

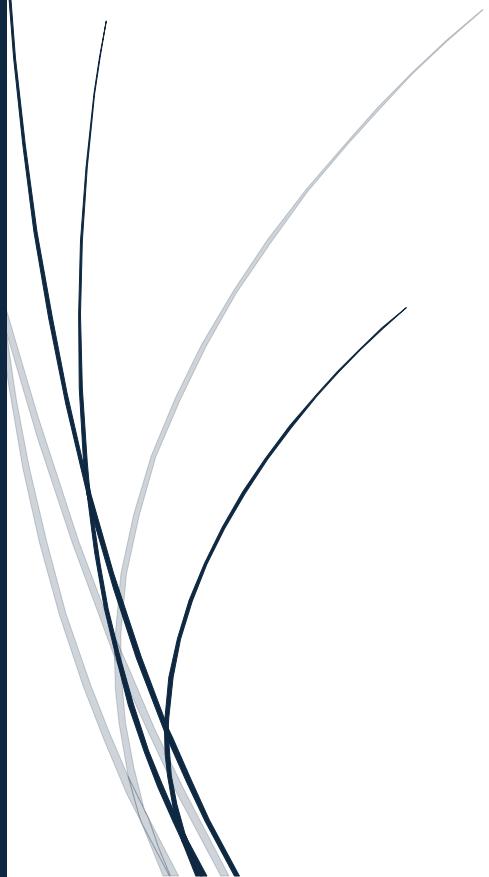


7-6-2024

Proyecto Deep Learning

Entrega 2

Profesor: Raúl Ramos P



YESID ORLANDO RAMIREZ CASTANO
ESTUDIANTE DE POSGRADO , MAESTRIA EN INGENIERIA

Universidad de Antioquia
Medellin-Antioquia
2024

Set de datos iniciales:

En la propuesta inicial del proyecto, se planteó usar los datos encontrados en una competencia de Kaggle (Kaggle, 2024), relacionados con el pronóstico de temperatura superficial a 2 metros. Estos datos son obtenidos por el instituto de investigación y servicio operativo europeo, ECMWF (European Centre for Medium-Range Weather Forecasts) de 500 estaciones meteorológicas de Alemania para un periodo de 6 años. Como objetivo principal de la competencia, Kaggle solicita realizar una corrección de los datos dados por el modelo meteorológico ECMWF (modelo basado en procesos físicos y dinámicos de la atmósfera) (ECMWF, 2024) a partir de los datos en tierra de estas 500 estaciones.

En primera instancia y con el objetivo de adaptarlo al presente proyecto, los resultados dados en la competencia reflejaban una implementación sencilla en costo computacional y en suficiencia de datos, razón que hace viable el uso de dicho set de datos.

Para cumplir con los objetivos del proyecto, se plantea entrenar con estos datos un modelo de redes neuronales recurrentes, ya que comúnmente para la implementación de este tipo de red se alimentan con variables en el tiempo (series de tiempo).

Hasta este punto, el proyecto reflejaba que iba por buen camino, se realiza la exploración inicial de los datos, donde se encuentran que el set de datos esta dispuesto inicialmente para un periodo de 6 años (Gráfico 1) donde las muestras son diarias, por cada una de las estaciones, sumando así un set de datos de aproximadamente un millón de datos de solo temperatura, adicionalmente el set de datos tenía mas variables atmosféricas que también podrían ser útiles según el enfoque que se requiera.

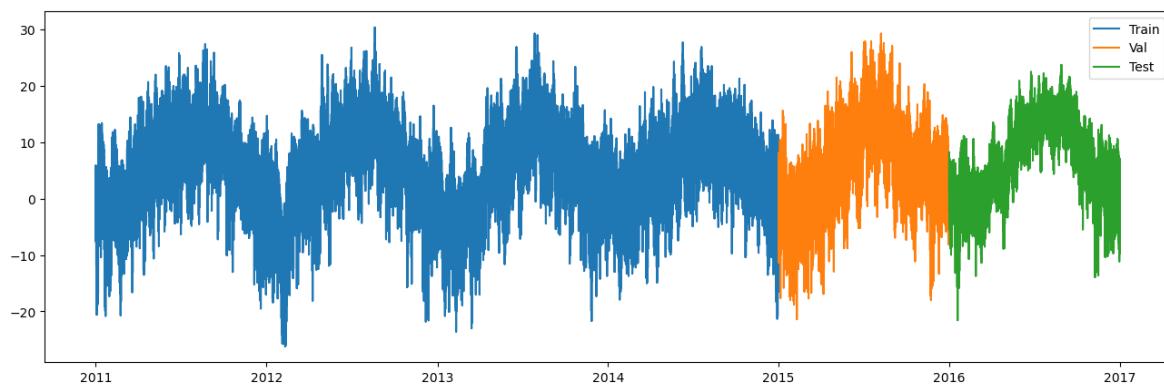


Gráfico 1. Exploración set de datos de 500 Estaciones meteorológicas en Alemania.

