

Universidad Complutense de Madrid

Facultad de Informática



Desarrollo de modelos de predicción de la generación de energía eólica y solar para la optimización de Centros de Datos en un entorno de Smart Grid

Alumno **Gabriel Galán Casillas**

Directores de proyecto **José Luis Ayala Rodrigo**
Marina Zapater Sancho

Trabajo de Fin de Grado

Ingeniería Informática

Septiembre de 2017

Índice general

Palabras clave	5
Resumen	7
1. Introducción	9
1.1. Motivación	9
1.2. Propósito	10
2. Conceptos básicos y metodología	11
2.1. Predicción energética sobre pronóstico	12
2.2. Solar	12
2.2.1. Modelos de predicción	12
2.3. Eólica	16
2.3.1. Modelos de predicción	16
3. Experimentación	19
3.1. Entorno	19
3.2. Pre-procesado	20
3.2.1. Fuentes de datos y obtención	20
3.2.2. Base de datos	21
3.2.3. Servidor	21
3.2.4. Extractor de datos	22
3.3. Simulación de planta solar/eólica	22
3.3.1. Matlab	22
3.3.2. Simulink	22
3.4. Modelos de predicción	24
4. Resultados	27
4.1. Solar: Modelos	27
4.1.1. EWMA	27
4.1.2. Adaptive power management (del ETHZ)	28
4.1.3. Optimal 2D prediction filter	29
4.1.4. WCMA	30

4.1.5. WCMA with PDR	31
4.1.6. Red neuronal artificial	32
4.1.7. N4SID	33
4.2. Eólica: Modelos	34
4.2.1. Eólica: N4SID	34
4.2.2. Eólica: ARIMA/ARMA	35
5. Conclusión	39
Bibliografía	41
Agradecimientos	43
Autorización de difusión	45

Palabras clave

Palabras clave en Español

- Smart Grid
- Predicción energética
- Modelos auto-regresivos
- Energías renovables

Keywords in English

- Smart Grid
- Power forecast
- Autoregressive model
- Renewable energies

Resumen

Resumen en Español

En la actualidad vivimos un continuo incremento de la demanda eléctrica, por otro lado, también se dedican más recursos renovables a abastecer dicho consumo debido a que, a diferencia de los combustibles fósiles, las fuentes renovables no contaminan ni favorecen el calentamiento global. Esto, junto al continuo abaratamiento de la infraestructura de una red de autoconsumo, resulta en una posible solución al problema de la demanda energética global: provisionar un sistema, íntegro o parcialmente, con una red de autoconsumo de energías renovables.

Una parte importante de este consumo es debida a los Centros de Procesamiento de Datos. Estos centros tienen como objetivo procesar toda la información de una organización. Dichos espacios procesan datos ininterrumpidamente durante periodos de tiempo muy extensos lo que implica un consumo prolongado en el tiempo. Este sistema puede ser abastecido si se combinan satisfactoriamente varios elementos: La red eléctrica convencional, una infraestructura de autoconsumo de energías renovables y un conjunto de baterías.

La complejidad radica en saber cuándo y cómo usar cada uno de estos elementos. (i.e. En un periodo de máxima producción solar, no tendría sentido consumir directamente de la red eléctrica). El elemento que añade mayor variedad al cuándo y cómo es la producción de energías renovables. Si tratamos con el sol, no se sabe con certeza cuándo se nublará, o tratando con el viento, cuánto y en qué dirección soplará. Por esto centraremos el trabajo en dicha complicación: Predecir la energía producida.

Para ello estudiaremos múltiples modelos de predicción enfocados a la predicción energética, no a la meteorológica, y evaluaremos los resultados obtenidos. Para ello tomaremos valores meteorológicos extraídos de una estación para, con un modelo detallado de la planta solar o un parque eólico, obtener valores energéticos lo más realistas posibles y aplicar los modelos predictivos a éstos con el fin de obtener y contrastar las predicciones. Con los resultados obtenidos se espera poder evaluar la capacidad y calidad actual de la técnica y en consecuencia, su viabilidad.

Summary in English

We are living a continuous increase in energy demand, on the other hand, more renewable resources are also focused on supplying such consumption because, unlike fossil fuels, renewable sources do not pollute or favor global warming. This, along with the continuous reduction of the infrastructure of a self-consumption grid, results in a possible solution to the problem of global

energy demand: provision of a system (totally or partially) with a grid of self-consumption of renewable energies.

An important part of this consumption is due to Data Centers. These centers aim to process all the organization's data. These spaces process data uninterruptedly for very long periods of time, which implies an extended consumption over time. This system can be supplied if several elements are satisfactorily combined: The conventional electricity grid, an infrastructure for self-consumption of renewable energies and a set of high capacity batteries.

The complexity lies in knowing when and how to use each of these elements. (i.e. In a period of maximum solar production, it would not make sense to consume directly from the grid). The element that adds more variety to when and how is the production of renewable energy. If we deal with the sun, we do not know for sure when it will cloud, or dealing with the wind, how much and in what direction it will blow. That is why we will focus the work on this complication: Predicting the energy produced.

For this purpose, we will study multiple prediction models focused on energy production, not meteorology, and evaluate the results obtained. To do this, we will take meteorological values extracted from a station to obtain the most realistic energy values with a detailed model of the solar plant or wind farm, and to apply the predictive models to these stations in order to obtain and contrast the predictions. With the results obtained it is hoped to be able to evaluate the current capacity and quality of the technique and, consequently, its viability.

Capítulo 1

Introducción

1.1. Motivación

Con la llegada del Internet de las Cosas, el Big Data o las aplicaciones de procesamiento de imagen en tiempo real, la demanda de capacidad de cómputo cada vez es mayor, y los centros de datos (CPD) cada vez son más, más grandes y consumen más¹ (esto último ha reducido el ritmo en los últimos años²). Además se sabe que el consumo de la mayor parte de los CPDs, al menos aquellos destinados a aplicaciones de altas prestaciones (HPC), se realiza de manera perpetua (en mayor o menor medida, pero siempre funcionando), por esto cada vez se opta por suministrar dichos centros con energía obtenida de fuentes renovables, ya que pueden suponer un ahorro energético considerable y aunque aún no es practicable alimentar un CPD completamente con energías renovables, se puede asegurar gran parte del consumo con estas fuentes.

Uno de los problemas del abastecimiento con energías renovables es la impredecibilidad de las fuentes, el sol y el viento. Estas dos fuentes no son constantes en el tiempo (El sol sale y se esconde cada día y el viento sopla con mayor o menor fuerza en función del estado de la atmósfera). En consecuencia la producción es muy inestable y aleatoria.

La solución clásica a estos problemas es almacenar la energía en baterías para el consumo en momentos de no producción.

Otro problema es que las baterías suelen tener una duración y no puede asegurarse todos los periodos de no producción. Así que finalmente se deberá consumir energía de la red eléctrica.

Pongamos un ejemplo: Si se acerca un periodo de no producción de 8 horas y las baterías duran 4 horas, ¿en qué 4 horas será mejor usar las baterías?

En principio nos debería dar igual, pero si nos encontramos en un país en el que hay tarifas horarias quizá convendría usar las baterías en el momento en que la electricidad es más cara. De este pensamiento surge la predicción energética.

Si pudiésemos predecir con moderada exactitud la energía que se producirá, podremos realizar un plan que permitirá ahorrar costes en gran medida

¹https://eta.lbl.gov/sites/all/files/publications/lbnl-1005775_v2.pdf

²<https://deepmind.com/blog/deepmind-ai-reduces-google-data-centre-cooling-bill-40/>

1.2. Propósito

Nuestro objetivo es examinar qué metodología tiene los mejores resultados observados en la predicción energética de una planta solar y una eólica con un horizonte de 24h.

Para la predicción se pueden usar 2 tipos de modelos de predicción: Modelos analíticos, que se basan en realizar un análisis exhaustivo del problema y proporcionar una solución detallada, y Modelos auto-regresivos, que se describen un proceso que proporciona una salida la cual depende linealmente de sus valores anteriores o "histórico".

Dado que queremos obtener un conjunto de reglas genéricas, optaremos por usar modelos auto-regresivos que se basan en el histórico para realizar predicciones y permiten abstraernos de la planta, el panel y las condiciones.

Con estos modelos, realizaremos predicciones con valores extraídos de una planta solar/eólica (Simulada) y tras extraer los resultados de la predicción y el error, analizaremos los diferentes puntos.

Además definiremos *Horizonte de predicción* como el límite máximo en términos temporales en los cuales se puede realizar una predicción válida. En nuestro caso será el periodo máximo para las predicciones. El horizonte de predicción puede adquirir cualquier valor, desde ultra-corto plazo como puede ser de minutos hasta a largo plazo como son semanas.

En materia de predicción energética, la precisión primará sobre la lejanía del horizonte de predicción, por esto barajaremos valores cercanos. Además, debido a la periodicidad del sol y la duración habitual de las baterías optaremos por 24 horas, que es el valor mínimo de calidad de días y está por encima de la duración de un equipo de baterías (Esto podría ser 8 horas).

Capítulo 2

Conceptos básicos y metodología

En un caso real, la planta generadora proporciona los valores de potencia que se introducen en cada modelo más el histórico que irá siendo almacenado. Debido a la falta de tiempo y recursos no se realizará la experimentación sobre una planta real, por ello en este trabajo simularemos con modelos de Simulink las plantas generadoras.

El gráfico 2.1 representa el proceso de generación de predicciones usado en el proyecto, desde las entradas hasta las salidas.

