



ITI

INSTITUTO TECNOLÓGICO
DE INFORMÁTICA

Entregable 2.1

INFORME DE TECNOLOGÍAS DE DETECCIÓN DE ANOMALÍAS APLICADAS A LA EFICIENCIA ENERGÉTICA

Línea de I+D En Big/Smart Data



Resumen

El Instituto Tecnológico de Informática cuenta con un Plan de Actividades de carácter no económico (PROMECE) que tiene como objetivo general potenciar las líneas de investigación en las que trabaja el Instituto, enmarcadas dentro del ámbito de las Tecnologías de la Información y las Comunicaciones (TIC).

Dentro de la Línea de I+D en Big/Smart Data, se trabaja en técnicas de análisis de grandes cantidades de datos con objeto de crear modelos predictivos que permitan detectar anomalías con antelación en distintos escenarios y en técnicas de visualización de datos para conseguir una transmisión óptima de los resultados de los análisis hacia los usuarios finales de las aplicaciones.

El presente documento propone técnicas para el análisis de datos con el objetivo de la detección de anomalías en el campo energético. Para ello, en primer lugar, se introducen brevemente las técnicas generales que se implementan y se explica la arquitectura del sistema. A continuación, se detallan las técnicas aplicadas dentro de los casos de uso contemplados en el proyecto (descripción del estado general, predicción, detección de anomalías). Finalmente, se diseña una herramienta genérica de visualización de datos que apoye a los usuarios en la toma de decisiones.

Abstract

Instituto Tecnológico de Informática has a non-economic Activities Plan (PROMECE) whose general objective is to strengthen the research lines in which the Institute works, within the scope of Information and Communication Technologies (ICT).

Within the Big/Smart Data R&D Line, we work on techniques to analyse large amounts of data, in order to obtain predictive models to detect anomalies in several scenarios. Another important topic is the visualization techniques to ease the analytics results transmission to the final users.

This document proposes some analytics techniques to detect anomalies on energy data. First, we briefly introduce the applied techniques and the system architecture. Then, we explain how we adapt those techniques to the proposed use cases (general status description, prediction, anomalies detection). Finally, we design a general visualization tool for decision-making.



Este documento está bajo una [Licencia Creative Commons Atribución-NoComercial-SinDerivar 4.0 Internacional](https://creativecommons.org/licenses/by-nc-nd/4.0/). Se permite libremente copiar, distribuir y comunicar públicamente esta obra siempre y cuando se reconozca la autoría y no se use para fines comerciales. No se puede alterar, transformar o generar una obra derivada a partir de esta obra.

Los derechos de autor de todas las marcas, nombres comerciales, marcas registradas, logos e imágenes pertenecen a sus respectivos propietarios.

Tabla de Contenidos

1. Introducción	6
1.1. <i>Objetivos del paquete de trabajo 2.....</i>	<i>6</i>
1.2. <i>Objetivo del presente documento.....</i>	<i>6</i>
2. Sistemas estadísticos sensibles al contexto aplicados a tareas de mantenimiento	6
3. Arquitectura del sistema	7
3.1. <i>Arquitectura Big Data</i>	<i>8</i>
4. Análisis e implementación de modelos de detección de anomalías aplicados a la eficiencia energética.....	12
4.1. <i>Corpus de datos</i>	<i>12</i>
4.2. <i>Preparación de los datos.....</i>	<i>13</i>
4.3. <i>Descripción del estado energético</i>	<i>16</i>
4.4. <i>Predicción.....</i>	<i>17</i>
4.5. <i>Detección de anomalías.....</i>	<i>21</i>
5. Herramienta de visualización.....	23
6. Referencias bibliográficas	27

Tabla de Figuras

Figura 1. Arquitectura general del sistema.	8
Figura 2. Arquitectura Big Data basada en Cloudera.	9
Figura 3. Inicialización del clúster.	10
Figura 4. Clúster con baja carga.	11
Figura 5. Clúster con alta carga.	11
Figura 6. Modelo del contador energético instalado en cada edificio.	13
Figura 7. Modelo del analizador instalado en cada línea.	13
Figura 8. Relación entre activa (P), reactiva (Q), aparente (S) y ángulo de fase (ϕ).	13
Figura 9. Mapa de calor de la potencia activa por metro cuadrado para cada edificio.	17
Figura 10. Red neuronal con capas de entrada, ocultas y de salida.	18
Figura 11. Predicción de activa y sus valores reales para un edificio.	19
Figura 12. Anomalías detectadas en un edificio a lo largo del tiempo.	22
Figura 13. Mockup de Dashboard.	23
Figura 14. Mockup de Eficiencia Energética.	24
Figura 15. Mockup de listado de sensores.	25
Figura 16. Resto de mockups de la herramienta de visualización.	27

1. Introducción

1.1. Objetivos del paquete de trabajo 2

El presente paquete de trabajo se centra en la aplicación de técnicas de predicción de anomalías aplicadas al campo de la eficiencia energética. En concreto, se plantea el contexto de edificios con dispositivos (sensores) que recogen mediciones continuamente. Se propone la aplicación de técnicas de aprendizaje automático para analizar dichos datos para la detección de anomalías desde el punto de vista energético. De este modo, se podrán identificar posibles problemas que repercutan en la eficiencia energética de los edificios.

Para conseguir este objetivo final, se han detectado los siguientes objetivos intermedios o casos de uso:

- Análisis descriptivo del estado actual.
- Sistema de predicción.
- Sistema de detección de anomalías.

A continuación se incluye una descripción más detallada de los sistemas a implementar para dar solución a cada uno de los objetivos señalados.

1.2. Objetivo del presente documento

Este documento resume las técnicas que se han aplicado para el análisis de los datos con el objetivo de la detección de anomalías en el campo energético.

Para ello, en primer lugar se introducen brevemente las técnicas generales que se implementan y se explica la arquitectura del sistema. A continuación, se detallan las técnicas aplicadas dentro de los casos de uso contemplados en el proyecto (descripción del estado general, predicción, detección de anomalías), haciendo hincapié en las técnicas empleadas y los resultados obtenidos. Finalmente, se muestran las tareas relacionadas con el diseño de una herramienta de visualización de datos que apoye a los usuarios en la toma de decisiones.

2. Sistemas estadísticos sensibles al contexto aplicados a tareas de mantenimiento

El sistema que se plantea en este trabajo se enmarca dentro de los denominados “sensibles al contexto”, que tratan de interpretar su entorno con un objetivo determinado. En general, la información contextual se obtiene a partir de medidas tomadas por sensores, que permiten al sistema tener una cierta percepción de su entorno. Así pues, el objetivo principal del modelado del contexto consiste en identificar cambios relevantes en su entorno para actuar acordeamente [7]. Un modelo de contexto debería:

- 1) Identificar precisamente la información necesaria,
- 2) Identificar y representar las relaciones entre la información capturada,

- 3) evaluar la adecuación de los sistemas de razonamiento que se emplearán, y
- 4) considerar la escalabilidad.

En este trabajo se aplican modelos sensibles al contexto basados en técnicas estadísticas para la detección de anomalías.

Tradicionalmente, los sistemas de mantenimiento se han clasificado en:

- Mantenimiento correctivo. Se detecta cuando un sensor falla, de manera que se conoce cuándo se debe llevar a cabo una reparación, reconfiguración o sustitución [8][9][10].
- Mantenimiento preventivo. Se intenta actuar antes de que falle el dispositivo o necesite ser limpiado, reajustado o reparado. Si se conoce la distribución de probabilidad del tiempo de vida, el sistema puede lanzar una alerta antes de que el dispositivo necesite mantenimiento. Dicha distribución debería ser idealmente proporcionada en la especificación del dispositivo. En otro caso, se podría estimar a partir de datos históricos de dispositivos similares [11][12].
- Mantenimiento predictivo. Intenta anticipar el fallo de un dispositivo provocado por razones diferentes a su distribución de tiempo de vida. Estos modelos son más complejos que los anteriores, ya que deben tener en cuenta variables independientes medidas en el contexto del dispositivo. Las técnicas de análisis estadístico multivariante pueden buscar contextos poco usuales dependientes de diversas medidas (datos internos o externos tomados en un instante dado) [13][14].

