TESIS DOCTORAL Volumen I

STUDY AND DESIGN OF CLASSIFICATION ALGORITHMS FOR DIAGNOSIS AND PROGNOSIS OF FAILURES IN WIND TURBINES FROM SCADA DATA

Alejandro Blanco Martinez

Directores de tesis:

Jordi Solé Casals
Pere Martí Puig
Jordi Cusidó Roura

Programa de doctorado: Ciencias Experimentales y Tecnología

2018

UNIVERSITAT DE VIC
UNIVERSITAT CENTRAL DE CATALUNYA

Escola de Doctorat

Resumen

Actualmente las operaciones de mantenimiento preventivo de los parques eólicos se soportan sobre técnicas de *Machine Learning* para reducir los costes de las paradas no planificadas. Por eso se necesita una predicción de fallos con cierta anticipación que funcione sobre los datos de SCADA, ya disponibles en los parques eólicos, sin necesidad de equipos adicionales. Los datos SCADA poseen una cierta cantidad de ruido y necesitan un proceso de distintas etapas hasta poder obtener una predicción de fallo con cierta precisión. Esta tesis comprende una secuencia de métodos utilizados por separado en distintos campos, siendo aplicados a los datos SCADA. Cada método ha sido evaluado, ajustado y modificado para su implementación en una plataforma automática de predicción de fallos. Los datos pasan por una primera etapa donde se limpian los valores extremos. Se ha encontrado que algunos métodos de filtrado automático pueden eliminar registros asociados a los fallos de las turbinas, por lo que se ha sugerido configuraciones y un método alternativo en una publicación. Acto seguido las distintas variables son seleccionadas por diversos métodos de selección de características, donde se ha hecho una comparativa en dos publicaciones para casos distintos con sus peculiaridades. Con las variables seleccionadas y filtradas, se han explorado métodos supervisados y no supervisados obteniendo resultados destacables en una publicación con el SOM y en *Deep Learning* con redes ANN y LSTM multicapa. Finalmente, se sugiere unas líneas que continúan el actual trabajo.

Resum

Actualment les operacions de manteniment preventiu dels parcs eòlics es recolzen sobre tècniques de Machine Learning per reduir els costos de les parades no planificades. Per això es necessita una predicció de fallades amb certa anticipació que funcioni sobre les dades de SCADA ja disponibles dels parcs eòlics, sense necessitat d'equips addicionals. Les dades SCADA tenen una certa quantitat de soroll i necessiten un procés de vàries etapes fins a poder obtenir una predicció de fallades amb certa precisió. Aquesta tesi comprèn una seqüència de mètodes utilitzats per separat en diferents camps, però sent aplicats a les dades SCADA. Cada mètode ha estat avaluat, ajustat i modificat per a la seva implementació en una plataforma automàtica de predicció de fallades. Les dades passen per una primera etapa on es netegen els valors extrems. S'ha trobat que alguns mètodes de filtratge automàtic pot eliminar registres associats a les fallades de les turbines, per la qual cosa s'ha suggerit configuracions i un mètode alternatiu a una publicació. Tot seguit les diferents variables són seleccionades per diversos mètodes de selecció de característiques, on s'ha fet una comparativa en dues publicacions per a casos diferents amb les seves peculiaritats. Amb les variables seleccionades i filtrades, s'han explorat mètodes supervisats i no supervisats obtenint resultats destacables en una publicació amb el SOM i en Deep Learning amb xarxes ANN i LSTM multicapa. Finalment es fan uns suggeriments amb unes línies que continuen l'actual treball.

Abstract

Nowadays, the preventive maintenance operations of wind farms are backing in Machine Learning techniques to reduce the costs of unplanned downtime. For this reason, an early fault prediction is needed that can work with the SCADA data that is already available in the wind farms, without the requirement of additional equipment. SCADA data has a certain amount of noise and requires a multi-stage process until a failure prediction can be obtained with a certain accuracy. This thesis comprises a sequence of methods that are found separated in diverse fields, with their application to SCADA data. Each technique has been evaluated, adjusted and modified for implementation on an automatic failure prediction platform. The data goes through a first stage where the extreme values are removed. It has been found that some automatic filtering methods can eliminate records associated with turbine failures, then configurations and an alternative method have been suggested inside a publication. Following, the different variables are selected by different methods of feature selection giving a result of two publications where a comparison of different methods is made for different cases with their peculiarities. With the variables selected and filtered, supervised and unsupervised methods have been explored, obtaining successful results in a publication with the SOM. The Deep Learning techniques with multilayer ANN and LSTM networks are also covered inside supervised section. Finally, a few lines are suggested that continue the current work.

Dedicado…

A mis padres que siempre me han apoyado a conseguir mis metas,
a mi novia por todos esos momentos que no he podido compartir.

AGRADECIMIENTOS

A mis directores de tesis por haberme ayudado, guiado y reconducido durante la elaboración de esta tesis. De forma más concreta, agradecer a Dr. Jordi Solé por cómo me ha aportado luz en cada paso importante a la hora de publicar y gestionar la tesis. A Dr. Pere Martí por su conocimiento y paciencia a la hora de analizar los resultados más complejos. Por otro lado y no menos importante, quiero agradecer la ayuda, confianza y facilidades prestadas por parte de la empresa SMARTIVE-ITESTIT donde pude desarrollar la presente tesis. En primer lugar, a Dr. Jordi Cusidó por aprender a gestionar adecuadamente el proyecto. Por otro lado, quisiera agradecer a Dr. Juan José Cárdenas por arrojar luz en la metodología y una visión positiva en cada resultado encontrado durante el proyecto, también debo agradecer a la persona que me aportó el conocimiento en el dominio de las turbinas eólicas, Dr. Isaac Jusiticia, sin el cual no pudiera haber validado muchos resultados.

A Dra. Karina Gibert por su visión e interpretación de los resultados en cuanto metodología no supervisada y filtrado de datos.

A la AGAUR y al programa de *Doctorats Industrials* fomentados por la Generalitat de Catalunya, por haberme dado la oportunidad mediante la beca concedida en realizar esta tesis.

A todas esas personas de la Universidad de Vic (OTRI) que me han ayudado con la gestión del doctorado.

Finalmente, a empresas como ACCIONA Energía entre otras que han facilitado el acceso a datos de cientos de turbinas y la confianza depositada en los resultados.

ÍNDICE GENERAL

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

CAPÍTULO 1

TESIS COMO COMPENDIO DE TRABAJOS PREVIAMENTE PUBLICADOS

La presente tesis doctoral, de acuerdo con el informe correspondiente, autorizado por los directores de tesis y el órgano responsable del programa de doctorado, se presenta como un compendio de tres trabajos previamente publicados. Las referencias completas de los artículos que pertenecen al cuerpo de la tesis son los siguientes:

*** Misma contribución**

- **Blanco-M, A.**, Gibert, K., Marti-Puig, P., & Cusidó, J. and Solé-Casals, J. (2018). Identifying Health Status of Wind Turbines by Using Self Organizing Maps and Interpretation-Oriented Post-Processing Tools. Energies, 11(4), 1-21. *(Impact Factor 2.262, Q2)*
- Marti-Puig, P.*, **Blanco-M, A.***, Cárdenas J. J., Cusidó, J. and Solé-Casals, J (2018). Effects of the Pre-processing Algorithms in Fault Diagnosis of Wind Turbines . Environmental Modelling and Software, ELSEVIER. ISSN 1364-8152, https://doi.org/10.1016/j.envsoft.2018.05.002 *(Impact Factor 4.404, Q1)*
- **Blanco-M, A.**, Solé-Casals, J., Marti-Puig, P., Cárdenas J. J., Justicia, I. and Cusidó, J. (2017) Impact of target variable distribution type over the regression analysis in wind turbine data, 2017 International Conference and Workshop on Bioinspired Intelligence (IWOBI), Funchal, 2017, pp. 1-7. (Indexado en IEEE Explore dentro de los Conference Proceeding.)

Se ha participado en tres pósters en dos congresos de gran interés en la eólica a nivel internacional (WindEurope, EWEA), donde se tratan metodologías aplicadas a casos reales:

- Cardenas J. J., **Blanco-M, A.**, Justicia, I., Cusidó, J., Marti-Puig, P. and Solé-Casals, J. (2016) PO.002 Cloud-based Platform for Wind Turbine's Health Estimation and Failure Prediction. EWEA Analysis of operating wind Farms, Bilbao, 14-15 April 2016. `http://www.ewea.org/events/workshops/analysis-of-operating-wind-farms-2016/posters/#002`
- **Blanco-M, A.**, Solé-Casals, J., Marti-Puig, P., Cardenas J. J., Justicia, I. and Cusidó, J. (2016) PO.066 Study of Feature-Selection-Algorithms with powerful 3D Insights for Wind Turbine Failure Prediction using SCADA Data. WindEurope Summit 2016, Hamburg, 27-30 September 2016. `https://windeurope.org/summit2016/conference/allposters/PO066.pdf`
- Cardenas J. J., **Blanco-M, A.**, Justicia, I., Cusidó, J., Solé-Casals, J.and Marti-Puig, P. (2016) PO.075 Wind Turbine Fault Forensic Analysis of SCADA Data Using Machine Learning Techniques. WindEurope Summit 2016, Hamburg, 27-30 September 2016. `https://windeurope.org/summit2016/conference/allposters/PO075.pdf`

Publicaciones que se encuentran actualmente en revisión y en preparación:

- **Blanco-M, A.**, Marti-Puig, P., Cardenas J. J., Cusidó, J. and Solé-Casals, J. (2018) SENSORS SELECTION IN WIND TURBINE PROGNOSIS. Journal Of Sensors, Hindawi. (En revisión)
- **Blanco-M, A.**, Solé-Casals, J., Marti-Puig, P., Cardenas J. J. and Cusidó, J. (2018) Deep Learning in Wind Turbine's Failure Prediction: comparison of multilayer ANN and LSTM techniques. (Artículo en preparación para presentar en Expert Systems with Applications)

CAPÍTULO 2

INTRODUCCIÓN

La energía eólica es la fuente de energías renovables que más crece [18], puesto a que ayuda a cumplir los objetivos, propuestos por la Unión Europea para el 2020, de reducir las emisiones y producir suficiente energía [16]. Además de estos objetivos se ha establecido que al menos un 20 % de la producción de electricidad, debe provenir de fuentes renovables [17] como las turbinas eólicas. En el caso de los parques eólicos, los costes de operación y mantenimiento (O&M) representan desde un 10 % a un 35 % del total de los costes de generación [45]. Si se reduce esta cantidad, los parques eólicos pueden llegar a ser más competitivos y acelerar la transición a las energías renovables [6].