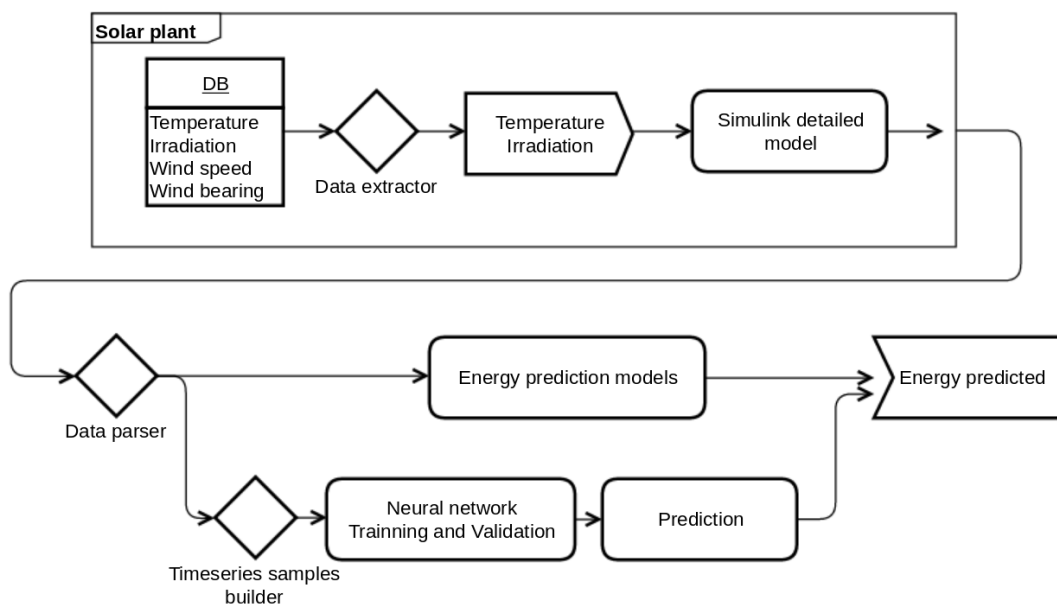


Figura 2.1: Pipeline de predicción

En el diagrama, la caja que contiene "Solar plant" representa lo que realmente sería una planta solar y que tiene como salida la "Potencia actual". Esta está compuesta por:

- La base de datos de donde extraeremos los valores de irradiación y temperatura, que hay sido previamente guardados para poder operar con ellos de manera más cómoda, pero que realmente sería un valor continuo.

- El extractor de datos, cuyo propósito es extraer los datos de la base de datos, revisar la coherencia e integridad (En caso de que alguna entrada en la base de datos no exista o se haya borrado, la rellenará con coherencia) y darle un formato apto para Matlab.
- Finalmente el modelo detallado de Simulink junto con Matlab, procesarán las entradas y producirá una salida.

El siguiente paso ya sería la propia predicción:

- Primero se trata la salida de la planta (Esta será un valor continuo que habrá que discretizar y darle un formato de vector o matriz, de valores, celdas o muestras de series temporales).
- Posteriormente cada modelo realizará la predicción oportuna dando una salida y unas gráficas de rendimiento (Comparativas y error absoluto).

2.1. Predicción energética sobre pronóstico

El gran problema de las energías renovables es su falta de estabilidad y su complejo pronóstico. En el caso de la energía solar, no habrá dos días que tengan la misma producción energética porque depende de la irradiación, de la temperatura, del ángulo, la contaminación, etc. Aunque sí se podrá expresar la periodicidad del sol; en el caso de la energía eólica, el viento dependerá de la presión atmosférica, de la posición de las borrascas, la zona, la altura, etc.

Por esto, se dedicará el trabajo a las predicciones basadas en modelos auto-regresivos. El resultado de la predicción será contrastado con los valores reales medidos del periodo precedido (Extraídos de la planta simulada).

Si por ejemplo queremos predecir el día siguiente, usaremos como histórico X días pasados y el resultado podremos contrastarlo con el valor real del día siguiente sin esperar a este por tener todos los valores ya guardados en la base de datos.

Otra duda que podría aparecer es "¿Por qué usar este modelo de predicción si se puede pronosticar el tiempo con muchísima más exactitud?". Es cierto que la predicción en lo que respecta al clima está mucho más avanzada, pero tener la predicción como entrada y la potencia como salida (A fin de contrastarla con la demandada) implicaría tener un modelo exacto de la planta generadora que permita obtener en función de las múltiples variables implicadas, la potencia. Esto, aunque es una solución óptima, queda fuera del ámbito del proyecto ya que los pilares son los modelos auto-regresivos, los cuales se abstraen de los detalles técnicos.

2.2. Solar

2.2.1. Modelos de predicción

Los modelos se expresan en términos de energía y tienen horizontes de predicción diferentes, horarios principalmente. Por esto, los modelos que tengan un horizonte de predicción de 1 hora, se mantendrán (No son capaces de proporcionar una predicción válida a 24 horas y modificarlos

podría inducir a errores) y los que tengan un horizonte definido por el usuario (Red neuronal, N4SID o ARMA) se establecerán a 24 horas.

La predicción de energía solar se abordará con los siguientes modelos:

- Exponentially weighted moving average
- Predictor for adaptive management developed at ETHZ
- Optimal 2D linear prediction filter
- Exponentially Weighted Moving Average
- Exponentially Weighted Moving Average with Phase Displacement Regulator
- Neural network
- N4SID

Exponentially weighted moving average

El modelo Exponentially Weighted Moving Average (EWMA [2]) es el más sencillo y tradicional. Para poder ajustarse al ciclo solar, se realizan subsegmentos de tamaño **numero de muestras por día**. Este se basa en una suma basculada de los valores Real y Pronosticado, siendo este último el resultado del predicho del día anterior.

$$E_{PRED}(d+1) = \rho E_{REAL}(d) + (1-\rho)E_{PRED}(d) \quad (2.1)$$

La energía predicha en un momento t y día d será la suma de la energía medida en el momento t del día anterior y la energía predicha en el momento t del día anterior.

En esencia, el algoritmo usa el histórico de los días anteriores para predecir el siguiente. Si la constante ρ esta más cerca del 1, la predicción tenderá a ser el valor del día anterior. Sin embargo, si ρ es más próxima al 0, se explotara el registro acumulado.

Pros: Bajo procesamiento y consumo de memoria

Contras: Predicción muy simple sin control de errores

Predictor for adaptive management developed at ETHZ

Este modelo desarrollado en el Swiss Federal Institute of Technology of Zurich (ETHZ) [3] de predicción combina la predicción a corto plazo (horas) con largo plazo (días) para producir en un instante t un vector de longitud L de predicciones. Sin embargo, para poder compararlo con los demás algoritmos que realizan una predicción, usaremos una versión a corto plazo (Horas) y una longitud de predicción fija.

$$E_S(t) = \min((1-\gamma)E_N(t) + (\gamma)R(t), E_N(t)) \quad (2.2)$$

$$R(t) = \beta R(t-1) + (1-\beta)E_S(t-1) \quad (2.3)$$

$$E_N(t) = \alpha E_N(t-1) + (1-\alpha)E_S(t-1) \quad (2.4)$$

E_S es la energía real producida por la planta fotovoltaica,

E_N es la energía promedio medida en T ultimas muestras con factor α Finalmente R es el promedio a corto plazo con factor β

Este algoritmo se basa en el mismo principio que el EWMA separando el corto plazo del largo plazo.

Pros: Tiene mejor factor de predicción a corto plazo que el EWMA

Contras: Sigue sin filtrar variaciones habituales y no es inmune a imprevistos

Optimal 2D linear prediction filter

El fundamento de este algoritmo [4] es la representación matricial de la energía. Siendo las filas los días y las columnas los momentos (horas en el estudio) ambas ordenadas en orden creciente, un nuevo dato de energía es igual a los valores de la columna (K días anteriores) más los valores de fila (K horas anteriores) más los valores previos ($(K - 1)^2$ horas anteriores en días anteriores).

Siendo el tamaño de venta K y el nuevo valor (K, K) , los valores $(_, K - 1)$ son los días, $(K - 1, _)$ son las horas y $([1, K - 1], [1, K - 1])$ son los valores previos no relacionados.

Cada uno de estos 3 elementos será multiplicado por unos coeficientes a_1, a_2 y a_3 .

Este sería un ejemplo para tamaño de ventana $K = 2$

$$X'_{i+1,j+1} = a_1 X_{i,j} + a_2 X_{i,j+1} + a_3 X_{i+1,j} \quad (2.5)$$

Para tamaño de ventana superior simplemente sumar la columna multiplicada por a_1 y dividida por K. Lo mismo con la fila y con la submatriz que queda de quitar la fila y columna.

Pros: Toma valores del histórico a corto (i.e. horario) y largo (i.e. diario) plazo. Contras: Igual que EWMA o ETHZ, no tiene control de errores ni modifica las variables dinámicamente.

Weather Conditioned Moving Average

El algoritmo más explotado en la literatura, WCMA [2], tiene sus bases en el EWMA con el valor añadido de que tiene una visión global del periodo y control de cambios.

$$E_{(d,n+1)} = \alpha E_{(d,n)} + (1 - \alpha) \frac{\sum_{i=1}^D E_{i,n+1}}{D} \quad (2.6)$$

$$v_k^{d,n} = E_{(d,n-k+1)} \frac{D}{\sum_{i=1}^D} E_{i,n-k+1} \quad (2.7)$$

$$p_k = 1 - \frac{K - k + 1}{K} \quad (2.8)$$

$$GAP_k^{d,n} = V^{d,n}(K) \times (P(K) / \sum P(K))^T \quad (2.9)$$

$$E_{d,n+1}^{GAP} = \alpha E_{(d,n)} + (1 - \alpha) GAP_K^{d,n} \frac{\sum_{i=1}^D E_{i,n+1}}{D} \quad (2.10)$$

$E_{(d,n+1)}$ esta es la energía calculada siguiendo el mismo estilo que EWMA.

Con $v_k^{d,n}$ se obtiene la mejora de las condiciones a lo largo de un periodo k previamente definido. Si $v > 1$ se considera que las condiciones del momento n del día d son mejores que los días anteriores. Iguales o peores en otro caso.

p_k es un vector de longitud k cuyo objetivo es dar más importancia a los valores más cercanos al momento actual. Esto se consigue con un vector creciente entre 0 y 1 que representa una recta creciente. e.g. [0, 0.33, 0.66, 1]

$GAP_k^{d,n}$ es el valor clave del algoritmo. Como se ve en la fórmula, contiene V y P . De esto se concluye que GAP representa la cantidad de energía de más o de menos que hay actualmente, ya que compara el momento actual con los valores anteriores, cuantificados con P (valor más cercano al actual, más peso) dando como resultado >1 si hay más energía disponible o <1 si hay menos.

Finalmente $E_{d,n+1}^{GAP}$ es el resultado de la predicción.

Pros: Permite tener constancia de los cambios más recientes gracias a la $v_k^{d,n}$

Contras: Si no se tiene control de posibles valores negativos ()

Weather Conditioned Moving Average with Phase Displacement Regulator

Esta mejora del algoritmo WCMA incluye la variable PDR que permite seguir el rastro de los errores además de reajustarlos. Esto se realiza gracias al histórico de errores. En función del error que se observó en ese momento de días, se suma o resta el error a fin de neutralizarlo.

$$PE(d, i) = \delta PE(d-1, i) + (1 - \delta) \frac{\sum_{i=1}^D PE_{i,n+1}}{D} \quad (2.11)$$

$$PDR_{d,n} = \left[\frac{\sum_{w_i=0}^{w-1} [PE(n-w_i) + PE(n+w_i)] \gamma^{(w_i+1)}}{\sum_{w_i=0}^{w-1} \gamma^{(w_i+1)}} \right] \quad (2.12)$$

$$E_{(}^{PDR} d, n+1) = \alpha E_{d,n} + (1 - \alpha) GAP_K^{d,n} \frac{\sum_{d,n}^D E_{i,n+1}}{D} + PDR_{d,n+1} \quad (2.13)$$

$PE(d, i)$ representa el error en cada momento, basculado con los errores pasados.