Se pueden encontrar multitud de casos reales que aplican estas técnicas en [15][16][17].

El objetivo de este proyecto se enmarcaría pues en el denominado “mantenimiento correctivo”, ya que se plantea la detección de anomalías una vez ya producidas.

En cuanto a las técnicas estadísticas que se plantea aplicar para implementar dicho mantenimiento correctivo, se contemplan tanto unidimensionales como multidimensionales. En ambos casos, se basarán en técnicas basadas en Deep learning. Se proporcionan más detalles en las secciones correspondientes (4.4 y 4.5).

3. Arquitectura del sistema

A continuación, se describe la arquitectura del sistema en el que se ejecutan las técnicas de análisis que detallan en el presente documento.

En resumen, el sistema recoge datos provenientes de los sensores desplegados, los analiza y ofrece las conclusiones de dichos análisis, como se muestra en la siguiente figura.

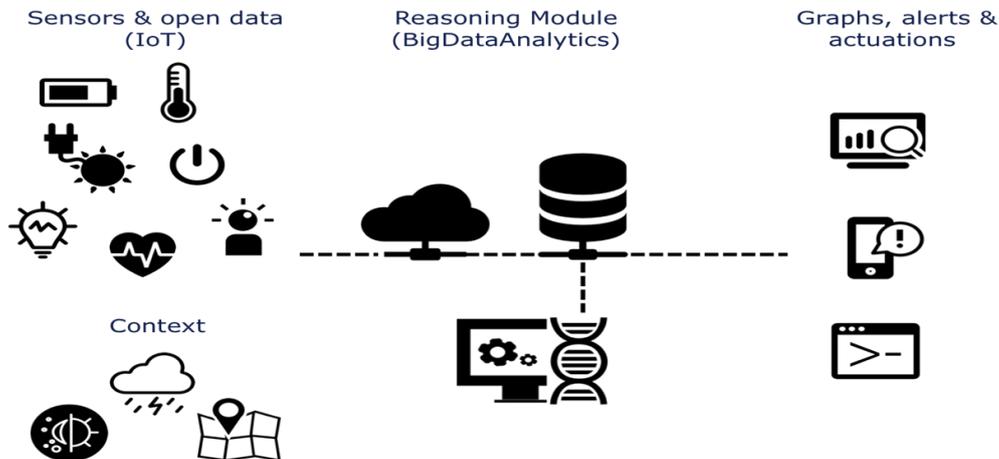


Figura 1. Arquitectura general del sistema.

3.1. Arquitectura Big Data

Los sensores desplegados en el edificio generarán una gran cantidad de datos heterogéneos de manera continua, que deben ser recibidos y procesados por el sistema. Así pues, se ha optado por una arquitectura basada en Big Data. De esta manera, se asegura la escalabilidad, el procesamiento de grandes cantidades de datos, y el análisis de nuevos datos en tiempo reducido.

En resumen, nuestro sistema se aloja en un clúster de cálculo compuesto por un nodo maestro y varios nodos trabajadores. El hardware está virtualizado con OpenStack, que permite un despliegue sencillo de nuevos nodos trabajadores (actualmente, hasta 6000) dependiendo de las necesidades de almacenamiento y procesamiento. En resumen, estos nodos se configuran como sigue:

- Nodo maestro: 8 VCPUs, 16GB RAM, 160GB HD
- Nodos trabajadores: 4 VCPUs, 6GB RAM, 80GB HD

La instalación general de cada nodo incluye:

- HDFS-Hadoop 2.6. Hadoop Distributed File System (HDFS) es un sistema de ficheros basado en Java que proporciona almacenamiento escalable y robusto en un clúster de servidores.
- YARN. YARN es el centro de la arquitectura de Hadoop que permite múltiples motores de procesamiento de datos como SQL interactivo, streaming en tiempo real, análisis de datos y procesamiento en lotes para manejar los datos almacenados en una única plataforma.

- MapReduce. MapReduce es un modelo de programación y su implementación asociada para el procesamiento y generación de grandes conjuntos de datos con algoritmos paralelos distribuidos en un clúster.
- Spark 1.3.1. Spark procesa datos en memoria, lo que permite la implementación de algoritmos iterativos rápidos para análisis avanzado, como por ejemplo la clasificación de conjuntos de datos.
- Sqoop. Apache Sqoop importa y exporta (Ingress/Egress) datos entre Hadoop y sistemas estructurados de almacenamiento como bases de datos relacionales.
- Hive 1.1. Hive está considerado el estándar de facto para peticiones SQL interactivas sobre grandes cantidades de datos en Hadoop.
- H2O 3.x. H2O es un software de código abierto para el análisis big data. Se emplea para la exploración y análisis de conjuntos de datos almacenados en sistemas de computación en la nube, en HDFS o en sistemas operativos convencionales.
- Cloudera. CDH (Cloudera Distribution Including Apache Hadoop) contiene los componentes principales de Hadoop para ofrecer procesamiento de grandes cantidades de datos de manera distribuida, escalable y robusta (ver Figura 2). También incluye otros componentes relacionados con la seguridad, la alta disponibilidad y la integración con hardware y software de terceros.

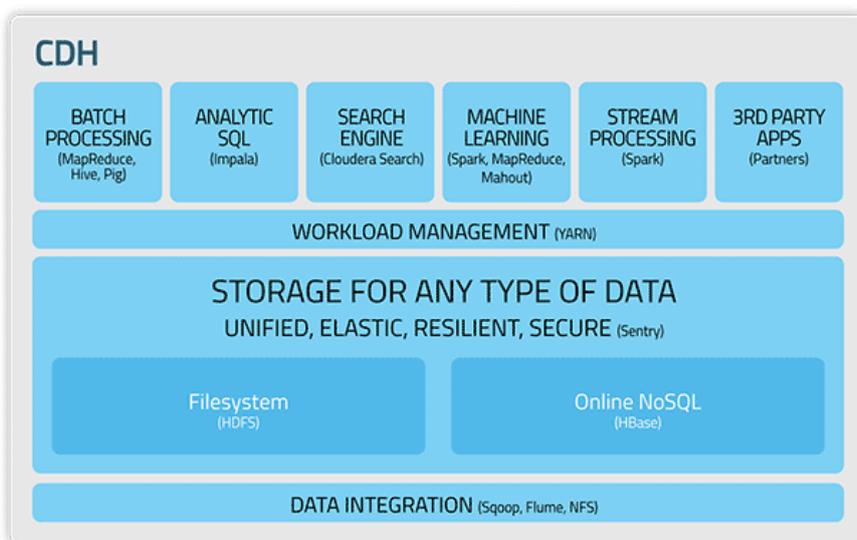


Figura 2. Arquitectura Big Data basada en Cloudera¹.

¹ Fuente: www.cloudera.com

Evaluación

Para evaluar la escalabilidad de esta arquitectura, se ha diseñado un algoritmo similar a los que implementarán las técnicas estadísticas en nuestro sistema. Se ha ejecutado en el clúster para comprobar cómo escala a medida que se cargan y procesan datos. La prueba consiste en los siguientes pasos:

- El algoritmo toma datos de varias fuentes, aprende una función para interpolar datos faltantes (uno de los procedimientos de data wrangling, como se ve en la sección correspondiente).
- El algoritmo en Python utiliza Spark in Memory y librerías Pandas.
- Se lanza Spark en el cluster del servidor, utilizando YARN.
- Subimos los datos en HDFS utilizando Hue, por lo que los datos se distribuyen entre los nodos.
- Ejecutamos el script Python mediante el comando spark-submit.
- Según lo va necesitando el algoritmo, los datos se cargan en memoria distribuida. Del mismo modo, la computación también se distribuye entre los nodos.
- En las siguientes figuras (Figura 3, Figura 4, Figura 5) se muestra cómo el clúster escala de manera transparente para el usuario y utiliza más recursos según la necesidad del algoritmo.
- El programa finaliza correctamente, interpolando con éxito los valores faltantes.