En el mantenimiento de los parques eólicos, se programan tareas de mantenimiento preventivo cada 2500 a 5000 horas. Esta frecuencia es insuficiente para detectar y predecir el estado de la turbina, ya que no permite anticipar posibles fallos que se da entre cada revisión. Esto provoca paradas inesperadas y producen pérdidas importantes, que en algunos casos requieren de días o semanas esperando el componente además de los recursos para efectuar la reparación. Para hacerse una idea de la situación, la sustitución de la multiplicadora de una turbina puede llegar a superar el 15 % [3] del coste total de la misma, siendo también la responsable de que el 25 % [43] de la vida útil de la turbina esté fuera de servicio.

Dado esto, obtener información mediante el monitorizado continuo de la turbina eólica para detectar deterioros en su estado o posibles fallos futuros como estrategia preventiva, permite reducir estos costes y tiempos de reparación. Es común que en la tarea de monitorizado, exista un operador que mediante conocimiento sobre los sistemas de las turbinas, haga una diagnosis del estado. Esta operación resulta ser costosa en tiempo y recursos, dado que una turbina puede aportar más de 200 variables [50] en intervalos de 5 a 10 minutos por medio del SCADA (Supervisory Control and Data Acquisition). El sistema SCADA aporta información sobre distintos sistemas en forma de variables e indicadores como; temperaturas, indicadores eléctricos, posiciones físicas, velocidades, vibraciones, etc [56]. Estos sistemas producen ingentes cantidades de datos [54]. Además de estas variables continuas, las turbinas proporcionan información sobre las *alarmas* ocurridas en las que se indica el problema que hubo en un instante determinado.

Para poder determinar el estado de las turbinas de forma preventiva, se utilizan técnicas de *Machine Learning* que efectúan el análisis de la forma más automática posible, ya que la tarea de efectuar un análisis manual de caso por caso, requiere de recursos ingentes por parte de una persona experta en eólica. Existen diversas aproximaciones de las técnicas de *Machine Learning*, que por lo general se dividen en supervisado y no supervisado [33].

En el caso de supervisado, técnica en la cual se debe proporcionar el marcado de los registros (*labeling*), se emplean las alarmas proporcionadas por la turbina o en ocasiones los registros históricos de reparaciones de otras turbinas ubicadas en el mismo parque. Estas otras turbinas deben ser del mismo modelo y se utilizan con el objetivo de tomar las lecturas realizadas antes de la rotura del sistema que se esté analizando. Por lo que este caso supervisado, requiere de una persona experta en el campo que defina las alarmas o los registros del histórico de reparaciones, que han ser utilizados para el marcado de casos, generalmente binarios (buen estado / mal estado). Por otro lado, se encuentran las técnicas agrupadas en el aprendizaje no supervisado, en el cual no se utiliza ningún registro de alarma o reparación. El aprendizaje no supervisado tiene como objetivo capturar la información de los datos como agrupaciones, comportamientos de las variables o inclusive limpieza de estas de acuerdo a la información que aportan.

Podemos encontrar diversos trabajos acerca de ambas variantes del *Machine Learning* utilizadas en la actualidad para tratar datos reales provenientes de parques eólicos [60]. Por el lado supervisado, encontramos sistemas de clasificación y regresión, desde métodos simples basados en modelos Bayesianos [27, 51] hasta técnicas más complejas basadas en *Deep Learning* [10, 34, 58]. Sin embargo, por el lado no supervisado, se identifican distintos métodos basados en clústerización de los registros de las turbinas que sirven para identificar zonas y puntos de operación de la turbina eólica, identificando así, zonas problemáticas. De entre estas técnicas de clústerización, destaca el SOM (Self Organizing Maps) [30, 15, 55, 64, 26, 63, 36, 21]. Dentro del mismo grupo se identifican otras técnicas que tienen como objetivo, comprimir la información mediante la composición de las distintas variables como las basadas en redes neuronales como RBM [57] o Autoencoders [35] y en técnicas que combinan las variables que maximizan la captura de la varianza en los datos como el PCA [38, 48, 8].

Antes de poder aplicar cualquiera de los métodos explicados anteriormente, se deben adquirir y preparar los datos, así como efectuar el etiquetado de casos explicado en la Sección 2.1.

Una vez se han adquirido los datos y etiquetado los casos, debido a la baja calidad de estos, ha sido necesario incluir un preprocesado de limpieza de datos para eliminar valores extremos (*outliers*) introducidos por fallos de sensores, producidos por la manipulación durante el mantenimiento, los fallos de comunicación o incluso de configuración en la adquisición de datos [24]. En esta fase se generó un artículo de revista (Sección 3.2) en la que se analiza, en el caso de las turbinas eólicas, la problemática de eliminar estos valores extremos sin contrastar que información se está eliminando, demostrando así que se borraban gran parte de los estados de fallo en las turbinas, necesarios para generar el modelo de predicción de fallos.

En una nueva fase, se encontró en diversos experimentos en los cuales las variables relacionadas con un fallo por su posición en el modelo físico de la turbina, no cambiaban necesariamente antes y después de un fallo y/o el sistema en análisis fuese reparado, por lo que se vio necesario incluir técnicas de selección de variables. Estas técnicas, permitieron descubrir relaciones entre variables ante un fallo y eliminar las que se presuponían con relación física a un fallo. Esta fase generó una publicación para un congreso (Sección 3.3), en el que se comparaba el uso de distintos métodos de selección para determinar las variables objetivo de un modelo de normalidad (regresión). En dicho trabajo se mostraba la necesidad de ser cauto al tomar los resultados, porque los métodos de selección estaban ligeramente desviados a seleccionar las variables con comportamiento discreto provocando que los modelos generados tuvieran un error mayor en la mayoría de casos. Sin embargo las seleccionadas por una persona experta presentaban un comportamiento más analógico y que además tenían mayor relación con el fallo, obteniendo resultados más sólidos. También se utilizaron otros métodos de reducción/construcción de características mediante PCA, generando un subset de variables reducidas y composiciones entre estas.

En la fase final de la presente tesis, se trabajó en los modelos supervisados y no supervisados, dedicando una mayor cantidad de recursos a los métodos no supervisados. En ella se generó un artículo de revista (Sección 3.1) acerca de la combinación de métodos de *clústering* y *SOM*, que permiten, mediante todos los datos disponibles sobre un parque del que se desconoce el estado y el comportamiento de las turbinas, extraer información sobre posibles turbinas que se comportan de forma similar, además de determinar o separar un conjunto de registros con un comportamiento fuera de lo común. Esto permite en primera instancia, una visualización de los resultados en las agrupaciones de las turbinas, en la que la persona experta pueda centrarse en una muestra de cada grupo, analizarlo en detalle y emitir un juicio del estado de todo el grupo. El segundo resultado que refleja esta metodología, es entender cuales son los comportamientos de las turbinas en el tiempo y en función de las variables de entrada al sistema, decir las variables independientes que condicionan el funcionamiento de la turbina. Estas variables suelen ser temperaturas y velocidades de viento, por lo que se pueden extraer patrones o puntos en el espacio N dimensional (tantas como variables tiene la turbina) para cada época estacional, permitiendo identificar registros de turbinas que se alejan de estas zonas en cada período, haciendo de ello una clasificación previa del estado de salud.

2.1. El origen de los datos de este trabajo

Los datos con los que se han trabajado en esta tesis, provienen de fuentes reales, es decir, no son simulaciones ni generados por un entorno de pruebas en laboratorio, por lo que conlleva una gran dificultad debido a la poca calidad y el ruido que traen como se puede ver en la Sección 4.1.1.

Los datos se capturan a partir de un sistema SCADA instalado en los distintos parques que siguen el formato del estándar IEC 61400-25 [31], el cual describe una estructura de dispositivos lógicos (identificado como turbinas) y nodos lógicos que representan los distintos sensores físicos y además la agrupación de estos en sistemas y subsistemas. Para poder acceder a estos datos, hay que utilizar un cliente que utiliza un protocolo OPC (Open Platform Communications) [46], el cual recibe con una frecuencia de 5 a 10 minutos valores de los distintos sensores de la turbina (de los nodos lógicos), además de los eventos de fallo ocurridos. Normalmente, se generan 4 indicadores estadísticos que resumen los datos de cada 5 o 10 minutos y suelen ser la media, mínimo, máximo y desviación estándar, puesto a que los datos se capturan con una mayor frecuencia dentro del PLC de la turbina [20], aunque debido a las limitaciones de las telecomunicaciones y el espacio se guardan de forma resumida [31].

Los datos que han sido capturados del SCADA mediante OPC, se guardan en una base de datos MYSQL en el cloud (Azure), la cual ha estado recogiendo información durante los más de 3 años que dura el proyecto de la tesis. Esta tesis ha sido propuesta, a partir de los distintos acuerdos alcanzados con empresas como EDPR, ACCIONA, ENHOL entre otras (un resumen de datos disponibles se puede ver en la Tabla 2.2), que han facilitado datos de varios años de los distintos parques eólicos, de los modelos y fabricantes de turbinas más utilizados, permitiendo así la prueba de la metodología desarrollada en la tesis, mediante el uso de distintos Datasets, con distinto grado de calidad de los datos.

Los datos que tienen como objetivo ser utilizados para el desarrollo de modelos de *Machine Learning*, se guardan en forma de tabla con las entradas en las filas y las distintas variables en las columnas, guardando lecturas de todas las variables disponibles en cada instante de tiempo, la Tabla 2.1 es un ejemplo. Los eventos de alarma producidos por el sistema de control de la turbina o los registros de las distintas intervenciones de mantenimiento o reparaciones efectuadas sobre ellas, se guardan en otra tabla con un formato distinto.