$PDR_{d,n}$ es el valor final que regula la salida del algoritmo. Toma como valor clave PE, más concretamente, un PE con tamaño de ventana $2w$ (los w PE anteriores y w PE siguientes). Esto permite equilibrar el error entre los que le rodean.

Pros: Tiene control del error pasado

Contras: Necesita un registro importante de valores para ajustarse bien

Neural network prediction

Consideraremos también una red neuronal. Estas al estar destinadas a resolver problemas cuyas entradas están fuera de lo esperado hacen aparentemente idónea la solución ya que la variabilidad de las entradas (el clima) depende de muchas variables difíciles de contemplar simultáneamente. Aprovecharemos la capacidad adaptativa y analítica con propagación hacia atrás para predecir valores con bajo coste de memoria.

La red neuronal estará diseñada con 2 capas de 4 neuronas y función sigmoidea $O = \frac{1}{(1+e^{-n})}$

El diagrama 2.2 representa la red neuronal que usaremos:

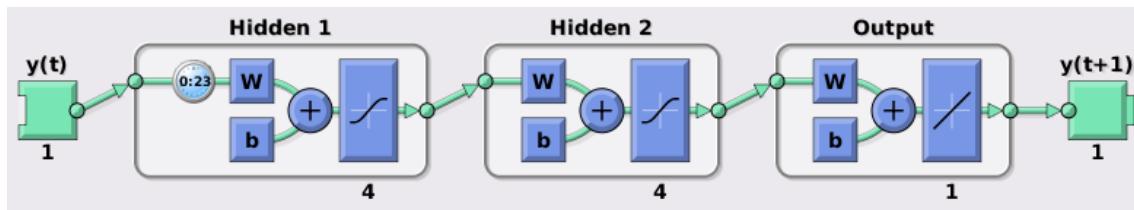


Figura 2.2: Diagrama de la red neuronal

Cada muestra de entrada seguirá el siguiente formato:

- D entradas con los valores de los valores anteriores
- K entradas con los valores de los días anteriores
- K-1 entradas con las diferencias entre los días anteriores

El entrenamiento será con el algoritmo de regularización Bayesiana, que mejorará el rendimiento en muestras con ruido.

Pros: Puede reaccionar y predecir mejor ante situaciones inesperadas

Contras: Requiere un entrenamiento continuo. Consume muchos recursos

N4SID

El algoritmo N4SID[8] (Sistema de identificación de subespacios de espacios de estados) permite estimar un modelo de espacio de estados usando subespacios.

Un modelo de espacio de estados es un modelo que usa variables de estado para describir un sistema con un sistema de ecuaciones diferenciales de primer orden o ecuaciones diferenciales.

Los modelos de espacio de estados son apropiados para realizar estimaciones (predicciones en este caso) ya que solo requieren del orden, el cual está relacionado con el retardo de las entradas y salidas a usar en la ecuación diferencial.

La representación del modelo estimado es un sistema como:

$$x(t) = Ax(t) + Bu(t) + Ke(t) \quad (2.14)$$

$$y(t) = Cx(t) + Du(t) + d(t) \quad (2.15)$$

Donde A,B,C y D son matrices del espacio de estados, K es la matriz de perturbaciones, $u(t)$ es la entrada, $y(t)$ es la salida, $e(t)$ es la perturbación y $x(t)$ es el vector de n_x estados, con n_x igual al orden.

2.3. Eólica

2.3.1. Modelos de predicción

La predicción de energía eólica es mucho más compleja que la solar ya que la irradiación a lo largo del día sigue una distribución normal. Luego se verá afectada por múltiples factores como

nubes o polución, pero el sol siempre sigue el mismo patrón. El viento, en cambio, no. Y esto dificulta mucho la tarea.

Abordaremos la predicción de energía eólica con los siguientes 2 modelos:

- Modelo ARIMA
- N4SID

ARMA/ARIMA

ARIMA¹: AutoRegressive Integrated Moving Average. Este modelo se usa, o para predecir valores (Su propósito aquí) o bien para entender los valores pasados.

Para entender un poco cómo funcionan los modelos ARIMA lo explicaremos por componentes.

El factor AutoRegressive, Auto regresivo o el AR de ARIMA define la variable en cualquier instante del tiempo como una combinación de los valores anteriores más un error.

El factor Moving Average, Media móvil o el MA de ARIMA define un valor de la variable como una suma de un valor α más la suma ponderada de los errores.

El factor Integration o la *I* de ARIMA indica que un valor de la variable es la diferencia entre un valor y su sucesor, pudiendo haber sido realizada la diferencia varias veces.

Sabiendo esto el modelo ARIMA se enuncia como ARIMA(p,d,q) siendo p,d y q números naturales que representan la distancia del retardo a usar, el número de veces que se realizó la resta en la diferencia y el orden de la ecuación de media móvil.

¹https://en.wikipedia.org/wiki/Autoregressive_integrated_moving_average

Capítulo 3

Experimentación

En este capítulo explicaremos todo el proceso de experimentación, desde la recogida de datos hasta la obtención de resultados como se ilustra en el diagrama 2.1.

3.1. Entorno

Para poder llevar a cabo unos resultados validos sería necesario tener una planta solar capaz de producir energía y la monitorización de esta. En este caso, dispondríamos de la potencia actual generada, la cual es la entrada a los algoritmos de predicción.

Dado que no es así, simularemos este comportamiento con un proceso manual de recogida y procesado de datos compuesto por una planta simulada con la cual obtendremos los valores de salida procesando valores almacenados en una base de datos. La temperatura y radiación será recopilada por un RESTful API¹, almacenada en una base de datos, extraída y formateada para ser introducida en el modelo de planta fotovoltaica de Matlab. Tras ejecutar la simulación, los valores de potencia resultantes serán pasados a los distintos modelos para obtener las predicciones. Una vez obtenidas las predicciones, serán contrastadas con los valores reales obteniendo así gráficas que prueben la certeza de estas.

A continuación se explican cada uno de los componentes del entorno.

- Pre-procesado
 - Fuentes de datos y obtención
 - Base de datos (DB)
 - Servidor
 - Extractor de datos (Traductor del contenido de la DB a Matlab)
- Simulador de planta solar y eólica
 - Matlab
 - Simulink
- Modelos de predicción

¹https://en.wikipedia.org/wiki/Representational_state_transfer

3.2. Pre-procesado

3.2.1. Fuentes de datos y obtención

El primer paso para la puesta en marcha del modelo de planta solar es recoger los valores de radiación y temperatura para los modelos solares, velocidad del viento y dirección para los eólicos. Estos son las variables más importantes a la hora de producir estas energías renovables, por eso nos centraremos en ellas y serán las introducidas a los modelos del Simulink².

Debido al esfuerzo requerido de tener que montar los propios sensores (De temperatura, irradiación, velocidad y dirección) y esperar a los valores predichos para ser contrastados, descartamos hacer un tratamiento de estas entradas en tiempo real. Aunque habría sido una opción perfectamente válida.

Por esto primero recogemos valores para posteriormente procesarlos y poder contrastarlos inmediatamente contra los valores del momento de la predicción.

Entonces para la recogida de datos barajamos las siguientes opciones:

1. Datos crudos de la estación meteorología de AEMET
2. API's públicas

A continuación detallamos la toma de decisiones.

Estación meteorológica de AEMET

Los datos en crudo proporcionados por la estación meteorológica de AEMET fue nuestra primera opción por ser una base de datos sólida a la que tenemos acceso. Tiene puntos fuertes como proximidad, soporte técnico y fiabilidad, sin embargo la insolación "hace referencia la radiación solar **porcentual** y sólo está disponible en Barajas y aunque podría ser un buen punto de partida, la temperatura sólo está disponible en Ciudad Universitaria y El Retiro. Por esto se decidió optar por otras fuentes de datos para el estudio.

API's públicas

Descartada la estación meteorológica, nos inclinamos por un API pública y gratuita que proporcionase los valores necesarios: Temperatura e Irradiación además de Velocidad del viento y Dirección.

Algunas de las opciones encontradas fueron: - Madrid AEMET opendata³ - El tiempo⁴ - solarelectricityhandbook.com/solar-irradiance.html⁵ - National Renewable Energy Laboratory (NREL)⁶ - dark sky weather⁷

²<https://es.mathworks.com/products/simulink.html>

³<https://opendata.aemet.es/>

⁴<http://api.tiempo.com/>

⁵<http://solarelectricityhandbook.com/solar-irradiance.html>

⁶<https://developer.nrel.gov/docs/solar/solar-resource-v1/>

⁷<https://darksky.net/dev/docs>

Tras optar por revisar otras fuentes, se observó que no es tan común disponer de un servicio que provinee la irradiación solar a nivel horario. El NREL dispone un API con la irradiación a nivel mensual, pero no es suficiente.

Tras ver que ninguno de éstos trae valores reales de irradiación (temperatura todos) y dado que el objetivo es ver cómo se comportan los modelos de predicción en un escenario real, se optó por combinar la oclusión (La cobertura del cielo/nubes) con una función normal de 5 valores. La cantidad de sol recibida al día dibuja casi la misma gráfica cada día a lo largo de todo el año (Desfasada y/o ampliada en función de la estación). Las nubes son el mayor agravante, por ello las tomaremos como el factor importante.

Finalmente nos quedamos con Dark Sky Weather que era capaz de proporcionar casi todo (Temperatura, Cobertura del cielo, Hora de amanecer/atardecer, Velocidad del viento y Dirección del viento) y tiene un RESTful API muy cómoda para recoger los datos.

Los datos necesarios para acceder a dicho API son la url y la API KEY.

Con estos datos ya es posible realizar llamadas en forma de peticiones HTTP y obtener los resultados.

Las funciones usadas fueron: Get token y Get weather

A la hora de consultar el tiempo y las predicciones incluimos varias opciones para obtener resultados más concisos.

3.2.2. Base de datos

Una vez obtenida la fuente de datos será necesario dotarla de cierta persistencia, ya que el API nos proporcionará los valores reales en un momento t . Para ello almacenaremos los resultados de las consultas en una base de datos de MongoDB⁸.

Este programa de base de datos está enfocado al paradigma de No-SQL⁹ (Not Only Structured Query Language o Not Structured Query Language, depende del enfoque) que permite tratar documentos (La materia prima de este tipo de base de datos) de manera rápida y sin necesidad de crear un esquema (En la analogía de las bases de datos relacionales o SQL, las tablas).

