urbana fondo

Medio Ambiente (AEE). En consecuencia, se priorizaron $PM_{2,5}$, NO_2 y O_3 . A escala global, se reafirma que $PM_{2,5}$, NO_2 y O_3 constituyen los principales contaminantes que presentan riesgos para la salud humana (Health Effects Institute (HEI) and Institute for Health Metrics and Evaluation (IHME), 2024). Según el informe de 2024 de la AEE, en 2022 estos contaminantes fueron responsables de aproximadamente 239.000 muertes prematuras por $PM_{2,5}$, 70.000 por O_3 y 48.000 por NO_2 en la Unión Europea. En contraste, CO y SO_2 , que tienen escasa incidencia en la mortalidad, se excluyeron del análisis para profundizar en el estudio de los contaminantes seleccionados. Para el caso de PM_{10} , este aporta información muy similar al de $PM_{2,5}$ ya que $PM_{2,5}$ constituye una fracción de PM_{10} .

El conjunto de datos empleado en este análisis abarca del 1 de julio de 2021 al 31 de diciembre de 2024. Durante la pandemia de COVID-19 se decretaron dos estados de alarma —el primero del 15 de marzo al 21 de junio de 2020, y el segundo del 25 de octubre de 2020 al 9 de mayo de 2021— en los que se impusieron confinamientos, toques de queda y cierres de actividades industriales y de movilidad. Las medidas provocaron reducciones de contaminación a niveles muy bajos y atípicos para la ciudad de Madrid, generando rupturas en la tendencia

y la estacionalidad de las series temporales. Por esta razón se optó por excluir los episodios extremos y así preservar la homogeneidad y validez de los modelos predictivos.

Los valores estadísticos de los contaminantes en los períodos de estudio se muestran en la Tabla 5.2. La media y la desviación estándar fueron calculadas a partir de la unión de los datos de todo el período de evaluación definido previamente. Para el caso del mínimo y máximo, se toma el de cada año y se realiza la media posteriormente para evitar la influencia de los posibles valores puntuales extremos. A su vez, se consideran los valores mayores a 0, ya que los menores no son posibles físicamente, y los iguales a 0, dado que en un entorno urbano estos valores reflejan ausencia de señal fiable y distorsionan el cálculo del mínimo real.

Tabla 5.2: Estadísticos descriptivos de los contaminantes seleccionados.

<i>Contaminante</i>	<i>Unidad</i>	<i># estaciones</i>	<i>Media</i>	<i>Desviación estándar</i>	<i>Mínimo</i>	<i>Máximo</i>
PM _{2,5}	$\mu\text{g}/\text{m}^3$	8	9.77	8.98	1.00	177.30
NO ₂	$\mu\text{g}/\text{m}^3$	24	26.92	19.31	2.08	133.50
O ₃	$\mu\text{g}/\text{m}^3$	13	54.37	31.93	1.52	180.51

A su vez, se tienen los valores estadísticos en la Tabla 5.3 desagregados por estación, aplicando el mismo criterio de filtrado. Se puede observar la variabilidad espacial de los contaminantes dependiendo de la estación de medición y su clasificación.

En la Figura 5.2 se presentan los histogramas de cada contaminante que permiten visualizar las distribuciones de una estación urbana de tráfico (Escuelas Aguirre) y una suburbana (Casa de Campo). Las gráficas se construyeron con 30 bins y filtrando los valores mayores o iguales a cero. En el eje x se observa la distribución y en el eje y la frecuencia con la que los valores se midieron en las estaciones determinadas. Para la visualización de histogramas en esta etapa del análisis, se limitó el eje x al percentil 99.9 de la distribución combinada de cada contaminante. Esta decisión impide que los valores atípicos puntuales distorsionen la escala y no afecta la interpretación del grueso de los datos. Se seleccionaron las dos estaciones porque miden simultáneamente los contaminantes objetivo y representan dos categorías distintas del sistema de monitoreo, lo que permite analizar las diferencias entre los entornos.

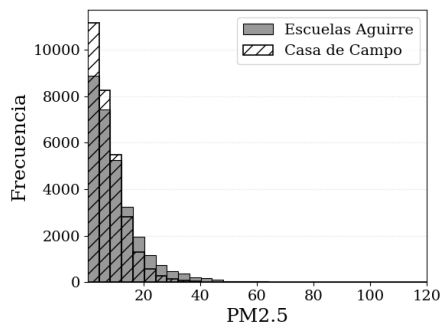
En los tres histogramas hay una asimetría positiva debido a que gran cantidad de las mediciones son valores moderados. Existen poca cantidad de episodios extremos que generan una cola hacia la derecha de distinta intensidad dependiendo del contaminante. En el caso de las partículas microscópicas PM_{2,5} (ver Fig. 5.2a) la estación suburbana tiene mayor concentración de observaciones en un rango de valores bajos y la frecuencia cae rápidamente a valores altos obteniéndose una cola corta. En la de tráfico, la distribución se desplaza hacia valores más altos y la cola derecha es más pesada. Respecto al NO₂ (ver Fig. 5.2b), la distribución de tráfico está notoriamente desplazada hacia la derecha y presenta una cola más pesada que la suburbana, lo que tiene sentido

Tabla 5.3: Estadísticos descriptivos por estación ($\#e$) para los contaminantes medidos.

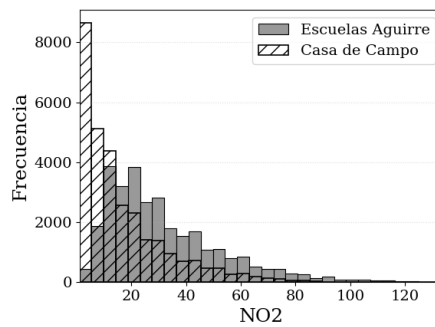
$\#e$	NO ₂		PM _{2,5}		O ₃	
	$\mu \pm \sigma$	Min–Max	$\mu \pm \sigma$	Min–Max	$\mu \pm \sigma$	Min–Max
4	26.0±19.8	1.0–157.0	–	–	–	–
8	32.4±22.0	1.0–179.5	11.36±35.7	1.0–440.5	52.0±31.0	0.9–336.6
11	29.9±24.5	1.5–196.5	–	–	–	–
16	26.0±21.7	1.0–168.5	–	–	53.1±32.6	1.0–234.5
17	34.3±28.0	1.0–195.0	–	–	51.3±35.0	1.0–213.5
18	27.5±21.4	1.0–171.0	–	–	52.3±34.2	0.9–261.9
24	16.5±16.0	1.0–134.0	7.8±6.9	1.00–81.0	59.0±33.4	0.9–255.0
27	30.8±23.4	1.0–152.0	–	–	51.0±36.1	0.9–245.6
35	30.0±19.8	1.5–150.5	–	–	55.6±33.3	1.0–219.0
36	28.6±21.7	1.0–150.5	–	–	–	–
38	28.0±23.6	1.0–185.5	9.9±10.2	1.0–130.5	–	–
39	27.7±24.7	1.0–191.5	–	–	55.7±34.2	1.0–210.5
40	29.0±23.2	1.5–148.5	–	–	–	–
47	26.2±22.4	1.0–168.0	9.6±20.2	1.0–281.5	–	–
48	26.6±20.5	1.0–170.5	10.1±13.7	1.0–517.0	–	–
49	19.9±17.0	1.0–129.0	–	–	53.5±32.8	1.0–410.8
50	30.2±20.6	1.5–190.0	9.4±14.9	1.0–492.5	–	–
54	28.9±26.1	1.0–251.5	–	–	52.1±32.9	0.9–178.1
55	27.4±22.9	1.0–169.0	–	–	–	–
56	36.8±25.1	1.0–186.5	11.5±17.7	1.0–809.0	–	–
57	23.8±22.3	1.5–169.0	8.1±6.5	1.0–65.0	–	–
58	13.4±11.1	1.0–162.0	–	–	56.9±35.9	1.0–255.7
59	21.4±19.8	1.0–140.5	–	–	58.0±33.6	1.0–214.5
60	23.3±18.9	1.0–150.5	–	–	58.0±32.2	1.0–215.5

debido a su tipología. En el caso del O₃ (ver Fig. 5.2c), el comportamiento se invierte y la estación suburbana presenta niveles más altos en casi todo el rango y una cola mayor. El ozono troposférico suele alcanzar concentraciones más altas en áreas suburbanas y rurales que en los núcleos urbanos con mayor intensidad de tráfico, donde el NO emitido tiende a consumir O₃ y a reducir sus niveles locales (Ministerio de Agricultura, Alimentación y Medio Ambiente, 2015).

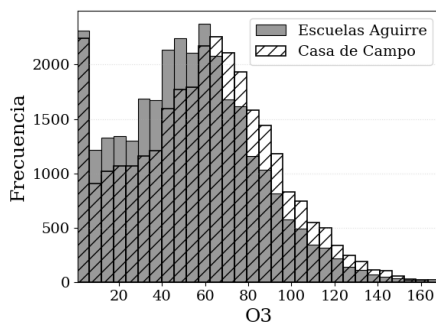
Además de analizar la distribución de los contaminantes objetivo se realizó un estudio del comportamiento semanal de sus concentraciones. En la Figura 5.3 se presentan las concentraciones de los contaminantes para la misma estación urbana tráfico (Escuelas Aguirre) y suburbana (Casa de Campo) para comparar las diferentes formas de distribución. Para cada estación y contaminante se calcula la mediana diaria sobre todos los años del conjunto de datos para tener un valor representativo. Cada punto corresponde a la mediana obtenida para cada día de la semana.



(a) $PM_{2,5}$



(b) NO_2



(c) O_3

Figura 5.2: Histogramas de los contaminantes objetivo.

En el caso de NO_2 (ver Fig. 5.3b) se observa un patrón claramente definido. En ambas estaciones las concentraciones durante los días laborales, de lunes a viernes, presentan valores más elevados que los registrados para el fin de semana. A partir del sábado se tiene un descenso sostenido de las concentraciones llegando a valores mínimos el domingo. Esta dinámica refleja la influencia del tráfico sobre el NO_2 ya que la reducción de la movilidad urbana el fin de semana reduce las emisiones del contaminante. Respecto a $\text{PM}_{2,5}$ (ver Fig. 5.3a) las variaciones semanales son casi constantes a lo largo de la semana. Ambas estaciones muestran concentraciones casi constantes a lo largo de los días sin mostrar un patrón semanal claro. Este comportamiento sugiere que las concentraciones de las partículas finas dependen de varios factores que son menos cíclicos que el tránsito como para el caso del NO_2 . En términos relativos, la estación urbana de tráfico muestra valores superiores a la suburbana. Para el O_3 (ver Fig. 5.3c) la relación entre las concentraciones es opuesta a la observada para los dos contaminantes anteriores. En este caso se obtienen valores más elevados en la estación suburbana que en la de tráfico. A su vez, los valores tienden a incrementar hacia el fin de semana obteniendo su pico en el domingo. Este patrón es coherente con lo observado en el histograma y la naturaleza fotoquímica de este contaminante en particular.

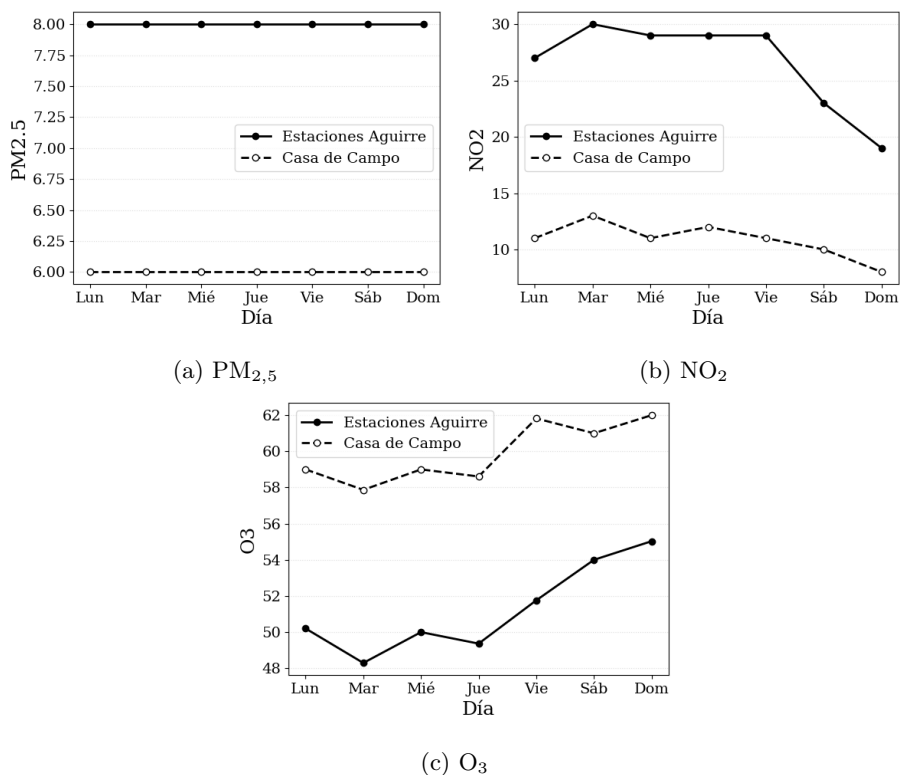


Figura 5.3: Patrones semanales de los contaminantes objetivo.

5.2. Preprocesamiento de los datos

En esta sección se aborda la limpieza y preprocesamiento de los datos. Este proceso implica la integración de las distintas fuentes de datos y el posterior manejo de valores faltantes y atípicos.

5.2.1. Integración de los datos

En este proyecto se combinan dos tipos de registros provenientes del Portal de Datos Abiertos del Ayuntamiento de Madrid: los contaminantes atmosféricos y las variables meteorológicas. Ambos conjuntos de datos están medidos con frecuencia horaria y clasificados por estación de monitoreo.

Los archivos de entrada aparecen en formato CSV, TXT, XML y JSON. Se elige el formato CSV por su apertura e interoperabilidad y porque es compatible con diferentes herramientas de aprendizaje automático. A diferencia de estructuras de datos jerárquicas, evita que haya una sobrecarga de metadatos, lo cual es importante a la hora de trabajar con gran cantidad de registros. Esta elección está alineada con las recomendaciones y estándares internacionales ya que

cumple con el RFC 4180, en donde se tiene una estructura tabular con número constante de columnas, delimitación por comas, reglas de entrecomillado y MIME type text/csv.

Cada archivo está estructurado de manera mensual y presenta un esquema que contiene metadatos: municipio, código de la estación, provincia, magnitud, punto de muestreo, año, mes, día y campos horarios (H01-H24), además de una columna de validación para cada hora. Inicialmente se eliminan los campos de provincia, municipio y punto de muestreo ya que el código de la estación es suficiente para identificarla. Se reorganiza la estructura del conjunto de datos, transformándola de un formato ancho a uno largo. Esta transformación se logra al homogenizar los nombres de las columnas, mapeando los códigos de los contaminantes o variables meteorológicas a las respectivas etiquetas descriptivas.

Se genera una marca temporal completa (YYYY-MM-DD HH:MM) que combina fecha y hora, la cual se utiliza como índice temporal. Esta marca permite reconfigurar los datos en una tabla pivote donde cada fila representa una observación única por estación y hora, y cada columna corresponde a una variable medida.