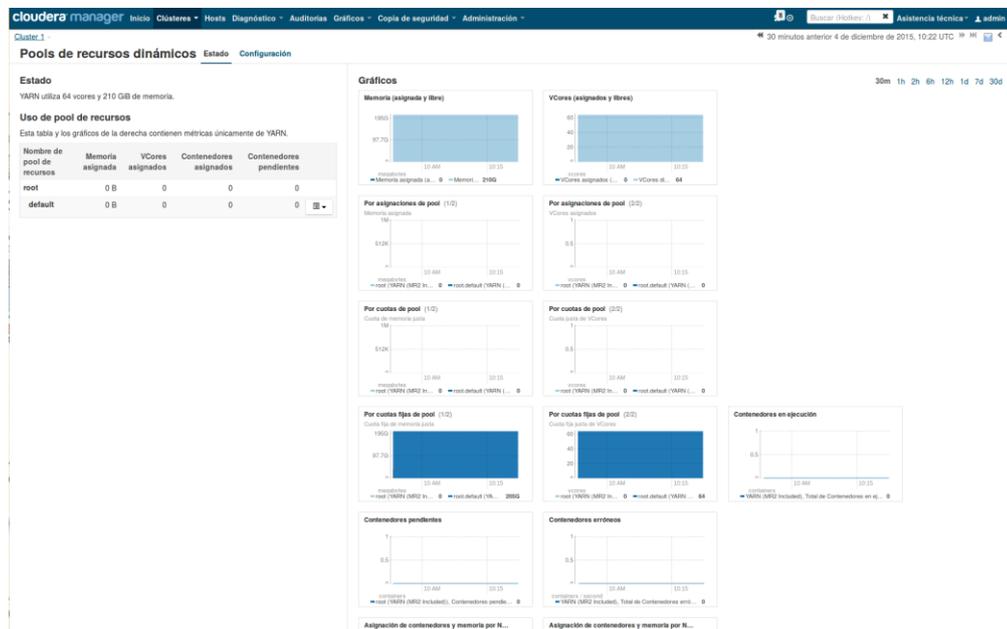


Figura 3. Inicialización del clúster.

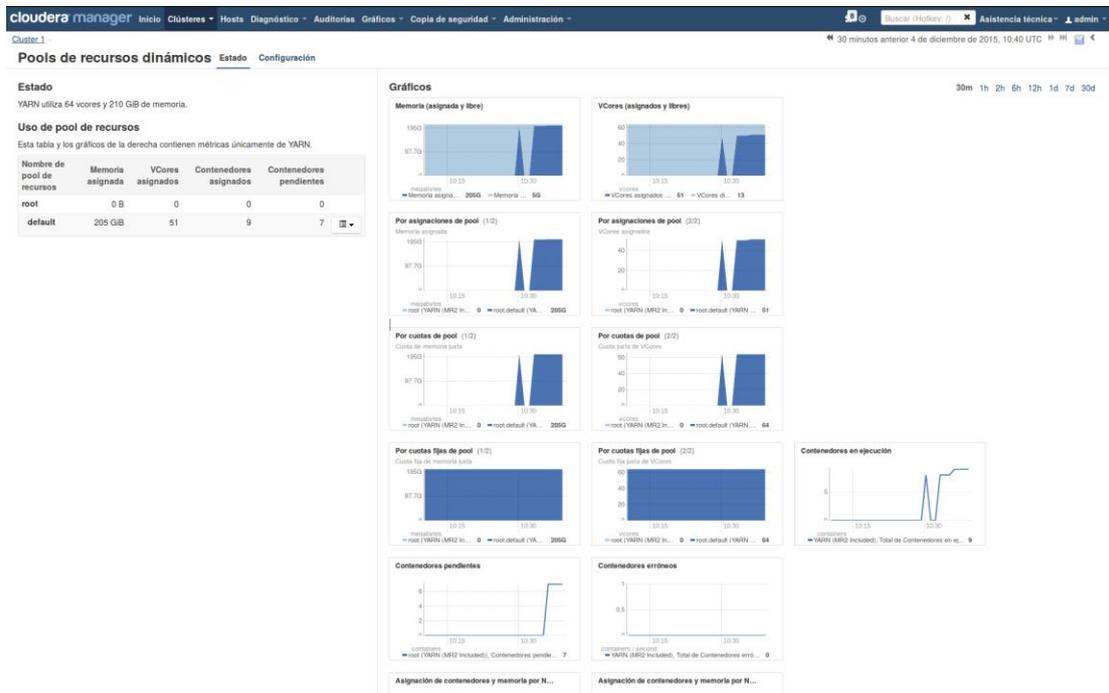


Figura 4. Clúster con baja carga.

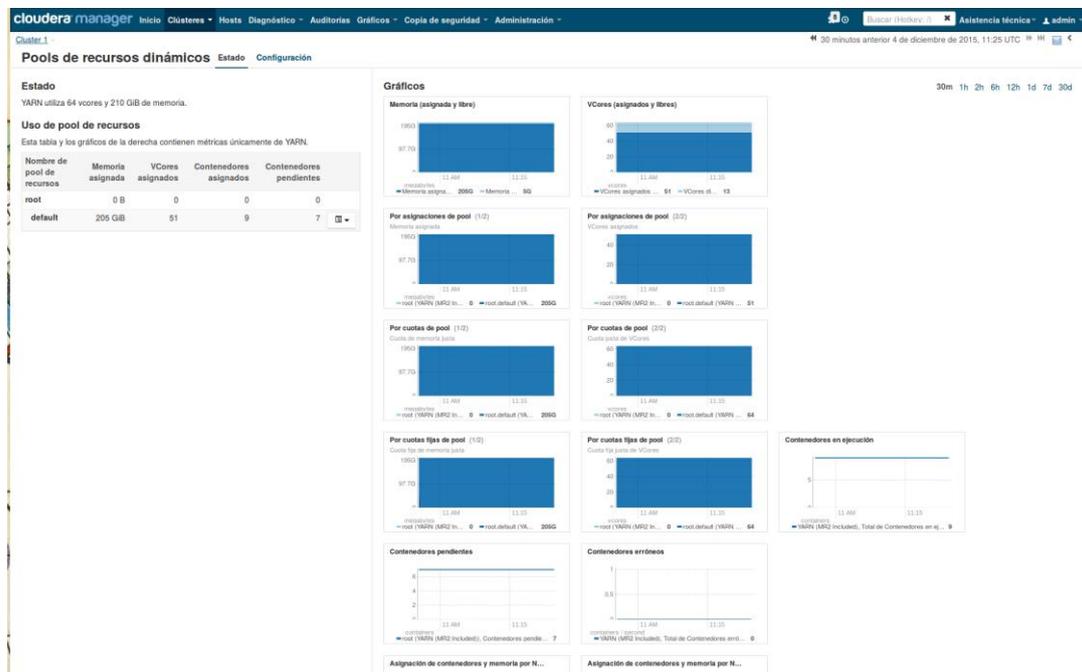


Figura 5. Clúster con alta carga.

4. Análisis e implementación de modelos de detección de anomalías aplicados a la eficiencia energética

En esta sección se detallan las técnicas que se han investigado e implementado para la detección de anomalías en el campo de la eficiencia energética.

El trabajo se ha centrado en tres casos de uso que se han considerado interesantes desde la perspectiva de la eficiencia energética: descripción del estado energético presente, predicción de valores futuros, y detección de anomalías.