Tabla 2.1: Ejemplo de dos variables y sus cuatro indicadores estadísticos, esta información es sólo una parte de una tabla.

date_time	Pot_avg	Pot_max	Pot_min	Pot_sdv	VelViento_avg	VelViento_max	VelViento_min	VelViento_sdv
2014-01-01 00:00	3042.49	3055	3026	5.77299	12.6732	14.7817	10.2277	0.757151
2014-01-01 00:10	3038.46	3109	2968	14.4987	13.0572	14.9977	10.037	0.790116
2014-01-01 00:20	3028.99	3114	2683	51.9811	11.9882	14.0099	8.47412	0.844361
2014-01-01 00:30	2721.04	3089	1945	353.451	11.0518	13.6694	7.69074	1.11926
2014-01-01 00:40	1712.22	2305	1289	255.599	9.1873	12.0117	6.60272	1.00914
2014-01-01 00:50	1611.69	1987	1219.71	201.704	8.82285	10.9603	6.27303	0.800605
2014-01-01 01:00	1415.67	1721	1126	177.691	8.52945	10.1275	5.26698	0.782771
2014-01-01 01:10	1448.35	1728	1204	125.751	8.86345	10.7914	6.91185	0.680807
2014-01-01 01:20	903.189	1247.88	597	190.03	7.52354	9.55344	4.41517	0.863314
2014-01-01 01:30	1166.66	1473	771	234.138	8.11257	10.0766	6.31044	0.719627
2014-01-01 01:40	1216.51	1520	1038	89.1022	8.35042	9.91983	6.60415	0.583451
2014-01-01 01:50	1393.17	1905	1066	205.113	8.6866	10.5981	6.54963	0.714473

Tabla 2.2: Datasets disponibles durante el desarrollo de la tesis

Modelo de Turbina	Num. de turbinas	Num. de años	Registros por año	Num. de variables	Num. de alarmas	Total registros evaluados
Fuhrlander2500 (SA)	5	4	105.120	303	72.422	2.102.400
Vestas V90 1.8 (RO)	7	4	52.560	194	9.681	1.471.680
Vestas V90 2.0 (PE)	13	4	52.560	63	5.063	2.733.120
Wfa H1	1	7	52.560	406	83.716	367.920
AW 3000 (ES)	16	3	52.560	181	1.266.457	2.522.880
AW 1500-77 (MO)	32	4	52.560	141	17.594	6.727.680
Gamesa G47-660 (IZ)	50	3	52.560	162	151.982	7.884.000
Vestas V90 2.0 (BA)	6	3	52.560	56	3.709	946.080
Vestas V90 2.0 (CE)	11	3	52.560	42	83.716	1.734.480
Vestas V90 2.0 (PI)	10	3	52.560	71	6.351	1.576.800
Vestas V90 2.0 (PR)	10	3	52.560	42	3.886	1.576.800
Vestas V90 2.0 (CA)	10	3	52.560	72	3.186	1.576.800
AW 1500-77 (VE)	10	3	52.560	141	6.150	1.576.800
AW 1500-77 (VI)	27	3	52.560	141	9.169	4.257.360
AW 1500-77 (CO)	22	3	52.560	141	48.938	3.468.960
AW 1500-77 (TA)	20	3	52.560	141	10	3.153.600
Total	276	57			2.141.248	45.061.920

Los eventos de alarma producidos por la turbina se analizan por separado, ya que cada uno tiene un identificador y simboliza un tipo de fallo. Estos eventos son inspeccionados por una persona experta que explora las relaciones y orden entre estos. Además del sentido físico, con esta exploración se pretende buscar los eventos de los sistemas más importantes que sirven para predecir los fallos, como son el generador y la multiplicadora [3], responsables de una buena cantidad del coste de mantenimiento y de largos tiempos de espera para su reemplazo [43].

2.2. Objetivos de la investigación

En función de lo descrito en la introducción, el objetivo general de la tesis doctoral es generar un proceso metodológico desde datos reales sin procesar desde los parques, incluyendo distintas metodologías presentes en el estado del arte que permitan tratar y extraer suficiente información como para emitir un resultado que pueda diagnosticar los fallos de las turbinas con la característica que sea práctico y aplicable al entorno industrial. Dado esto, se tendrá que evaluar, modificar metodologías y buscar mecanismos que permitan ajustar de forma automática los parámetros y variables presentes en los distintos algoritmos del estado del arte. Se pueden desglosar en los siguientes objetivos:

- Determinar qué datos de los disponibles son necesarios captar de las turbinas, para poder llevar a cabo el análisis tanto en cantidad, como en calidad.
- Determinar los indicadores necesarios de salud de la turbina, para poder construir y entrenar el modelo.
- Analizar qué proceso de transformación y filtrado se ha de efectuar en los datos para cada parque eólico.
- En el caso de predicción de fallos, encontrar con que antelación se puede trabajar para cada caso, teniendo en cuenta que se espera márgenes de más de 1 semana para que tenga utilidad industrial.
- Generar las herramientas, implementaciones y modificaciones necesarias de los métodos utilizados en la tesis, para que cada paso funcione de forma automática pero siendo configurable.

2.3. Aportaciones del doctorando

Las contribuciones del doctorando en el proceso de predicción de fallos en las turbinas eólicas, se soporta sobre el resultado de las distintas publicaciones y su aplicación industrial:

- El filtrado de los datos considerado como valores extremos (*outliers*) se debe tratar con precaución, ya que como muestra el trabajo de la Sección 3.2, la eliminación de los registros marcados como valores extremos eliminan información de casos de turbinas defectuosas. Esto hace que el ajuste (*Accuracy*) del modelo generado a partir de estos datos, presente buenos resultados incluso utilizando *crossvalidation*, pero sin embargo limita de forma notable la capacidad de generación de nuevos casos no filtrados. Esto es debido a que en más del 97 % [14] de los datos, las turbinas eólicas no presentan estados de fallo lo suficientemente graves, quedando sólo el 2-3 % de los datos que representaban las anomalías en el funcionamiento de la turbina, los cuales eran borrados a causa del filtrado.
- Algunos métodos de selección de variables aplicados en el caso de turbinas eólicas con datos provenientes de SCADA, tienden a seleccionar variables predictoras de fallo de peor calidad, debido a la notable preferencia que tienen estos métodos por las variables con una distribución más discreta. Esto ha sido validado examinando la exactitud de los resultados arrojados por modelos de regresión, en los que se han obtenido peores resultados en general con las variables seleccionadas por estos métodos automáticos, que con las variables determinadas por una persona experta en el campo del análisis de fallos.
- La elaboración de una metodología que permite por un lado, encontrar grupos de turbinas que se comportan de forma similar en el tiempo y por otro, crear grupos en los que se pueden analizar una o más turbinas representativas, reduciendo los recursos necesarios a la hora de preclasificar las turbinas según

su estado de salud, para generar los modelos de clasificación. Además, la metodología aporta información sobre el comportamiento y las interacciones de las variables, dejando ver patrones en conjunto de datos fuera del funcionamiento normal, de manera que permite elaborar un preetiquetado de casos de posible fallo.

- La aportación a nivel industrial, es la implementación de todos los métodos cubiertos en la tesis para que funcionen de forma automática, siendo necesario en algunos casos, añadir técnicas para generar los parámetros que varios métodos tienen asignado de forma manual en el estado del arte, como por ejemplo el número de clústers o el tamaño del mapa SOM, mediante la aproximación de distintas métricas. Además, se ha aportado una interfaz gráfica que permite visualizar los resultados en términos de análisis científico (en R y reportes automáticos en HTML) y una resumida industrial en una interfaz web (cast.smartive.eu y cm.smartive.eu).

Además de estas aportaciones, el sistema de predicción implementado se ha ido mejorando en distintas iteraciones añadiendo mejoras a cada parte del proceso, que generarán publicaciones fuera de la tesis doctoral. El doctorando ha tenido un papel clave en el nexo entre los distintos grupos de investigación que han ido participando en el proyecto de la empresa durante estos 3 años; UPC (dpto. Ciencias de la computación), UVIC (Tractament de Dades i Senyals), aportándoles los datos con el preprocesado requerido para caso, y ayudando en la interpretación de resultados para cada uno de los grupos cuando fuese necesario. El doctorando, ha aportado en todo momento a la empresa donde se desarrollaba la tesis una visión de implementación industrial a cada uno de los algoritmos desarrollados.

CAPÍTULO 3

COPIA DE LOS TRABAJOS PUBLICADOS

3.1. Identifying Health Status of Wind Turbines by Using Self Organizing Maps and Interpretation-Oriented Post-Processing Tools

energies

Article

Identifying Health Status of Wind Turbines by Using Self Organizing Maps and Interpretation-Oriented Post-Processing Tools

Alejandro Blanco-M. [1,2,†], **Karina Gibert** [3,4,*,†], **Pere Marti-Puig** [1], **Jordi Cusidó** [2] and **Jordi Solé-Casals** [1]

[1] Data and Signal Processing Group, U Science Tech, University of Vic-Central University of Catalonia, 08500 Vic, Catalonia, Spain; alejandro.blanco@uvic.cat (A.B.-M.); pere.marti@uvic.cat (P.M.-P.); jordi.sole@uvic.cat (J.S.-C.)

[2] Smartive Wind Turbine's Diagnosis Solutions, 08204 Sabadell, Barcelona, Catalonia, Spain; jordi.cusido@smartive.eu

[3] Department of Statistics and Operations Research, Universitat Politècnica de Catalunya-BarcelonaTech, Knowledge Engineering and Machine Learning Research group at Intelligent Data Science and Artificial Intelligence Research Center, UPC, 08034 Barcelona, Catalonia, Spain

[4] Institute of Science and Technology of Sustainability, UPC, 08034 Barcelona, Catalonia, Spain

* Correspondence: karina.gibert@upc.edu;Tel.: +34-93-4017323

† These authors contributed equally to this work.