Los subconjuntos de contaminantes y meteorología se fusionan por el código de la estación y marca de tiempo mediante una unión externa, así se preserva la información incluso cuando alguna de las dos fuentes presenta datos faltantes en determinados momentos. El resultado es una estructura integrada por estación (implementada como colección tipo diccionario indexada por el identificador de estación), en donde los valores son tablas indexadas por tiempo que combinan, para cada hora, concentraciones de contaminantes y condiciones meteorológicas. Para cada estación se recuperan las coordenadas geográficas de latitud y longitud a partir de catálogos auxiliares del portal. Los campos de las coordenadas geográficas habilitan el posterior análisis espacio-temporal.

5.2.2. Detección y manejo de valores atípicos

Las observaciones que se alejan significativamente del comportamiento general de los datos y aquellas que son físicamente imposibles son detectadas y tratadas para evitar que distorsionen y confundan a los modelos predictivos.

En esta etapa no se realiza un análisis global porque las contribuciones de estaciones en zonas de baja polución no se verían representadas apropiadamente en el proceso de entrenamiento. Por el contrario, se realiza la detección de valores atípicos de modo local.

Inicialmente se establecen umbrales que son propios de los valores de los contaminantes y variables meteorológicas, para descartar los registros con valores físicamente imposibles. En la Tabla 5.4 se presentan los mínimos y máximos establecidos para cada variable, ya sea objetivo o predictiva, en base a datos registrados históricamente y definidos en la literatura. En el caso de las variables meteorológicas, la temperatura se establece en un umbral de plausibilidad más amplio que los valores históricos registrados en Madrid por la AEMET (-15.2 y 42.7 °C en la estación de Barajas), para no eliminar posibles valores extremos futuros o errores de calibración. La humedad relativa está medida en

porcentajes, por lo que el rango se acota a 0–100 y para la presión barométrica se adoptaron límites de plausibilidad de 850–1050 hPa, suficientes para abarcar las oscilaciones atmosféricas. Para la velocidad del viento se estableció un rango de 0-60 m/s, lo que incluye ráfagas extremas poco frecuentes pero físicamente posibles. De manera similar, para la radiación solar y la precipitación se definieron umbrales lo suficientemente amplios como para cubrir episodios máximos documentados.

Para el caso de los contaminantes atmosféricos se fija un valor mínimo de 0 ya que no puede haber mediciones de valores negativos cualquiera sea el contaminante. Para los máximos se aplican umbrales anti error muy altos para permitir picos extremos que pueden ser posibles pero evitar errores de medición generados por los sensores o unidades mal registradas. Para $PM_{2,5} \leq 2000 \mu\text{g}/\text{m}^3$, $PM_{10} \leq 3000 \mu\text{g}/\text{m}^3$, $NO_2 \leq 2000 \mu\text{g}/\text{m}^3$, $SO_2 \leq 2000 \mu\text{g}/\text{m}^3$, $CO \leq 50 \text{ mg}/\text{m}^3$ y $O_3 \leq 1000 \mu\text{g}/\text{m}^3$. Todos los casos se establecen como NaN para ser tratados por la etapa de valores faltantes.

Tabla 5.4: Rangos físicos plausibles para cada variable según literatura y normativa.

<i>Variable</i>	<i>Unidad</i>	<i>Mínimo</i>	<i>Máximo</i>
CO	mg/m ³	0	50
PM ₁₀	μg/m ³	0	3000
PM _{2,5}	μg/m ³	0	2000
SO ₂	μg/m ³	0	2000
NO ₂	μg/m ³	0	2000
O ₃	μg/m ³	0	1000
Radiación UV	índice	0	12+
Velocidad del viento	m/s	0	60
Dirección del viento	grados	0	360
Temperatura	°C	-20	50
Humedad relativa	%	0	100
Presión barométrica	hPa	850	1050
Radiación solar	W/m ²	0	1200
Precipitación	mm/h	0	300

5.2.3. Valores faltantes

Para que los modelos de redes neuronales artificiales funcionen correctamente se necesita que el conjunto de datos de entrada no tenga datos incompletos. La presencia de valores faltantes no permite que se realicen las operaciones matemáticas inherentes a la propagación y retropropagación, lo que afecta en la convergencia del modelo.

Previo a la imputación, se cuantifica el porcentaje de valores faltantes de los contaminantes por estación tras una reindexación a malla horaria.

Con base en las cifras y en las buenas prácticas de aprendizaje profundo para series temporales se hace la elección del método para tratar con los datos ausentes. La técnica más simple para lidiar con los valores faltantes es eliminar los registros incompletos. Sin embargo, esta técnica reduce el tamaño de la muestra y puede introducir sesgos si los datos no son completamente aleatorios, como sucede en el caso de los sensores que miden tanto los contaminantes como las variables meteorológicas. Por otro lado, elegir la media o la mediana para completar los valores tiende a distorsionar la varianza, las correlaciones entre variables, y por lo tanto, la variabilidad de la serie (Joel y Doorsamy, 2022).

Para el caso en el que el porcentaje de valores faltantes es mayor al 40% se descarta el par estación-contaminante si es variable objetivo y si es una variable de entrada solamente esa columna, por considerar que la imputación introduciría un sesgo excesivo. Para el resto de los casos se discriminan en dos dependiendo del largo de los huecos. Cuando los huecos de valores faltantes son hasta 3 horas continuas es por valores físicos atípicos y tramos planos, por lo que se adopta como estrategia la interpolación lineal sobre el eje temporal. Esta técnica es comúnmente usada en el tratamiento de series horarias temporales debido a que se mantiene la estructura temporal y la dinámica local de los datos (Tlameo et al. 2021).

Para el resto de los casos, se detecta que los huecos horarios en los que faltan valores son todos mayores o iguales a un día completo, 24 horas. Este comportamiento confirma que los datos inexistentes no son ruido aleatorio y pueden deberse a una falla en los sensores o cortes operativos. Por este motivo se define utilizar máscaras explícitas para que el modelo identifique la ausencia de datos. Se reindexa cada serie a malla horaria y se construyen tres señales por variable: (i) x_t^{imp} , una imputación mínima usada solo para completar la forma del tensor que se rellena con el valor 0 luego de normalizarse la serie; (ii) m_t , una máscara binaria que vale 1 cuando hay medición y 0 cuando falta; y (iii) Δ_t , el tiempo desde la última observación válida en horas. Las señales se alimentan juntas a los modelos de modo que aprenda a modular la confianza en x_t^{imp} cuando $m_t = 0$ y Δ_t es grande. Además, toda imputación y normalización se realiza dentro de cada partición (entrenamiento/validación) para evitar fuga de información (Lipton et al. 2016).

5.3. Transformación de los datos

En esta sección se preparan y transforman los datos para que tengan un formato que sea más acorde al análisis, aplicando operaciones de limpieza, normalización y reorganización de las series temporales.

En primer lugar, se realiza una winsorización local por cuantiles: para cada par estación-contaminante, los datos se limitan a $[Q_\alpha, Q_{1-\alpha}]$ con Q_α y $Q_{1-\alpha}$ estimados solo en el conjunto de entrenamiento. En este proyecto se adopta $\alpha = 0,001$, lo que genera un recorte en los percentiles 0.1 y 99.9. De este modo, los valores más extremos se sustituyen por los cuantiles de corte. Solo se realiza para las variables de entrada y no para las objetivo, para permitirle al modelo ver

los episodios extremos reales. Este procedimiento reduce la influencia de colas largas sin eliminar mediciones potencialmente informativas. Este procedimiento se realiza previo a la normalización para que los valores extremos no dominen la escala.

La normalización se aplica tanto a las variables predictoras como a las variables objetivo, lo que facilita la convergencia del entrenamiento y mejora la estabilidad numérica de las redes neuronales. Se aplica una normalización Min-Max que transforma cada variable x al intervalo $[0, 1]$ utilizando únicamente las estadísticas del conjunto de entrenamiento para evitar fugas de información. Para cada variable, se calcula el mínimo y el máximo, $x_{\text{mín,train}}$ y $x_{\text{máx,train}}$, y se normaliza según la Ecuación 5.1.

$$x_{\text{norm}} = \frac{x - x_{\text{mín,train}}}{x_{\text{máx,train}} - x_{\text{mín,train}}} \quad (5.1)$$

Dado que es una transformación lineal y reversible, los resultados pueden desnormalizarse y obtenerse en las unidades originales utilizando la Ecuación 5.2.

$$x = x_{\text{norm}} (x_{\text{máx,train}} - x_{\text{mín,train}}) + x_{\text{mín,train}} \quad (5.2)$$

Capítulo 6

Modelos predictivos

En este capítulo se presentan los modelos predictivos de redes neuronales desarrollados para predecir las concentraciones de los contaminantes atmosféricos y el procedimiento de entrenamiento empleado.

6.1. Modelos de redes neuronales

En esta sección se describen las arquitecturas de los diferentes modelos de redes neuronales artificiales. Se desarrollan tres modelos con distinta complejidad. El primero corresponde a un perceptrón multicapa, utilizado como base para establecer una referencia de rendimiento sin tener en cuenta la naturaleza secuencial de los datos. El segundo modelo adopta una arquitectura codificador–decodificador basada en redes LSTM, que permite capturar dependencias temporales a largo plazo presentes en los datos. Por último, se incorpora información espacial mediante una arquitectura híbrida GCN-LSTM, donde las relaciones entre las estaciones de monitoreo se representan mediante un grafo, lo que permite al modelo aprender de forma conjunta las interacciones espaciales y temporales.

6.1.1. Perceptrón multicapa

El perceptrón multicapa constituye el modelo base desarrollado. Este modelo presenta una estimación de los resultados, captando relaciones no lineales a través de la capa oculta. El modelo es utilizado como punto de referencia para la evaluación del desempeño de las arquitecturas secuenciales y espaciotemporales implementadas.

El diseño de la arquitectura del perceptrón multicapa se ilustra en la Figura 6.1. Es una red no recurrente que trata a la ventana de entrada como un conjunto de características estáticas. La primera capa de aplanamiento convierte los datos secuenciales en un vector unidimensional, transformando el tensor de entrada bidimensional (W, F) en un único vector de tamaño $W \times F$, donde W es la cantidad de pasos considerados y F el número de variables de entrada.

Esta capa se conecta con la capa oculta del modelo compuesta de ocho unidades, que comprime toda la información de los valores de entrada en un solo valor. Esta capa usa una función de activación no lineal ReLU, que le permite aprender patrones no lineales simples entre las variables de entrada. Luego se tiene la capa densa de salida que toma el valor comprimido de la capa anterior y lo proyecta linealmente a la cantidad total de valores $OUT_STEPS \times P$, donde OUT_STEPS es el horizonte de predicción y P la cantidad de contaminantes objetivo. Finalmente, se tiene la última capa que reestructura la salida a la forma que necesitan los datos para la comparación con las etiquetas.

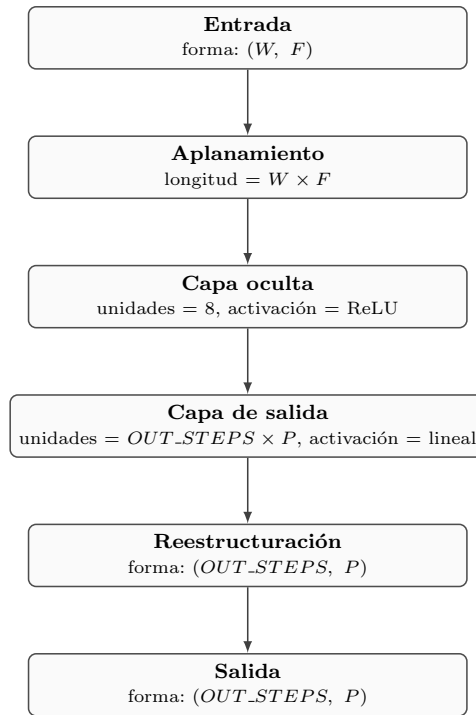


Figura 6.1: Arquitectura del modelo perceptrón multicapa.

6.1.2. LSTM codificador–decodificador

Con el objetivo de incorporar la dimensión temporal en la predicción, se implementó una red neuronal basada en la combinación de una arquitectura LSTM con un esquema codificador–decodificador. Este diseño, ilustrado en la Figura 6.2, permite capturar dependencias temporales a largo plazo naturales de los datos utilizados, modelando la evolución de las variables meteorológicas y de contaminación a lo largo del tiempo.

Dada una ventana de entrada $X \in \mathbb{R}^{W \times F}$, donde W es la longitud de la ventana temporal y F la cantidad de variables de entrada, se implementa un

esquema codificador–decodificador basado en redes LSTM con 64 unidades. El codificador lee la secuencia y la resume en un vector de contexto junto con sus estados finales (h, c) , que representan la memoria a corto y largo plazo respectivamente. El vector de contexto se replica $H = OUT_STEPS$ veces mediante una capa RepeatVector, generando una secuencia de longitud igual al horizonte de predicción. La secuencia se utiliza como entrada del decodificador con 64 unidades. Se inicializa con los estados finales del codificador (h, c) , procesa el contexto repetido y produce una secuencia de salida de igual longitud. Finalmente, una capa TimeDistributed asigna cada paso del decodificador a los P contaminantes objetivo, para generar predicciones con forma $\hat{Y} \in \mathbb{R}^{H \times P}$. La regularización ℓ_2 ($1,4910^{-6}$) se aplica a las capas LSTM del codificador y del decodificador. El modelo genera los valores de todos los OUT_STEPS en una sola vez hacia adelante.

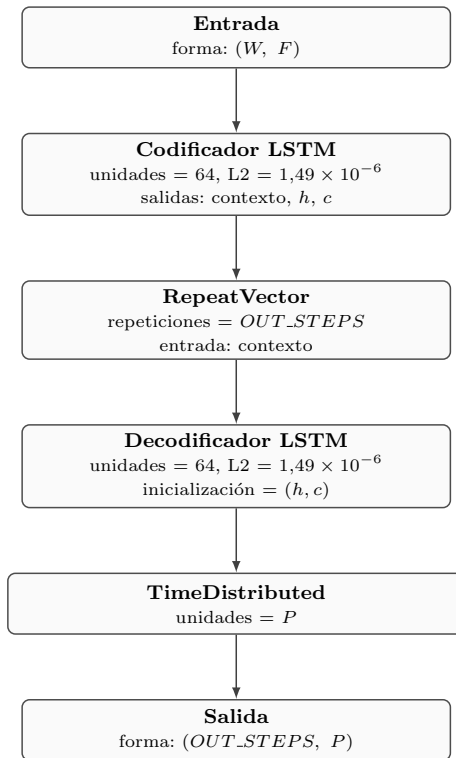


Figura 6.2: Arquitectura del modelo codificador–decodificador LSTM.

6.1.3. GCN-LSTM codificador–decodificador

Los datos utilizados tienen dependencias temporales y espaciales. Las relaciones temporales son tomadas en cuenta por el modelo presentado en la Subsección 6.1.2. A su vez, la concentración de los contaminantes puede verse afectada

por el cambio de las variables meteorológicas y de los propios contaminantes en estaciones cercanas, por lo que es relevante incorporar esta dimensión. Para modelar las relaciones espaciales entre las estaciones, se diseñó un modelo híbrido que combina el modelo LSTM codificador-decodificador con una red convolucional sobre grafos.

La representación espacial del sistema de monitoreo se modela mediante un grafo que describe las relaciones entre las estaciones. Se define un grafo no dirigido $G = (V, E)$ donde los vértices $V = \{v_1, v_2, \dots, v_N\}$ son las estaciones y las aristas E representan la conexión espacial entre ellas. Se considera que todas las estaciones están interconectadas, entonces cada par de vértices (v_i, v_j) se une mediante una arista e_{ij} . La relevancia de cada conexión, así como cuánto influyen los cambios de los valores de las variables en estaciones cercanas, se determina a través de los pesos w_{ij} asociados a las aristas. La ponderación de las aristas se define tomando en cuenta la distancia geográfica entre las estaciones, de manera que las más lejanas tienen conexiones más débiles. Los pesos se organizan en una matriz $W = [w_{ij}] \in \mathbb{R}^{N \times N}$, donde N es el número total de estaciones.