3.2.3. Servidor

Habiendo elegido la fuente de datos y el lugar donde debe persistir la información, es necesario automatizar la recogida y su almacenado.

Para ello optaremos por una aplicación creada con el framework Nodejs¹⁰ (Un muy versátil framework de desarrollo de JavaScript enfocado al desarrollo web) y una base de datos MongoDB.

El servidor se encarga de realizar periódicamente las llamadas al API de Dark Sky, añadir la fecha de la llamada (por claridad. Se recibe también las fechas pero al estar en formato POSIX¹¹ no es legible para un humano), quitar las predicciones horarias excepto del principio del día e insertar las respuestas en la base de datos.

⁸<https://www.mongodb.com/>

⁹<https://en.wikipedia.org/wiki/NoSQL>

¹⁰<https://nodejs.org/en/>

¹¹https://en.wikipedia.org/wiki/Unix_time

Este proceso deberá estar corriendo ininterrumpidamente durante todo el periodo del cual quiera obtenerse valores, y un error que deje al servidor caído podrá no realizar la consulta al API y perder los datos de ese momento, por ello dotaremos al servidor de cierta resiliencia.

La arquitectura estará montada en un Ubuntu Desktop 16.04 y dockerizada con la herramienta Docker¹² que permitirá trabajar sobre contenedores. Sin entrar en detalles, crearemos un contenedor que contendrá el servidor, otro la base de datos y si se cayese alguno de los dos, con la configuración adecuada se recrearían ambos y se volverían a lanzar en cuestión de segundos.

Finalmente el servidor estará emplazado físicamente en el Centro de Datos de la Facultad de Informática.

3.2.4. Extractor de datos

Una vez tenemos la base de datos con contenido suficiente en continuo crecimiento, ya se puede proceder a extraer los datos de la base de datos y pasárselos al modelo de planta fotovoltaica de Simulink para que proporcione una potencia de salida.

Para ello, primero realizaremos una consulta a la base de datos y escribiremos los resultados en un archivo de texto que reconozca Matlab y facilite su manipulación, que en el proyecto será JSON¹³.

A lo largo del desarrollo se han implementado 2 tipos: Como texto donde cada línea tiene datos separados por tabulaciones y como JSON.

Dado que la base de datos no contiene el 100% de los datos (Se escapan alrededor de 2-4 entradas por mes) se rellenan esos datos con el valor más próximo a fin suprimir posibles outliers.

Con el archivo en formato JSON creado, introducir los valores en el modelo de Matlab será tan sencillo como `importdata(filename)`.

3.3. Simulación de planta solar/eólica

3.3.1. Matlab

Para la implementación de los modelos se ha optado por usar Matlab¹⁴ ya que a diferencia de C++ u otros lenguajes, tiene un toolbox de diseño de redes neuronales muy completo y práctico complementado con una excelente documentación.

La primera función de Matlab será extraer los valores de los ficheros producidos en la fase anterior 3.2 y crear una señal para el modelo de Simulink. Una vez hecho esto se podrá comenzar la simulación y obtener los resultados. Finalmente se discretizará la señal de salida para ajustarla al formato de horas y días para pasárselo a los modelos.

3.3.2. Simulink

Como anotamos anteriormente, usaremos un modelo teórico de una planta solar y otra eólica para obtener valores de potencia coherentes con los que podría dar una planta real. Por esto

¹²<https://www.docker.com/what-docker>

¹³<http://www.json.org/>

¹⁴<http://www.json.org/>

optamos por Simulink, que además de ser software completo de diseño y simulación eléctricos, tiene una comunidad muy grande que nos ha permitido encontrar un modelo detallado tanto de una planta solar como una eólica.

Se puede encontrar toda la información sobre estos modelos en

La figura 3.1 ilustra la simulación de una planta fotovoltaica de 100kW.

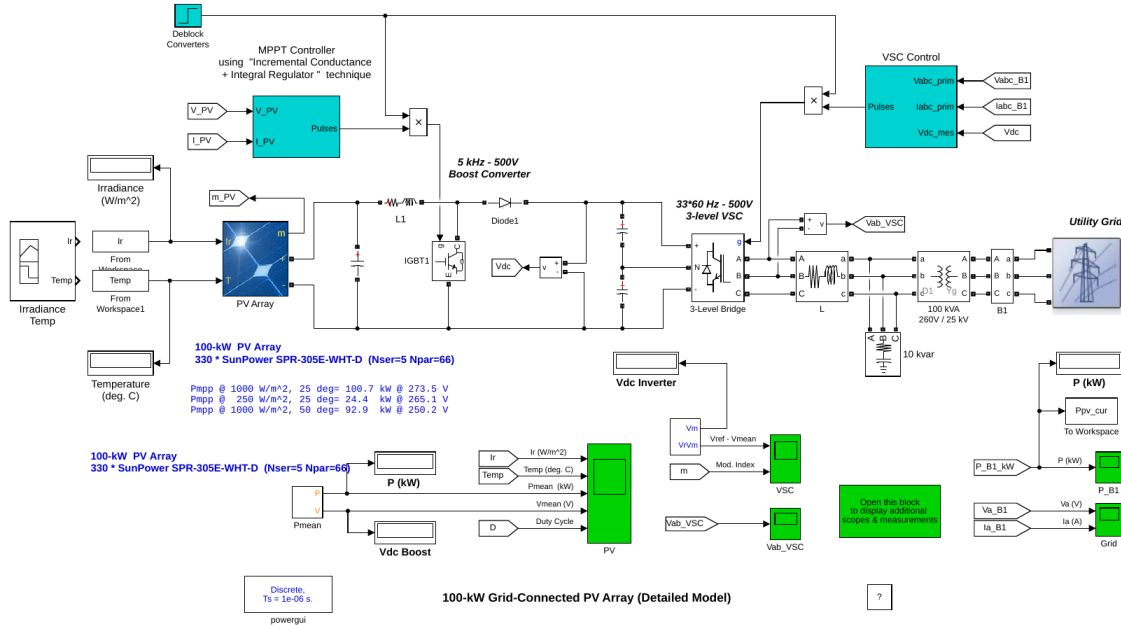


Figura 3.1: Diagrama de la planta fotovoltaica de 100kW conectada a una red

El modelo simula una planta fotovoltaica con las siguientes características:

- 100kW a $1000W/m^2$
- Conectada a una red de 25kV de tres fases
- Conexión DC-DC con amplificador de tensión
- 1980-Hz con VSC de 3-niveles y 3-fases
- Condensador de 10kvar para el filtrado

Para llevar a cabo la simulación es necesario introducir las variables más relevantes para el panel fotovoltaico: Radiación y Temperatura. Tiene como resultado la potencia instantánea en vatios. El modelo evalúa las entradas 1000 veces por segundo dando como resultado una detallada gráfica de potencia producida.

La importación y exportación de los valores se ha realizado desde el workspace, donde se pre y post procesan las señales resultantes: Las entradas (Radiación y Temperatura) se formatean de acuerdo a las entradas del modelo y las salidas (La potencia) se discretiza a valor por hora, ya que se evalúa la potencia a un ratio de 1000 veces por 1 segundo de simulación.

La quincena de potencia que usaremos es la descrita entre el 1 y el 16 de abril. Se puede observar en la gráfica 3.2 donde el eje x son los días y el y la potencia generada en vatios.

El resultado de la simulación puede observarse en la figura 3.2 :

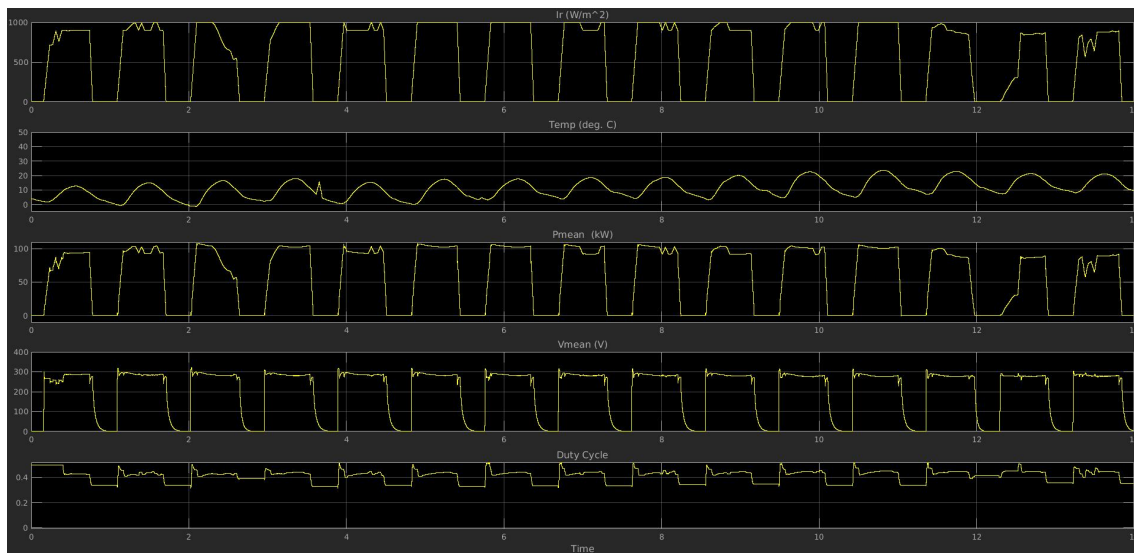


Figura 3.2: Salida de la planta fotovoltaica

3.4. Modelos de predicción

A fin de desacoplar lo máximo posible la predicción de otras etapas (Como importación del fichero, simulación o procesado de la señal de salida), los modelos leerán la variable del workspace E_{cur} cuyo formato será filas = días, columnas = horas

En función del algoritmo y la cantidad de operaciones a nivel horario se optará por trasponer y hacer los accesos aprovechando el acceso de índice único de Matlab a fin de poder tratar de manera más cómoda.

Todos los modelos están preparados para realizar una única predicción con las constantes definidas (Sin etapa de ajuste como podría ser realizar una predicción con cada constante y elegir la menor dispersión) y mostrar el resultado final con una comparación entre valores reales - predichos y el error.

N4SID

Consideraremos también un N4SID (Algoritmo numérico para identificación de sistemas de espacios de subespacios de estados) y su implementación de Matlab. Tanto este modelo como el ARMA/ARIMA están basados en modelos de estados. Con la función $m = n4sid(data, order, 'Property1', Value1, \dots)$, siendo m el modelo resultante [idss], $data$ la entrada y salida [idss], $order$ el valor de n_x que en este caso dejaremos en 'bestél' cual escogerá el mejor entre 1 y 10, y finalmente pares propiedad-valor, que únicamente usaremos N4Horizon, que define el horizonte de predicción y será 24h.