Al iniciar el preprocessamiento de datos (Entregable ‘01_Exploración_de_datos.opynb’) para alimentar el modelo, surgieron varios inconvenientes. Inicialmente la presencia de datos faltantes, que se solucionó eliminando los registros incompletos, lo que redujo ligeramente el conjunto de datos en cuanto a estaciones. Luego, se procedió a preprocessar subconjuntos de series temporales para alimentar el modelo y realizar el pronóstico. Este fue el punto crítico

que llevó a cambiar el set de datos, ya que el modelo requiere una considerable cantidad de series temporales consecutivas y del mismo tamaño.

El conjunto de datos inicial consistía en registros diarios de 500 estaciones meteorológicas de un periodo de 6 años, formando una matriz de 500 valores por día. Por lo que la primera alternativa para disponer los datos consecutivamente para alimentar el modelo, eran subconjuntos de matrices consecutivos, lo que resultaba en una reducción significativa del conjunto de datos. También se consideró la opción de promediar estos valores, pero esto no tenía sentido desde el punto de vista meteorológico por la ubicación física de las estaciones, e igualmente se daría la reducción de datos, finalmente otra opción era analizar los datos de una sola estación, pero esto también implicaba un conjunto de datos reducido, (2000 registros), insuficientes para calibrar el modelo adecuadamente.

Debido a estos problemas, se decidió usar datos de una estación meteorológica del sistema de alerta temprana del Valle de Aburrá (SIATA) de un periodo de 9 años, con el mismo objetivo de realizar una predicción de la temperatura con una arquitectura de RNN.

Set de datos SIATA :

Con el objetivo de no desviar el objetivo principal del proyecto, e implementar el mismo modelo de RNN, se elige un set de datos adquirido de la página principal de SIATA (SIATA, 2024) consistió en datos de temperatura del Valle de Aburrá de una de las estaciones centrales geográficamente hablando, en este caso la estación 202 AMVA (Área Metropolitana del Valle de Aburrá). Bajo el criterio de selección de menor cantidad de datos ausentes o mal procesados, ya que éstos cuentan con un índice de calidad, adicionalmente que cumplieran con la suficiencia de datos y consecución requeridos para este tipo de arquitecturas.

Los datos estaban dispuestos cada minuto, por lo que según el periodo que eligiera, podría convertirse en un problema de procesamiento y costo computacional, convirtiendo el proyecto en algo poco práctico para su ejecución. Por lo anterior, a los datos se les realiza un preprocesamiento de limpieza de datos e interpolación, finalmente un remuestreo promediando los datos por hora, reduciendo los datos considerablemente y permitiendo así un periodo de evaluación mayor. Obteniendo 87648 datos para un periodo de 10 años (2012-2022), esto se puede ver en mayor detalle en el entregable ‘01_Exploración_datos_Siata.ipynb’.