Las distancias entre estaciones se calculan a partir de sus coordenadas geográficas mediante la fórmula de Haversine, que determina la distancia entre dos puntos de la superficie terrestre, dadas las coordenadas de longitud y latitud. De esta manera se construye una matriz de distancias $D = [d_{ij}] \in \mathbb{R}^{N \times N}$, donde cada elemento d_{ij} representa la distancia en kilómetros entre las estaciones i y j . A partir de esta matriz se define la matriz de pesos W utilizando la inversa de la distancia, donde ε es un valor pequeño para evitar las divisiones por cero, según la Ecuación (6.1).

$$w_{ij} = \frac{1}{d_{ij} + \varepsilon}, \quad i, j = 1, \dots, N, \quad (6.1)$$

Los valores de la diagonal se fijan inicialmente en cero para impedir que una estación se conecte consigo misma con pesos excesivos derivados de la distancia nula entre estaciones. Posteriormente, se agregan lazos propios de peso unitario para que la información de una estación influya en la predicción de los valores futuros en la misma estación. Luego se aplica una normalización simétrica para permitir que todas las estaciones contribuyan de forma balanceada. La matriz de adyacencia normalizada A se define en la Ecuación (6.2).

$$A = \tilde{D}^{-\frac{1}{2}} W \tilde{D}^{-\frac{1}{2}}, \quad \tilde{D} = \text{diag} \left(\sum_j W_{ij} \right). \quad (6.2)$$

La matriz A resultante se usa como matriz de adyacencia normalizada en la red convolucional sobre grafos, lo que permite que la información se propague entre estaciones cercanas con una intensidad mayor cuanto más próximas se encuentran. En la Figura 6.3 se representan las estaciones y la conexión entre ellas, lo que forma el grafo completo con los nodos y aristas.

Como no todos los contaminantes son medidos por todas las estaciones, se consideró una partición del grafo inicial en subgrafos. Para cada contaminante

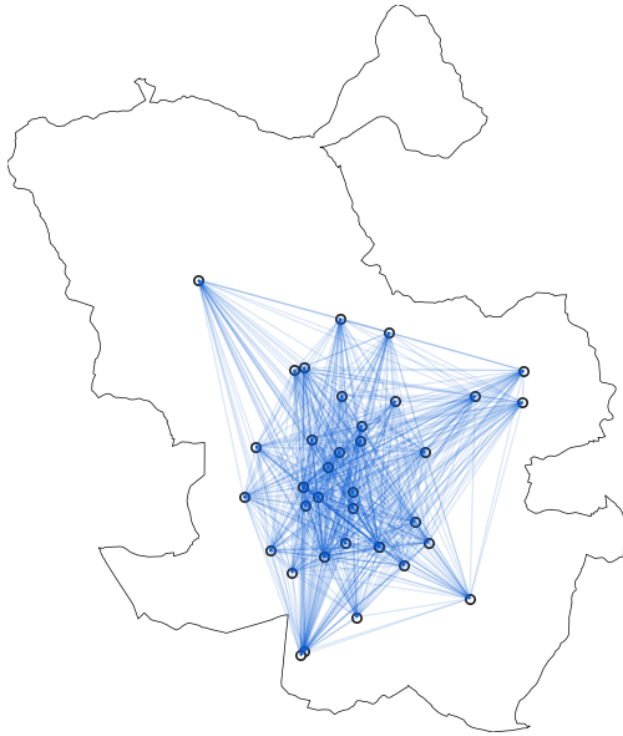


Figura 6.3: Estructura espacial del grafo empleado en el modelo GCN–LSTM, con conexiones entre estaciones (datos de datos.madrid.es).

c se restringió el grafo al subgrupo de estaciones que lo miden y se renormalizó dentro del subgrafo, tomando $W_c = W[S_c, S_c]$. A su vez, debido a la diferencia de variables meteorológicas y contaminantes medidos por cada estación, no todas las estaciones comparten el mismo conjunto de características. Para evitar que se tuviera una gran cantidad de variables nulas y tener que completarlas se definió un umbral de presencia del 80 %. Únicamente se incluyeron en el subgrafo las variables que estuvieran disponibles en al menos 80 % de las estaciones que midieran el contaminante objetivo. GCN—LSTM se entrenó y evaluó sobre cada subgrafo para garantizar que las predicciones se comparen solamente con valores reales del contaminante correspondiente.

Una vez definido el grafo, se usó para la construcción del modelo híbrido GCN-LSTM codificador–decodificador. Así, las relaciones espaciales capturadas por el GCN se combinan con las temporales modeladas por la LSTM. Este diseño se ilustra en la Figura 6.4.

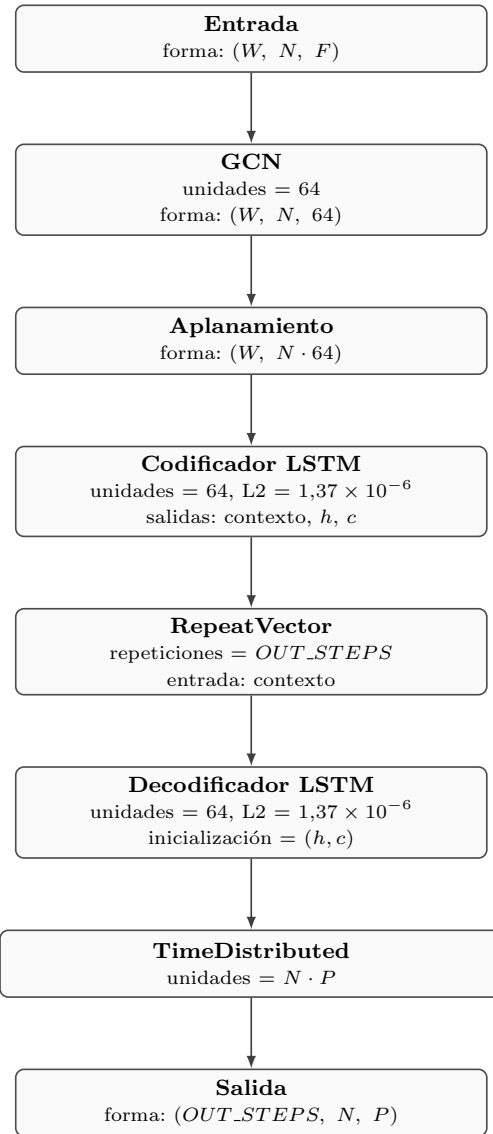


Figura 6.4: Arquitectura del modelo GCN-LSTM codificador-decodificador.

Capítulo 7

Experimentación

En este capítulo se presentan los experimentos realizados para la selección de los hiperparámetros y la evaluación del desempeño de los modelos predictivos. En la Sección 7.1 se describe el entorno de ejecución utilizado para los experimentos. En la Sección 7.3 se presenta el procedimiento de selección de hiperparámetros. Por último, en la Sección 7.5 se presenta el esquema de evaluación y el análisis de los resultados obtenidos para el Perceptrón multicapa, el modelo LSTM codificador–decodificador y el GCN–LSTM codificador–decodificador en las Subsecciones 7.5.1, 7.5.2 y 7.5.3, respectivamente.

7.1. Entorno de desarrollo y ejecución

Los experimentos realizados fueron ejecutados en un entorno con los recursos computacionales adecuados para el procesamiento de los datos y el entrenamiento de las redes neuronales.

La ejecución se realizó sobre un equipo con sistema operativo macOS, con chip Apple M1 y 8GB de memoria RAM. Los modelos fueron implementados utilizando el lenguaje de programación Python y principalmente las bibliotecas NumPy y pandas para el procesamiento de datos, y TensorFlow y scikit-learn para la implementación y el entrenamiento de los modelos de redes neuronales.

7.2. Configuración experimental

En esta sección se presentan las configuraciones definidas para el entrenamiento y la evaluación de los modelos predictivos, que se mantuvieron fijas en todos los experimentos para tener una comparación justa entre las diferentes arquitecturas. Las configuraciones no forman parte del proceso de selección de hiperparámetros y fueron definidas a partir de experimentos preliminares y del análisis exploratorio de los datos, considerando la presencia de patrones temporales característicos.

7.2.1. Esquema experimental

Ventanas temporales deslizantes. Para el entrenamiento y evaluación de los modelos predictivos y la elección de sus hiperparámetros, se utilizaron ventanas temporales deslizantes para transformar el conjunto de datos temporal original en pares entrada-salida. Cada entrada corresponde a una secuencia de observaciones pasadas y la salida una secuencia de valores a predecir. Así, el problema de predicción de series temporales se reformula como un problema de aprendizaje supervisado.

Para manejar las ventanas temporales se implementó una clase específica llamada WindowGenerator que se encarga de la creación de los pares entrada-salida. En la Figura 7.1 se muestra cómo cada ejemplo de entrenamiento se genera a partir de una ventana de entrada de longitud W que contiene las observaciones pasadas hasta el instante t y una secuencia de salida de longitud OUT_STEPS , correspondiente a los valores futuros a predecir. El parámetro desplazamiento es la separación entre el último instante de la ventana de entrada y el inicio de la secuencia de salida, permitiendo controlar el punto futuro desde el cual comienza la predicción.

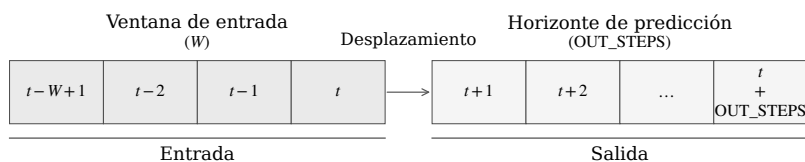


Figura 7.1: Esquema de generación de las ventanas temporales deslizantes.

Para la configuración de las ventanas se utilizó un desplazamiento igual al horizonte de predicción para que la información de entrada no se superponga con los valores futuros a predecir, lo que evitó que se filtre información futura y se preservara el orden temporal.

Por otro lado, la longitud de la ventana de entrada se fijó en 168 observaciones, que corresponden a una semana de datos horarios. Esta elección se basó en que los datos presentan patrones temporales asociados a ciclos diarios y semanales. Una ventana de 168 observaciones permitió capturar información temporal importante para la predicción sin extender la ventana de entrada, lo que evitó el aumento del costo computacional.

Configuración del entrenamiento. El entrenamiento de todos los modelos se realizó utilizando una cantidad máxima fija de épocas y un criterio de parada temprana basado en la pérdida del conjunto de validación para que el modelo no se sobreajustara en el entrenamiento. El número de épocas se fijó en 200, lo que permitió a los modelos alcanzar la convergencia durante el entrenamiento. El criterio de parada se definió en 10 épocas para poder detener el entrenamiento

cuando la pérdida de validación dejara de mejorar de forma sostenida.

El tamaño de lote se fijó en 32 observaciones y se mantuvo para todos los modelos y sus respectivos entrenamientos, para poder compararlos consistentemente. Este valor permitió obtener estabilidad en el entrenamiento y un adecuado costo computacional.

A su vez, se utilizó el optimizador Adam por su capacidad de ajustar la tasa de aprendizaje de manera individual para cada parámetro durante el entrenamiento, dado que los modelos involucran varios parámetros y dinámicas complejas. Se evaluaron otros optimizadores, como el descenso de gradiente estocástico, pero no se observaron mejoras en la convergencia y el desempeño de los modelos, por lo que no fueron utilizados en la etapa de búsqueda de hiperparámetros.

7.3. Selección de hiperparámetros

En esta sección se describe el procedimiento utilizado para la selección de los hiperparámetros de los modelos predictivos y los resultados obtenidos.

7.3.1. Estrategia de búsqueda

Para la selección de hiperparámetros se realizó una etapa inicial manual, seguida de una etapa de búsqueda automática. La búsqueda permitió entender el comportamiento de los modelos frente a diferentes combinaciones de parámetros y, posteriormente, explorar el espacio de hiperparámetros de manera sistemática y reproducible.

En la primera etapa se realizó un ajuste manual de los hiperparámetros teniendo en cuenta el análisis del error de entrenamiento y del de validación. El ajuste manual requirió entender la relación entre los hiperparámetros, la capacidad del modelo y el impacto tanto en el error de entrenamiento como en el error de generalización (Goodfellow et al. 2016). Algunos hiperparámetros influyen en la capacidad del modelo para representar relaciones complejas, como el número de unidades ocultas. Los hiperparámetros de regularización, como el coeficiente ℓ_2 y el uso de dropout, permiten controlar el sobreajuste. Otros hiperparámetros, como la tasa de aprendizaje, están asociados a la optimización. Este hiperparámetro es importante porque los valores grandes pueden impedir la convergencia del entrenamiento y los valores pequeños pueden generar entrenamientos excesivamente lentos o soluciones subóptimas.

Aunque el ajuste manual permitió identificar configuraciones razonables y descartar valores inadecuados, es un enfoque limitado. Para poder explorar de manera sistemática el espacio de hiperparámetros cuando se utilizan múltiples parámetros y su relación no es sencilla de entender, se requiere un método automático. Por esta razón se adoptó una estrategia de búsqueda aleatoria para la selección de hiperparámetros. La búsqueda aleatoria ha demostrado ser más eficiente que la búsqueda en grilla (Goodfellow et al. 2016) teniendo en cuenta el número de evaluaciones necesarias para alcanzar configuraciones adecuadas.

La búsqueda aleatoria permite que en cada experimento se utilice una configuración completamente diferente, evitando repetir experimentos similares. La búsqueda aleatoria se implementó mediante la ejecución de múltiples experimentos independientes denominados trials. Cada trial correspondió al entrenamiento completo del modelo con una configuración de hiperparámetros muestreada aleatoriamente, seguido de su evaluación sobre el conjunto de validación.

El perceptrón multicapa fue utilizado como modelo base por lo cual no se realizó una búsqueda de hiperparámetros para esta arquitectura. El objetivo del modelo base fue proporcionar una referencia simple contra la cual comparar el desempeño de modelos más complejos y no maximizar su rendimiento predictivo.

7.3.2. Espacio de hiperparámetros

El espacio de hiperparámetros fue definido teniendo en cuenta los parámetros considerados en la etapa manual y aquellos con más influencia en la capacidad del modelo, la regularización y la optimización. Para cada modelo predictivo se seleccionó un conjunto de hiperparámetros para mantener un equilibrio entre la exploración del espacio de configuraciones y el costo computacional asociado al entrenamiento.

Algunos hiperparámetros tienen valores continuos, por lo tanto se utilizó un esquema basado en distribuciones continuas en escala logarítmica. Este enfoque permitió explorar de manera más efectiva órdenes de magnitud distintos evitando la discretización de los parámetros continuos (Goodfellow et al. 2016).

La tasa de aprendizaje no se usó directamente, sino que se muestreó su logaritmo en base 10 de forma uniforme y se transformó al espacio original, como se define en las Ecuaciones 7.1 y 7.2. Este esquema de muestreo permitió asignar la misma importancia a los distintos órdenes de magnitud del parámetro.

$$u \sim \mathcal{U}(a, b) \tag{7.1}$$

$$learning_rate = 10^u \tag{7.2}$$

El coeficiente de regularización ℓ_2 tampoco se utilizó directamente y se siguió el mismo procedimiento que con la tasa de aprendizaje, como se define en las Ecuaciones 7.3 y 7.4.

$$v \sim \mathcal{U}(c, d) \tag{7.3}$$

$$\lambda = 10^v \tag{7.4}$$

El número de unidades ocultas y la tasa de dropout toman valores discretos y fueron muestreados a partir de conjuntos finitos predefinidos. Esta elección responde a que dichos parámetros producen cambios discretos en la capacidad del modelo y en el esquema de regularización, por lo que resulta suficiente evaluar un conjunto reducido de configuraciones representativas. El muestreo se realizó de manera uniforme sobre los valores considerados.