Received: 6 February 2018; Accepted: 21 March 2018; Published: 22 March 2018

Abstract: *Background:* Identifying the health status of wind turbines becomes critical to reduce the impact of failures on generation costs (between 25–35%). This is a time-consuming task since a human expert has to explore turbines individually. *Methods:* To optimize this process, we present a strategy based on Self Organizing Maps, clustering and a further grouping of turbines based on the centroids of their SOM clusters, generating groups of turbines that have similar behavior for subsystem failure. The human expert can diagnose the wind farm health by the analysis of a small each group sample. By introducing post-processing tools like Class panel graphs and Traffic lights panels, the conceptualization of the clusters is enhanced, providing additional information of what kind of real scenarios the clusters point out contributing to a better diagnosis. *Results:* The proposed approach has been tested in real wind farms with different characteristics (number of wind turbines, manufacturers, power, type of sensors, ...) and compared with classical clustering. *Conclusions:* Experimental results show that the states *healthy*, *unhealthy* and *intermediate* have been detected. Besides, the operational modes identified for each wind turbine overcome those obtained with classical clustering techniques capturing the intrinsic stationarity of the data.

Keywords: wind farms; Supervisory Control and Data Acquisition(SCADA) data; self organizing maps (SOM); clustering; fault diagnosis; renewable energy; interpretation oriented tools; post-processing; data science

1. Introduction

Wind energy, the most growing renewable source [1], helps to meet the demanding climate and energy targets for 2020 set by the EU Commission [2]. Together with these targets, it was established that at least 20% of electricity production must come from sustainable sources [3], among which wind farms are. Wind farms operation and maintenance costs (O&M) represents from 10% to 35% of the overall generation costs [4]. Reducing this amount, the wind farms will be more competitive concerning fossil fuels and accelerate this transition [5].

In the management of wind farms, turbines are scheduled to be maintained every 2500 to 5000 h with preventive maintenance. However, the preventive maintenance operation frequency is insufficient to detect and predict device status and anticipate potential failures. Unexpected stop of turbines has significant costs since they often are placed far from urban areas and several days may be required to wait for the necessary new component and make in situ reparations. To get an idea, about 15% of total turbine cost [6] raises every time that a gearbox needs to be replaced unexpectedly, this representing about a 25% of the total downtime [7].

Getting accurate information about potential failures requires continuous monitoring and diagnosis of turbines health status, and the development of preventive maintenance strategies, which avoid unexpected failures of wind turbines. Expert knowledge plays a fundamental role in diagnosing turbines. However, exhaustive analysis of the whole set of wind turbines of a given wind farm cannot be made by a human expert. When a wind farm starts being monitored, and mainly if it contains a large number of turbines, the first big challenge is to identify a reduced set of representative turbines for detailed inspection. These require, as a first stage, grouping the turbines according to the status of each of their primary subsystems.

Modern wind turbines record more than 200 analog variables [8] at intervals of 5 to 10 min using their SCADA (Supervisory Control and Data Acquisition) system. The SCADA system provides information about temperatures, electrical indicators, physical positions, speeds, vibration, etc. [9]. The analogic variables are the continuous readings from the different wind turbine's sensors along the time; the SCADA also provides discrete variables which are generated by failure events. Through SCADA-based condition monitoring, detailed data is provided, and this data is suitable to be exploited to find the different wind turbines operation regimes that allow grouping by turbines of similar health status. The exhaustive handmade exploration of turbine variables becomes an unfeasible task. There are many manufacturers, and there is no standardization on how event data is reported. This means that the different variables names and also were they are physically is different from manufacturer to manufacturer. The failure events are also heterogeneous in format and meaning not having a generic code to reference a specific type of physical failure like a gearbox breakdown. Because of this significant amount of data has to be checked, and the number of different working conditions is high, this is why the attention of human experts can only focus on a few turbines, and why groups of turbines associated with similar health status need to be identified. SCADA data is a rich source of information. Taking advantage of a proper analysis of these data, automatic monitoring systems, and decision support tools can be developed, thus contributing to the better planning of maintenance operations and, as a consequence, to decrease operating costs. Data Science and machine learning techniques offer appropriate methods and approach to tackle this tasks.

The purpose of this work is to propose a new methodology based on data science and automatic interpretation techniques to identify a reduced set of wind turbines, representative enough of a complete wind farm, to be carefully inspected by human experts in a reasonable time, by providing support to decision-making about preventive maintenance of the park. The significance of the work is high, as exhaustive inspection of all wind turbines in the farm is no affordable, and the economic impact of reducing unexpected failures is considerable. The primary hypothesis of this work is that the proposed methodology allows identification of distinct turbine operation regimes, by grouping the turbines of a park accordingly, in such a way that bad health regimes appear in separate groups. These groups should be understood in terms of certain indicators that will support the expert decision to schedule a maintenance operation and, as a consequence the number of unexpected failures is expected to be reduced, overcoming the current state of the art.

This study focuses on a particular type of failure, for simplicity, but the proposed methodology is general. Thus, in this work, the identification of distinct groups of turbines according to the status of the gearbox is pursued, because this is an expensive wind turbine subsystem, with frequent breakdowns that are challenging to repair and is the responsible for expensive maintenance costs due to its components, as explained before.

Several strategies exist for implementing Condition Monitoring Systems (CMS). One of the most popular methods comes from the machine learning field, which is based on Artificial Neural Networks (ANN), is the Self Organizing Maps (SOM) [10]. SOM runs as an unsupervised system and is envisaged as a promising tool due to its sensitivity to detect abnormal operation registers. Therefore, an ANN approach based on SOM can provide a clustering that reflects the nature of the entire set of turbines and significantly reduces the human factor in the consistency criterion. Discovering turbines whose characteristics deviate from normal behavior is useful for experts, who can then focus their attention on them. At the same time, finding turbines in better and more stable conditions allows to take them as a reference in trend systems.

However, as happens with the other ANN systems, the simple use of SOM have some limitations concerning capturing a particular type of complex stationarities and providing a good understanding of the nature of proposed clusters to the experts. Few works have been done on complementing the results provided by SOM with additional tools that bridge the gap between raw data mining results and decision-making processes. In this paper, a data-driven process is proposed with the objective of close this gap. The process combines the clusters discovery using SOM with some further elaboration of the proposed clusters and additional interpretation. The further interpretation is made with oriented tools like Class panel graphs (CPG) [11] or Traffic Lights panels (TLP) [12], both introduced in Section 3, with the objective of identifying a reduced set of turbines to be inspected in situ, using the available SCADA measurements monitoring.

The structure of the paper is the following: Section 2 provides results of the application of the proposed approach to real data, while they have discussed in Section 3 pointing also to future work. Finally, in Section 4 the methodological approach and the context of the resented research in a real wind farm is described.

2. Results

2.1. SOM Dimensions for the Experiments

In this work, data coming from the SCADA system of several wind farms are considered (see Section 4 for details on data used). For conduct the experiments for the wind farm *'Wf1'* (see Table 7), the first step is to decide the size of the SOM. Having $R = 17.536$ registers (16 turbines × 3 years × 365.3 days), and according to the rule $n = \lceil 5\sqrt{R} \rceil$ [13], the number of units (neurons) to be used should be 663, which represents a SOM of size 25 × 25, approximately. However, as explained in Section 4.4, we will use Topographic error (*TE*), and the Quantization error (*QE*) metrics to set the optimal size. Therefore, SOM maps of different sizes have been generated with the data from wind farm *'Wf1'*. The (normalized) evolution of both metrics *TE* and *QE* is plotted in Figure 1, for sizes ranging from 20 × 20 (400 neurons) to 100 × 100 (10,000 neurons). We observe how *TE* drops exponentially when the number of neurons increases, while *QE* increases with it. The crossing point of both curves is 52, which will be used as the optimal SOM size.

To verify the adequacy of the SOM dimensions, we compared results obtained with a SOM generated, for wind farm *'Wf1'*, using the optimal (52 × 52) size and a sub-optimal one (70 × 70). Figure 2a,b represent the U-matrix for sizes 52 × 52 and 70 × 70, respectively. Although they are not equal, the peaks of both maps are located in the same areas and show similar values, indicating that both U-matrix identify the same kind of structure despite being independently created from a different number of neurons.

Figure 3a,b show clustering performed on the SOM codes (neuron weights) using the *Hierarchical clustering* technique for a fixed number of 5 clusters, for 52 × 52 and 70 × 70 maps, respectively. In this figure, the clustering result is plotted over the corresponding U-matrix for each case, to ease the interpretation of the clustering.

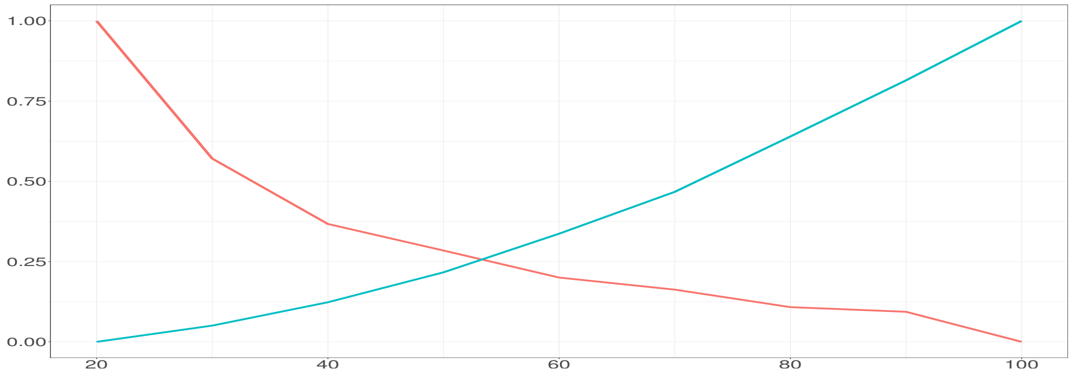


Figure 1. Normalized *TE* (blue) and *QE* (red) metrics of map sizes from 20 to 100 neurons. The horizontal axis indicates the different SOM map sizes. Vertical axis indicates the normalized error (0-1).