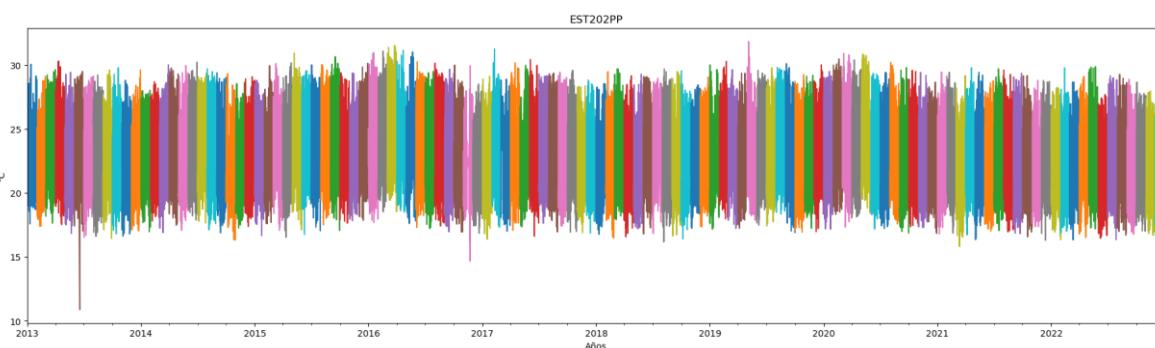


Gráfico 2. Datos SIATA de temperatura 2012-2022

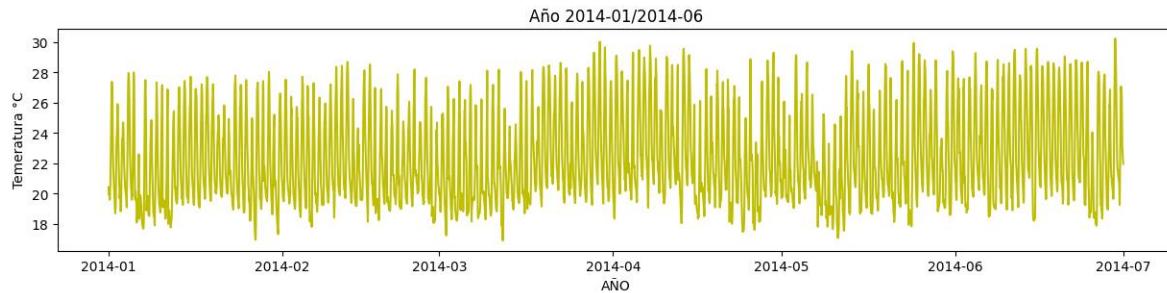


Gráfico 3. Sección de datos SIATA

Al estar en una zona tropical , Medellín no cuenta con estaciones como en otros países más alejados de la Línea del Ecuador, por lo que su ciclo diario oscila en promedio entre 20°C a 30°C aproximadamente como se puede apreciar en el Gráfico 3. En el código anexo '01_Exploración_datos_siata.ipynb' entregados con este proyecto se realizan procesos adicionales donde se muestra valores promedios, desviación estándar, cuartiles, máximos y mínimos , gráficos adicionales de promedio anual , autocorrelación e histograma.

Descripción notebooks y archivos entregados

Para el proyecto se entregan los siguientes archivos.

Descargador.py: En este archivo hay un pequeño código que facilitó la descarga de datos de la página de SIATA, debido a que eran alrededor de 120 archivos csv por cada mes.

links.txt: Debido al cambio del set de datos explicado previamente, los archivos dispuestos en SIATA están dados por mes, y en este caso para el periodo en cuestión , implicó la descarga de 120 archivos, esto hacerlo manualmente hubiera sido dispendioso, por lo que se optó una automatización con el archivo Descargador.py que recibe este archivo como argumento, donde estan plasmados los 120 links de los archivos csv de los registros de temperatura de dicho periodo.

TemperaturaEst202_2013_2022.csv: Este archivo es el producto final posterior para descargar los archivos de SIATA y concatenarlos todos en uno solo.

01_Exploración_de_datos.ipynb: En este notebook se realiza la exploración inicial de los datos descargados de kaggle de las 500 estaciones alemanas. Allí se abren los archivos, se exploran sus variables, el tipo de datos, se miran las principales métricas de distribución de datos, se realizan gráficos para observar la distribución, se realizan histogramas, un gráfico de autocorrelación se hace una primera etapa de limpieza de datos para proceder con gráficos de promedios anuales para cada una de las variables dispuestas en el set de datos. En estos primeros gráficos se evidencia el ciclo anual y estacional dados en estas latitudes. Finalmente se constata una frecuencia constante en el muestreo de los datos , para observar que se cumpliera con la secuencialidad de estos.

01_Exploración_datos_Siata.ipynb: Como su nombre lo indica, el archivo esta destinado en la manipulación inicial de los datos descargados de la página de SIATA. Se leen los archivos y se pasan por una función que reemplaza datos faltantes, valores por defecto y valores en cero, por ‘np.nan’ y se aplica un proceso de interpolación, se aclara que el porcentaje de datos defectuosos es pequeño y se opta por esta alternativa para poder tener los datos dispuestos como se requieren para alimentar el modelo posteriormente. Se concatenan todos los archivos y generar uno solo (TemperaturaEst202_2013_2022.csv) y trabajar desde este, facilitando la manipulación de los datos. Se realiza un gráfico exploratorio inicial (Gráfico 32)