Cada configuración de hiperparámetros se definió como un trial independiente que abarcó el entrenamiento del modelo y la evaluación sobre el conjunto de validación. El desempeño obtenido en la evaluación se utilizó para comparar

configuraciones y elegir los valores finales de los hiperparámetros para cada modelo. La Tabla 7.1 resume los tipos de hiperparámetros considerados durante la búsqueda aleatoria.

Tabla 7.1: Espacio de hiperparámetros considerado.

<i>Hiperparámetro</i>	<i>Descripción</i>
Tasa de aprendizaje	Tasa de aprendizaje del optimizador
Regularización ℓ_2	Penalización sobre los pesos del modelo
Capacidad del modelo	Número de unidades ocultas
Dropout	Regularización por desactivación

7.3.3. Resultados de la selección de hiperparámetros

Para la selección de las configuraciones más adecuadas del modelo temporal y del modelo espacio-temporal se realizaron 20 ejecuciones y se seleccionaron los valores de los hiperparámetros que minimizaron la función de pérdida en el conjunto de validación.

7.3.3.1. Modelo LSTM

Para LSTM se realizó una búsqueda aleatoria del número de unidades de las capas LSTM, la tasa de aprendizaje, el coeficiente de regularización ℓ_2 y la tasa de dropout siguiendo los valores y distribuciones presentados en la Tabla 7.2.

Tabla 7.2: Valores de hiperparámetros considerados para el modelo LSTM.

<i>Hiperparámetro</i>	<i>Distribución / Valores</i>
Tasa de aprendizaje	$\log_{10}(lr) \sim \mathcal{U}(-4, -3)$
ℓ_2	$\log_{10}(\lambda) \sim \mathcal{U}(-6, -3)$
Unidades codificador y decodificador	{32, 64, 128}
Dropout	{0,0, 0,2, 0,3}

La búsqueda de hiperparámetros se realizó para los dos horizontes de predicción de una y 24 horas, con el objetivo de analizar el comportamiento del modelo en predicciones inmediatas y evaluar las configuraciones óptimas cuando se requiere predecir varias horas en el futuro.

Los resultados obtenidos en ambas configuraciones se muestran en la Tabla 7.3. Los valores de los hiperparámetros seleccionados fueron similares, lo que sugiere que la configuración de LSTM es relativamente estable independientemente de la cantidad de horas a predecir. La capacidad del modelo es la misma en ambas configuraciones. En cambio, la tasa de aprendizaje y el coeficiente de regularización son menores para las predicciones a horizontes más

lejanos, aunque del mismo orden de magnitud. Dado que la forma de generalizar las predicciones del modelo es considerar horizontes más lejanos, se adoptó como configuración final la obtenida al probar con el horizonte de predicción de 24 horas. El Apéndice A.1 muestra los resultados de todas las ejecuciones con las diferentes configuraciones y horizontes de predicción.

Tabla 7.3: Hiperparámetros seleccionados en los horizontes de predicción.

<i>Hiperparámetro</i>	<i>OUT_STEPS = 1</i>	<i>OUT_STEPS = 24</i>
Unidades LSTM	64	64
Tasa de aprendizaje	$1,53 \times 10^{-4}$	$1,43 \times 10^{-4}$
ℓ_2	$8,18 \times 10^{-6}$	$1,49 \times 10^{-6}$
Dropout	0.0	0.0

7.3.3.2. Modelo GCN–LSTM

Para GCN–LSTM se exploraron mediante la búsqueda aleatoria distintos valores para la cantidad de unidades ocultas de las capas LSTM del codificador y del decodificador, la tasa de aprendizaje, el coeficiente de regularización ℓ_2 , la tasa de dropout aplicada al bloque GCN y el número de unidades ocultas de la capa GCN. En la Tabla 7.4 se presentan los valores y distribuciones utilizados en las ejecuciones.

Tabla 7.4: Valores de hiperparámetros considerados para el modelo GCN–LSTM.

<i>Hiperparámetro</i>	<i>Distribución / Valores</i>
Tasa de aprendizaje	$\log_{10}(lr) \sim \mathcal{U}(-4, -3)$
ℓ_2	$\log_{10}(\lambda) \sim \mathcal{U}(-6, -3)$
Unidades GCN	{32, 64, 128}
Unidades LSTM	{32, 64, 128}
Dropout GCN	{0,0, 0,2, 0,3}

El modelo espacio-temporal se entrenó sobre grafos específicos para cada contaminante, como se describe en la Sección 6.1.3. Para cada contaminante, se construyó un subgrafo con las estaciones que lo miden. Sin embargo, la selección de los hiperparámetros se realizó de manera conjunta para todos los contaminantes objetivo. Para cada configuración de la búsqueda aleatoria se entrenaron modelos independientes utilizando los respectivos subgrafos, manteniendo la misma configuración de hiperparámetros. Cada configuración se evaluó a partir del error de validación promedio sobre todos los contaminantes, haciendo que la selección final sea generalizable y no optimice únicamente el desempeño en la predicción de un contaminante en particular.

Al igual que para LSTM, se ejecutó la búsqueda aleatoria para ambos horizontes de predicción. La Tabla 7.5 presenta los resultados para ambos horizontes. Los resultados completos de la búsqueda aleatoria, considerando todas las configuraciones y horizontes de predicción, se incluyen en el Apéndice A.2.

Tabla 7.5: Hiperparámetros seleccionados en los horizontes de predicción.

<i>Hiperparámetro</i>	<i>OUT_STEPS = 1</i>	<i>OUT_STEPS = 24</i>
Tasa de aprendizaje	$6,43 \times 10^{-4}$	$1,57 \times 10^{-4}$
Coficiente ℓ_2	$8,20 \times 10^{-6}$	$1,37 \times 10^{-6}$
Unidades GCN	64	64
Unidades LSTM	64	64
Dropout GCN	0.0	0.0

7.4. Esquema de evaluación

Para evaluar los diferentes modelos implementados se aplicaron dos esquemas complementarios: un hold-out temporal y un esquema de validación cruzada temporal.

En el esquema hold-out, el conjunto de datos entero se dividió respetando el orden cronológico de la serie, utilizando el 70 % de los datos para entrenamiento, el 20 % para validación y el 10 % restante para la prueba. Con este enfoque el conjunto de prueba está formado por el período más reciente disponible y se pueden comparar directamente las distintas arquitecturas utilizando todos los datos.

Para el esquema de validación cruzada temporal se utilizaron tres particiones consecutivas, cada una con 18 meses de entrenamiento, 6 meses de validación y 6 meses de prueba. Este enfoque permite analizar la estabilidad del modelo en diferentes períodos de la serie. En la primera partición, el período de entrenamiento abarcó desde el 01/07/2021 hasta el 23/11/2022, el conjunto de validación desde el 23/11/2022 hasta el 22/05/2023 y el conjunto de prueba desde el 22/05/2023 hasta el 18/11/2023. En la segunda partición, el entrenamiento se realizó entre el 27/01/2022 y el 21/07/2023, la validación entre el 21/07/2023 y el 17/01/2024, y la prueba entre el 17/01/2024 y el 15/07/2024. Finalmente, en la tercera partición, el período de entrenamiento se extendió desde el 24/09/2022 hasta el 17/03/2024, la validación desde el 17/03/2024 hasta el 13/09/2024 y la prueba desde el 13/09/2024 hasta el 31/12/2024. Este enfoque siguió un orden cronológico, evitando que se filtraran comportamientos futuros y permitiendo la evaluación en distintos intervalos temporales. Los resultados finales son reportados como el promedio de la métrica de evaluación obtenida en las tres particiones.

Se utilizaron diferentes métricas que permiten analizar el comportamiento de los modelos. En primer lugar MAE que mide el error promedio en las predicciones. También se considera RMSE que penaliza más los errores de gran

magnitud, lo cual es útil para entender el comportamiento del modelo cuando hay picos de concentración. Además de las métricas anteriores, para el esquema de hold-out temporal se calcularon métricas específicas para el horizonte de predicción $t+k$. Las métricas fueron MAE_{t+k} y $RMSE_{t+k}$, que permiten evaluar el error asociado al último paso que el modelo predice, que lo diferencia del error promedio sobre toda la secuencia de salida. En el caso de predicción de una hora hacia adelante las métricas asociadas al último paso de predicción coinciden con las métricas del error promedio.

Para el análisis de las predicciones se seleccionó la estación 8 como caso representativo, ya que corresponde a una estación urbana de tráfico y cuenta con mediciones de los tres contaminantes objetivo. Esta estación permite analizar el comportamiento de los modelos en un entorno urbano influenciado por diferentes fuentes de emisión. Las gráficas adicionales para estaciones suburbanas y urbanas de fondo se encuentran en el Apéndice B y las métricas en el Apéndice C.

7.5. Análisis de resultados

En esta sección se presenta el análisis de los resultados obtenidos para las diferentes arquitecturas implementadas.

7.5.1. Modelo Perceptrón multicapa

El modelo del perceptrón actuó como línea base y se usó para definir una referencia de desempeño frente a las arquitecturas temporales y espacio-temporales más complejas. En la Figura 7.2 se muestra la pérdida del entrenamiento y de la validación para ambos horizontes de predicción. En el caso del horizonte de una hora la pérdida disminuyó progresivamente. En cambio, para el horizonte de 24 horas la disminución fue más abrupta. En ambos casos, las curvas de entrenamiento y validación fueron cercanas, indicando que no hubo un sobreajuste significativo.

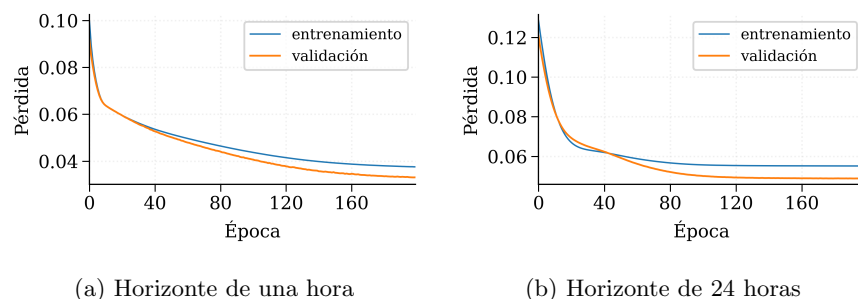


Figura 7.2: Convergencia de la función de pérdida del modelo MLP en la estación urbana de tráfico Escuelas Aguirre

La Tabla 7.6 muestra los valores de las métricas obtenidos para una y 24 horas. Los errores obtenidos se interpretan en relación con la magnitud y la variabilidad de los contaminantes. El contaminante $\text{PM}_{2,5}$ tuvo alta variabilidad temporal, con una concentración media de aproximadamente $9.8 \mu\text{g}/\text{m}^3$ y una desviación estándar de $9.0 \mu\text{g}/\text{m}^3$. La serie de $\text{PM}_{2,5}$ es ruidosa y presenta valores abruptos debido a picos en la contaminación. Para el horizonte de una hora, el MAE obtenido fue elevado en comparación con la variabilidad de la serie, indicando que el MLP tuvo dificultad para capturar la dinámica temporal. El RMSE fue mayor que el MAE, lo que significa que hubo errores puntuales de mayor magnitud que pueden estar asociados a picos de concentración y que se deben a que el modelo no logró anticiparlos. Para NO_2 y O_3 , el comportamiento del modelo fue similar. En ambos casos, los errores obtenidos para el horizonte de una hora fueron elevados en relación con los niveles característicos de cada contaminante y el RMSE superó al MAE, indicando la presencia de errores puntuales de mayor magnitud. En NO_2 la diferencia entre RMSE y MAE fue más notoria y en el caso de O_3 , aunque el error fue alto, la diferencia entre MAE y RMSE fue menor, lo que indica una menor cantidad de errores extremos en comparación con $\text{PM}_{2,5}$ y NO_2 .

En el caso del horizonte de 24 horas los valores de MAE y RMSE fueron similares para todos los contaminantes. A su vez, los valores de MAE y MAE_{t+24} , y RMSE y RMSE_{t+24} fueron cercanos entre sí, indicando que el error no incrementó al final del horizonte de predicción. Esto sugiere que el MLP generó predicciones suavizadas y cercanas al valor medio.

Tabla 7.6: MAE y RMSE del modelo MLP para horizontes de una y 24 horas en la estación Escuelas Aguirre ($\mu\text{g}/\text{m}^3$).

Contaminante	1 h		24 h			
	MAE	RMSE	MAE	MAE_{t+24}	RMSE	RMSE_{t+24}
$\text{PM}_{2,5}$	6.30	7.59	6.78	6.89	8.06	8.03
NO_2	12.87	15.97	16.94	16.22	21.45	21.60
O_3	8.71	10.90	22.93	23.05	26.72	26.56

La Figura 7.3 muestra el comportamiento general del MLP en el horizonte de una hora. En los tres contaminantes, el modelo predijo valores suavizados cercanos al valor medio de cada serie y tuvieron una variación menor que los valores reales observados, lo que explica los valores de la Tabla 7.6. En el caso de O_3 , la predicción siguió más de cerca la tendencia general de la serie, aunque el modelo también suavizó los valores y no capturó completamente los picos de concentración.

Como se muestra en la Figura 7.4, el comportamiento del MLP fue similar al predecir 24 pasos en el futuro. Las predicciones del modelo base fueron suavizadas y cercanas al valor medio de cada contaminante. El modelo base subestimó los picos y no reprodujo adecuadamente la amplitud de los valores reales.