Para este modelo de predicción tomaremos los valores de salida del modelo de simulink $pe_{asm_generator_2}$ separándolos en 75 % para entrenamiento y 25 % para validación.

Lo más importante para que funcione adecuadamente, no solo el modelo sino también un molino real, es que la velocidad se encuentre entre 3 y 23m/s. De otro modo la salida será 0.

Entrenamos realizando un bucle entre 8 y 168 que representan los valores del pasado tomados para el entrenamiento. Entre 1/3 de día y 7 días. Guardando en cada iteración el valor de ajuste para finalmente crear el modelo con los valores finales.

ARMA/ARIMA

Para realizar la predicción con alguno de estos modelos primero habría que estudiar si la serie a tratar es estacionaria (Se realiza solo con la de entrenamiento). Si es estacionaria, se puede omitir la Integración (ARMA) si no, ARIMA.

Primero habrá que asegurarse de que la velocidad del viento ni baja de 3m/s ni excede los 23m/s ya que por debajo no producirá nada y por encima, para evitar riesgos de rotura del molino, se detendrá.

A continuación se estudiará si la serie es estacionaria, en cuyo caso se suprimirá la fase de Integración (ARMA) o si no lo es, en cuyo caso habrá que transformarla para extraer las líneas de tendencia.

Si se da que no es estacionaria, pueden diferenciarse dos tipos: En varianza, donde la transformación será cox-box o en media, donde la transformación será diferenciación de la serie. Si finalmente resulta ser no estacionaria en media, y por tanto habrá que calcular las diferencias, el número de veces que se realicen estas será el índice de I de ARIMA.

Finalmente se realizará un bucle doble sobre la q y la p a fin de encontrar el mejor valor de fitness y así definir el modelo como AR(I)MA(p,d,q).

Capítulo 4

Resultados

A continuación se exponen los resultados de la experimentación y observaciones.

4.1. Solar: Modelos

Para la experimentación se ha usado el mismo periodo de 15 días, entre el 15 y el 30 de marzo del 2017 para los entrenamientos y 1 al 15 de abril para las predicciones.

Los resultados mostrados a continuación se presentan como energía real producida más energía predicha y, error instantaneo más RSME.

4.1.1. EWMA

Este algoritmo debido a simplicidad no ofrece los mejores resultados, aunque tienen una precisión en lo que a la predicción solar respecta aceptables. En la gráfica 4.1 se puede observar que usa los valores del día anterior con una ligera atenuación de la media de las predicciones pasadas. El nivel de error es especialmente alto durante el amanecer y el ocaso. Este fenómeno veremos que sucede en múltiples modelos y es ocasionado por no ser capaz de adaptarse al ciclo solar.

En el caso del EWMA, al usar los valores anteriores, tras adaptarse a la noche, el cambio al día ocasiona un notable error de predicción. Además es especialmente sensible a cambios no graduales o espontáneos (Como puede ser un día nublado o peor aún, una nube).

$$\alpha = 0,75$$

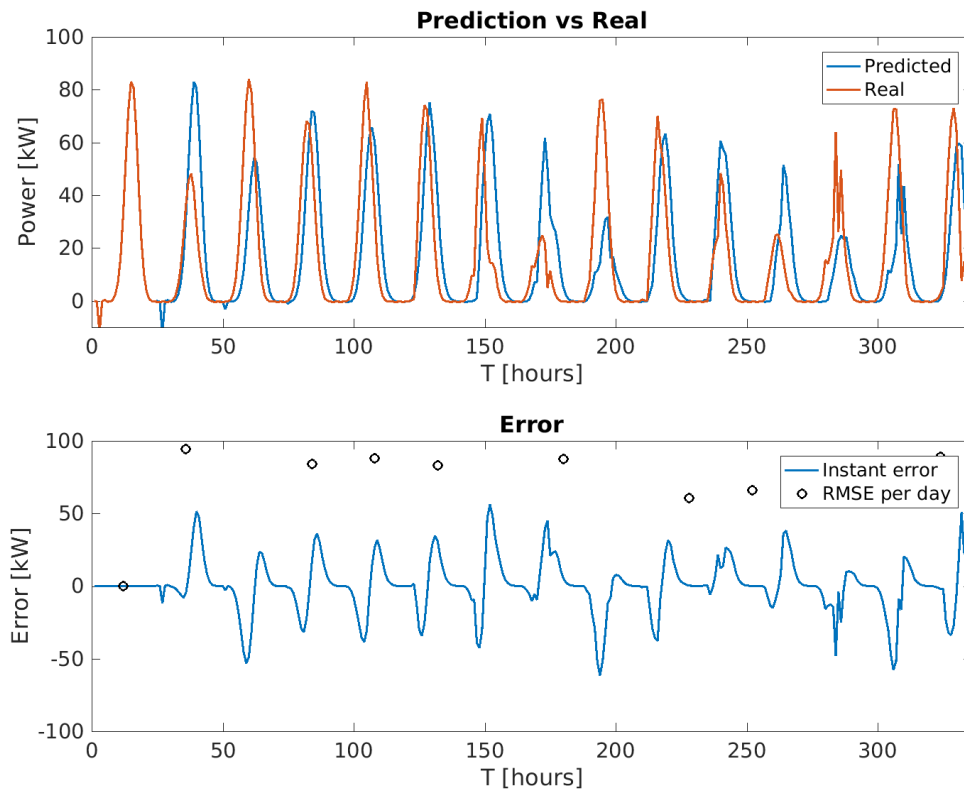


Figura 4.1: EWMA Prediction accuracy

4.1.2. Adaptive power management (del ETHZ)

El resultado de este tipo de predictor es muy similar al EWMA ya que realiza un cálculo basculado del valor del día anterior, pero sutilmente mejorado ya que también añade a la predicción los valores inmediatamente anteriores. El error en este caso es más pronunciado en los extremos del día pero más estable en los puntos de irradiación estable.

$$\alpha = 0,2$$

$$\beta = 0,25$$

$$\gamma = 0,85$$

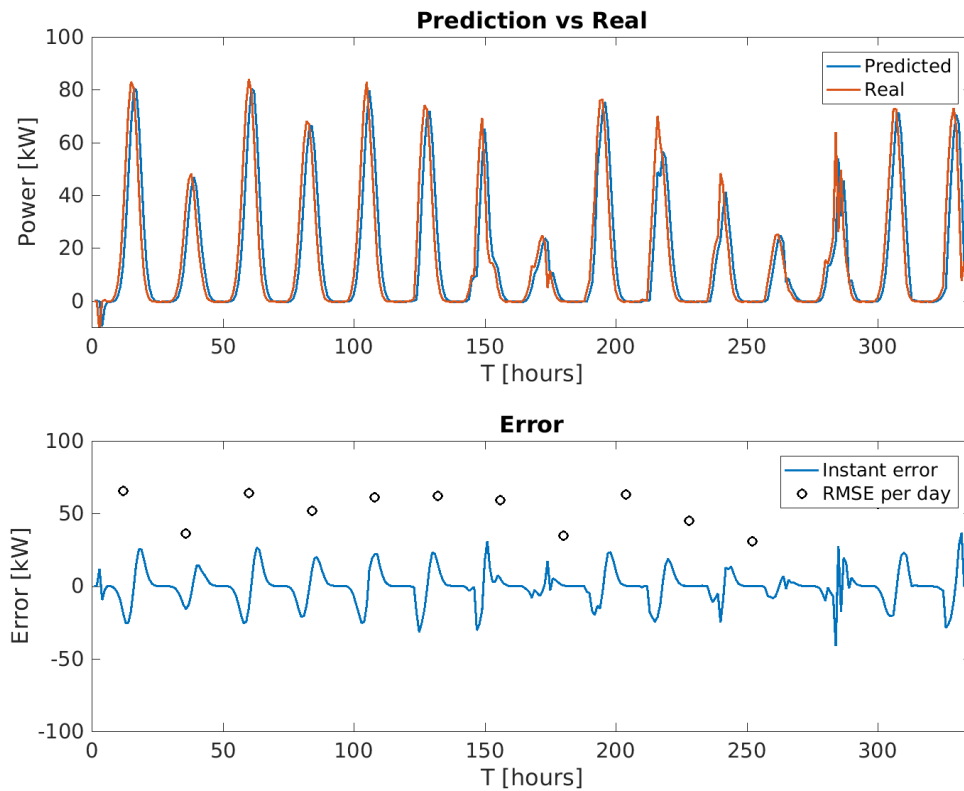


Figura 4.2: ETHZ Prediction accuracy

4.1.3. Optimal 2D prediction filter

Los resultados obtenidos son muy similares a los ya vistos, ya que los principales problemas son, los extremos de luz (atardecer y amanecer) y los imprevistos (Día nublado o una nube que eclipse el sol momentáneamente). Sin embargo este modelo hace uso de los instantes anteriores en los que el sol estuvo en el mismo sitio i.e. Misma hora, día anterior. Por esto se adapta mejor al primer problema. Además se observa en la figura 4.3 que ante cambios de potencia a lo largo de los días, proporciona una predicción pesimista. Esto es un punto a favor pues en caso de un fallo notable en la predicción, siempre es mejor estimar por debajo y tener más energía disponible que enfrentarse a un posible caso de abastecimiento.

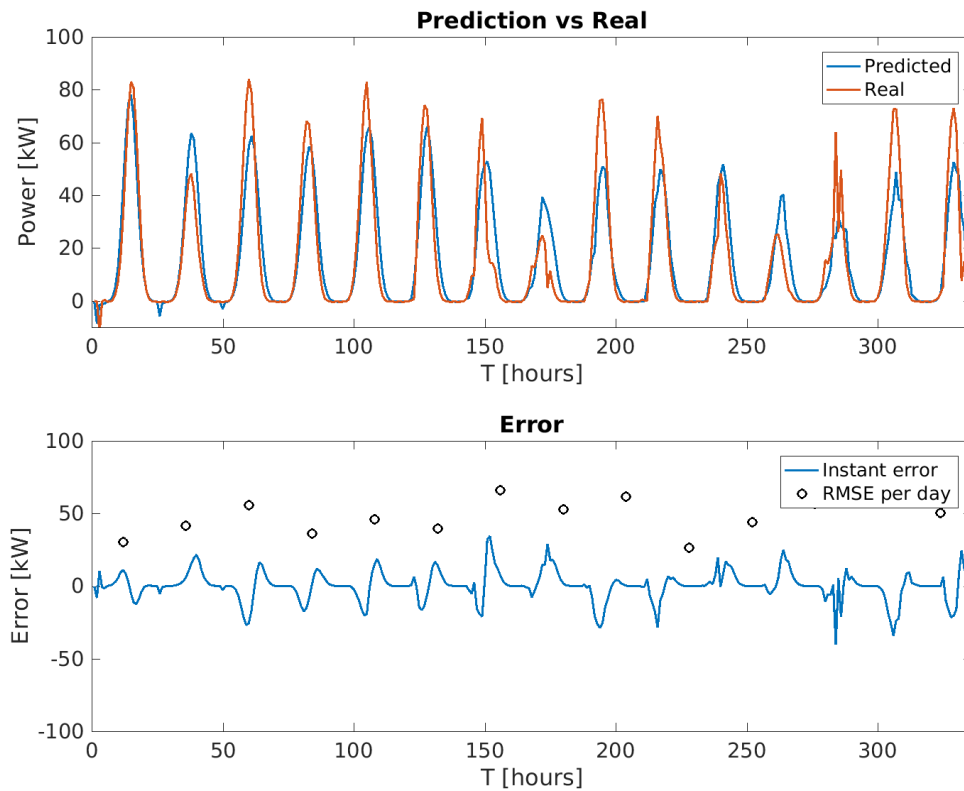


Figura 4.3: 2D filter Prediction accuracy

4.1.4. WCMA

El modelo WCMA hemos observado en 4.4 que opera especialmente bien durante el que consideraremos periodo de estabilidad (Primeros 7 días) con picos de error de hasta el 30%. Cabe destacar que en este periodo el entrenamiento (Periodo en el cual las medias aun no tienen todos los valores para que sean válidas) está aún latente, razón para más para hacer hincapié en este modelo.