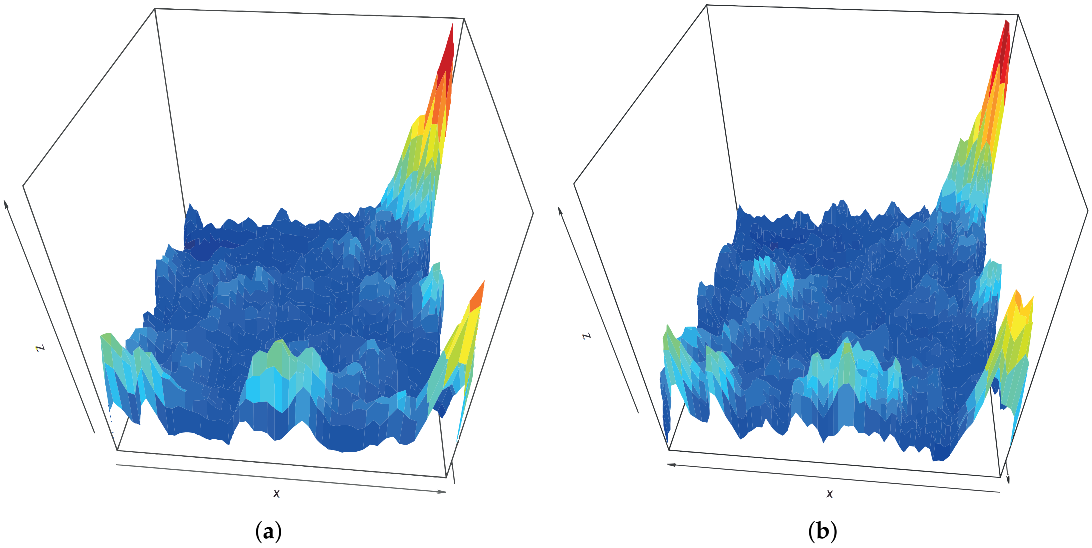


Figure 2. U-Matrix for SOM sizes of 52 × 52; (**a**) and 70 × 70 (**b**).

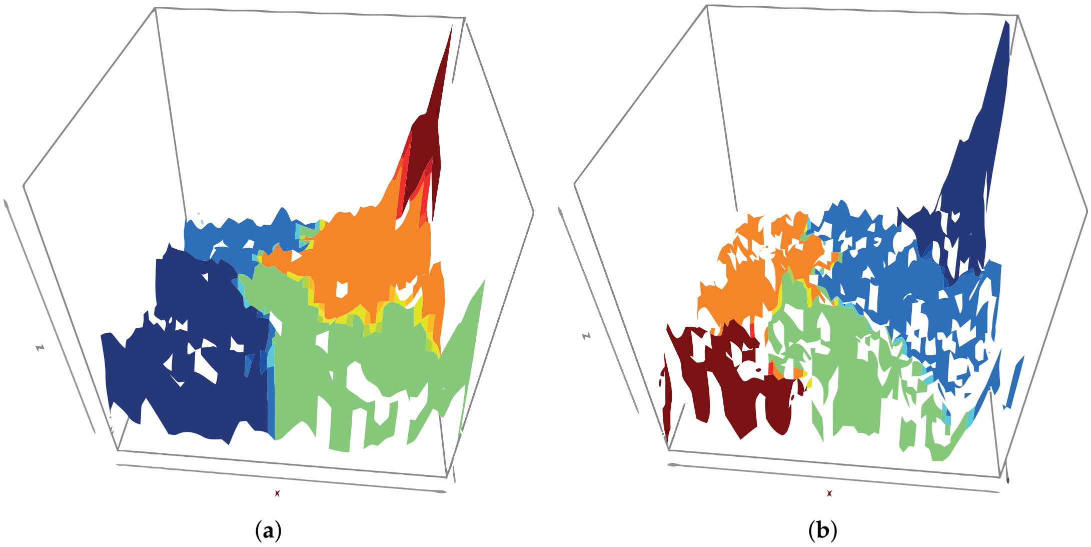


Figure 3. U-Matrix colored according the results of SOM clustering for a SOM dimension of 52 × 52; (**a**) and 70 × 70 (**b**).

In both cases, the clusters in the upper right and lower left map corners appear. These Sections can also be seen in the U-Matrices and show areas with high distance values. Provided that computational costs of SOM increase with the number of neurons and that the impact of increasing the neurons on the identified structures is low, we will use 52 × 52 size.

2.2. Understanding the Results of SOM Clustering

The clustering performed over the SOM codes contains information about the wind turbines. The TLP of the resulting clustering is shown in Table 1(see details and meaning of colors in Section 4.7); the corresponding class panel graph with super-imposed TLP is in Table 2. Both are performed to support the conceptualization process of the clusters.

Looking into details of each one of the clusters in Figure 3a we identified the following (listed from most general to most particular) cases:

Cluster 1-High-performance regime due to strong wind

(bottom left corner in Figure 3a): This scenario can take place all along the year on windy days, and therefore a variety of ambient temperatures are registered. Its main characteristic is the presence of higher wind. Thus the rotor is in full movement, the wind production is high, the oil temperature is high and so is the temperature of the bearing. The best performance of all the groups.

Cluster 4-Low-performance low wind regime

(top right area in Figure 3a): In this scenario, there is low wind; rotor does not rotate at maximum speed and the power generated is small. Except Cluster 5, this is the weakest generation case, and it can happen all along the year, so air temperatures range widely while bearing and oil temperatures are not very high. Low performance.

Cluster 3-Moderate performance regime in summer due to moderate wind

(bottom right in Figure 3a): Due to an intermediate wind level, the rotor is rotating adequately but not at high speed, so the energy production is low. It is summer time, with high air temperatures, and the oil is warmer than in Cluster 2, but the bearing has its same temperature. Intermediate performance.

Cluster 2-A regime of moderate performance in winter by moderate wind

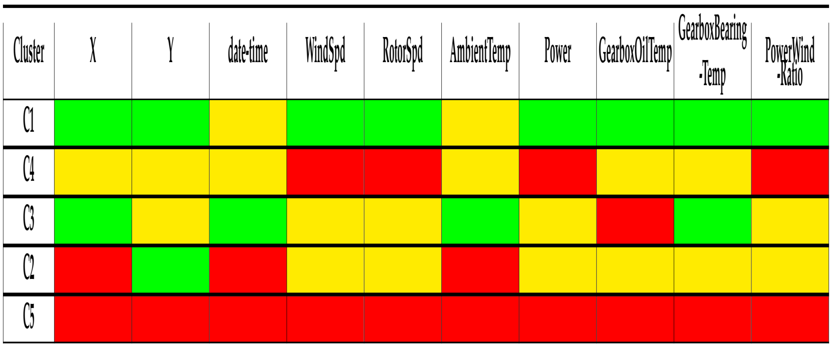
(top left in Figure 3a): Moderate wind; rotor rotating adequately but not at high speed. It is winter time and therefore the air temperature is cold, the energy production is low, the oil temperature is colder than in Cluster 3 while the bearing temperature is moderate. Intermediate performance. The difference between Cluster 3 and Cluster 2 is the ambient temperature: in both cases, similar wind forces, rotor speed, rotor and bearing temperatures, power and similar performances associated with warmer oil in C3.

Cluster 5-Turbine regimen stopped due to lack of wind on winter days

(supper top right in Figure 3a): A particular scenario in which there is no wind; therefore the rotor is stopped. It occurs cyclically in winter and cold days (low ambient temperature). The power production is sometimes negative, meaning that there is no production but consumption which can be the consequence of the oil heater system or when the wind turbine enters in a start-up phase. Bearing temperature is the lowest among all the clusters. As the oil is heated, the bearing temperature is also heated. The turbines are stopped because there is no wind at all. Zero or negative performance.

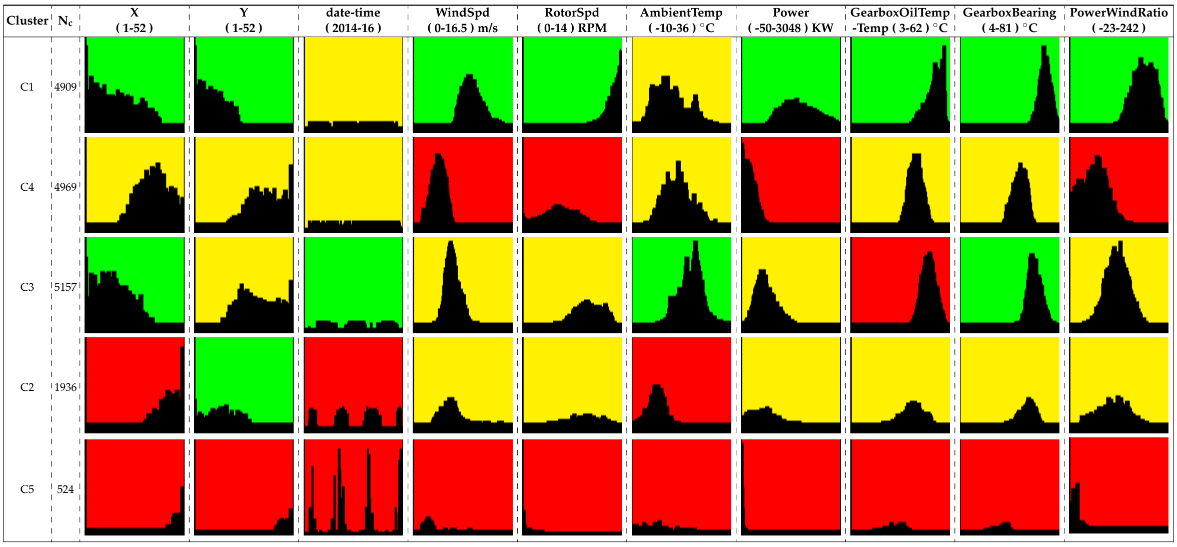
With the use of CPG and TLP a clear interpretation of SOM areas is now obtained. Since the CPG also contains the coordinates of the SOM neurons involved in each class, Figure 4 shows the interpretation of the SOM map.