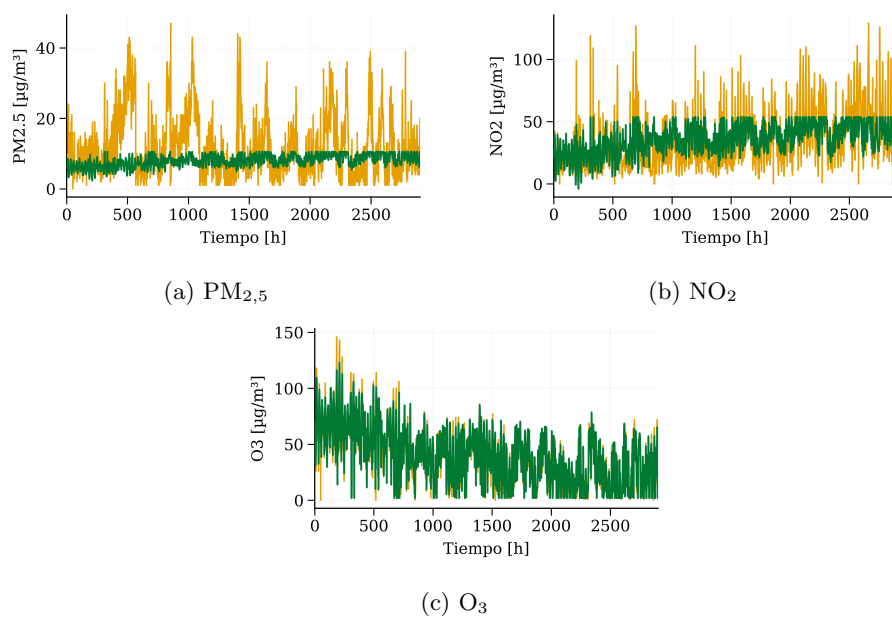


Figura 7.3: Predicciones a una hora del modelo MLP en la estación urbana de tráfico Escuelas Aguirre
 (— : concentraciones reales, — : concentraciones predichas)

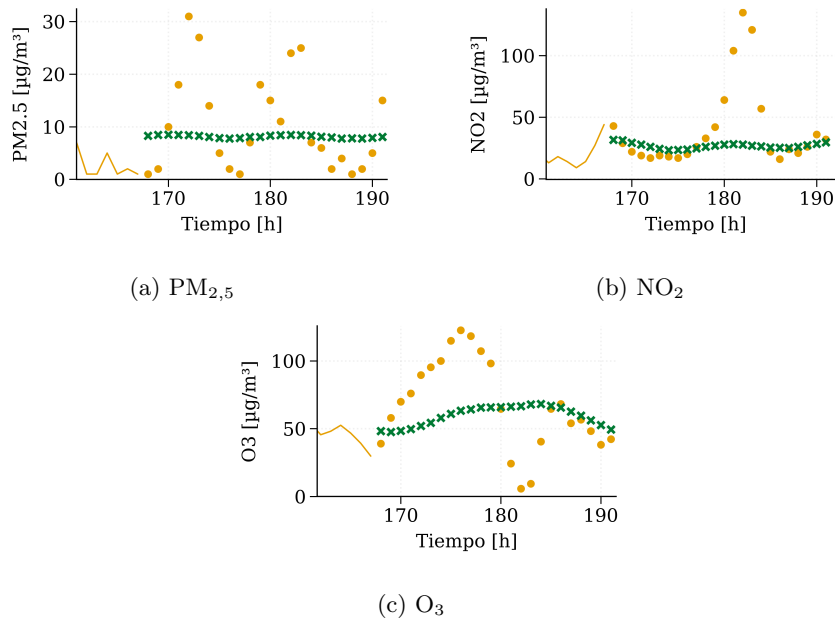


Figura 7.4: Predicciones a 24 horas del modelo MLP en la estación urbana de tráfico Escuelas Aguirre
(— : concentraciones reales, — : concentraciones predichas)

7.5.2. Modelo LSTM codificador–decodificador

La evaluación del modelo LSTM codificador-decodificador también consistió en realizar el hold-out temporal para los horizontes de una y 24 horas. El modelo es multivariado, por lo que predijo simultáneamente las concentraciones de NO_2 , O_3 y $\text{PM}_{2.5}$.

La Figura 7.5 muestra la convergencia de las pérdidas de entrenamiento y validación para la estación de tráfico urbano Escuelas Aguirre. Para ambos horizontes de predicción, la pérdida disminuyó consistentemente a medida que avanzan las épocas en el conjunto de entrenamiento y en el de validación. Ambas curvas disminuyen hasta un punto estable, con una pequeña diferencia entre los valores finales. El comportamiento de las curvas indica que el modelo aprendió sin perder la capacidad de generalización. La convergencia para el horizonte de 24 horas fue más lenta, indicando la mayor complejidad del problema cuando aumenta la cantidad de pasos futuros a predecir.

La Tabla 7.7 muestra las métricas obtenidas para los horizontes de una y 24 horas en la estación Escuelas Aguirre. En comparación al MLP, LSTM redujo los errores para todos los contaminantes en ambos horizontes de predicción.

Para el horizonte de una hora, LSTM obtuvo errores menores para todos los contaminantes objetivo en comparación con el modelo base. En todos los casos,

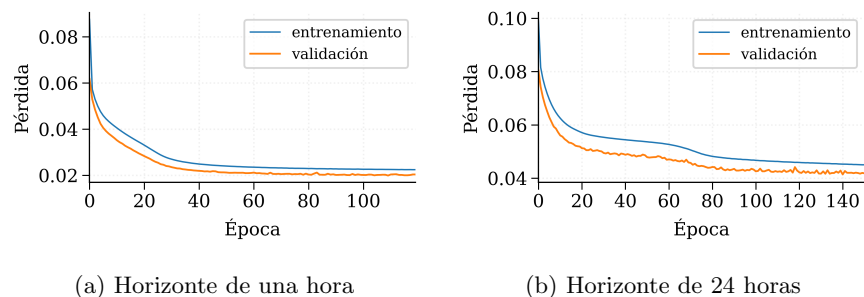


Figura 7.5: Convergencia de la función de pérdida del modelo LSTM en la estación urbana de tráfico Escuelas Aguirre

Tabla 7.7: MAE y RMSE del modelo LSTM codificador–decodificador para horizontes de 1 h y 24 h en la estación Escuelas Aguirre ($\mu\text{g}/\text{m}^3$).

Contaminante	1 h		24 h			
	MAE	RMSE	MAE	MAE _{t+24}	RMSE	RMSE _{t+24}
PM _{2,5}	2.81	3.62	5.31	6.03	6.50	7.05
NO ₂	5.47	7.80	11.89	12.90	15.60	16.52
O ₃	4.94	6.80	12.38	13.83	15.21	16.57

los errores fueron menores que la variabilidad de la serie, lo que demuestra la capacidad del modelo para capturar dinámicas temporales a corto plazo. En términos de MAE, la reducción fue de aproximadamente un 55 % para PM_{2,5}, un 57 % para NO₂ y un 43 % para O₃.

En el caso de PM_{2,5}, los errores se redujeron a la mitad respecto al MLP. A su vez, en la Figura 7.6a se muestra que el modelo siguió más de cerca la tendencia del contaminante, capturando los incrementos y descensos de los valores reales. Aún así, no logró predecir exactamente los picos de concentración ni las disminuciones abruptas. Para el contaminante NO₂ se obtuvo un comportamiento consistente con el de PM_{2,5}. MAE fue bajo en comparación con la variabilidad de la serie y los errores fueron menores respecto al modelo base. Al igual que para PM_{2,5}, en la Figura 7.6b el modelo siguió de forma más consistente las variaciones de los valores, pero con algunas dificultades durante picos de concentración, lo cual fue consistente con los valores más altos de RMSE en comparación con MAE. En el caso de O₃, en la Figura 7.6c se muestra que las predicciones de LSTM se acercaron con mayor precisión a los valores reales en comparación con el MLP. Aunque el modelo base pudo capturar la tendencia general y parte de la dinámica de la serie, LSTM redujo los errores obtenidos.

En el horizonte de 24 horas, MAE y RMSE aumentaron para todos los contaminantes, lo que sugiere una degradación del desempeño del modelo cuando incrementa el número de pasos en el futuro. Aún siendo mayores, los errores se mantuvieron menores a los errores obtenidos por el modelo base. Los valores de

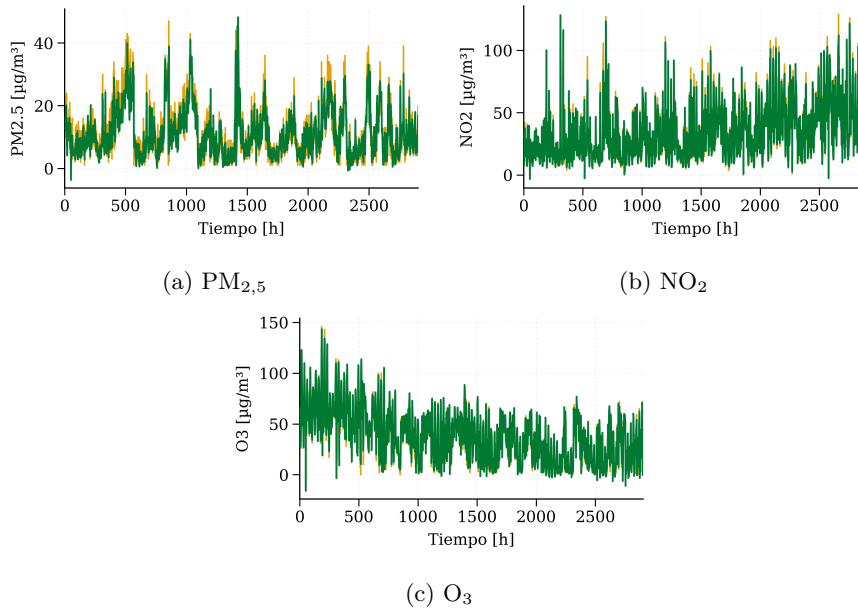


Figura 7.6: Predicciones a una hora del modelo LSTM codificador–decodificador en la estación urbana de tráfico Escuelas Aguirre (— : concentraciones reales, — : concentraciones predichas)

MAE_{t+24} y RMSE_{t+24} fueron mayores que los errores promedio para 24 horas para todos los contaminantes. Además, la diferencia entre los errores promedio y los del último paso del horizonte fue mayor que en el modelo base. En LSTM, el error cambió según cuán lejos se predijo, lo que indica que el modelo tuvo en cuenta el paso del tiempo, a diferencia del MLP que mantuvo el error constante.

La Figura 7.7 muestra un tramo de las predicciones del horizonte de 24 horas. En el caso del $\text{PM}_{2.5}$, el modelo siguió la tendencia descendente de los valores reales, aunque suavizó las variaciones abruptas de la serie. Las predicciones fueron más estables que los valores reales, lo que explica que los valores más extremos tuvieron mayor impacto en RMSE en comparación con MAE. Para NO_2 , el modelo capturó la dinámica general de la serie y, a diferencia de lo que ocurrió con el material particulado, logró capturar el mínimo y máximo en el tramo. Aún así, la magnitud del pico fue poco menor que la real. El modelo tuvo un comportamiento similar para el caso del O_3 ya que siguió la tendencia de la serie, incluyendo los cambios abruptos y reprodujo el máximo y el posterior descenso.

7.5.3. Modelo GCN–LSTM codificador–decodificador

El modelo más complejo incorporó la dimensión espacial de los datos construyendo subgrafos específicos para cada contaminante y se entrenó un modelo

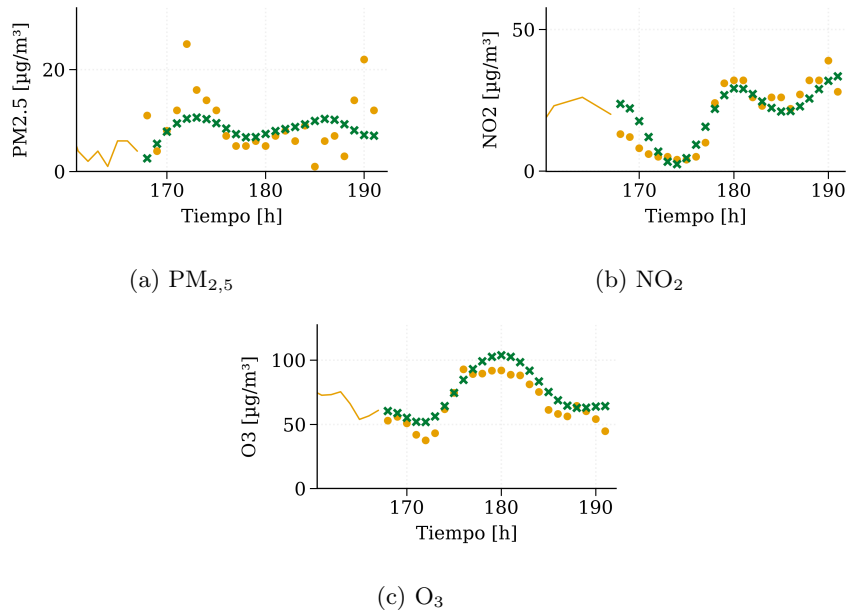


Figura 7.7: Predicciones a 24 horas del modelo LSTM codificador—decodificador en la estación urbana de tráfico Escuelas Aguirre (— : concentraciones reales, — : concentraciones predichas)

independiente por cada uno. Como consecuencia se obtuvieron curvas de pérdida diferentes para cada contaminante objetivo. La Figura 7.8 muestra las curvas de pérdida de entrenamiento y validación para el contaminante $PM_{2,5}$ en horizontes de predicción de una y 24 horas. En las primeras épocas la pérdida disminuyó y luego se estabilizó. Hubo un equilibrio entre los errores de entrenamiento y validación, sugiriendo que el modelo generalizó correctamente sin sobreajustar.

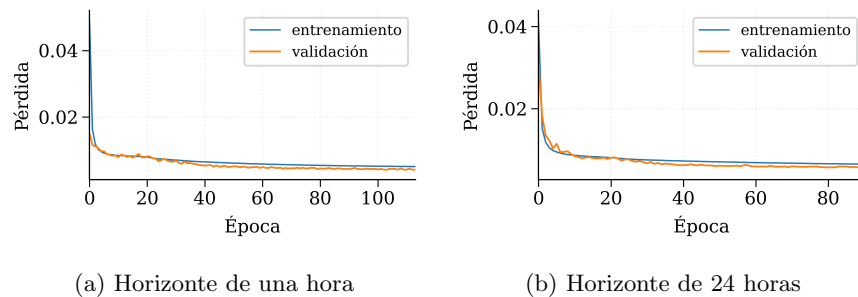


Figura 7.8: Convergencia de la función de pérdida del modelo GCN–LSTM en la estación urbana de tráfico Escuelas Aguirre para $PM_{2,5}$

En la Tabla 7.8 se presentan los resultados obtenidos para el hold-out temporal realizado para el modelo espacio-temporal en la estación Escuelas Aguirre. En todos los casos GCN-LSTM superó al base y redujo el error entre 25 % y 50 % dependiendo del contaminante y horizonte de predicción.

En comparación con LSTM, GCN-LSTM no obtuvo mejoras en el horizonte de una hora en términos de MAE y RMSE. Este resultado puede explicarse por el hecho de que las concentraciones futuras de los contaminantes dependen fuertemente del estado inmediato de la serie. En esta situación, las relaciones espaciales no agregan información en las predicciones a corto plazo y el patrón únicamente temporal captó adecuadamente las dinámicas presentes en los datos.

Por el contrario, en el horizonte de 24 horas el modelo espacio-temporal presentó mejoras de hasta el 10 % en comparación a LSTM. Este comportamiento sugiere que a medida que aumenta el horizonte de predicción, los valores a predecir no dependen únicamente del estado inmediato, sino también de condiciones meteorológicas acumuladas, transporte de contaminantes y otros procesos acumulativos. En este caso, hay evidencia de que la estructura espacial aporta información útil en predicciones de mayor alcance temporal.

Tabla 7.8: MAE y RMSE del modelo GCN-LSTM codificador-decodificador para horizontes de 1 h y 24 h en la estación Escuelas Aguirre ($\mu\text{g}/\text{m}^3$).

Contaminante	1 h		24 h			
	MAE	RMSE	MAE	MAE _{t+24}	RMSE	RMSE _{t+24}
PM _{2,5}	3.36	4.09	4.77	5.19	6.05	6.40
NO ₂	6.26	8.92	10.94	11.87	15.75	16.76
O ₃	6.05	8.16	11.58	12.20	13.99	15.62

Las predicciones a una hora se ilustran en la Figura 7.9, donde el modelo siguió de cerca los valores de la serie y logró predecir los picos de contaminación y los descensos, aunque suavizando los valores altos. Para O₃ la diferencia con el modelo base no fue grande, por lo que el aporte temporal y espacial no es notoria en las gráficas de las predicciones.

En la Figura 7.10 se muestran ejemplos de predicción de 24 horas obtenidos. En todos los casos el modelo logró capturar la tendencia general de las series temporales. Sin embargo, para todos los contaminantes hubo una subestimación de los picos máximos.