Sin embargo, en el segundo periodo, el escenario es diferente. El octavo día comienza con un día ligeramente nublado. Este suceso ocasiona una predicción no válida y es la clase de situaciones que se tiene como objetivo evitar ya que un predictor ideal se habría "preparado" durante la noche para este día nublado.

Posteriormente al día nublado, vuelve a ser soleado para lentamente volver a un día nublado de insuficiente producción y una rápida recuperación. El WCMA es capaz de seguir el rastro de estos "malos" días a costa de predicciones muy optimistas.

Podríamos concluir la calidad de predicción de este modelo como moderada; Capaz de ser usada en producción en algunos casos particulares.

$$\alpha = 0,35$$

$$K = 3$$

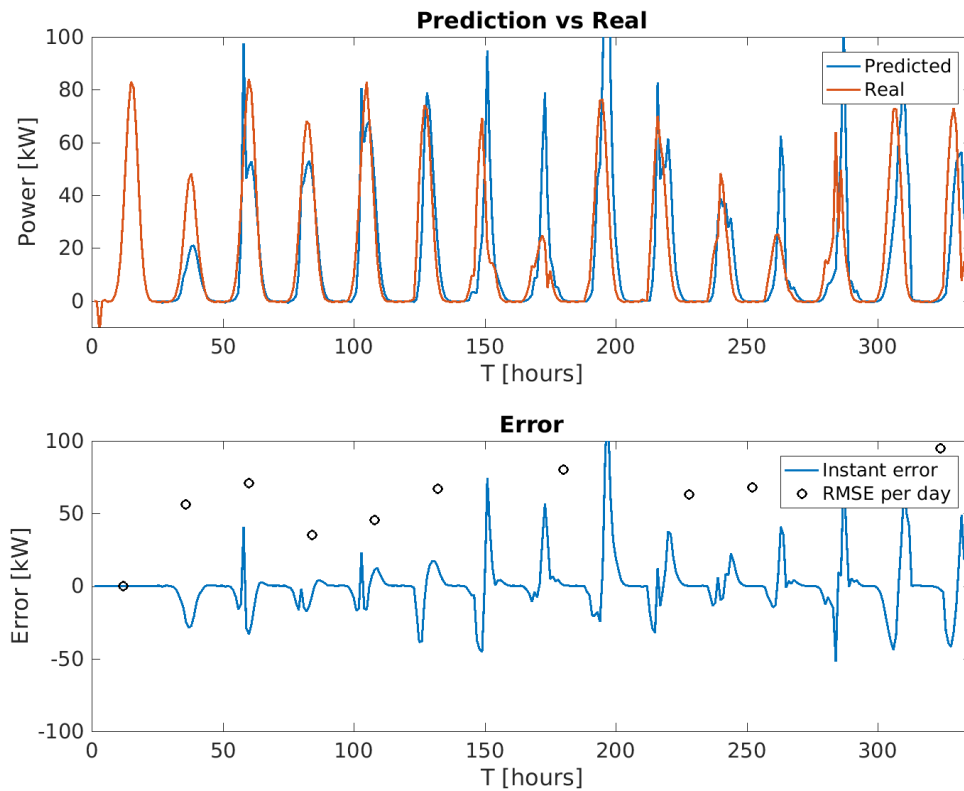


Figura 4.4: WCMA Prediction accuracy

4.1.5. WCMA with PDR

Este modelos de predicción basado en el WCMA y mejorado en el paper[6] con el atributo PDR realmente permite apurar la calidad de la predicción explotando los errores. Más concretamente, regulando el valor predicho con los errores de momentos anteriores.

Es importante definir un valor de α alto para que tenga un peso elevando el valor del momento anterior, pues es el valor que mejor apunta a una buena predicción.

Durante el primer periodo la predicción tiene la misma forma que cualquier otro predictor, sin embargo donde brilla es en los momentos que se salen de lo común, como es el segundo periodo.

En 4.5 vemos que aunque tiene puntos en los que rebasa los valores reales, la calidad de la predicción mejora en función del tiempo proporcionando errores inferiores al 25 %.

$$\alpha = 0,99 \quad s = 0,75 \quad y = 0,25 \quad K = 12 \quad W = 24 \quad Ad = ,60 \quad K = 3$$

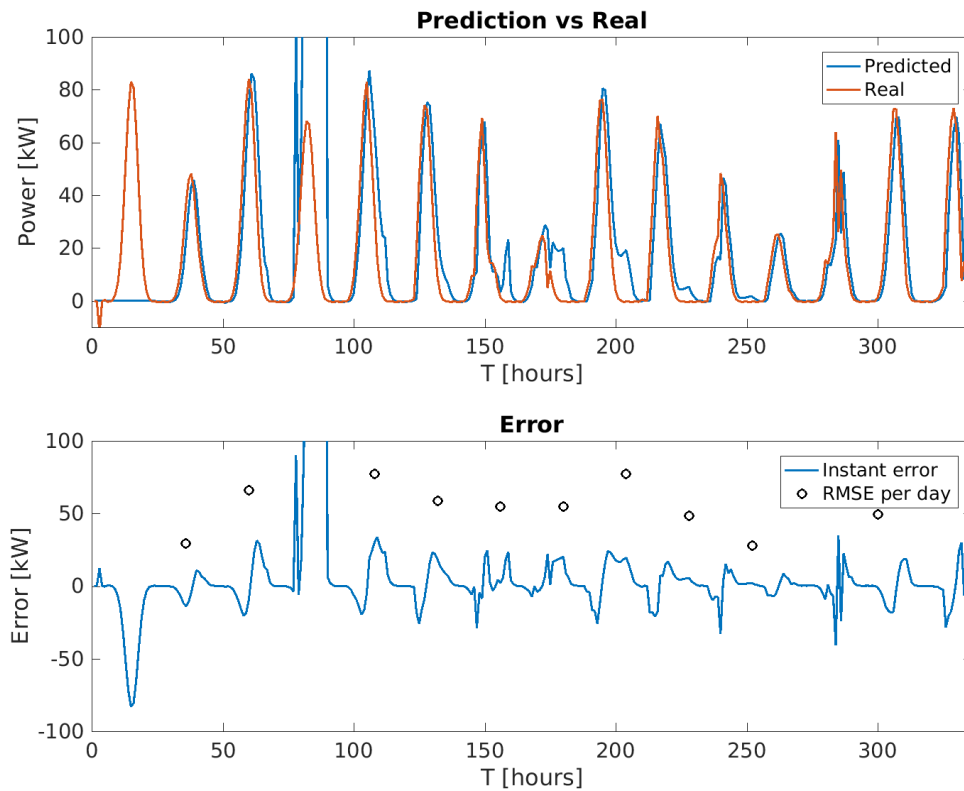


Figura 4.5: WCMA with PDR Prediction accuracy

4.1.6. Red neuronal artificial

El resultado de la red neuronal es especialmente bueno si consideramos valores estables, pero el tiempo de procesado y aprendizaje (El cual deberá ser continuo) son un importante impedimento ya que para que la red neuronal se reajuste con cada nuevo valor, tendremos que realizar un aprendizaje en cada instante.

El resultado de la predicción se puede resumir como una versión no calibrable del Opimal2D ya que toma valores similares (Días anteriores, Instantes anteriores y el valor actual de energía) pero no permite dar peso a unos u otros. Por esto deposita el mismo peso en días anteriores dando como resultado predicciones no acertadas como el primer día nublado, el cual posteriormente toma y realiza otra predicción errónea.

A modo de conclusión, aunque puede proporcionar una predicción válida igual que otros modelos, el coste computacional lo desecha.

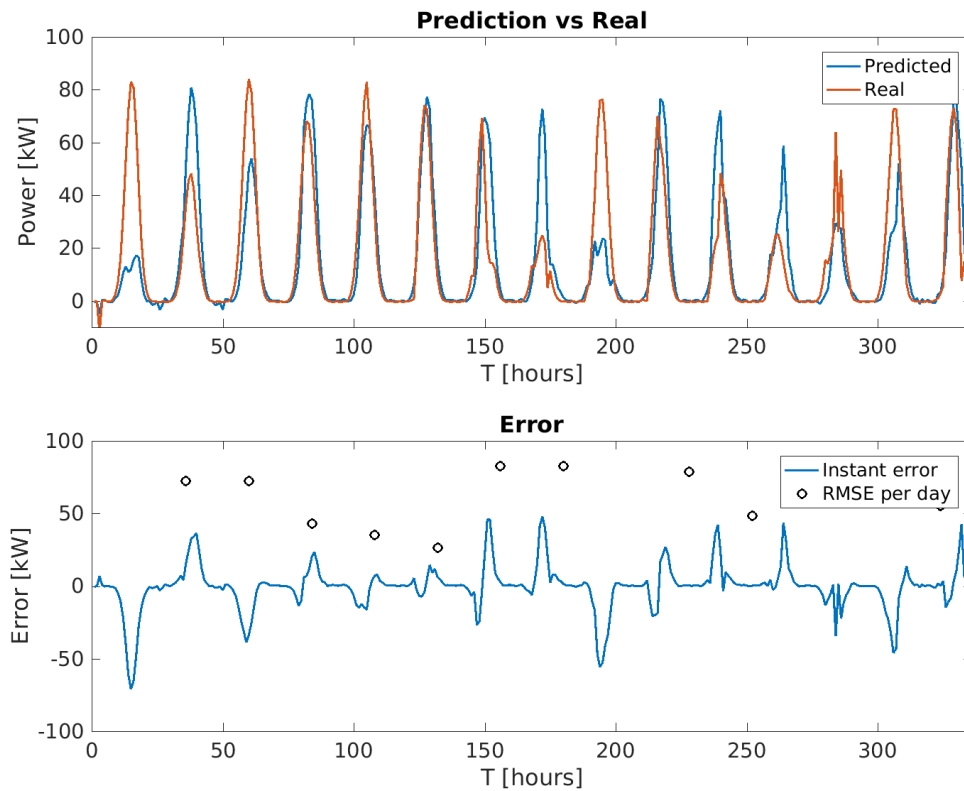


Figura 4.6: neural network Prediction accuracy

4.1.7. N4SID

Este modelo basado en un modelo de estados ha proporcionado un grato resultado. Como se ha observado en otros casos como el de eólica, el N4SID retorna predicciones en forma de armónicos, siendo especialmente notorio en los días con una predicción más complicada.