02_Preprocesado_y_Aquitectura.ipynb: en este notebook se crea una función para separar los datos en entrenamiento, validación y prueba, destinando respectivamente 60% , 20% y 20% por defecto. Se crea otra función para organizar y disponer los datos según se deseé entrenar el modelo , este permite crear series de las horas que se deseen y tantos momentos en el tiempo a predecir como se sea conveniente, retornando los arreglos X y Y respectivamente. Se procede con el preprocesado y se crean los datasets de entrenamiento, validación y prueba. Como se requiere escalar los datos , se crea una función para tal fin, de modo que reciba todos los datos de entrada desde un diccionario que se crea con todos los datasets previamente preprocesados y retorna un diccionario con los datasets escalados entre 0 y 1 para este caso ya que los datos no bajan de cero en estas latitudes.

03-Arquitectura_variacion2.ipynb: Este archivo cuenta con una estructura similar al anterior con la variación particular de que se hace una predicción de 8 horas hacia adelante, incluyéndose así un gráfico adicional para ver el error del modelo en la predicción

Descripción de la solución e iteraciones.

Para este proyecto por la disposición y características de los datos, se emplean varias arquitecturas de redes neuronales recurrentes; SimpleRNN, GRU y LSTM basados en la literatura, donde de las más usadas es la LSTM (Fan et al., 2013; Freeman et al., 2018) debido a su capacidad de memoria a largo plazo, por lo anterior se decide realizar varios ensayos con variaciones entre ellas y de los hiperparámetros.

La arquitectura se define con 80 neuronas, 80 épocas y un tamaño de lote de 256, inicialmente , al ejecutar el modelo se presenta overfitting , lo que implicó reducir las iteraciones y el número de neuronas, por lo que se define para las 3 redes mencionadas usar 25 neuronas , con 30 épocas para cada una , tambien se hacen ensayos con diferentes optimizadores.

Resultados

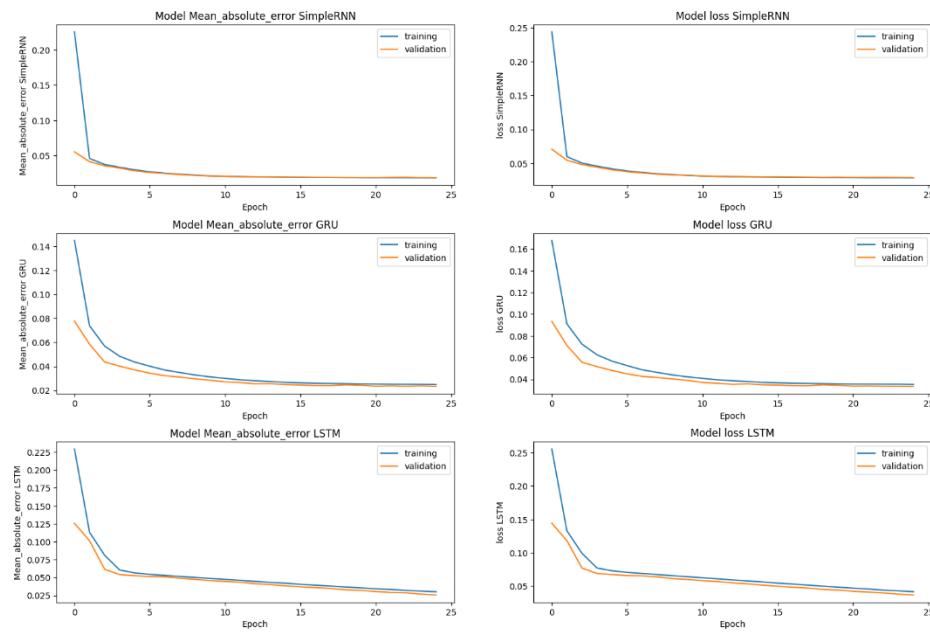


Gráfico 4. MAE y loss para las 25 épocas y los 3 modelos

En el Gráfico 4 se observa que el modelo más acertado o con menor error es el de la SimpleRNN pero las otras otras no se alejan mucho