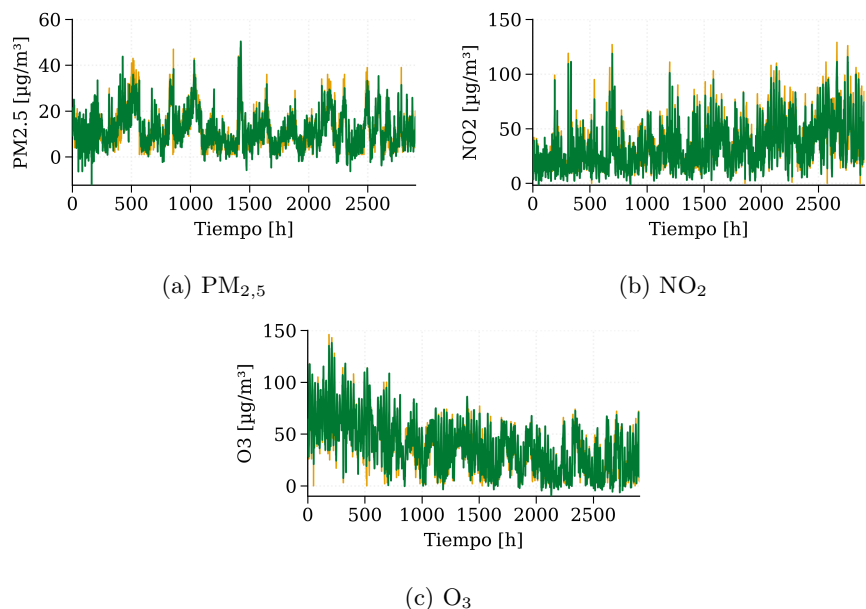


Figura 7.9: Predicciones a una hora del modelo GCN-LSTM codificador-decodificador en la estación urbana de tráfico Escuelas Aguirre (— : concentraciones reales, — : concentraciones predichas)

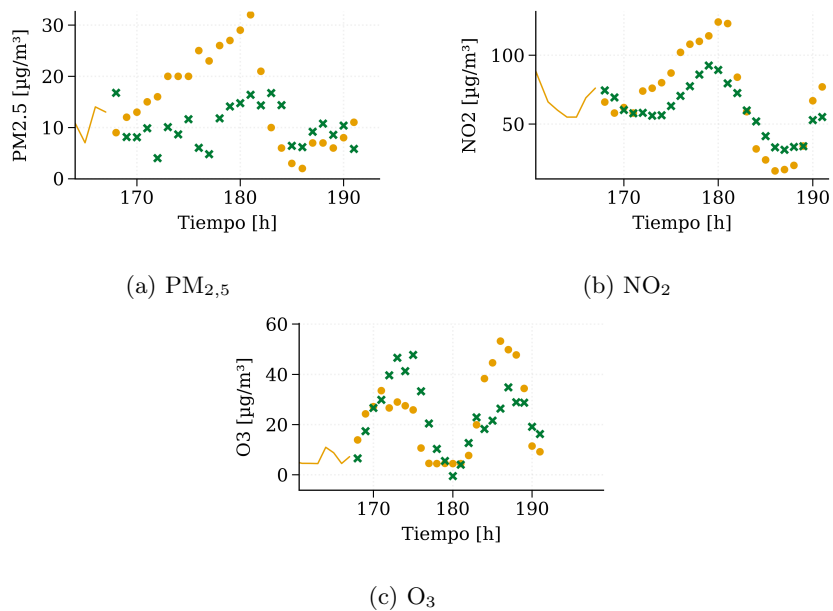


Figura 7.10: Predicciones a 24 horas del modelo GCN-LSTM codificador-decodificador en la estación urbana de tráfico Escuelas Aguirre (— : concentraciones reales, x : concentraciones predichas)

7.5.4. Validación cruzada temporal

Para analizar la estabilidad de los modelos y la variación del error en diferentes periodos de entrenamiento y prueba, se utilizaron las particiones definidas en la Sección 7.4. En las tablas 7.9 y 7.10 se presentan los promedios de las métricas obtenidas en las tres particiones para cada modelo, en ambos horizontes de predicción. Estos resultados son consistentes con los obtenidos en el hold-out temporal realizado.

En las predicciones de corto y largo plazo, los modelos secuenciales superaron el desempeño del modelo base en los tres contaminantes. Las arquitecturas recurrentes lograron capturar la dinámica temporal de mejor manera. LSTM obtuvo los menores errores en el horizonte de una hora en MAE y RMSE. Aunque los errores obtenidos por el modelo espacio-temporal no estuvieron alejados, los resultados sugieren que la información espacial no tuvo un gran impacto en la mejora de las predicciones a corto plazo.

En cuanto a la estabilidad de los modelos, LSTM y GCN-LSTM presentaron un desempeño similar en los distintos períodos evaluados. Las menores desviaciones estándar indican que los modelos no dependen de un período específico y tienen capacidad de generalización. Por el contrario, el perceptrón multicapa presentó una alta desviación en las mediciones de todos los contaminantes, indicando que su desempeño depende fuertemente del período de evaluación utilizado. En algunas particiones los errores fueron mayores que en otras, sugiriendo que el modelo no pudo adaptarse y que es menos estable temporalmente.

En el horizonte de 24 horas el comportamiento fue diferente. Los modelos secuenciales superaron al modelo base, pero en este caso el componente espacial presentó mejoras. GCN-LSTM obtuvo los menores valores de MAE y RMSE en PM_{2,5} y O₃. En NO₂ LSTM alcanzó un RMSE apenas menor, pero GCN-LSTM tuvo una desviación estándar inferior, indicando mayor estabilidad temporal. Estos resultados sugieren que la incorporación de información espacial fue útil a medida que aumenta el horizonte de predicción. La dimensión espacial ayudó a capturar patrones más complejos que los únicamente temporales.

En términos de estabilidad, las desviaciones estándar en el horizonte de largo plazo fueron menores para los modelos recurrentes que para el modelo base. Las mejoras obtenidas con GCN-LSTM no se limitaron a un período específico porque el comportamiento se repitió en las distintas particiones temporales.

Tabla 7.9: Resultados validación cruzada a una hora en la estación Escuelas Aguirre ($\mu\text{g}/\text{m}^3$)

Modelo	PM _{2,5}		NO ₂		O ₃	
	MAE	RMSE	MAE	RMSE	MAE	RMSE
MLP	7.88±4.21	9.72±5.59	10.41±1.26	13.20±1.27	13.48±5.89	16.69±6.97
LSTM	4.37±0.86	4.78±1.60	5.92±0.53	8.33±0.77	6.30±0.78	8.50±1.02
GCN-LSTM	5.05±1.47	5.68±1.01	6.33±0.62	8.93±0.75	7.61±2.17	8.52±1.73

Tabla 7.10: Resultados de la validación cruzada a 24 horas en la estación Escuelas Aguirre ($\mu\text{g}/\text{m}^3$)

Modelo	PM _{2,5}		NO ₂		O ₃	
	MAE	RMSE	MAE	RMSE	MAE	RMSE
MLP	6.48±0.74	8.17±1.30	15.43±0.96	19.61±1.27	21.17±4.79	25.24±5.51
LSTM	5.53±1.27	7.13±1.87	13.24±2.09	17.30±2.75	15.03±1.90	18.51±2.09
GCN-LSTM	4.80±1.56	6.98±1.19	13.19±2.05	17.71±1.65	14.80±1.18	17.24±0.88

Capítulo 8

Consideraciones finales

En este capítulo se presentan las conclusiones del proyecto, se analizan las implicancias de los resultados obtenidos y se proponen posibles líneas de trabajo futuro.

8.1. Conclusiones

En este proyecto se desarrolló una metodología para la predicción de concentraciones de contaminantes atmosféricos utilizando arquitecturas de redes neuronales recurrentes y modelos espaciotemporales. La metodología fue evaluada mediante un caso de estudio basado en datos horarios de calidad del aire y variables meteorológicas de estaciones de monitoreo de la ciudad de Madrid, España. El enfoque propuesto permitió incorporar información temporal de cada estación con información espacial entre estaciones de monitoreo. El desempeño de los modelos se evaluó mediante esquemas de evaluación temporal hold-out y validación cruzada temporal. Los resultados obtenidos mostraron que los modelos propuestos lograron mejoras en la predicción de las concentraciones de $\text{PM}_{2,5}$, NO_2 y O_3 respecto a un modelo base de tipo perceptrón multicapa.

Para el caso del horizonte de una hora, LSTM obtuvo menores valores de MAE y RMSE para los tres contaminantes objetivo. En comparación con MLP, las mejoras obtenidas fueron entre 37 % y 57 % dependiendo del contaminante y la métrica. Los resultados indicaron que en horizontes de predicción de corto plazo la información de la propia estación fue suficiente para capturar la dinámica del contaminante. Aunque GCN-LSTM obtuvo resultados cercanos a LSTM, la incorporación de relaciones espaciales no produjo mejoras significativas en este horizonte de predicción. Además, se identificaron algunas dificultades relacionadas con la disponibilidad de datos entre estaciones ya que no todas contaban con registros de los mismos contaminantes y variables meteorológicas. La heterogeneidad limitó la cantidad de información que GCN-LSTM pudo utilizar para modelar las relaciones entre estaciones.

Para el caso del horizonte de 24 horas se obtuvieron resultados diferentes.

GCN-LSTM mostró mejoras en la predicción de $\text{PM}_{2,5}$ y O_3 . En comparación con el modelo base, las reducciones del error fueron entre 30 % y 50 % y respecto a LSTM se obtuvieron mejoras de hasta 10 %. Esta mejora puede atribuirse a que la información espacial entre estaciones tiene mayor influencia cuando aumenta el horizonte de predicción. Con respecto a NO_2 , ambos modelos alcanzaron resultados similares de MAE y RMSE. Este comportamiento sugiere que la información espacial varía según el contaminante objetivo y su aporte es más notorio cuando aumenta el horizonte de predicción.

Además, el uso de validación cruzada permitió analizar la estabilidad del desempeño de los modelos en diferentes particiones del conjunto de datos. En las distintas particiones se obtuvieron resultados similares, lo que evidenció comportamientos consistentes en las métricas de evaluación. Este resultado sugiere que los modelos desarrollados tuvieron capacidad de generalización y no dependieron únicamente de una partición específica de los datos.

Los resultados obtenidos mostraron que la incorporación de información temporal es fundamental para la predicción de la contaminación del aire. Las redes neuronales recurrentes permitieron capturar la dinámica temporal de las concentraciones de los tres contaminantes analizados y redujeron los errores de predicción respecto al modelo base. Asimismo, se evidenció que el aporte de la información espacial depende tanto del horizonte de predicción como del contaminante analizado. En el horizonte de una hora la información de una única estación fue suficiente para modelar el comportamiento de los contaminantes. En cambio, en el horizonte de 24 horas las relaciones espaciales entre estaciones permitieron mejorar las predicciones en algunos casos, particularmente para $\text{PM}_{2,5}$ y O_3 . Estos resultados sugieren que la utilidad del modelado espacio-temporal varía según la dinámica del contaminante y que su aporte es más relevante a medida que aumenta el horizonte de predicción.

8.2. Trabajo futuro

Los resultados obtenidos en este proyecto de grado sirven como referencia para profundizar en el modelado espacio-temporal de la calidad del aire y explorar diferentes arquitecturas que permitan mejorar la precisión de las predicciones.

Respecto al análisis y procesamiento de los datos, una posible extensión es incorporar variables externas adicionales. Aunque en este proyecto se utilizaron variables meteorológicas, se podrían incluir datos respecto al tráfico vehicular, eventos particulares que afecten las concentraciones de contaminantes o información adicional sobre fuentes de emisión. La incorporación de nuevas variables permitiría modelar de manera más completa los factores que influyen en las variaciones de las concentraciones de los contaminantes atmosféricos y podría contribuir a mejorar el desempeño de los modelos.

En cuanto a las arquitecturas de redes neuronales consideradas, es interesante explorar otras variantes de los modelos desarrollados, como incorporar varias capas de convolución sobre grafos para capturar relaciones más complejas entre estaciones de monitoreo. También es interesante explorar modelos basados en

mecanismos de atención que permitirían identificar dinámicamente qué observaciones pasadas o qué estaciones de monitoreo resultan más importante para realizar las predicciones en cada instante.

Finalmente, la metodología desarrollada en este proyecto no depende únicamente de las características particulares de la ciudad y de las estaciones de donde provienen los datos. Por lo tanto, su aplicación en distintos escenarios geográficos permitiría evaluar la capacidad de generalización de los modelos en diferentes contextos y su robustez en otras dinámicas temporales y espaciales. Asimismo, el enfoque podría adaptarse a otros problemas de predicción, como la estimación de la calidad del agua en redes fluviales y la predicción de la demanda energética.

Referencias bibliográficas

- Aditya, C. R., Deshmukh, C. R., Nayana, D. K., y Gandhi, P. V. (2018). Detection and Prediction of Air Pollution using Machine Learning Models. *International Journal of Engineering Trends and Technology (IJETT)*, 59(4), 204-207. <https://doi.org/10.14445/22315381/IJETT-V59P238>
- Aggarwal, C. C. (2018). *Neural Networks and Deep Learning: A Textbook*. Springer. <https://doi.org/10.1007/978-3-319-94463-0>
- Box, G. E. P., Jenkins, G. M., Reinsel, G. C., y Ljung, G. M. (2015). *Time Series Analysis: Forecasting and Control* (5.^a ed.). John Wiley & Sons.
- Cáceres-Tello, J., y Galán-Hernández, J. J. (2024). Analysis and Prediction of PM2.5 Pollution in Madrid: The Use of Prophet–Long Short-Term Memory Hybrid Models. *AppliedMath*, 4(4), 1428-1452. <https://doi.org/10.3390/appliedmath4040076>
- Cordova, C. H., et al. (2021). Air quality assessment and pollution forecasting using artificial neural networks in Metropolitan Lima-Peru [Accessed: 2025-05-17]. *Scientific Reports*, 11(1), 1-19. <https://doi.org/10.1038/s41598-021-03650-9>
- Deepan, S., y Saravanan, M. (2024). Air quality index prediction using seasonal autoregressive integrated moving average transductive long short-term memory. *ETRI Journal*, 46(5), 915-927. <https://doi.org/10.4218/etrij.2023-0283>
- European Environment Agency. (2024, 10 de diciembre). Harm to human health from air pollution: Burden of disease 2024 [Accessed September 15, 2025]. *European Environment Agency*. [%5Curl%7Bhttps://www.eea.europa.eu/en/analysis/publications/harm-to-human-health-from-air-pollution-2024%7D](https://www.eea.europa.eu/en/analysis/publications/harm-to-human-health-from-air-pollution-2024%7D)
- Goodfellow, I., Bengio, Y., y Courville, A. (2016). *Deep Learning*. MIT Press.
- Hamilton, W. L. (2020). *Graph Representation Learning* (Vol. 14). Morgan & Claypool Publishers.
- Health Effects Institute (HEI) and Institute for Health Metrics and Evaluation (IHME). (2024). *State of Global Air 2024* (Technical Report) (Accessed: 2025-09-03). Health Effects Institute, Institute for Health Metrics y Evaluation. <https://www.healthdata.org/sites/default/files/2024-06/soga-2024-report.pdf>