Table 1. Traffic Lights Panel of the SOM codes clustering result. The *Cluster* column indicates the clusters found in Section 2.1.



Cluster	X	Y	date-time	WindSpd	RotorSpd	AmbientTemp	Power	GearboxOilTemp	GearboxBearing-Temp	PowerWind-Ratio
C1	🟩	🟩	🟨	🟩	🟩	🟨	🟩	🟩	🟩	🟩
C4	🟨	🟨	🟨	🟥	🟥	🟨	🟥	🟨	🟨	🟥
C3	🟩	🟨	🟩	🟨	🟨	🟩	🟨	🟥	🟩	🟨
C2	🟥	🟩	🟥	🟨	🟨	🟥	🟨	🟨	🟨	🟨
C5	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥

Table 2. Class Panel Graph of the SOM codes clustering result vs input variables with TLP super-imposed.



As mentioned in Section 4.5, different turbines activate different areas of the SOM map. In Figure 4 (middle and right) two different turbines show different patterns or active neurons. In fact, the plot in the middle corresponds to a healthy turbine (H) and the plot in the right to an unhealthy turbine (U). Here we see that some neurons of cluster 2 of the turbine H are the single ones intersecting with the low-performance behavior due to low wind (sector C4 in Figure 4, left), whereas for the turbine U the whole cluster 2 is practically concentrated over regions of low performance or stopped turbine. Also, other topological differences are observed between the two maps. Even if these analyses are accessible by an expert, an automatic procedure to evaluate if this SOM sub-maps are similar or not is required.

2.3. *Discrimination of Wind Turbines According to the Neuron Activation in SOM Maps*

A local analysis of each turbine is performed by following the methodology presented in Section 4.6. To illustrate the feasibility of the method, the activation of neurons in the SOM maps of two preselected turbines of the same model and wind farm is compared. Turbine H (Figure 4) is in excellent conditions, and we know it has had very few failures. In contrast, turbine U (in Figure 4) had many shortcomings and suffered repairs, among which we highlight a breakdown in the gearbox, which is the system we are analyzing. The results can be seen for a 52 × 52 map in Figure 4, showing how the maps exhibit a near complementary assignation of BMUs. This is the key point to identify turbines with a similar state of health. If we manage to separate the turbines according to how the BMU activations resemble among them, we will be able to group turbines according to their state, and this will make possible to discriminate the unhealthy turbines from the healthy ones. To simplify the comparison between turbines and to have a non-subjective measure, clustering is applied to each wind turbine, and the cluster centroids are calculated. As we will detail in the next sub-Section, these centroids will be later used to group turbines of similar health status.

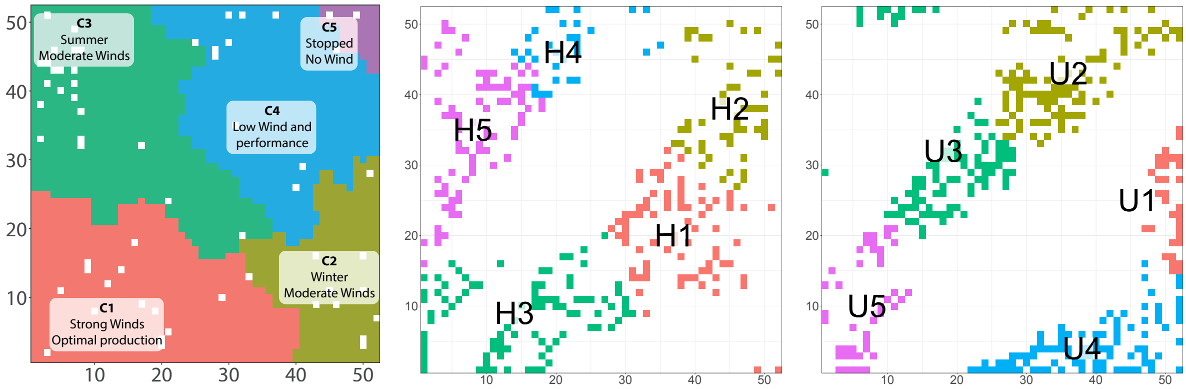


Figure 4. Interpretation of clusters built over the SOM map based on TLPs (left) and active BMU for turbines H (middle) and U (right) with local colored clusters, generated as detailed in Section 4.5. Both axes on all the subfigures indicate the neuron id for a SOM-map of 52 × 52.

2.4. Understanding Results of BMU Clustering

For the new local clusters of each turbine, CPGs and TLPs are also developed (see Table 3). The resulting local patterns shown in each turbine are analyzed.

The post-processing performed with CPGs and TLPs elicits a relationship between clusters local to a wind turbine (built over the BMUs) and global clusters (built over the SOM codes). The operating points do not disappear when analyzing wind turbines separately but might take slightly different behaviors in each local cluster. In the following lines we interpret the relationship between the CPG in Table 2 (prefix C for the general clusters) and the two turbines (H and U) with their local clusters with prefix H for Healthy for the turbine with identifier 119 and U for Unhealthy for the turbine with identifier 133.

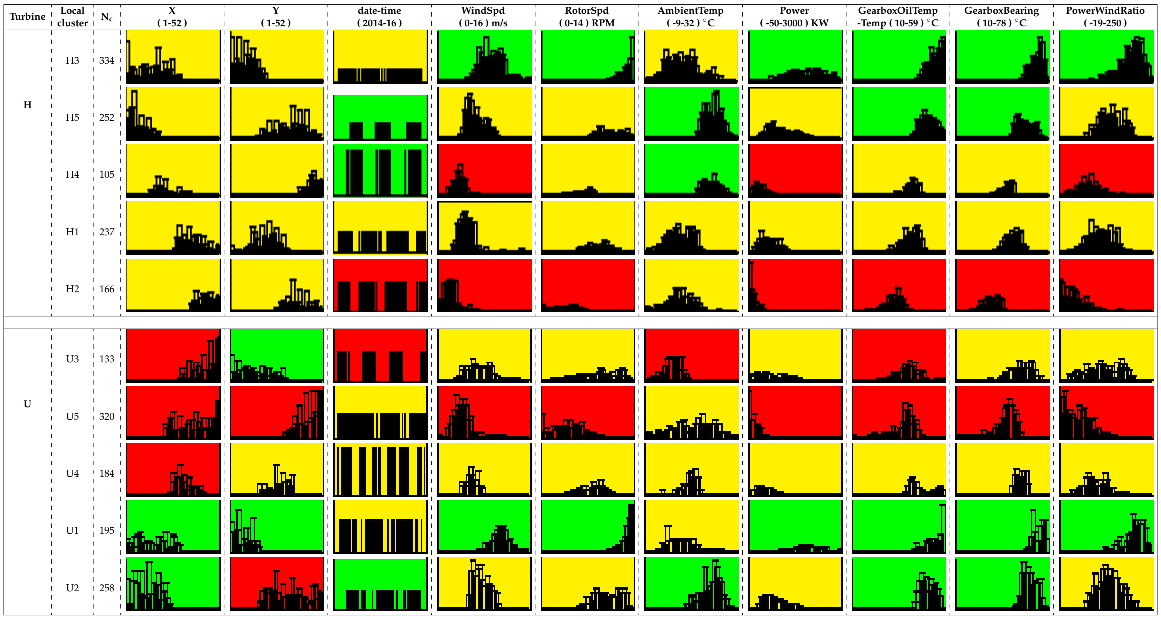
Looking first at the similarities of the clusters found with the general CPG, the local clusters *U4* and *H3* are representing the same operational regime as *C1: optimal performance*. Also, we observe that local clusters *U1* and *H1* are pointing to the same pattern as *C2: winter, moderate wind*. Clusters *U5* and *H5* are reflected in *C3: summer and soft wind*. Finally, cluster *H1* and *U1* can be seen in cluster *C2: winter, moderate wind*, even if *H1* has the variable *AmbientTemp* slightly higher.

However, we observe that *H4* and *U3* are similar to *C3: summer, soft wind*, although each one with a different characteristic, *H4* shows *AmbientTemp* slightly higher and *WindSpd* slightly lower than *C3*, however, *U3* has *AmbientTemp* slightly lower. So they could be placed between *C3* and *C4*.

This means that local analysis might elicit specific behaviors or operating conditions of particular turbines and provides more detailed information about the wind farms.

Going further, centroids of all the *N* clusters of each turbine can be compared together to built a distance matrix between turbines that allows a further turbine regrouping based on considering two turbines similar when they show similar clustering results, i.e., similar sets of *N* clusters each. As computing the distance between two turbines indeed involving the comparison of two sets of *N* centroids, the simplex algorithm has been used for this purpose.

Table 3. Class panel graphs for turbines H and U with Traffic Lights Panel super-imposed.



2.5. Generating Groups of Turbines Using the Average Distance Between Centroids

As commented on previously, pairwise comparisons of turbines are performed using the distance between their centroids. A global distance for each pair of turbines (calculated as indicated in Section 4.6) is presented in Table 4.

Turbines are now regrouped according to their distances and using the algorithm presented in Section 4.6. In this work, the *p-threshold* is optimized to generate between 3 to 5 groups, because it is a range of clusters that the experts can manage well (as they expect to identify between 3 to 5 prototypical turbines to visit for in situ inspection).

Table 4. Average distances between the different turbine pairs, calculated as indicated in Section 4.7.

