UNIVERSIDAD DE MÁLAGA
Dpto. Lenguajes y Ciencias de la Computación

Desarrollo de Software Crítico

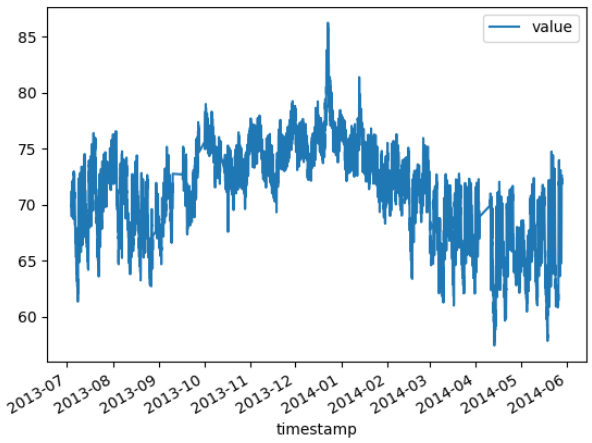
Detección de Anomalías con Machine Learning

Disponemos de un fichero con datos de temperaturas registrados en un dispositivo industrial (datos.csv). Vamos a utilizar diferentes técnicas de Machine Learning para identificar posibles anomalías en este fichero.

Se muestran a continuación ejemplos de mediciones del fichero y una gráfica con todas las mediciones:

```
timestamp,value
2013-07-04 0:00:00,69.88083514
2013-07-04 1:00:00,71.22022706
2013-07-04 2:00:00,70.87780496
2013-07-04 3:00:00,68.95939994
```

...



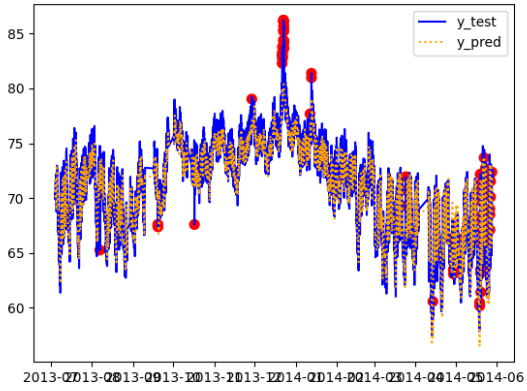
La práctica va a constar de las 3 partes que se mencionan a continuación y el lenguaje de programación a utilizar será Python.

1. Construcción de modelo LSTM y detección de anomalías (5 puntos)
    a. Construcción de modelo básico LSTM (2 puntos). En este apartado se construirá un modelo con redes LSTM. La red tomará como entrada secuencias de datos (ventanas) del fichero con un tamaño determinado, por ejemplo [69.88, 71.22, 70.87] y devolverá una predicción del siguiente valor de la secuencia (p.ej. [68.86]).

b. Detección de anomalías (2 puntos). Una vez está el modelo entrenado, si le pasamos una nueva ventana, por ejemplo [71.88, 70.22, 72.87], nos devolverá la predicción del siguiente valor de la secuencia (p.ej. [72.3[).

   Vamos a aprovechar esto para detectar anomalías en el fichero. Habrá que determinar un criterio para identificar las anomalías y aplicarlo al fichero de datos para que se muestren las mismas, tanto en forma textual como gráfica. El resultado debe ser similar al siguiente:

   El número de anomalias es _ sobre (7257,)
   2013-08-06T20:00:00.000000000
   2013-08-19T22:00:00.000000000
   2013-10-16T22:00:00.000000000
   2013-12-21T20:00:00.000000000
   …

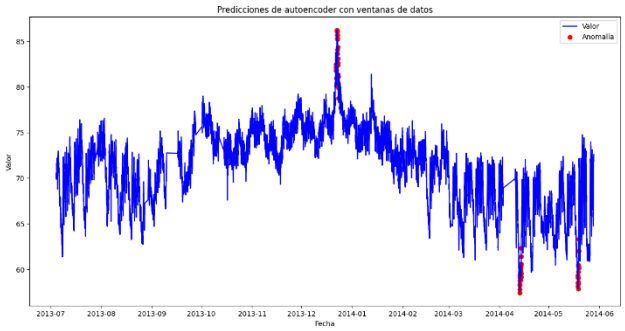


   Como las anomalías aplican a una ventana de datos (es la ventana la que es anómala o no), en el listado de texto se puede indicar, por ejemplo, la primera fecha de la ventana para indicar cuándo se produce la anomalía.

c. Mejora del modelo (1 punto). El estudiante debe intentar mejorar el modelo inicialmente desarrollado aplicando **2** cambios. Por ejemplo, se puede usar escalado de datos (MinMaxScaler), cambiar la estructura de la red, el tamaño de las ventanas o incorporar más características en el modelo de entrenamiento. Un ejemplo de más características en el modelo puede ser la incorporación en el dataframe de hora del día, día de la semana, etc. Como resultado final se debe indicar qué aspectos se han modificado y un análisis de los resultados obtenidos.

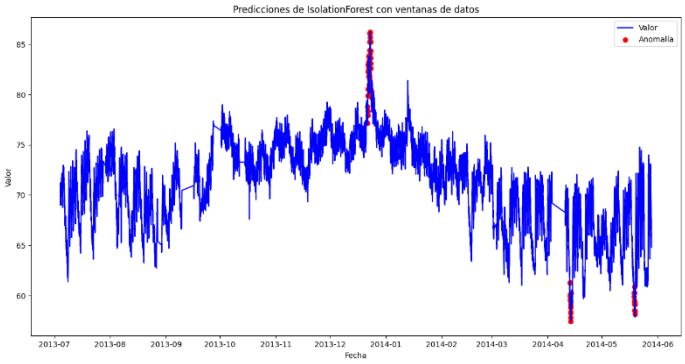
2. Autoencoder (2,5 puntos)

Trabajando sobre los mismos datos del apartado anterior, se debe construir un modelo basado en autoencoders para la detección de anomalías. Se deben mostrar los mismos resultados que en el apartado anterior en cuanto a fechas de las anomalías y una gráfica donde se puedan identificar. El resultado (gráfico) debe ser similar al siguiente:



3. Isolation forest (2,5 puntos)

Trabajando sobre los mismos datos del apartado anterior, se debe utilizar la técnica de IsolationForest para la detección de anomalías. Se deben mostrar los mismos resultados que en el apartado anterior en cuanto a fechas de las anomalías y una gráfica donde se puedan identificar. El resultado (gráfico) debe ser similar al siguiente:



Para los apartados 2 y 3 se pueden utilizar librerías como scikit-learn y TensorFlow, o bien, investigar el uso de PyOD (https://pyod.readthedocs.io/en/latest/) para la detección de las anomalías.

La entrega debe incluir:

- Jupyter notebook o Memoria en PDF indicando todos los aspectos relevantes que se quieran incluir.
- Código fuente (ficheros .py o notebook). Se tendrá en cuenta la organización del código entregado. Su legibilidad, modularidad, comentarios en el código, etc. Instrucciones para su funcionamiento incluyendo versión de Python y fichero requirements.txt