Para ello, se han estudiado diversas técnicas de aprendizaje automático (machine learning) susceptibles de ser implementadas en una arquitectura Big Data como la descrita en secciones anteriores.

4.1. Corpus de datos

Las técnicas basadas en aprendizaje automático que se aplican en este trabajo son capaces de aprender las características de los datos de ejemplo (corpus de muestras) que se le proporcionan. Dicho conocimiento se plasma en una serie de modelos estadísticos en una fase previa de aprendizaje (entrenamiento) que son capaces de aprender las relaciones existentes entre una serie de variables explicativas y una o varias variables objetivo. Una vez generados, estos modelos se pueden aplicar sobre nuevas muestras de datos para analizar sus características con respecto a los empleados en la fase de entrenamiento, o para realizar predicciones sobre su comportamiento.

Así pues, el entrenamiento de los modelos constituye una fase esencial en cualquier aplicación de machine learning. Es necesario seleccionar un conjunto de muestras significativas (corpus), similares a los datos que se pretenden analizar cuando el sistema se encuentre “en producción”.

Para el presente trabajo, se ha seleccionado un corpus de medidas energéticas tomadas en edificios. En concreto, cada uno de los 378 edificios instala un contador inteligente (modelo Landis&Gyr ZMD410CTSAT en Figura 6), con el que se ha tomado un año de mediciones mensuales de potencia activa, carga inductiva y carga capacitiva. Además, algunos de los edificios (26), tienen analizadores para cada una de sus líneas (general, climatización, hornos, iluminación y frío industrial, ver Figura 7) que cada 6 horas proporcionan la carga reactiva inductiva, la potencia activa y el cos (ϕ) (ver Figura 8). Finalmente, se conoce información descriptiva adicional de cada edificio, como su tamaño, localización, tipo, nombre del director y del encargado de mantenimiento.



Figura 6. Modelo del contador energético instalado en cada edificio.



Figura 7. Modelo del analizador instalado en cada línea.

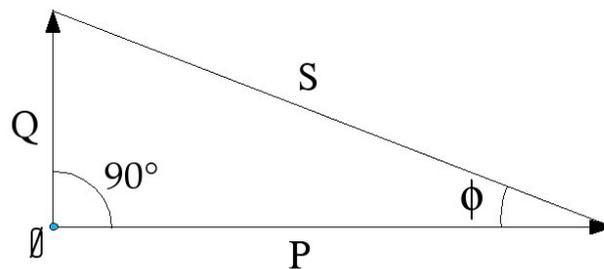


Figura 8. Relación entre activa (P), reactiva (Q), aparente (S) y ángulo de fase (ϕ)².

4.2. Preparación de los datos

Los datos deben ser preparados para la adecuada aplicación de las técnicas de aprendizaje automático. Este proceso transforma los datos “en crudo” homogeneizándolos y eliminando inconsistencias para facilitar el entrenamiento de los modelos estadísticos. En general, la preparación de los datos (data wrangling o munging) debería contemplar al menos los siguientes pasos:

² Fuente: Wikipedia.

- **Integración.** Conversión de los datos en crudo de tal manera que:
 - Cada muestra del corpus (usualmente, una fila en un fichero o base de datos) se corresponda con un individuo de la población estadística bajo estudio.
 - Cada muestra tenga un conjunto de características o variables (usualmente, columnas en un fichero o base de datos), con lo que se dispondrá de individuos con características multidimensionales sobre las que se establecerán las relaciones pertinentes.

En el caso del corpus de datos considerado en este trabajo, los datos en crudo son ficheros csv provenientes del volcado de una base de datos. Se han integrado en ficheros en los que cada muestra es un conjunto de mediciones realizadas en un edificio al mismo tiempo, añadiendo la información descriptiva del edificio. Así, cada muestra se compone de un conjunto de características como las mediciones de un sensor (por ejemplo, el consumo de energía activa representada en un número racional), una fecha, una categoría numérica para conocer el tipo de edificio, las coordenadas geográficas del edificio, un código correspondiente al encargado de mantenimiento, etc.

- **Alineamiento temporal.**

El anterior proceso de integración combina datos provenientes de fuentes diversas. Habitualmente, esta combinación se realiza de acuerdo a la dimensión temporal, es decir, agrupando en un individuo todas las mediciones obtenidas en un momento dado.

Dicho alineamiento debe tener en cuenta varios aspectos, especialmente cuando las frecuencias de muestreo son diferentes para cada medida. Además, algunas técnicas estadísticas (como las series temporales) necesitan que la frecuencia de muestreo sea constante, esto es, que tiempo transcurrido entre cada par de medidas consecutivas sea siempre el mismo. En aquellos casos en los que dicha frecuencia no se mantenga de manera fija, se pueden aplicar técnicas de agregación de datos cada cierto tiempo fijo o ciclo de trabajo específico.

El conjunto de datos utilizado en este trabajo se encontraba alineado temporalmente, pues se trataba de datos recogidos en periodos concretos de tiempo preestablecidos inicialmente.

- **Normalización.**

Los datos pueden ser continuos (p.ej. temperatura), discretos (p.ej. el número de personas en una habitación), o cualitativos (p.ej. si una puerta está abierta o no). Puede ser conveniente estandarizar los datos cuantitativos para que las características sean comparables entre sí independientemente de sus unidades.

Algunas técnicas de aprendizaje automático llevan a cabo una estandarización inicial, que habitualmente elimina las dimensiones de las variables en estudio. Éste es el caso de las técnicas aplicadas en este trabajo, por lo que no ha sido necesaria una normalización previa de los datos.

- **Datos anómalos.**

Los datos anómalos son aquéllos que se apartan del resto, debido en muchas ocasiones a una mala introducción por fallos del aparato de medida o algún tipo de anomalía que no debería de producirse dentro de un régimen estacionario. La lejanía de estos datos respecto a lo considerado como normal provoca que los estimadores de los modelos varíen enormemente, lo cual dificulta la obtención de los modelos. Esta problemática justifica una fase previa de análisis de dichos datos, necesaria para localizar los datos anómalos y eliminarlos antes del entrenamiento. De esta manera, los modelos estadísticos se podrán entrenar con mayor precisión, pues aprenderán lo que se considera poblacionalmente correcto, necesitando menor cantidad de muestras de ejemplo.

En el caso del corpus de datos seleccionado para este trabajo, este paso no ha sido necesario.

- **Datos faltantes.**

En general, los algoritmos de aprendizaje necesitan que cada muestra tenga el mismo número de características. Sin embargo, es habitual que haya características de valor desconocido, especialmente cuando se ha hecho un alineamiento temporal y las frecuencias de muestreo no coinciden entre las medidas. También se puede deber a errores en la comunicación, ficheros incompletos, sensores estropeados, etc.

En estos casos, los datos faltantes deberían simularse de acuerdo a su correlación con el resto de características para el individuo concreto, o interpolándolos a partir de sus valores anteriores. Algunos algoritmos permiten el marcado de los datos faltantes con un carácter especial de manera que se pueden tener en cuenta en la fase de aprendizaje, como es el caso de la técnica *drop out* del *Deep learning*.

No había datos faltantes en el conjunto seleccionado para este trabajo.

- **Maldición de la dimensionalidad.**

Una elevada dimensionalidad en los individuos produce una necesidad de incrementar el número de muestras de aprendizaje para poder llenar poblacionalmente el espacio creado al introducir cada dimensión. Este incremento en el número de muestras necesarias tras la introducción de una nueva dimensión es exponencial, ya que cada nueva dimensión amplía el espacio disponible de manera multiplicativa. La habitual dificultad para conseguir un número suficiente de muestras provoca la conocida como maldición de la dimensionalidad, que consiste básicamente en que nuestro espacio está prácticamente vacío o con un número insuficiente de muestras en algunos puntos como para poder extraer conclusiones estadísticas de dichas zonas. Para solucionarlo,

existen técnicas que permiten la reducción de dichas dimensiones, bien mediante la selección de las características originales que mejor nos dirijan hacia el objetivo buscado, bien mediante la proyección de las características originales en un nuevo espacio de menor dimensionalidad.