	119	120	121	122	123	124	125	126	127	128	129	130	131	132	133	134
119		5.4	4.7	9.8	3.0	7.1	14.0	11.3	8.4	12.0	7.2	10.7	9.0	9.1	8.4	7.7
120	5.4		7.2	8.7	4.7	4.7	13.5	10.8	10.0	10.9	8.3	11.8	6.8	7.0	9.7	9.6
121	4.7	7.2		9.2	2.7	9.9	12.9	7.8	6.4	10.9	6.9	7.9	8.4	8.6	10.4	6.2
122	9.8	8.7	9.2		9.7	10	5.8	4.5	5.4	7.4	8.3	9.2	8.7	5.5	5.0	8.6
123	3.0	4.7	2.7	9.7		7.9	13.9	9.5	6.5	12.9	6.5	9.3	8.1	8.9	9.5	6.9
124	7.1	4.7	9.9	10.0	7.9		13.7	12.4	12.5	10.9	9.3	12.4	9.2	9.4	10.5	10.5
125	14.0	13.5	12.9	5.8	13.9	13.7		8.1	8.4	9.6	11.7	10.8	11.4	10.1	7.2	12.8
126	11.3	10.8	7.8	4.5	9.5	12.4	8.1		6.3	7.0	8.4	7.3	8.5	6.7	6.5	7.5
127	8.4	10.0	6.4	5.3	6.5	12.5	8.4	6.3		9.2	9.4	8.2	12.9	9.5	6.1	9.8
128	12.0	10.9	10.9	7.4	12.9	10.9	9.6	7.0	9.2		12.1	10.6	7.4	5.1	8.0	11.9
129	7.2	8.3	6.9	8.3	6.5	9.3	11.7	8.4	9.4	12.1		6.6	7.2	9.1	8.0	4.0
130	10.7	11.8	7.9	9.2	9.2	12.4	10.8	7.3	8.2	10.6	6.6		10.0	12.4	9.6	5.2
131	9.0	6.8	8.4	8.7	8.1	9.2	11.4	8.5	12.9	7.4	7.2	10.0		4.8	11.0	8.7
132	9.1	7.0	8.6	5.5	8.9	9.4	10.1	6.7	9.5	5.1	9.1	12.4	4.8		8.3	9.6
133	8.4	9.7	10.4	5.0	9.5	10.5	7.2	6.5	6.1	8.0	8.0	9.6	11.0	8.3		10.0
134	7.7	9.6	6.2	8.6	6.9	10.5	12.8	7.5	9.8	11.9	4.0	5.2	8.7	9.6	10.0	

The Table 5 contains the results of grouping turbines into 3, 4 and 5 groups (column *Number of groups*) . The group is shown in the *Group Id* column. Column *Turbine identifiers* indicates the turbine label. Columns within *Expert probability* indicate the probability of failure estimated by an expert for the system under evaluation during in situ inspections. Columns within *Maintenance events* indicates the number of interventions to repair the system under analysis (gearbox).

Table 5. Results for different p-threshold for wind farm *'Wf1'*, together with the expert-based probability of failure and the number of maintenance events generated by each group.

Number	Group	Turbine Identifiers	Expert Probability			Maintenance Events			
of Groups	Id		Mean	Median	sdv	Count	Mean	Median	sdv
3	1	119, 120, 121, 123, 127, 129, 134	0.282	0.176	0.197	166	23.714	24	6.047
	2	122, 125, 126, 128, 130, 131, 132, 133	0.452	0.394	0.219	191	23.875	22	8.855
	3	124	0.332	0.332	NA	35	35	35	NA
4	1	119, 121, 123, 127, 134	0.254	0.176	0.186	123	24.6	27	6.95
	2	122, 125, 126, 128, 129, 132, 133	0.49	0.554	0.232	176	25.143	23	8.802
	3	120, 124, 131	0.29	0.332	0.124	77	25.667	24	8.622
	4	130	0.356	0.356	NA	16	16	16	NA
5	1	119, 121, 123	0.151	0.172	0.04	68	22.667	19	9.074
	2	129, 130, 134	0.383	0.356	0.159	63	21	19	6.245
	3	122, 125, 126, 127, 132, 133	0.471	0.488	0.248	143	23.833	25	5.811
	4	120, 124, 131	0.29	0.332	0.124	77	25.667	24	8.622
	5	128	0.621	0.621	NA	41	41	41	NA

Since historical wind farm data is available and all events have been collected, the status of the wind turbines at each timestamp is known and can be used as a ground-truth for the evaluation of the discovered clusterings. The turbines that the specialist reported as the worse ones are the 125, 126, 128, 130, 131 and 133. In particular, the wind turbine 133 had broken the gearbox system and the wind turbine 128 had the gearbox changed before it broke. On the contrary, the turbines that we know that are the best ones are the 119 and 121. When analyzing the number of repairs, we see that regardless of the number of groups (3, 4 or 5), the group that had the more repairs always contains most of the damaged turbines, while the group that had fewer repairs contains most of the healthy turbines.

2.6. Comparison with Alternative Clustering Methods

Although that the use of SOM and further clustering over the SOM codes is the current state of the art in wind farm data-driven analysis. An elaborated proposal based on local analysis by wind turbine followed by a global regrouping of similar clusters is presented, this Section is devoted to comparing the achieved results with a much more classic approach that finds clusters avoiding intermediate SOM construction.

Hierarchical classical clustering has been performed over the normalized dataset for *Wf1* data, by using Ward's method [14] and Euclidean distance, as all variables are numerical. The resulting dendrogram is shown in Figure 5 and Calinski-Harabasz index [15] has been optimized to determine the resulting number of clusters. A cut in 4 clusters is suggested. CPGs and TLPs build over the resulting clusters apparently show good results and provide clusters with quite clear interpretation. However, when the daily classification of the wind turbine is temporally plotted, as we can see in Figure 6, the clusters change chaotically from one day to another, as if the wind turbine experimented pattern changes asynchronously along time. This pattern seems not to be realistic and makes difficult the understanding of the wind turbine operation regime. In Figure 7 it can be seen the corresponding temporal evolution of the daily classification of the wind turbine operation, obtained by applying the local analysis methodology proposed in Section 4.5. It is clear that the proposed method is able to capture much better the intrinsic stationarity of the aero-generation phenomenon.

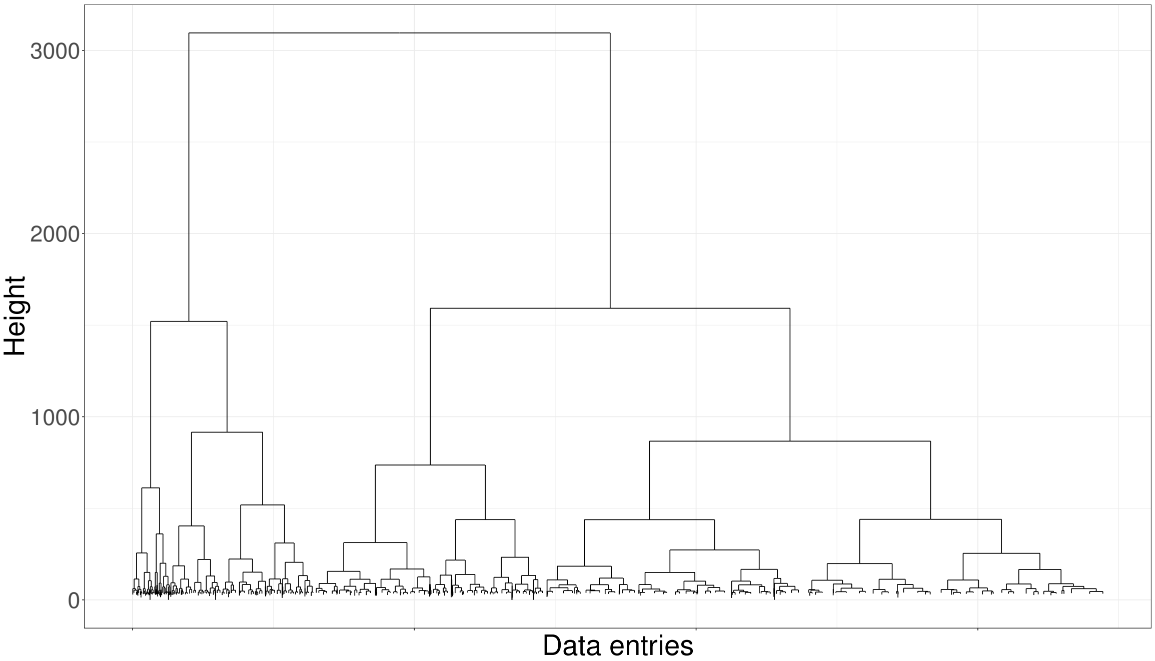


Figure 5. Dendrogram of applying Wards method to the complete wind turbine dataset of ’Wf1’. Only the top 40 trees are shown for easier visualization.

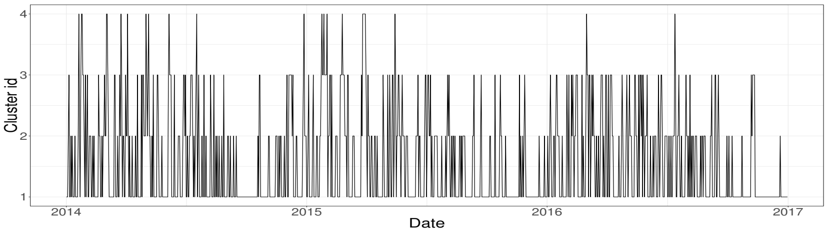


Figure 6. Class sequencing according to Wards method for the turbine’s id 132.

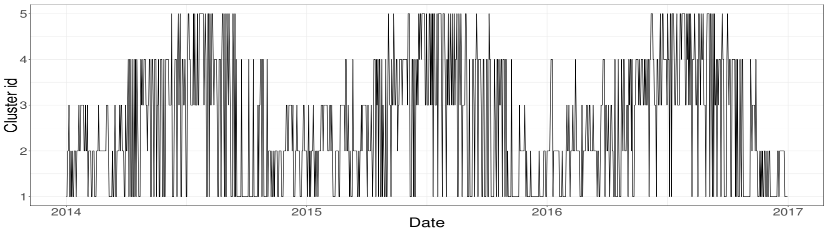


Figure 7. Class sequencing according to the method proposed in this work for the turbine’s id 132.

2.7. Validation with Additional Wind Farms

Based on these evidences, we extend the application of the proposed local analysis methodology to the rest of the wind farms.