En la gráfica 4.7 se observa que durante el primer periodo, donde los días son muy similares, la tasa de acierto es muy alta llegando al 80 % (Lo contrario al error; 20 %). En cambio en el segundo periodo, para ser exactos en el primer día nublado y el anterior que es donde al final del día se observa un indicio de nubes, la predicción mantiene la tendencia del histórico dando como salida una predicción de 60kW contra unos 20kW producidos. Este caso es el más complejo de predecir y requiere algo más que explotar el histórico. Continuando con la predicción, se durante los días posteriores se restablece la calidad lentamente para obtener finalmente una predicción pesimista de la última semana con un error máximo del 50 % a favor del valor real y con una media del 25 %.

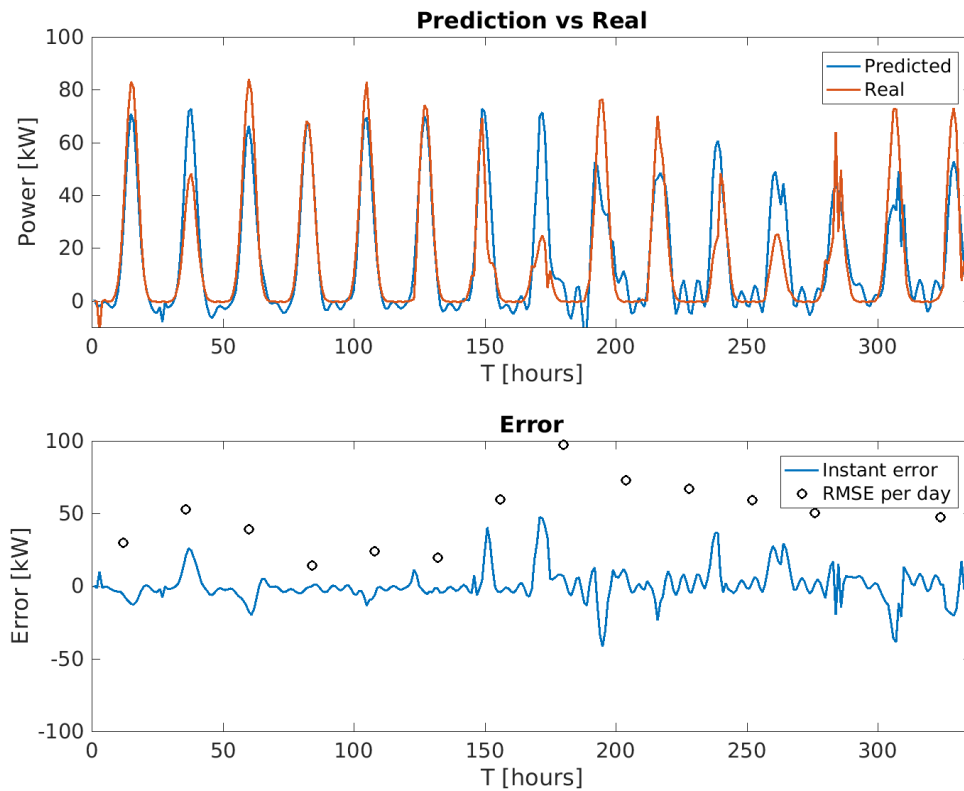


Figura 4.7: N4SID Prediction accuracy

4.2. Eólica: Modelos

Para el estudio de predicción de energía eólica se han probado los siguientes modelos.

- N4SID
- ARIMA/ARMA

En el caso de la predicción eólica uno de los mayores problemas ha sido encontrar los valores adecuados para las pruebas debido a que solo son válidos si la velocidad es válida. Por esto, tras obtener los valores necesarios y comprobar que debido a la localización y el periodo recogido que los valores no son del todo aptos, optaremos por desplazar todos los valores de forma que

4.2.1. Eólica: N4SID

Estos resultados han sido producto de un periodo de entrenamiento entre el 15 y el 30 de mayo y probados con un periodo entre el 1 y el 15 de junio. Los mismos que usaremos para la predicción del modelo ARMA.

La predicción tiene varios valores con picos de potencia predicha que empeoran la calidad de la predicción. Son estos puntos donde haremos hincapié.

Entre la hora 70 y la 100 se observa un pico de potencia predicha de hasta 0.5W, siendo la media de 0.2W. Igual que en este punto, hay más entorno a la hora 160, 280 o 310 y los resultados son similares. La predicción apunta a dichos valores a causa del entrenamiento. En base a esto podemos considerar que la calidad de la predicción en estos puntos es media ya que está altamente influenciada por el entrenamiento, el cual sería ideal que fuese periódico e.g. Cada día o semana.

Dejando de lado los valores anteriores, en puntos donde aunque oscila continuamente podemos considerar estable, la predicción es capaz de ajustarse hasta proporcionar una calidad del 20 % como se observa en 4.8 entre las horas 225 y 275.

Este modelo de predicción con un entrenamiento adecuando podría obtener aún mejores resultados.

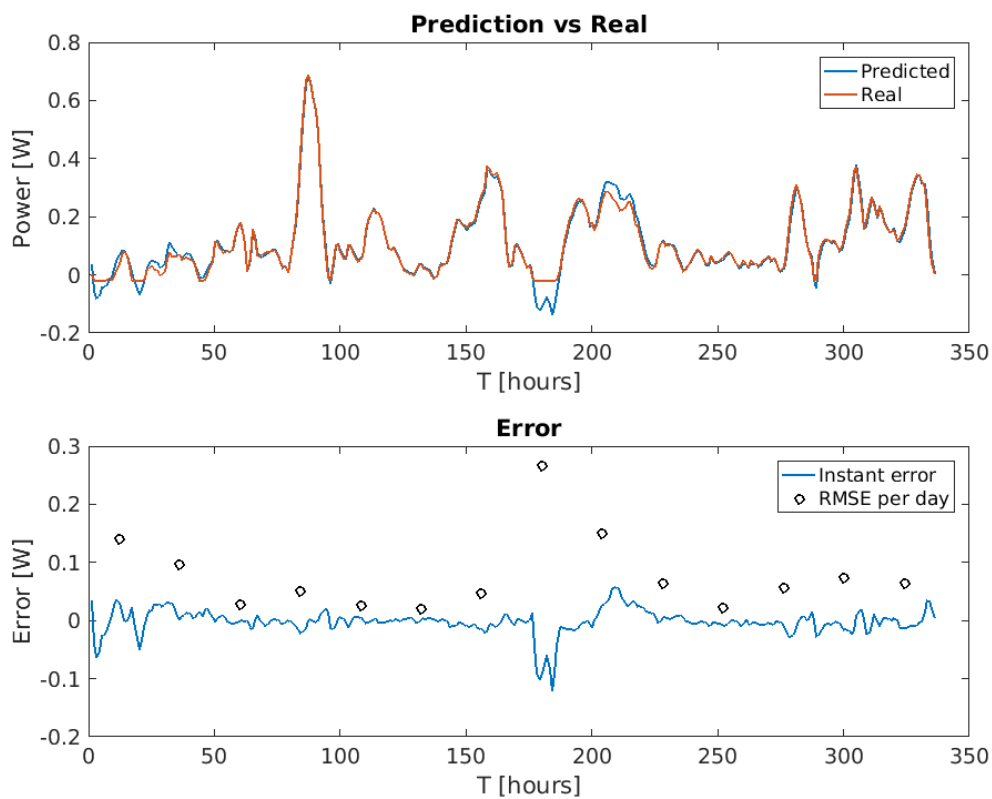


Figura 4.8: N4SID Prediction accuracy

4.2.2. Eólica: ARIMA/ARMA

Para obtener resultados primero tendremos que comprobar si es estacionaria, que en este caso no es así. Por ello continuaremos con un modelo ARMA y la elección de los parámetros p y q . En el caso de solar podríamos obtener resultados donde la longitud sea muy limitada ya que la periodicidad puede interpretar un papel importante. Sin embargo en la eólica donde es más complicado encontrar patrones del mismo tipo, muchas de las veces los valores de p y q deberán

ser altos.

Por esto, los valores escogidos son 10 y 8.

Los resultados obtenidos por este predictor concluyen de forma muy similar al N4SID: Predicción ajustada en periodos de cierta estabilidad con la excepción de que produce cierto suavizado en la predicción resultante.