En el corpus empleado en este trabajo, la cantidad de características utilizadas no ha sido excesivamente elevada, por lo que no ha sido necesaria la aplicación de este tipo de técnicas.

- **Multicorrelación.**

La multicorrelación entre las variables explicativas es otro de los problemas que pueden surgir en la obtención de modelos. Cuando las variables no son independientes, unas ocultan o incrementan el efecto de otras, por lo que los modelos pueden no extraer relaciones significativas existentes en los datos, o por el contrario, aprender relaciones significativas inexistentes. La solución de esta problemática, como sucedía con la maldición de la dimensionalidad, se suele resolver mediante la aplicación de técnicas de selección de características o proyección en espacios de menor dimensionalidad.

En los modelos presentados no se han aplicado técnicas que resuelvan dicha problemática, quedando su posible aplicación para estudios posteriores de mejora de los modelos ya existentes.

- **Transformación de variables.**

En muchas ocasiones, las variables originales no tienen las relaciones lineales apropiadas para los objetivos buscados. Sin embargo, ciertas transformaciones de éstas sí que pueden llegar a encontrar relaciones lineales apropiadas. En otras ocasiones, las variables simples no son los factores más relevantes para explicar un objetivo concreto, sino la interacción entre varias de estas variables. Para poder encontrar estas relaciones relevantes, es necesaria la aplicación de técnicas de transformación que permitan encontrar dichas más allá de las variables originales.

En los modelos planteados aquí no se han aplicado dichas transformaciones, quedando pendiente su uso en estudios posteriores de mejora de los modelos ya existentes.

4.3. Descripción del estado energético

Previamente al planteamiento del sistema de detección de anomalías, es aconsejable una primera aproximación puramente descriptiva al problema de la eficiencia energética. De este modo, se puede conseguir una idea clara del estado real de las infraestructuras y detectar posibles anomalías simplemente comparando las mediciones realizadas en diferentes edificios. Así, a partir de este análisis inicial, se pueden establecer medidas correctoras aprendiendo de la experiencia de los edificios que consiguen mejores mediciones y aplicándola al resto.

Tras el análisis del conjunto de datos empleado en este trabajo, se han elegido tres criterios útiles para establecer un ranking según sus consumos de activa, inductiva y capacitiva

- Planificar costes futuros.
- Detección de anomalías al comparar el consumo real con el predicho o esperado para un contexto dado.

Para la implementación de este sistema de predicción, se propone el uso de series temporales, técnica que analiza el comportamiento de valores pasados para inferir valores futuros.

En un entorno big data como el utilizado en este trabajo, las series temporales se pueden implementar mediante técnicas de Deep learning, similares a redes neuronales (Figura 10) [1]. De esta manera, el sistema será capaz de estimar y emplear estos modelos en tiempo razonable independientemente de la cantidad de datos a considerar.

Las técnicas de Deep learning han demostrado su eficacia en múltiples áreas (visión por computador, reconocimiento de imágenes y voz, etc.) [1]. Estas técnicas están experimentando un renacimiento debido a la disponibilidad de mayor capacidad de cómputo, mejores algoritmos y más datos.

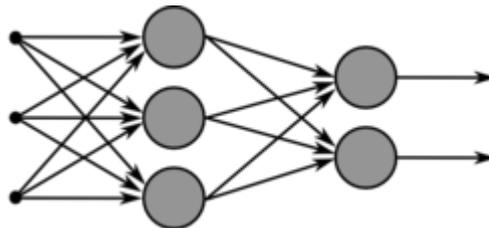


Figura 10. Red neuronal con capas de entrada, ocultas y de salida.

Una de las características del corpus de datos utilizado en este trabajo es que sólo dispone de información de un año. Por ello, los ciclos y estacionalidades en una medición de un edificio con frecuencias mayores no se podrían capturar por la serie temporal. No se trata de un problema aislado cuando se trabaja con datos reales, especialmente cuando los sensores se han instalado recientemente. Para abordar este asunto, se ha optado por generar un modelo “general” de edificio, es decir, entrenado a partir de datos de todos los edificios con el objetivo de aprender el comportamiento común a todos ellos. De esta manera, se resuelve la falta de datos para un edificio dado asumiendo que se comportará de manera similar a otros parecidos a él.

La estructura de los modelos de Deep learning aplicados al aprendizaje de series temporales que se ha diseñado en este trabajo es la siguiente:

- Capa de entrada (24 neuronas): Activa, inductiva, capacitiva de los últimos 5 meses. Descripción del edificio (superficie, director, encargado de mantenimiento,...).
- Capa de salida (1 neurona): Consumo de potencia active del siguiente mes.
- Capas ocultas: se han probado distintos números de capas internas con diverso número de neuronas cada una.

La Figura 11 compara los valores predichos de consumo de potencia active con respecto a sus valores reales para un edificio determinado.

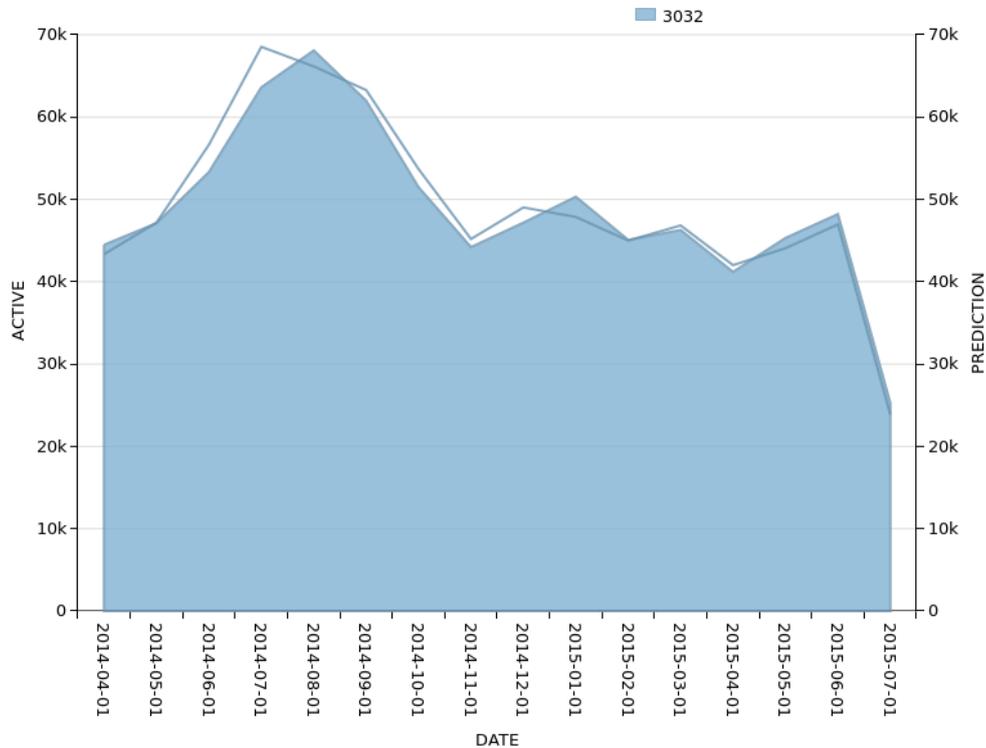


Figura 11. Predicción de activa y sus valores reales para un edificio.

Evaluación

Para la evaluación de esta técnica, se ha dividido el corpus en dos conjuntos aleatorios. El primero, con el 75% de las muestras, se ha empleado para entrenar los modelos. El segundo, con el 25% restante, se ha empleado para evaluar los modelos previamente obtenidos. De esta manera, al evaluar con muestras no utilizadas durante el entrenamiento, se obtiene una medida objetiva de la capacidad de aprendizaje de la técnica, similar al resultado que se encontraría en un entorno real en producción. Además, así se evita el sobreajuste de los modelos al conjunto de datos de entrenamiento, evaluando mejor la capacidad de generalización de la técnica.