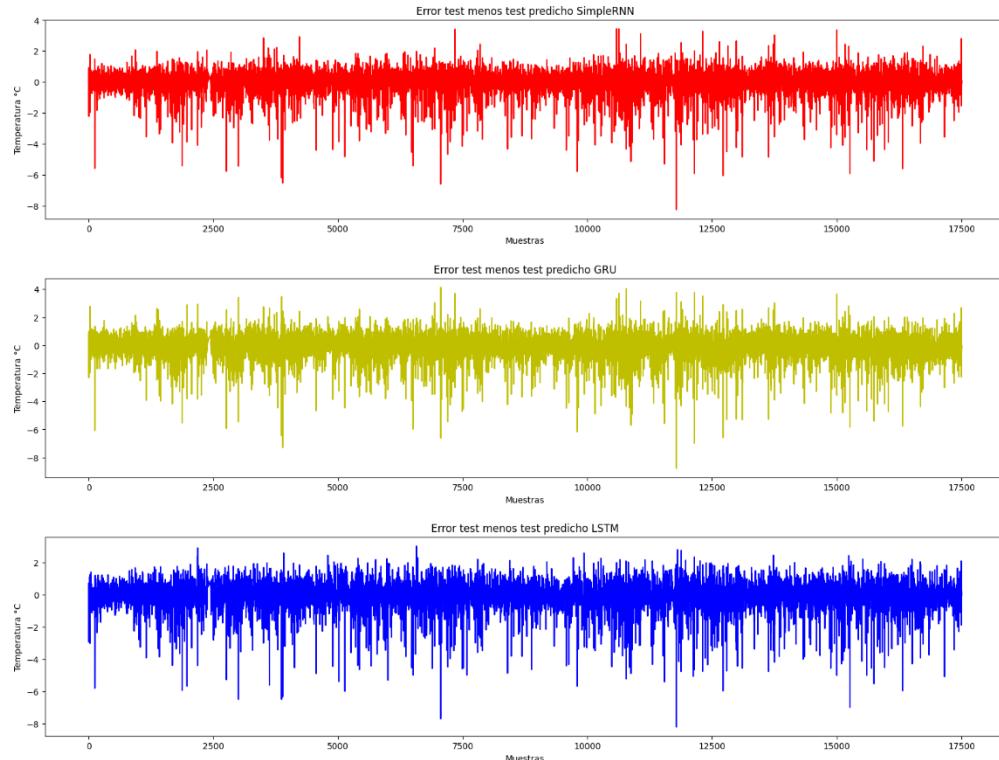


Gráfico 5 Errores de las 3 redes

Para este gráfico se puede observar que para todas las redes los errores oscilan respecto a cero, concentrándose entre 2 y -2 pero con mayor variabilidad aparente en la red LSTM. Lo que sugiere que puede requerir más épocas para ser entrenada o variaciones en los hiperparámetros para que capture la variación de los datos

Para esta parte del entregable el modelo funciona como tipo regresión lineal , debido a que solo está prediciendo un momento del tiempo hacia adelante, haciendo que no haya un aprovechamiento real de las capacidades de las redes, y por tanto reflejan comportamientos similares en las predicciones y calibración de estas.