- Iskandaryan, D., Ramos, F., y Trilles, S. (2023). Graph Neural Network for Air Quality Prediction: A Case Study in Madrid. *IEEE Access*, 11, 2729-2742. <https://doi.org/10.1109/ACCESS.2023.3234214>
- Jana, S., Middy, A. I., y Roy, S. (2024). Short-term air pollution prediction using graph convolutional neural networks. *Technological Forecasting and Social Change*, 208, 123684. <https://doi.org/10.1016/j.techfore.2024.123684>
- Joel, M., y Doorsamy, W. (2022). A review of missing data handling techniques for machine learning. *Journal of Big Data*, 9(1), 1-34.
- Kipf, T. N., y Welling, M. (2017). Semi-Supervised Classification with Graph Convolutional Networks. *International Conference on Learning Representations (ICLR)*.
- Kök, İ., Şimşek, M. U., y Özdemir, S. (2017). A deep learning model for air quality prediction in smart cities. *2017 IEEE International Conference on Big Data (Big Data)*, 1983-1990. <https://doi.org/10.1109/BigData.2017.8258144>
- Li, T., Hua, M., y Wu, X. (2020). A Hybrid CNN-LSTM Model for Forecasting Particulate Matter (PM_{2.5}). *IEEE Access*, 8, 26933-26940. <https://doi.org/10.1109/ACCESS.2020.2971348>
- Lipton, Z. C., Kale, D. C., y Wetzel, R. (2016). Directly Modeling Missing Data in Sequences with RNNs: Improved Classification of Clinical Time Series. *Proceedings of the 1st Machine Learning for Healthcare Conference*, 56, 253-270.
- Marinov, E., Petrova-Antonova, D., y Malinov, S. (2022). Time Series Forecasting of Air Quality: A Case Study of Sofia City. *Atmosphere*, 13(5), 788. <https://doi.org/10.3390/atmos13050788>
- Ministerio de Agricultura, Alimentación y Medio Ambiente. (2015). *Perfil Ambiental de España 2014: 2.2. Calidad del aire* (inf. téc.) (Capítulo 2.2 del Perfil Ambiental de España 2014). Ministerio de Agricultura, Alimentación y Medio Ambiente. Madrid. Consultado el 24 de agosto de 2025, desde https://www.miteco.gob.es/content/dam/mitesco/es/calidad-y-evaluacion-ambiental/publicaciones/02_02_tcm30-185130.pdf
- Tlavelo, M., et al. (2021). A review of missing data handling methods in machine learning. *Journal of Big Data*, 8(1), 1-29.
- Toutouh, J., Lebrusán, I., y Nesmachnow, S. (2020). Computational Intelligence for Evaluating the Air Quality in the Center of Madrid, Spain. En *Optimization and Learning* (pp. 115-127). Springer International Publishing. https://doi.org/10.1007/978-3-030-41913-4_10
- World Health Organization. (2021). *WHO global air quality guidelines: particulate matter (PM_{2.5} and PM₁₀), ozone, nitrogen dioxide, sulfur dioxide and carbon monoxide* (inf. téc.) (September 2025). World Health Organization. Geneva.
- Xiao, J., Wang, Q., Cui, J., y Yu, J. (2022). Multi-feature PM_{2.5} Prediction with ARIMA-LSTM. *2022 International Conference on Machine Learning, Cloud Computing and Intelligent Mining (MLCCIM)*, 229-233. <https://doi.org/10.1109/MLCCIM55934.2022.00046>

Apéndice A

Selección de hiperparámetros

Este Apéndice presenta los resultados de todas las configuraciones de búsqueda aleatoria evaluadas para la selección de los hiperparámetros de los modelos predictivos.

A.1. Modelo LSTM

Tabla A.1: Resultados de la búsqueda aleatoria de hiperparámetros para el modelo LSTM en la estación Escuelas Aguirre en el horizonte de una hora.

<i>Dropout</i>	ℓ_2	<i>Tasa de aprendizaje</i>	<i>Unidades LSTM</i>	<i>Pérdida</i>
0.2	$1,17 \times 10^{-6}$	$1,02 \times 10^{-4}$	128	0.0287
0.3	$1,49 \times 10^{-5}$	$1,49 \times 10^{-4}$	32	0.0387
0.0	$1,49 \times 10^{-6}$	$1,43 \times 10^{-4}$	32	0.0180
0.0	$8,18 \times 10^{-6}$	$1,53 \times 10^{-4}$	64	0.0175
0.0	$8,12 \times 10^{-4}$	$1,05 \times 10^{-4}$	32	0.0190
0.2	$5,29 \times 10^{-6}$	$1,70 \times 10^{-4}$	128	0.0313
0.3	$2,62 \times 10^{-6}$	$4,09 \times 10^{-4}$	32	0.0365
0.0	$4,99 \times 10^{-6}$	$9,41 \times 10^{-4}$	32	0.0176
0.0	$7,11 \times 10^{-4}$	$2,37 \times 10^{-4}$	128	0.0185
0.3	$3,25 \times 10^{-6}$	$4,05 \times 10^{-4}$	64	0.0395
0.2	$3,97 \times 10^{-6}$	$6,10 \times 10^{-4}$	64	0.0279
0.3	$2,51 \times 10^{-5}$	$9,62 \times 10^{-4}$	128	0.0379
0.3	$6,17 \times 10^{-5}$	$6,02 \times 10^{-4}$	32	0.0378
0.3	$3,06 \times 10^{-5}$	$1,32 \times 10^{-4}$	64	0.0385
0.2	$5,05 \times 10^{-5}$	$1,61 \times 10^{-4}$	128	0.0284
0.0	$1,01 \times 10^{-6}$	$8,68 \times 10^{-4}$	64	0.0182
0.3	$2,68 \times 10^{-6}$	$2,16 \times 10^{-4}$	128	0.0407
0.3	$8,20 \times 10^{-6}$	$6,43 \times 10^{-4}$	32	0.0391
0.0	$6,71 \times 10^{-4}$	$1,03 \times 10^{-4}$	32	0.0189
0.0	$8,61 \times 10^{-6}$	$4,60 \times 10^{-4}$	128	0.0181

Tabla A.2: Resultados de la búsqueda aleatoria de hiperparámetros para el modelo LSTM en la estación Escuelas Aguirre en el horizonte de 24 horas.

<i>Dropout</i>	ℓ_2	<i>Tasa de aprendizaje</i>	<i>Unidades LSTM</i>	<i>Pérdida</i>
0.2	$1,17 \times 10^{-6}$	$1,02 \times 10^{-4}$	128	0.0414
0.3	$1,49 \times 10^{-5}$	$1,49 \times 10^{-4}$	32	0.0499
0.0	$1,49 \times 10^{-6}$	$1,43 \times 10^{-4}$	64	0.0369
0.0	$8,18 \times 10^{-6}$	$1,53 \times 10^{-4}$	64	0.0371
0.0	$8,12 \times 10^{-4}$	$1,05 \times 10^{-4}$	32	0.0384
0.2	$5,29 \times 10^{-6}$	$1,70 \times 10^{-4}$	128	0.0427
0.3	$2,62 \times 10^{-6}$	$4,09 \times 10^{-4}$	32	0.0475
0.0	$4,99 \times 10^{-6}$	$9,41 \times 10^{-4}$	32	0.0373
0.0	$7,11 \times 10^{-4}$	$2,37 \times 10^{-4}$	128	0.0388
0.3	$3,25 \times 10^{-6}$	$4,05 \times 10^{-4}$	64	0.0505
0.2	$3,97 \times 10^{-6}$	$6,10 \times 10^{-4}$	64	0.0424
0.3	$2,51 \times 10^{-5}$	$9,62 \times 10^{-4}$	128	0.0478
0.3	$6,17 \times 10^{-5}$	$6,02 \times 10^{-4}$	32	0.0467
0.3	$3,06 \times 10^{-5}$	$1,32 \times 10^{-4}$	64	0.0506
0.2	$5,05 \times 10^{-5}$	$1,61 \times 10^{-4}$	128	0.0441
0.0	$1,01 \times 10^{-6}$	$8,68 \times 10^{-4}$	64	0.0370
0.3	$2,68 \times 10^{-6}$	$2,16 \times 10^{-4}$	128	0.0484
0.3	$8,20 \times 10^{-6}$	$6,43 \times 10^{-4}$	32	0.0474
0.0	$6,71 \times 10^{-4}$	$1,03 \times 10^{-4}$	32	0.0388
0.0	$8,61 \times 10^{-6}$	$4,60 \times 10^{-4}$	128	0.0378

A.2. Modelo GCN–LSTM

Tabla A.3: Resultados de la búsqueda aleatoria de hiperparámetros para el modelo GCN–LSTM en el horizonte de una hora.

<i>Dropout</i>	ℓ_2	<i>Tasa de aprendizaje</i>	<i>Unid. GCN</i>	<i>Unid. LSTM</i>	<i>Pérdida</i>
0.0	$8,20 \times 10^{-6}$	$6,43 \times 10^{-4}$	64	64	0.0169
0.0	$5,14 \times 10^{-6}$	$5,92 \times 10^{-4}$	64	64	0.0173
0.2	$3,27 \times 10^{-6}$	$4,80 \times 10^{-4}$	128	64	0.0218
0.3	$2,51 \times 10^{-5}$	$9,62 \times 10^{-4}$	128	128	0.0249
0.0	$2,91 \times 10^{-6}$	$3,55 \times 10^{-4}$	32	64	0.0196
0.2	$9,44 \times 10^{-6}$	$6,01 \times 10^{-4}$	64	128	0.0187
0.3	$6,17 \times 10^{-5}$	$6,02 \times 10^{-4}$	128	64	0.0228
0.0	$6,87 \times 10^{-6}$	$2,44 \times 10^{-4}$	64	128	0.0204
0.2	$1,95 \times 10^{-6}$	$4,13 \times 10^{-4}$	32	32	0.0226
0.3	$3,06 \times 10^{-5}$	$1,32 \times 10^{-4}$	32	128	0.0261
0.0	$7,01 \times 10^{-6}$	$5,10 \times 10^{-4}$	128	128	0.0175
0.2	$3,97 \times 10^{-6}$	$6,10 \times 10^{-4}$	64	64	0.0191
0.3	$2,62 \times 10^{-6}$	$4,09 \times 10^{-4}$	32	64	0.0237
0.0	$1,49 \times 10^{-6}$	$1,43 \times 10^{-4}$	64	64	0.0190
0.2	$5,29 \times 10^{-6}$	$1,70 \times 10^{-4}$	128	128	0.0212
0.3	$1,49 \times 10^{-5}$	$1,49 \times 10^{-4}$	64	32	0.0240
0.0	$1,01 \times 10^{-6}$	$8,68 \times 10^{-4}$	64	64	0.0182
0.2	$5,05 \times 10^{-5}$	$1,61 \times 10^{-4}$	128	128	0.0221
0.3	$8,20 \times 10^{-6}$	$6,43 \times 10^{-4}$	32	64	0.0219
0.0	$6,71 \times 10^{-4}$	$1,03 \times 10^{-4}$	64	64	0.0278

Tabla A.4: Resultados de la búsqueda aleatoria de hiperparámetros para el modelo GCN-LSTM en el horizonte de 24 horas.

<i>Dropout</i>	ℓ_2	<i>Tasa de aprendizaje</i>	<i>Unid. GCN</i>	<i>Unid. LSTM</i>	<i>Pérdida</i>
0.0	$1,37 \times 10^{-6}$	$1,57 \times 10^{-4}$	64	64	0.0348
0.0	$8,20 \times 10^{-6}$	$2,04 \times 10^{-4}$	64	64	0.0352
0.2	$3,11 \times 10^{-6}$	$3,89 \times 10^{-4}$	64	64	0.0387
0.3	$2,45 \times 10^{-6}$	$4,61 \times 10^{-4}$	64	64	0.0406
0.0	$4,92 \times 10^{-6}$	$4,76 \times 10^{-4}$	32	64	0.0379
0.2	$6,45 \times 10^{-6}$	$5,88 \times 10^{-4}$	128	64	0.0391
0.3	$1,05 \times 10^{-5}$	$6,72 \times 10^{-4}$	128	128	0.0435
0.0	$2,71 \times 10^{-6}$	$2,98 \times 10^{-4}$	32	128	0.0395
0.2	$9,81 \times 10^{-6}$	$1,92 \times 10^{-4}$	128	64	0.0374
0.3	$7,06 \times 10^{-6}$	$1,79 \times 10^{-4}$	64	128	0.0410
0.0	$5,32 \times 10^{-6}$	$3,27 \times 10^{-4}$	128	128	0.0368
0.2	$2,03 \times 10^{-6}$	$4,13 \times 10^{-4}$	32	32	0.0412
0.3	$4,66 \times 10^{-6}$	$3,55 \times 10^{-4}$	32	64	0.0420
0.0	$7,14 \times 10^{-6}$	$1,85 \times 10^{-4}$	64	128	0.0359
0.2	$1,92 \times 10^{-6}$	$2,44 \times 10^{-4}$	128	128	0.0365
0.3	$9,44 \times 10^{-6}$	$2,16 \times 10^{-4}$	128	64	0.0415
0.0	$1,01 \times 10^{-6}$	$8,68 \times 10^{-4}$	64	64	0.0392
0.2	$5,05 \times 10^{-5}$	$1,61 \times 10^{-4}$	128	128	0.0446
0.3	$3,06 \times 10^{-5}$	$1,32 \times 10^{-4}$	32	128	0.0461
0.0	$6,71 \times 10^{-4}$	$1,03 \times 10^{-4}$	64	64	0.0488

Apéndice B

Resultados gráficos

Este Apéndice presenta gráficos adicionales de la predicción en horizontes de una y 24 horas para los distintos modelos en una estación urbana de fondo y una suburbana.