Los modelos se han evaluado calculando la variabilidad de la variable objetivo (en este caso, el consumo de activa del mes siguiente que se muestra en la neurona de salida de la red) que es capaz de capturar el modelo. Esta variabilidad SS_{tot} (del inglés *total sum of squares*, suma de cuadrados total) se puede dividir en la variabilidad capturada por las variables introducidas en el modelo (SS_{reg} , suma de cuadrados explicada), y por otras variables desconocidas (suma de cuadrados residual, SS_{res}).

$$SS_{tot} = SS_{reg} + SS_{res}$$

En un buen modelo, SS_{res} será baja, ya que el modelo será capaz de explicar la mayor parte de la variabilidad. En general, el coeficiente de determinación (R^2) evalúa su relación:

$$R^2 = 100 * SS_{reg} / SS_{tot}$$

R^2 es un estadístico que proporciona información sobre la bondad del ajuste de un modelo. Por ejemplo, en regresión, el coeficiente de determinación R^2 es una medida estadística de lo bien que se aproxima la línea de regresión a los puntos de los datos reales. Si R^2 es 1, significa que la línea de regresión se ajusta perfectamente a los datos.

En nuestro caso, tomaremos un modelo base sin ninguna información, que predice valores futuros utilizando la media de una distribución de probabilidad normal. Si nuestro modelo de Deep learning consigue mejores resultados que dicho modelo base, podremos afirmar que nuestra implementación es capaz de aprender. De este modo, el modelo base con el que nos compararemos tiene los siguientes valores:

$$SS_{reg} = 0$$

$$SS_{tot} = SS_{res}$$

$$R^2 = 0$$

En cambio, los conseguidos por nuestro modelo de Deep learning son los siguientes:

$$R^2_{train} = 72\%$$

$$R^2_{test} = 64\%$$

El modelo ha sido capaz de capturar un 64% de la variabilidad de dicha variable para el conjunto de test. Es decir, pese a la escasa cantidad de datos, se consigue predecir un 64% de variabilidad ante datos que no se han visto en la fase de entrenamiento. Este resultado mejora claramente el modelo base, lo cual demuestra la capacidad de la técnica para aprender y generalizar. Cabe destacar que se puede considerar un buen resultado, a pesar de no haber entrenado con datos de más de un año de antigüedad. Esto habría permitido conocer el consumo de activa del mismo mes del año anterior, que previsiblemente habría aportado información significativa.

Alertas

Como subproducto de este sistema de predicción, se puede detectar una situación inusual comparando los valores predichos con los reales medidos. Proponemos emplear el error cuadrático medio (MSE, Mean Square Error o varianza) para decidir si esta diferencia (residuo, e) debería lanzar una alerta.

Cuando el valor absoluto del residuo dividido entre el MSE es mayor que 1.96, significa que se encuentra en la parte exterior de una distribución normal con un 95% de confianza. Así, estas alertas indican que el valor medido es mayor o menor que el esperado en una situación normal.

Alerta si: $|e/\sigma| > 1,96$

Dado: $e = \text{real-predicho}$

$\sigma = \sqrt{\text{MSE}}$

4.5. Detección de anomalías

Finalmente, se ha estudiado la aplicación de técnicas de Deep learning para la detección de anomalías multidimensionales en un corpus en el que no se ha etiquetado previamente las muestras anómalas, como es el que se utiliza en este trabajo. Este tipo de corpus no etiquetado es bastante habitual, ya que el proceso de etiquetado es costoso al implicar la intervención humana. Sin embargo, es más común disponer de un corpus de medidas tomadas de manera automática para el que se desea discernir situaciones anómalas.

Para resolver este problema, se ha implementado la aproximación basada en *auto-encoders* [2]. Un auto-encoder es una red neuronal en la que existe el mismo número de neuronas en la capa de entrada y en la capa de salida. Al contrario que en la aproximación empleada en la predicción, explicada en la sección anterior, en este caso, se hace coincidir los valores de las neuronas de salida con las de entrada. Sin embargo, al tener las capas internas menor número de neuronas, se fuerza al modelo a realizar una reducción de dimensionalidad al pasar de la capa de entrada a la primera capa oculta. Se puede considerar que se lleva a cabo una reducción similar a la realizada por otras técnicas estadísticas como el PCA (Análisis de Componentes Principales) [5]. Inversamente, el modelo deberá expandir las neuronas intermedias para producir la capa de salida.

Así pues, una vez entrenado el modelo, ante una nueva muestra, se reducirán sus dimensiones y se volverán a expandir. Es decir, el modelo habrá aprendido las correlaciones entre las neuronas de entrada para ser capaz de reducirlas en las capas intermedias y recuperar posteriormente los valores iniciales. Si este proceso se realiza correctamente, la salida de la última capa estará muy próxima a la entrada que se le proporcionó. Sin embargo, si la muestra que se analiza es diferente a las que se han empleado en la fase de entrenamiento del modelo, es decir si es anómala con respecto a las muestras consideradas “normales”, no se mantendrán las correlaciones aprendidas en el entrenamiento, por lo que esta reducción y expansión producirá un resultado distinto a la muestra de entrada.

Siguiendo esta filosofía, se han entrenado un auto-encoder para cada edificio con analizadores, con 14 neuronas de entrada y sus correspondientes 14 neuronas de salida (inductiva, activa y cos (ϕ) para cada una de las líneas: climatización, hornos, iluminación, frío industrial y general).

Alertas

Con el objetivo de establecer la similitud entre los valores de la capa de salida y la de entrada, se ha optado por calcular su distancia de Mahalanobis [4], que puede entenderse como una distancia euclídea en un espacio multidimensional. Según el valor de esta distancia se puede establecer la probabilidad de que exista una anomalía en la muestra. Por otra parte, se puede

establecer cuáles de los factores intervinientes son responsables de la anomalía analizando la contribución normalizada de cada una de ellos sobre la distancia multidimensional.

La Figura 12 muestra una posible visualización del resultado de estos modelos a lo largo del tiempo para un edificio determinado. Se han establecido categorías según el grado de probabilidad de la existencia de anomalías. Para ello, se han obtenido automáticamente umbrales a partir de los percentiles (0.95, 0.99) de las distancias.



Figura 12. Anomalías detectadas en un edificio a lo largo del tiempo.

Verde significa comportamiento normal, amarillo significa posible anomalía, rojo significa anomalía muy probable.

Evaluación

Se han evaluado los modelos entrenados dividiendo el corpus de datos en conjuntos de entrenamiento y test, de manera similar a como se explicó en la sección de predicción. En este caso, se ha optado por medir la precisión del modelo calculando su desviación estándar, obteniéndose un RMSE de 0,012. Así pues, es de esperar que las distancias de este modelo se distribuyan conforme a una distribución normal con media 0 y desviación típica 0,012. Además, podría utilizarse esta distribución para detectar posibles anomalías, de igual forma que se realizarían en un sistema de control estadístico de calidad clásico basado en gráficos de control [6].

5. Herramienta de visualización

Se ha diseñado una herramienta para la visualización de los resultados y la navegación gráfica avanzada a través de los mismos para facilitar la toma de decisiones.

En esta herramienta, se presentan en primer lugar los datos recogidos en las mediciones históricas y las que se reciben en tiempo real. Para ello, se emplean mapas y gráficos similares a los mapas de calor presentados en secciones anteriores, además de diversos widgets para representar los diversos tipos de datos.

Por otra parte, también se muestran los resultados de las técnicas explicadas en este documento, de manera que se le proporciona al usuario una herramienta para comprender el estado de las medidas disponibles, así como la predicción de estados futuros y la detección de anomalías.