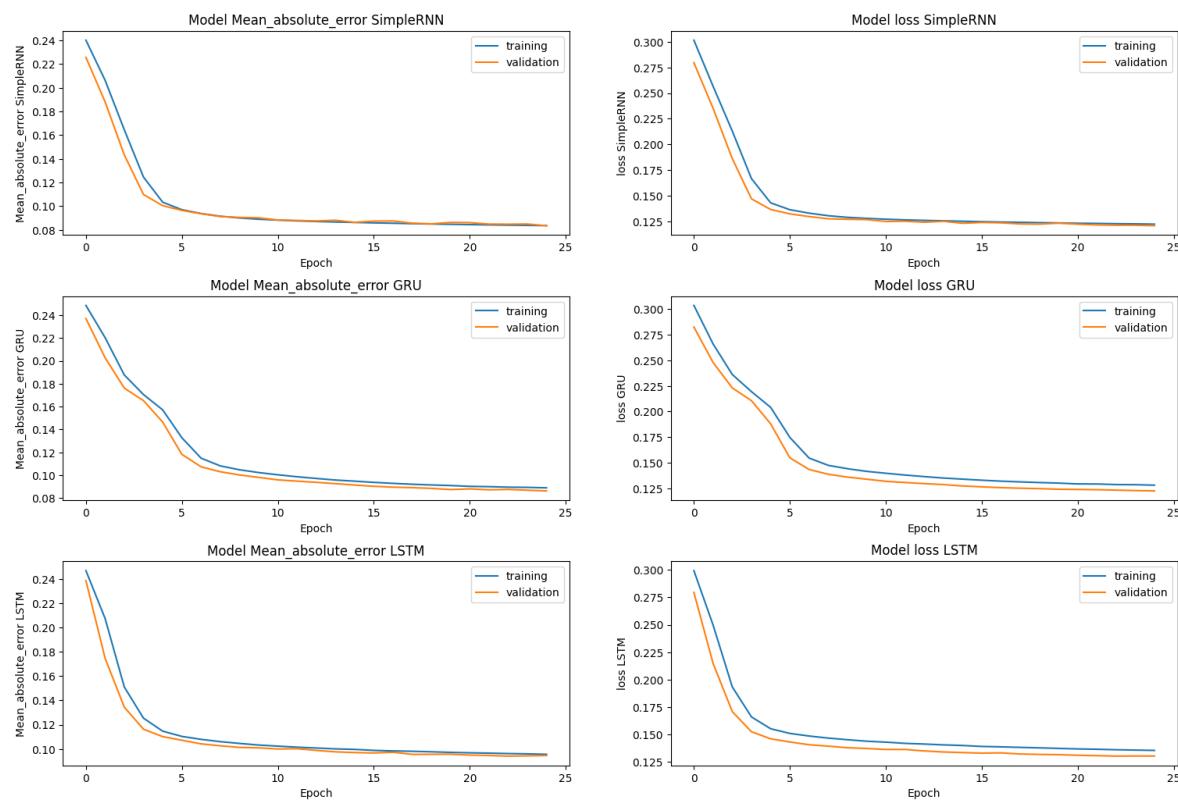


Gráfico 6. Error de las 3 redes para 8 horas hacia delante de forma general

En este caso la red se entrena con set de datos de 48 momentos de tiempo anterior o lookback y se predice 8 momentos hacia delante , de forma general y tambien soportados en los resultados numéricos, sugiere igualmente un mejor comportamiento por parte de la SimpleRNN. Pero con el Gráfico 7 se observa que la red LSMT tiende a estabilizarse un poco más rápido que las otras dos, sugiriendo de esta forma que ante la modificación de algunos hiperparámetros se puede ajustar mejor éste modelo, como lo sugiere la literatura y su configuración de memoria para series temporales.

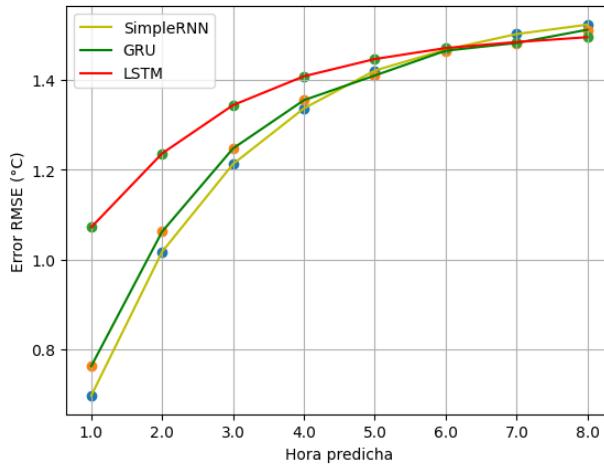


Gráfico 7. Errores de cada momento de manera individual para las 8 horas predichas

Referencias y resultados previos

Bibliografía

- ECMWF. (03 de 04 de 2024). Obtenido de <https://www.ecmwf.int/en/about>
- Kaggle. (05 de 04 de 2024). *Weather Postprocessing*. Obtenido de <https://www.kaggle.com/competitions/weather-postprocessing/overview>
- SIATA. (01 de 05 de 2024). SIATA. Obtenido de SIATA: https://siata.gov.co/descarga_siata/index.php/index2/estaciones/

Use the "Insert Citation" button to add citations to this document.

- Fan, J., Li, Q., Hou, J., Feng, X., Karimian, H., & Lin, S. (2013). A spatiotemporal prediction framework for air pollution based on deep RNN. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 4(4W2), 15–22. <https://doi.org/10.5194/isprs-annals-IV-4-W2-15-2017>
- Freeman, B. S., Taylor, G., Gharabaghi, B., & Thé, J. (2018). Forecasting air quality time series using deep learning. *Journal of the Air and Waste Management Association*, 68(8), 866–886. <https://doi.org/10.1080/10962247.2018.1459956>