Architectural Decisions Document
Anomaly Detector Project

Carlos Granados

1 Introduction

In this document we describe the outline of several of the architectural decisions taken to develop our deep learning model (DLM), used to detect anomalies in the power generation of solar panels.

2 Data Source

2.1 Technology Choice

For this application we decided to use *CSV data files*, obtained from the kaggle.com website, and uploaded by Ani Kannal. Individual files contain the power generation measured for different solar panels, for different weather conditions.

2.2 Justification

The data are measured every 15 minutes, and then it is saved in CSV form. It offers the advantage to be very easy to read and to share. For our application, we consider to perform monitoring every four hours, then this kind of format offers the perfect flexibility. The desired variables can be saved and then transferred to our application.

3 Enterprise Data

3.1 Technology Choice

We decided to use a *secure gateway*. The server that control the measurements should have the ability to upload every four hours the collected data to a data center and our application can read it from it.

3.2 Justification

It offers a high level of security and privacy. The data is protected and can be accessed only by the application and the administrators of the project.

4 Streaming Analytics

4.1 Technology Choice

We decided to use *Node-RED* based streaming engines to transfer the data.

4.2 Justification

This kind of technology can run on small machines, as a Raspberry PI, making it the perfect candidate for our purposes. The machine can save the data, select the required variables and then transfer them every four hours to our servers. Latency usually is not a big factor for our considerations.

5 Data Integration

5.1 Technology Choice

The *IBM Data Stage on Cloud* is a good candidate for the data integration.

5.2 Justification

Usually the data is already received in the desired format, but with small and efficient scripts the required small changes, as normalization and formatting, can be performed straightforwardly.

6 Data Repository

6.1 Technology Choice

To maintain our data repositories we choose a *relational database*.

6.2 Justification

For our application we require to save the data for only a defined short period of time. The DLM is retrained only after deep maintenance of the units, so no large volume of data have to be considered.

7 Discovery and Exploration

7.1 Technology Choice

To explore the data, we use the traditional *Jupyter* notebooks, together with several of *Python* tools.

7.2 Justification

Each time that a new data set is received, we can plot the power output for each solar panel, and then check how it performs compared with our DLM. The losses generated in the model can be visualized in real-time using the notebooks.

8 Actionable Insights

8.1 Technology Choice

For our DLM we use *Keras*, *TensorFlow* and *Python*.

8.2 Justification

The full training of our DLM is high time consuming, so the tools for paralellization offered by Keras and TensorFlow made them the ideal candidates. In principle the models are relative easy to train as the ideal deep neural network. We use LSTM nodes, in order to make the predictions in a four hours interval. This technology allows us to have a wide variety of analytic tools for the performance; it has the possibility too of the retraining of our DLM.

9 Applications / Data Products

9.1 Technology Choice

For the development we can use *Node-RED*.

9.2 Justification

Once the DLM is trained and set up to work, the results can be streamed back to its source, making the Node-RED option ideal, since it can use the same communication channels as before. No large amounts of date are required to be transmitted, and the code to generate the corresponding plots are simple.

10 Security, Information Governance and Systems Management

10.1 Technology Choice

For the security management of the users, we decided to use *IBM Cloud PaaS/SaaS*.

10.2 Justification

This service manage automatically the user identity, with different registry services, and it can be integrated easily with the data repositories. In particularly a grained access is not required, and a small number of services/users are required to have access to the data or the DLM.