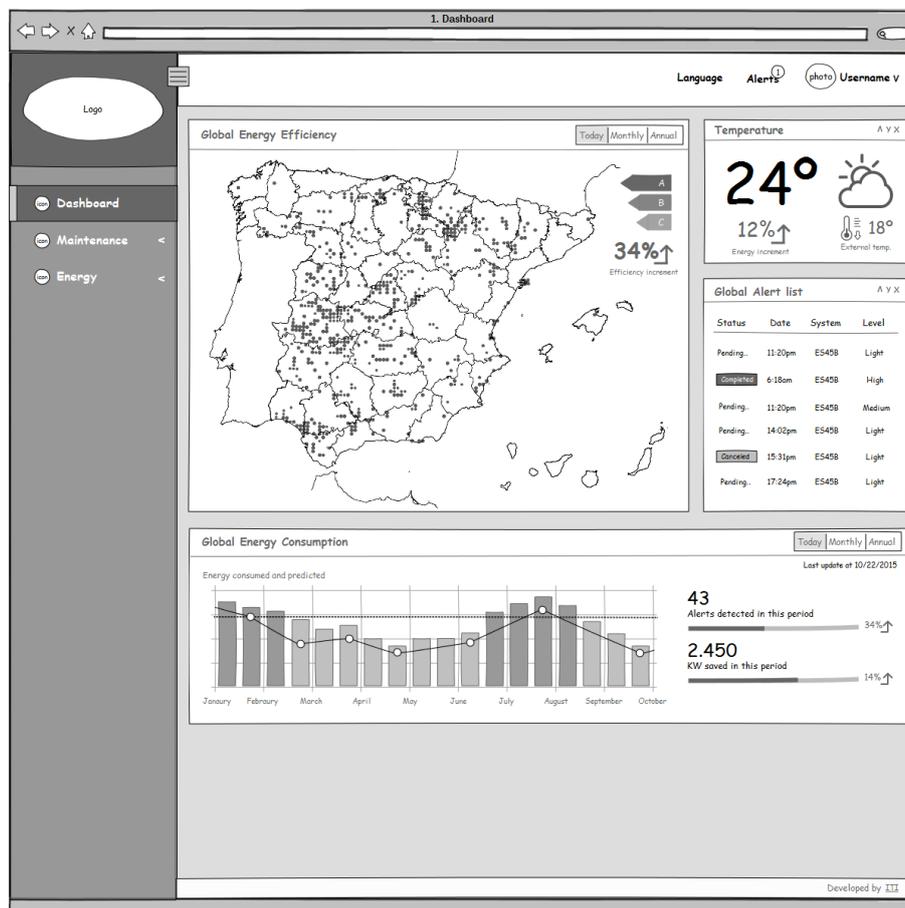


Figura 13. Mockup de Dashboard

El dashboard muestra tres widgets (aunque en el futuro puedan ser más) que recogen el comportamiento general en materia eficiencia energética del sistema. En concreto, muestra un mapa de calor que representa una clasificación de la eficacia de un conjunto de edificios

ubicados en diferentes lugares, un widget que permite conocer el consumo energético a lo largo del tiempo, un listado de las alertas y las acciones correctivas llevadas a cabo para solucionarlas.

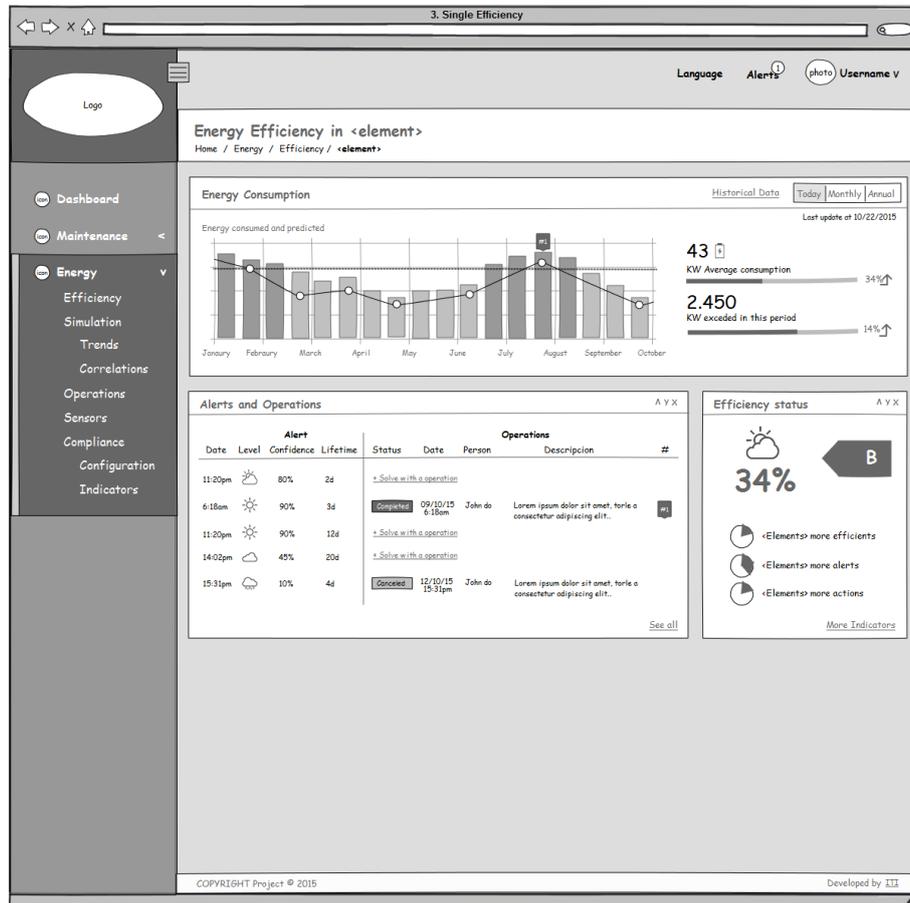


Figura 14. Mockup de Eficiencia Energética

Este mockup representa el estado del edificio en cuanto a eficiencia energética se refiere. En concreto, representa tres widgets en los que se recoge la energía consumida a lo largo del tiempo, un conjunto de indicadores de la calidad de eficiencia energética y un conjunto de alertas y acciones correctivas relacionadas con solucionar problemas detectados por los motores de predicción.

Este mockup será instanciado para cada uno de los edificios que se quiera monitorizar, permitiendo así construir una arquitectura de vistas que irá desde la generalidad (mapa con un conjunto de edificios) hasta la particularidad (edificio o partes del mismo).

De nuevo, este mockup será un paso intermedio antes de llegar a los datos que generaron las lecturas de consumo energético, representados por la siguiente figura.