B.1. Estaciones suburbanas

B.1.1. LSTM

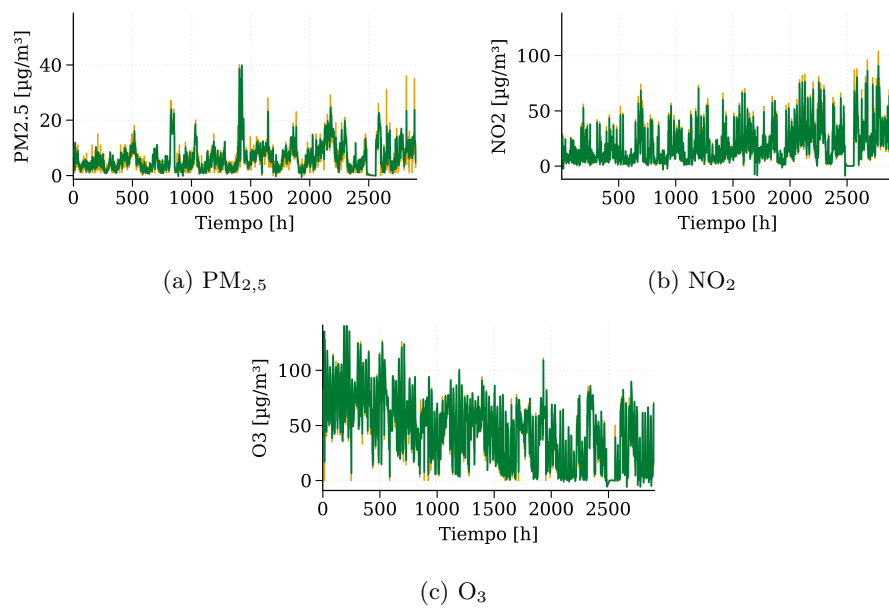
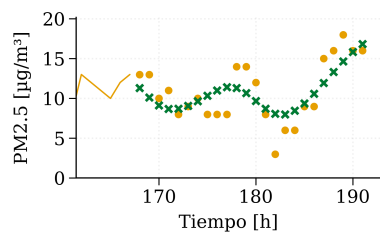
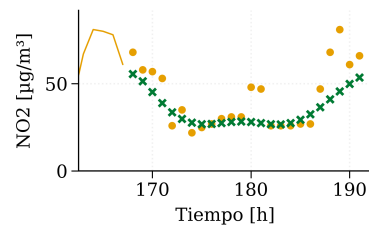


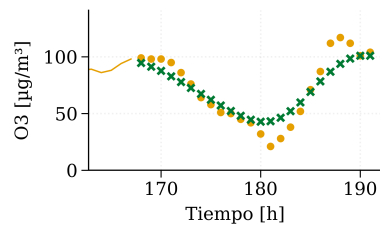
Figura B.1: Predicciones a una hora del modelo LSTM en la estación suburbana Casa de Campo
 (— : concentraciones reales, — : concentraciones predichas)



(a) $PM_{2,5}$



(b) NO_2



(c) O_3

Figura B.2: Predicciones a 24 horas del modelo LSTM en la estación suburbana Casa de Campo
 (— : concentraciones reales, — : concentraciones predichas)

B.1.2. GCN-LSTM

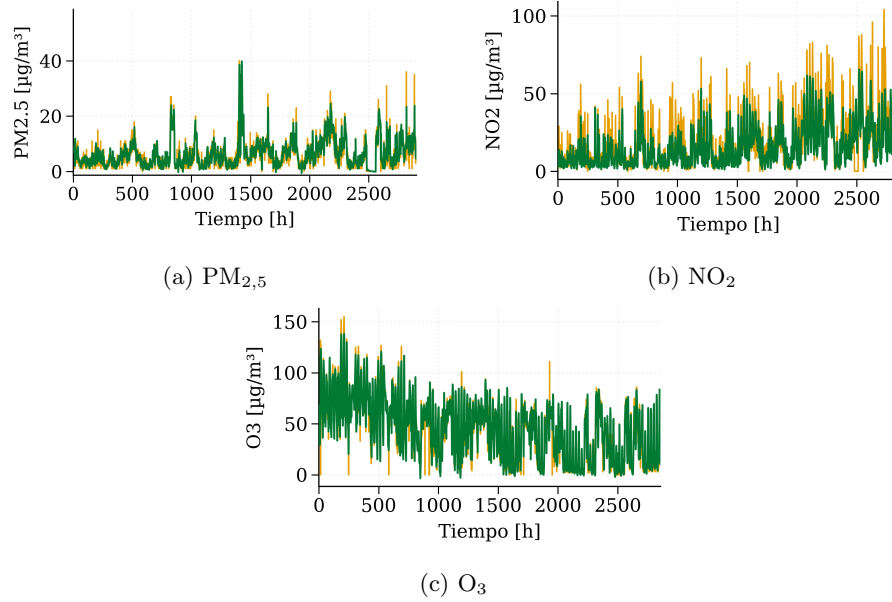


Figura B.3: Predicciones a una hora del modelo GCN-LSTM en la estación suburbana Casa de Campo
(— : concentraciones reales, — : concentraciones predichas)

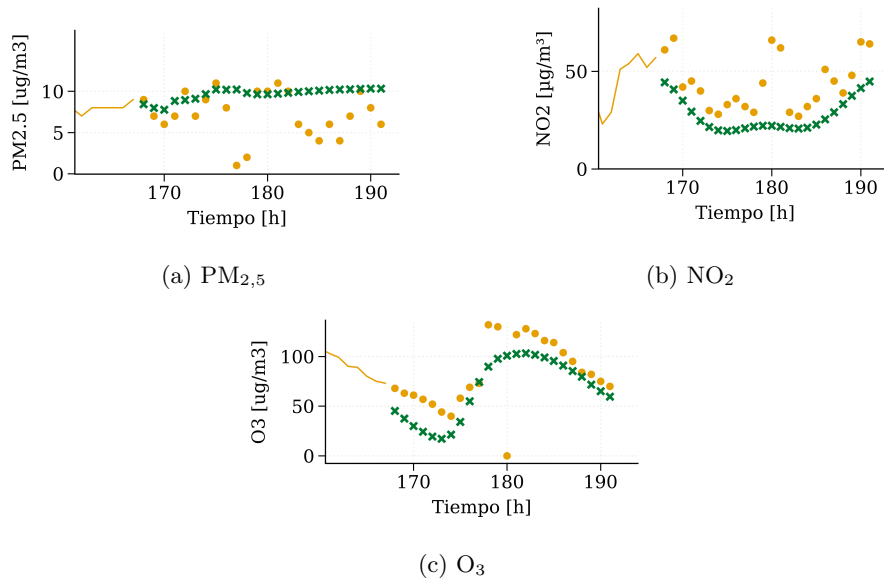


Figura B.4: Predicciones a 24 horas del modelo GCN-LSTM en la estación suburbana Casa de Campo
 (— : concentraciones reales, — : concentraciones predichas)

B.2. Estaciones urbanas fondo

B.2.1. LSTM

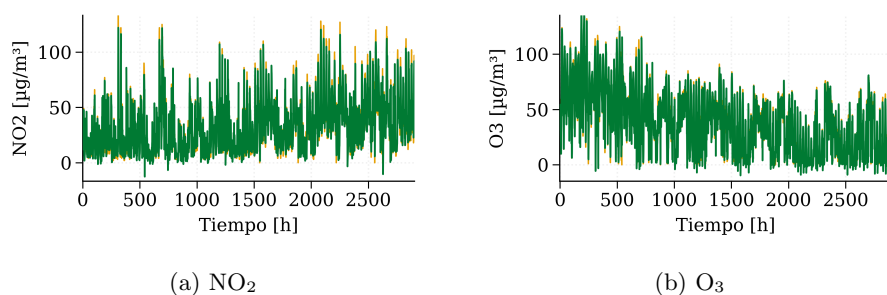


Figura B.5: Predicciones a una hora del modelo LSTM en la estación urbana fondo Ensanche de Vallecas
(— : concentraciones reales, — : concentraciones predichas)

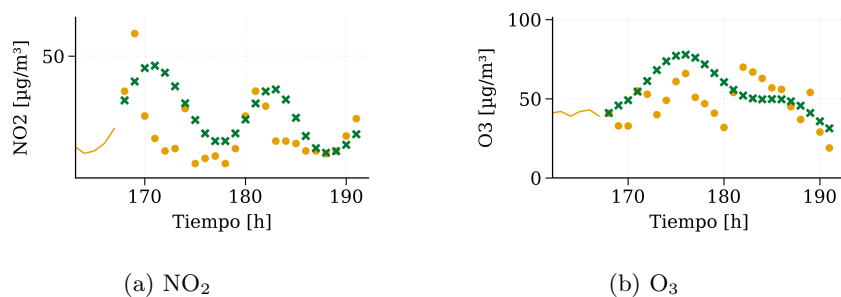
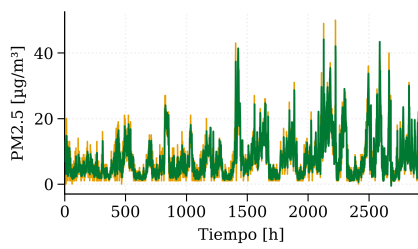
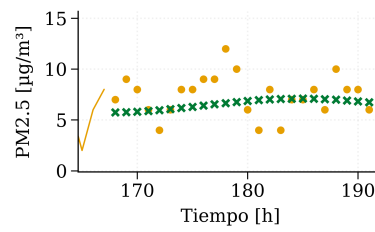


Figura B.6: Predicciones a 24 horas del modelo LSTM en la estación urbana fondo Ensanche de Vallecas
(— : concentraciones reales, — : concentraciones predichas)



(a) $PM_{2,5}$ - 1 hora



(b) $PM_{2,5}$ - 24 horas

Figura B.7: Predicciones a una y 24 horas del modelo LSTM en la estación urbana fondo Sanchinarro

(— : concentraciones reales, — : concentraciones predichas)

B.2.2. GCN-LSTM

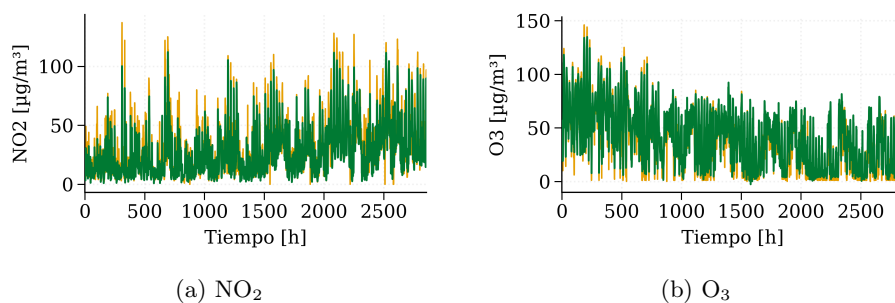


Figura B.8: Predicciones a una hora del modelo GCN-LSTM en la estación urbana fondo Ensanche de Vallecas (— : concentraciones reales, — : concentraciones predichas)

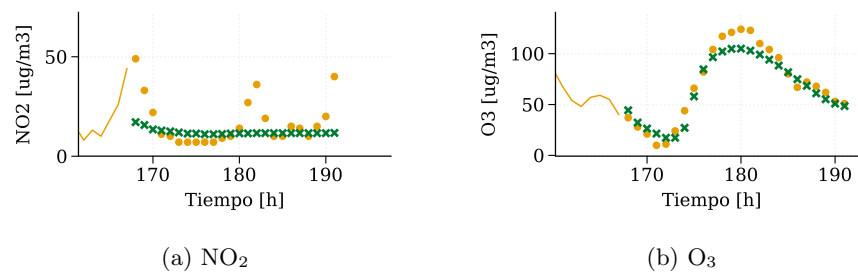
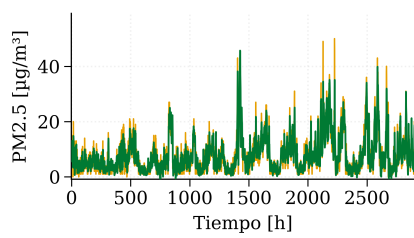
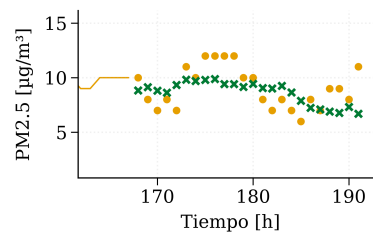


Figura B.9: Predicciones a 24 horas del modelo GCN-LSTM en la estación urbana fondo Ensanche de Vallecas (— : concentraciones reales, — : concentraciones predichas)



(a) $PM_{2,5}$ - 1 hora



(b) $PM_{2,5}$ - 24 horas

Figura B.10: Predicciones a una y 24 horas del modelo GCN-LSTM en la estación urbana fondo Sanchinarro

(— : concentraciones reales, — : concentraciones predichas)

Apéndice C

MAE y RMSE por estación

Este Apéndice presenta los valores de MAE y RMSE obtenidos por los modelos LSTM y GCN-LSTM para todas las estaciones de monitoreo, en los horizontes de predicción de una y 24 horas.

C.1. PM_{2,5}

Tabla C.1: MAE y RMSE para PM_{2,5} en todas las estaciones para horizontes de una y 24 horas ($\mu\text{g}/\text{m}^3$).

Estación	1 h				24 h			
	LSTM		GCN-LSTM		LSTM		GCN-LSTM	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
8	2.81	3.62	3.36	4.09	5.31	6.50	4.77	6.05
24	1.72	2.33	2.38	2.73	3.01	3.88	3.46	4.78
38	2.17	2.89	2.63	3.01	4.52	5.65	4.12	5.68
47	2.17	2.80	2.74	3.78	3.82	4.73	3.84	5.19
48	3.79	6.23	4.19	6.68	5.70	8.62	5.21	7.82
50	3.46	4.79	3.98	5.61	7.17	8.09	6.82	7.36
56	3.02	4.22	3.26	4.77	4.79	6.15	4.76	6.12
57	2.12	2.84	2.93	3.68	4.47	5.44	4.35	5.07

C.2. NO₂

Tabla C.2: MAE y RMSE para NO₂ en todas las estaciones para horizontes de una y 24 horas ($\mu\text{g}/\text{m}^3$).

Estación	1 h				24 h			
	LSTM		GCN-LSTM		LSTM		GCN-LSTM	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
4	5.39	7.60	7.26	8.51	13.91	16.62	12.46	15.27
8	5.47	7.80	6.26	8.92	11.89	15.60	10.94	15.75
11	6.77	9.51	7.91	10.11	16.02	19.25	14.69	17.39
16	4.32	6.11	5.23	7.24	9.78	12.69	9.58	12.28
17	5.90	7.71	6.37	8.23	16.24	20.91	17.00	21.00
18	5.30	7.57	6.16	8.47	12.04	16.53	11.63	15.36
24	4.36	6.33	5.71	7.85	10.21	12.10	9.98	11.52
27	7.54	9.29	8.45	10.55	13.83	17.31	12.93	16.85
35	5.88	8.07	6.38	8.56	11.99	15.13	12.51	15.65
36	5.72	7.90	6.25	8.70	14.17	16.67	13.42	15.21
38	6.07	8.67	6.72	9.73	14.25	18.73	13.47	17.58
39	5.82	8.46	5.96	8.97	12.01	15.26	11.80	15.60
40	6.19	8.69	7.58	9.02	16.06	18.06	13.52	17.88
47	5.30	7.70	6.29	8.68	12.57	15.39	13.05	16.79
48	5.05	6.97	6.03	7.94	12.14	16.37	11.89	15.26
49	5.96	7.47	6.24	8.65	10.27	13.03	9.55	11.95
50	5.15	7.05	7.83	8.12	13.41	15.44	11.87	14.89
54	7.69	9.34	8.73	10.58	15.87	19.91	14.02	17.99
55	7.88	9.56	9.51	10.84	16.77	18.95	13.69	17.94
56	6.09	8.48	7.24	9.98	15.30	18.98	14.41	18.10
57	5.52	7.95	6.17	9.14	12.89	16.72	13.09	16.49
58	3.08	4.52	4.66	5.85	6.72	8.07	5.77	7.52
59	6.15	8.78	7.68	10.95	12.58	15.17	13.48	14.95
60	5.17	7.26	5.68	8.18	11.87	15.28	11.86	16.50

C.3. O₃

Tabla C.3: MAE y RMSE para O₃ en todas las estaciones para horizontes de una y 24 horas ($\mu\text{g}/\text{m}^3$).

Estación	1 h				24 h			
	LSTM		GCN-LSTM		LSTM		GCN-LSTM	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
8	4.94	6.80	6.05	8.16	12.38	15.21	11.58	13.99
16	5.57	7.72	6.57	8.75	12.81	15.76	11.98	14.45
17	5.22	7.95	6.41	8.01	16.10	19.51	15.94	19.23
18	5.50	8.21	5.91	9.09	13.97	17.51	12.89	16.07
24	5.33	7.93	6.16	8.36	15.91	19.40	15.51	18.85
27	6.47	9.05	7.40	9.86	15.77	19.08	14.92	17.83
35	5.67	7.90	6.40	8.17	14.97	18.05	14.05	17.13
39	5.96	8.36	6.73	9.38	16.10	18.36	15.30	17.80
49	5.87	8.13	6.95	9.25	15.89	19.12	15.81	19.06
54	5.92	8.44	6.44	9.25	15.90	18.67	14.73	16.97
58	5.32	7.87	5.92	8.56	14.85	17.66	13.19	16.35
59	5.87	8.07	6.67	9.11	15.28	18.47	13.90	17.17
60	6.00	8.69	7.03	9.43	15.80	19.11	15.49	19.58