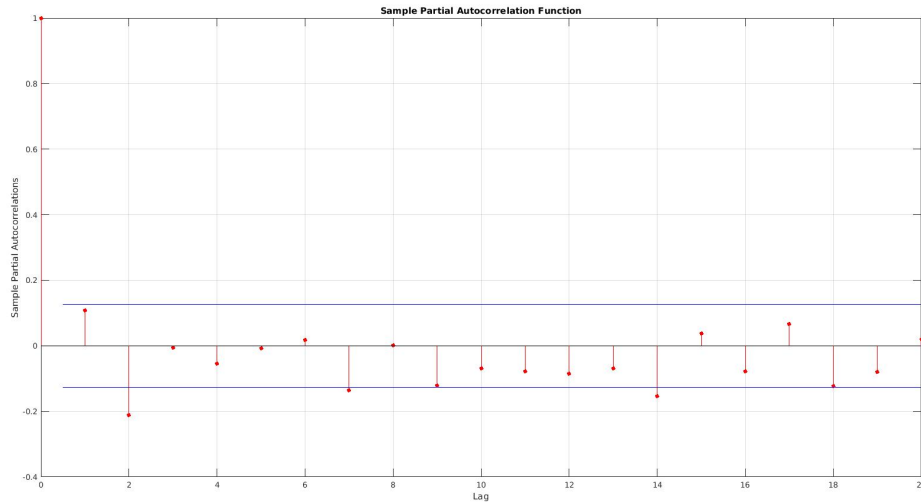


Figura 4.9: N4SID Partial autocorrelation

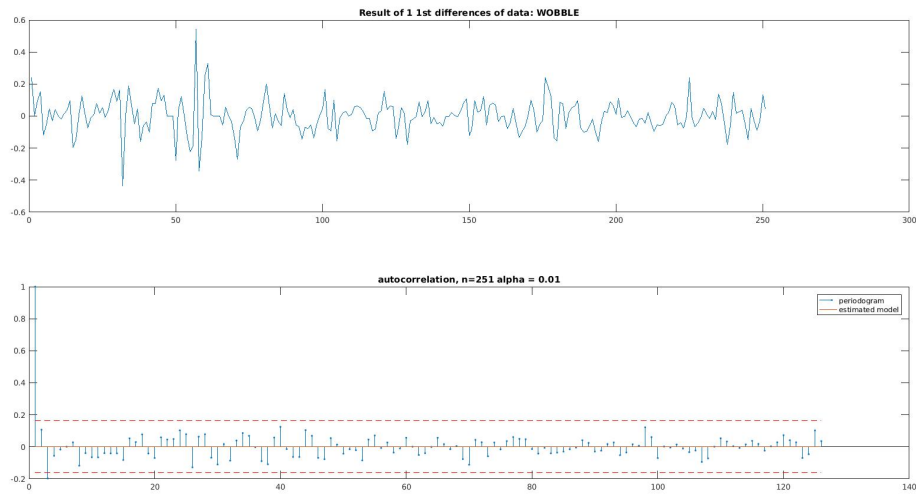


Figura 4.10: N4SID Differentiation

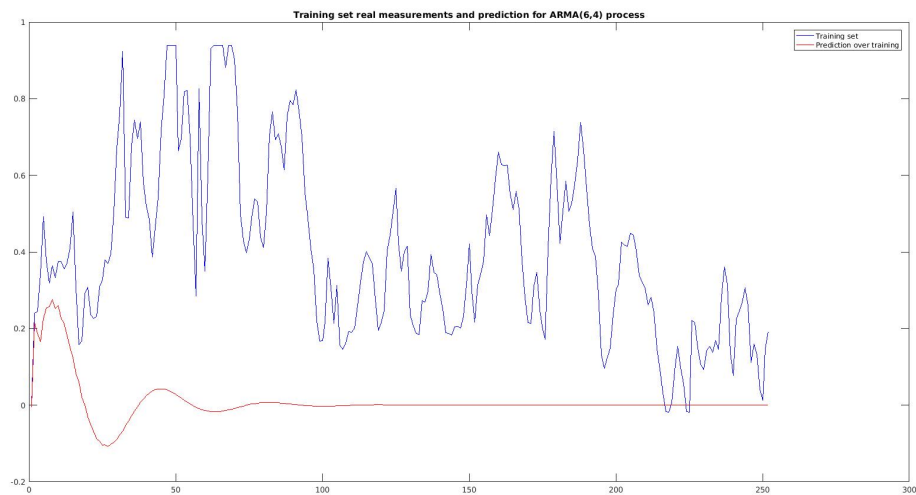


Figura 4.11: N4SID Train and Prediction over Train

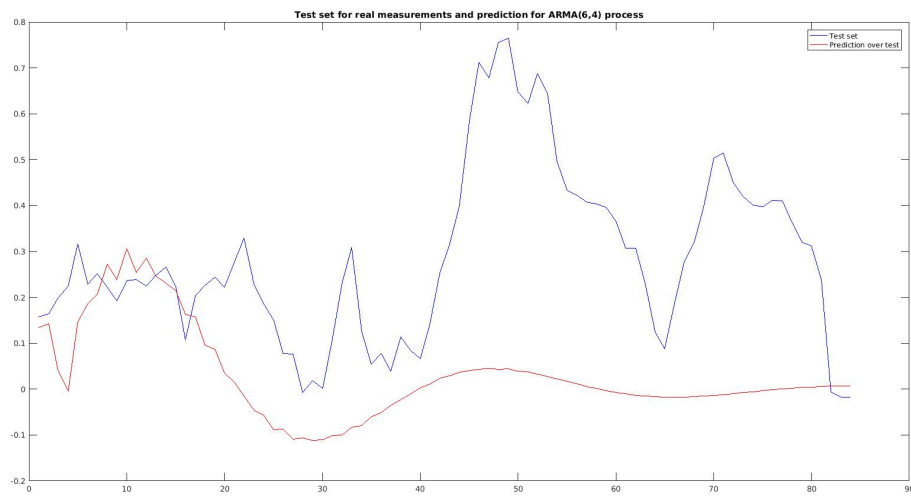


Figura 4.12: N4SID Test and Prediction over Test

Capítulo 5

Conclusión

Es un hecho que las energías renovables están desbancando gradualmente a los combustibles fósiles (Mírese el caso de los coches). Por esto, los sistemas dependientes de no solo a los combustibles fósiles si no también las fuentes convencionales como la red eléctrica barajan incluir en el proceso de adquisición energética energías renovables.

En este caso nos hemos centrado en un contexto de CPD para dicha mejora energética ya que cubre una parte importante del consumo en el sector de las TI que continua en crecimiento. La razón por la que puede combinarse efectivamente los CPDs y las energías renovables es debido a su perpetuidad.

Para llevar esto a cabo es muy recomendable mejorarlo con un sistema de predicción. De esta forma podremos adelantar la producción, y junto a un sistema de almacenamiento, aumentar el rendimiento energético y reducir costes.

Por esto hemos evaluado la eficacia de ciertos modelos de predicción con el objetivo de valorar la validez de estos de acuerdo al enfoque practico en el escenario de los CPDs.

Hemos abordado tanto modelos auto-regresivos como basados en aprendizaje ya que al no requerir conocer los detalles de la planta, simplifican la predicción y permiten generalizar, abstrayéndonos así de esta. De estos hemos extraído tanto las ventajas como los inconvenientes en su aplicabilidad en el citado escenario. En algunos la perpetuidad del consumo es un inconveniente, en otros el nivel de procesamiento es una ventaja al precio de la precisión.

Como conclusión podemos extraer que hay una basta cantidad de maneras de aumentar la eficiencia energética de los CPDs tanto por la parte del consumo como por la parte de la producción, y en este caso y desde el lado de la producción, podemos concluir que los métodos evaluados junto a sus resultados obtenidos permitirán su integración en entornos de simulación con el fin de confirmar su aplicabilidad en un escenario real.

Conclusions

It's a fact that renewable energies are ready to work alongside fossil fuels. For this reason, systems that depends not only on fossil fuels but also conventional sources such as the electricity grid, look for including renewable energies in the process.

In this case we have focused on a context of Data Center for such energy improvement as it

covers a significant part of consumption in the IT sector that continues to grow. The reason why Data Centers and renewable energies can effectively be combined is because of their perpetuity (Power consumption may vary but it will not cease).

To accomplish this it's highly recommended to improve it with a prediction system. In this way we can advance production, and together with a storage system, increase energy efficiency and reduce costs.

For this reason, we have evaluated the efficacy of certain prediction models in order to assess their validity according to the practical approach in the Data Center scenario.

We have approached both self-regressing and learning-based models, since they do not need to know the details of the plant, simplifying the prediction and allowing us to generalize, thus abstracting from it. From these we have extracted both the advantages and the drawbacks in their applicability in the aforementioned scenario. In some algorithms the perpetuity of the consumption is a disadvantage (neural network), in others the level of processing is an advantage even losing precision (Optimal 2D linear prediction filter).

Closing, we can conclude that there is a sufficient number of ways to increase the energy efficiency of the Data Centers both for the consumption part and for the production part, and in this case from the production side, we can conclude that the methods evaluated together with their results will allow their integration in simulation environments in order to confirm their applicability in a real scenario.

Bibliografía

- [1] MARINA ZAPATER ATA TURK JOSE M. MOYA JOSE L. AYALA AYSE K. COSKUN *Dynamic Workload and Cooling Management in High-Efficiency Data Centers*
- [2] D.R. COX *Prediction by exponentially weighted moving averages and related methods, Journal of the Royal Statistical Society, Ser. B 23 (1961) 414–422*
- [3] C. MOSER, L. THIELE, D. BRUNELLI, L. BENINI *Adaptive power management in energy harvesting systems, in: Design, Automation and Test in Europe Conference and Exhibition, 2007*
- [4] FATIH O. HOCAOGLU, OMER N. GEREK, MEHMET KURBAN *Hourly solar radiation forecasting using optimal coefficient 2-D linear filters and feed-forward neural networks*
- [5] J.K. PIORNO C. BERGONZINI, D. ATIENZA, T.S. ROSING *Prediction and management in energy harvested wireless sensor nodes, 2009*
- [6] CARLO BERGONZINI, DAVIDE BRUNELLI, LUCA BENINI *Comparison of energy intake prediction algorithms for systems powered by photovoltaic harvesters*
- [7] MAURIZIO ROSSI, ALESSANDRO TOPPANO, DAVIDE BRUNELLI *Real-time optimization of the battery banks lifetime in Hybrid Residential Electrical Systems*
- [8] PETER VAN OVERSCHEE, BART DE MOOR *N4SID: Subspace Algorithms for the Identification of Combined Deterministic-Stochastic Systems*

Agradecimientos

A todos los que me han ayudado a ser mejor persona, tanto a nivel tecnico como personal.

Autorización de difusión

Autorización para la difusión del Trabajo Fin de Grado y su depósito en el Repositorio Institucional E-Prints Complutense

Los abajo firmantes, alumno y tutores del Trabajo Fin de Grado (TFG) en Ingeniería de Informática de la Facultad de Informática, autorizan a la Universidad Complutense de Madrid (UCM) a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a su autor, el Trabajo Fin de Grado (TFG) cuyos datos se detallan a continuación. Así mismo autorizan a la Universidad Complutense de Madrid a que sea depositado en acceso abierto en el repositorio institucional con el objeto de incrementar la difusión, uso e impacto del TFG en Internet y garantizar su preservación y acceso a largo plazo.

TÍTULO del TFG: Desarrollo de modelos de predicción de la generación de energía eólica y solar para la optimización de Centros de Datos en un entorno de Smart Grid.

Curso académico: 2016/2017

Nombre del Alumno: **Gabriel Galán Casillas.**

Tutores del TFG: **José Luis Ayala Rodrigo y Marina Zapater Sancho**, Departamento de Arquitectura de Computadores y Automática.

Fdo.: Gabriel Galán Casillas

Fdo.: José Luis Ayala Rodrigo

Fdo.: Marina Zapater Sancho