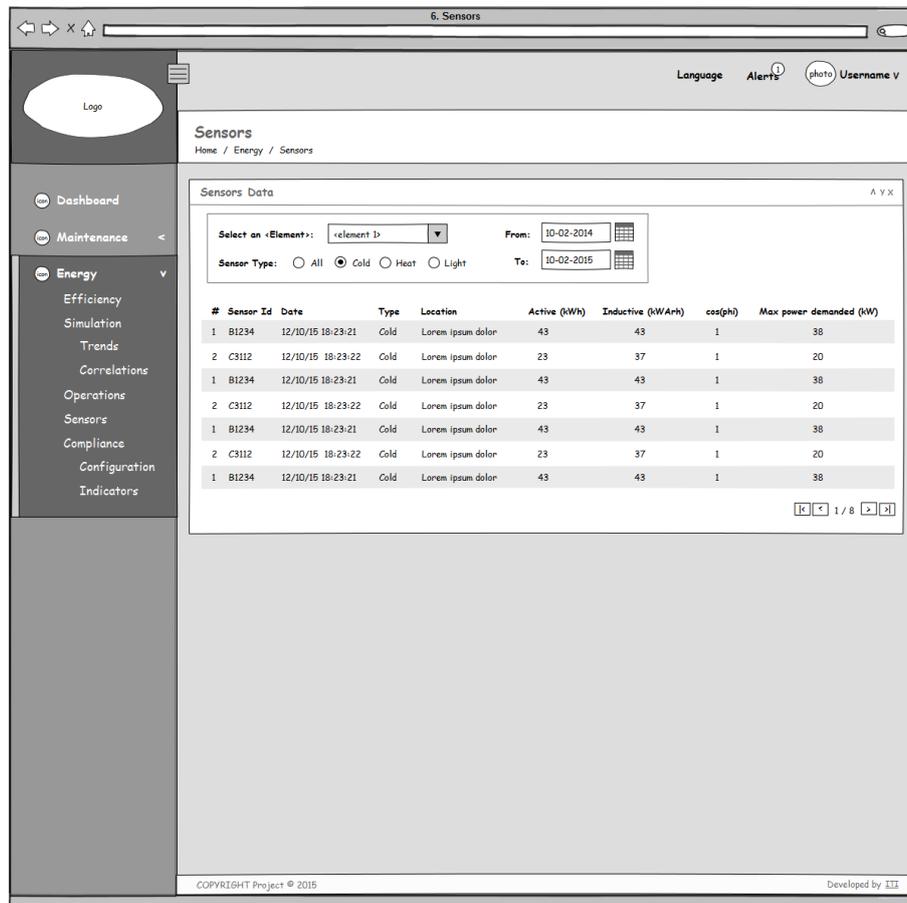


Figura 15. Mockup de listado de sensores

El mockup representa el acceso a los valores de las mediciones de los sensores ubicados en un edificio. Estos pueden ser filtrados por un conjunto de parámetros como el tipo de sensor, parte del edificio o la fecha de la medición. Esta vista nos permite bucear en el océano de datos que serán recolectados por los sistemas de medición.

Representados los mockups más importantes de la herramienta de visualización, se muestran a continuación el resto de funcionalidades relacionadas con la gestión de las operaciones/acciones correctivas, objetivos que se persiguen y los indicadores de rendimiento.

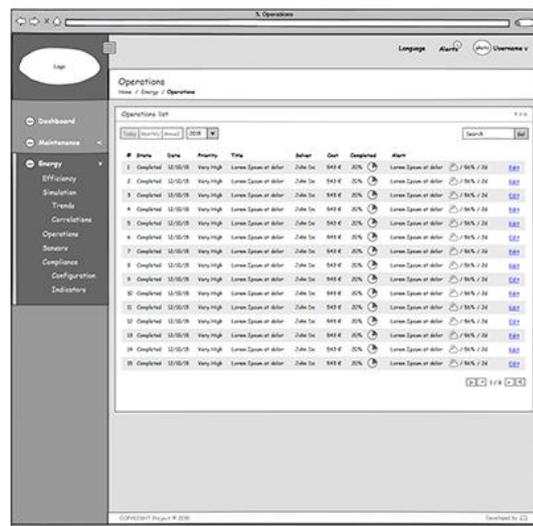
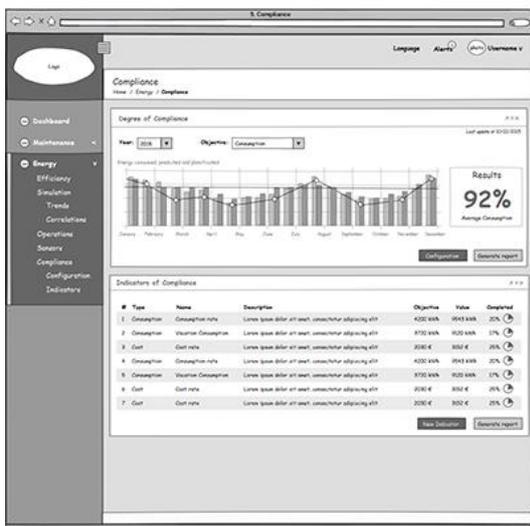
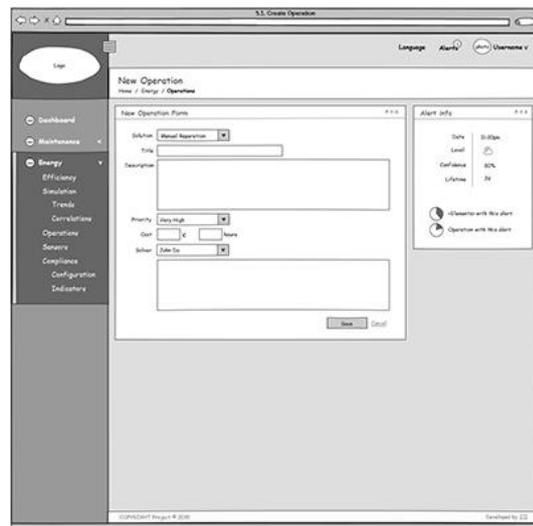
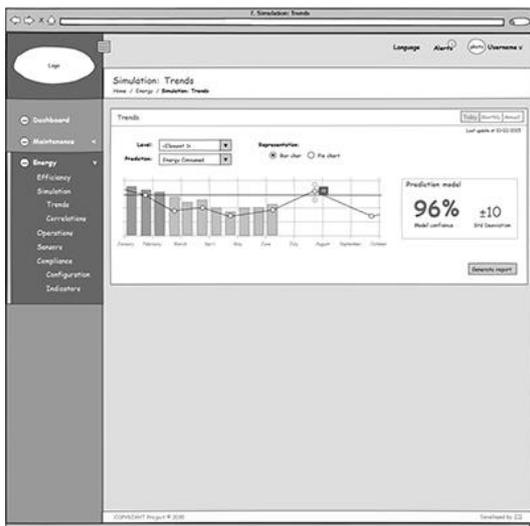


Figura 16. Resto de mockups de la herramienta de visualización

6. Referencias bibliográficas

- [1] M. Långkvist, L. Karlsson, A. Lourfi. A review of unsupervised feature learning and deep learning for time-series modelling. *Pattern Recognition Letters* 42, 11-24.
- [2] A. Jaokar. An introduction to Deep Learning and its role for IoT / future cities, May 2015.
- [3] A. Candel and V. Parmar. Deep Learning with H2O.
- [4] P.C. Mahalanobis (1936). "On the generalised distance in statistics". *Proceedings of the National Institute of Sciences of India* 2 (1): 49–55.
- [5] Pearson, K. (1901). "On Lines and Planes of Closest Fit to Systems of Points in Space". *Philosophical Magazine* 2 (11): 559–572.
- [6] <http://www.statisticalprocesscontrol.com/>
- [7] Context awareness in mobile computing environments. Christos B. Anagnostopoulos, Athanasios Tsounis, Stathes Hadjiefthymiades. *Wireless Personal Communications* 42(3). Páginas 445-464. 2007.
- [8] Deng, H.; Runger, G.; Tuv, Eugene (2012). "System monitoring with real-time contrasts". *Journal of Quality Technology* 44 (1): 9–27.
- [9] Mandel, B. J. (1969). "The Regression Control Chart". *Journal of Quality Technology* 1 (1): 1–9.
- [10] Oakland, J. (2002). *Statistical Process Control*. ISBN 0-7506-5766-9.
- [11] Wu, S. and Zuo, M.J. (2010). Linear and nonlinear preventive maintenance, *IEEE Transactions on Reliability*, Volume 59, No. 1, 2010, 242-249.
- [12] Wagner, Harvey M. and Giglio, Richard J. and Glaser, R. George. *Preventive Maintenance Scheduling by Mathematical Programming*
- [13] Innovation Award for "Simplifying Predictive Maintenance", The Institute of Engineering and Technology, United Kingdom, Dec 2007
- [14] R. Keith Mobley. *An Introduction to Predictive Maintenance*.
- [15] <http://www.winspc.com>
- [16] <http://www.reliasoft.com>
- [17] <http://www.stat-design.com/Software/Apogee.html>