The same procedure was applied to the gearbox system failures for the other two wind farms. Results are shown in Table 6. For each wind farm, turbines have been clustered according to the proposed methodology in 4 clusters. In the Table, the column “Group Id” indicates the class identifier. The number of turbines involved in each of these classes is shown in column “N^{o} of turbines”. “Expert probability” columns provide the mean, median and standard deviation of the probability of failure estimated by the expert for the turbines of the class, whereas columns “Maintenance events” contains statistics related to the real number of maintenances required in the turbines of each group.

Table 6. Results for a group size of 4 for ′codes2′ and ′moncayuelo′ wind farms; includes the probability generated by an expert per group and the real number of maintenance events observed.

Wind	Group	Nr of	Expert Probability			Maintenance Events			
Farm	**Id**	**Turbines**	**Mean**	**Median**	**sdv**	**Count**	**Mean**	**Median**	**sdv**
moncayuelo	1	7	0.14	0.14	0.09	8	1.14	1.15	2
	2	20	0.29	0.26	0.16	46	2.3	1.53	2
	3	4	0.16	0.15	0.04	16	4	2.45	4
	4	1	0	0	0	0	0	0	0
codes2	1	37	0.24	0.20	0.15	49	1.32	1.29	1
	2	9	0.11	0.10	0.09	9	1	0.84	1
	3	3	0.67	0.7	0.15	4	1.33	1.41	2
	4	1	0.2	0.2	NA	1	1	NA	1

In the first wind farm, *moncayuelo*, we can see that group 2 has the higher probability of failure given by the expert (mean and median), besides the number of repairs (mean and median) is also the highest of the four groups. Therefore, this group will contain the unhealthiest turbines. In group 3, even though the expert determines an intermediate failure probability, the amount of repairs is high, indicating that it is a group that could also be considered as unhealthy (or close to) when creating a classification of turbines. In group 1, both the expert and the number of repairs are among the lowest, so it groups turbines in excellent condition. Group 4 contains the turbine in the best condition of the farm since the number of repairs is zero and also the expert has assigned the lowest value (0).

In the second wind farm, *codes2*, group 1 contains the turbines in an intermediate state of health according to both failure probability of the expert and repairs. Group 2 has the turbines in better condition according to both criteria. In group 3 we have the turbines with the highest failure rate and that the expert also considers that the probability of failure is high. Finally, group 4 contains a turbine that does not resemble any of the other groups. Even if it is not considered as damaged, the turbine belonging to this group needs to be further analyzed to clarify whether it is another mode of operation or hides some other problem.

3. Discussion and Future Work

Identifying health status of wind turbines is a severe problem that cannot be tackled by using simple data analysis methods because the interactions between the factors impacting in particular kind of failures are too complex. As it has been seen in the paper, the plain hierarchical clustering is not able to tackle the stationarity involved in the process.

The main contribution of our system is the proposal of an intensive data-driven methodology able to automatize the identification of groups of turbines with similar behaviors, that can support the company staff in selecting a reduced number of representative turbines for in situ inspections. This solves a critical issue in the company, related to human and economic resources involved.

The proposal provides a data-driven methodology based on a strategic combination of SOM, hierarchical clustering, post-processing and simplex-based matching, that was resulting successful in grouping turbines according to its healthy state for different group sizes providing an understanding of this status. The groups contain turbines with a similar number of maintenance interventions, also in accordance with the expert evaluation, validating that the groups are well derived.

To do that, we develop a strategy based on the comparison of the centroids of the local BMUs, which facilitates the characterization of each turbine as a vector of operational status (the N local centroids). This allows a further re-grouping of turbines by merging in groups those that behave similarly as a whole, i.e., have similar vectors of operational regimes. The introduction of CPG and TLP as interpretation oriented tools was of major importance to elicit the meaning of the patterns identified and supporting the final diagnoses made by the experts about operational regimes of the turbines.

The importance of our proposed method relays in the fact that this initial clustering of turbines can be done automatically, generating 3 (or more) groups, each one with turbines in a similar healthy state. Thanks to the application of interpretation tools such as CPG and TLP, it has been possible to understand the information captured by the SOM, clearly identifying at least four different types of turbine operating modes that directly impact in energy production rates. Therefore, the human expert can focus his/her work only on a subset of turbines, according to the problem to be solved. Thus we save precious and expensive time, especially when large farms or many different farms have to be handled by the same specialist.

Moreover, our system allows for identifying interactions in the behavior of the variables involved, from an N-dimensional analysis and particular areas of some problematic turbines. Therefore, after the identification of the unhealthy classes, in which the use of CPG and TLPs is supporting the conceptualization of the clusters, our system allows monitoring the time evolution of any turbine, by visualizing how their clusters/centroids evolve and identifying if they are moving towards the distribution of an unhealthy class. This automatic process is of paramount importance to reduce costs and handle an important number of turbines and wind farms.

The process has been automatized and scaled to be in production in a real company, and it provides a helpful framework to identify a reduced set of turbines to be inspected in situ.

The proposed method has also been applied to two additional wind farms to validate their real usability.

To the best of our knowledge, this is the first exploratory work that combines SOM, clustering based on BMUs and turbine characterization through CPG and TLPs altogether. Many aspects would need an in-depth, and other possibilities can be considered. For example, the clustering algorithm used on BMUs has a real effect on the final clusters, and also the way we group turbines based on the distance between centroids by means the simplex algorithm. Here, several measures of distance/correlation could be used and will be explored in future work. Also, the variables considered for the problem to be modeled could be automatized through a feature selection algorithm, instead of using a human expert. This feature selection algorithm will have a significant effect on the result. Hence an in-depth investigation should also be carried on. Finally, the optimum number of clusters derived in Section 4.5 could also be determined by evaluating the quality of the clusters generated in each turbine. Possible relevant metrics to do so are the *Davies–Bouldin index* [15,16] and the *Silhouettes index* [17]. These metrics should be computed for each dendrogram, exploring a reasonable range for the number of clusters, and for each turbine individually. Then, we could calculate the average for each metric for turbines within the same amount of clusters. The trade-off between the two results could be used to determine the optimal number of clusters to be applied for all the turbines. Regarding the CPG and the TLP, there is work in progress to implement several automatic criteria to built the TLPs from some overlapping indicators between the local distributions of variables inside each class. The degree of overlapping between classes will determine the three levels of each variable, the assignment of a color to each cell of the TLP. The automatic interpretation of the patterns would also be included in the standard automatic processing of the wind farms to define strategic in situ inspections.

4. Materials and Methods

4.1. Data

The SCADA data used in this work follows the IEC 61400-25 format [18]. The data was gathered via an OPC (OLE for Process Control) [19] with frequencies of 5 or 10 min, for a rich set of variables. Each sensor usually provides *minimum*, *mean*, *maximum* and *standard deviation* values for each variable.

The dataset is stored in a local database, which has been recording values from the SCADA over the years. The dataset is structured as a table, with the time evolution in rows and sensors variables in columns. The wind farms used in this work are detailed in Table 7.

Table 7. Summary of the dataset used in this work. The Table shows the number of wind turbines, number of years of historical data available, the frequency of each wind turbine, number of variables, number of events (alarms in this case) and the total number of registers evaluated by each experiment. Each experiment corresponds to a different wind farm.

Turbine	Number of Turbines	Years	Rows / Year	Variables	Triggered Alarms	Total Registers Evaluated
Wf1 3MW (confidential)	16	3	52.560	181	709.972	2.522.880
Acciona Wind Power AW-1500 'codes2'	50	3	52.560	163	80.194	7.884.000
Acciona Wind Power AW-1500 'moncayuelo'	32	4	52.560	142	21.742	6.727.680
Total	98	10			811.908	17.134.560

According to the fault to be detected, an expert decides which variables will be used to analyze the system. In this work, gearbox problems will be focused because, as already mentioned above, it is one of the main important turbine systems, being the responsible for expensive maintenance costs due to its components. These variables could also be obtained through different *Feature Selection* algorithms see Table 8, although according to previous works the variables selected by an expert give excellent results [20]. All the analysis carried on will be in daily scale.

Table 8. Different *Feature Selection* algorithms used in [20] to identify relevant variables, as an alternative to expert-based variables selection.

Algorithm	Author
Mutual Information Feature Selection (MIFS)	Battiti [21]
Conditional Mutual Information (CMI)	Cheng et al. [22]
Joint Mutual Information (JMI)	Yang and Moody [23]
Min-Redundancy Max-Relevance (mRMR)	Peng et al. [24]
Double Input Symmetrical Relevance (DISR)	Mayer and Bontempi [25]
Conditional Mutual Info Maximisation (CMIM)	Fleuret [26]
Interaction Capping (ICAP)	Jakulin [27]

Maintenance interventions directly related to the gearbox have been kept on the database, as well as a failure probability analysis obtained by an expert after his analysis of oil and temperature. This information will be used to evaluate the quality of the groups generated by our proposed procedure.

The variables selected by an expert as relevant for the gearbox operation are introduced below. Figure 8 provides an overview of a wind turbine and these variables:

Power The power generated by the wind turbine in KW.
GearboxOilTemp The temperature of the gearbox oil , in degree Celsius.
GearboxBearingTemp The temperature of the gearbox bearing (output side) , in degree Celsius.
AmbientTemp The external temperature of the environment, in degree Celsius.
RotorSpd The speed of the rotor main shaft before gearbox, in revolutions per minute (RPM).
WindSpd The wind speed in m/s measured by the anemometer at the wind turbine's nacelle.

Additionally, a new variable is internally created to evaluate the results, as a non-linear combination of two of the variables provided by the SCADA system. It reflects a parameter often used by experts in the interpretation of the health status of wind turbines and enhances the interpretation process.

PowerWind_ratio The ratio of the *Power* variable divided by the wind speed.

As we try to discover abnormal behaviors and wind turbines are different among them, data is normalized to Z-score to eliminate wind turbine heterogeneity from the analysis. However, the normalization factors must be saved to reconstruct the original variables for graphical analysis. The extreme values (outliers) are set to NA, and then the rows that contain any variable with NA is removed since it represents less than the 5% of the total number of registers. Further analysis focalized to registers with missing data are in progress. A complete set of guidelines to be taken into account in preprocessing can be found in [28].

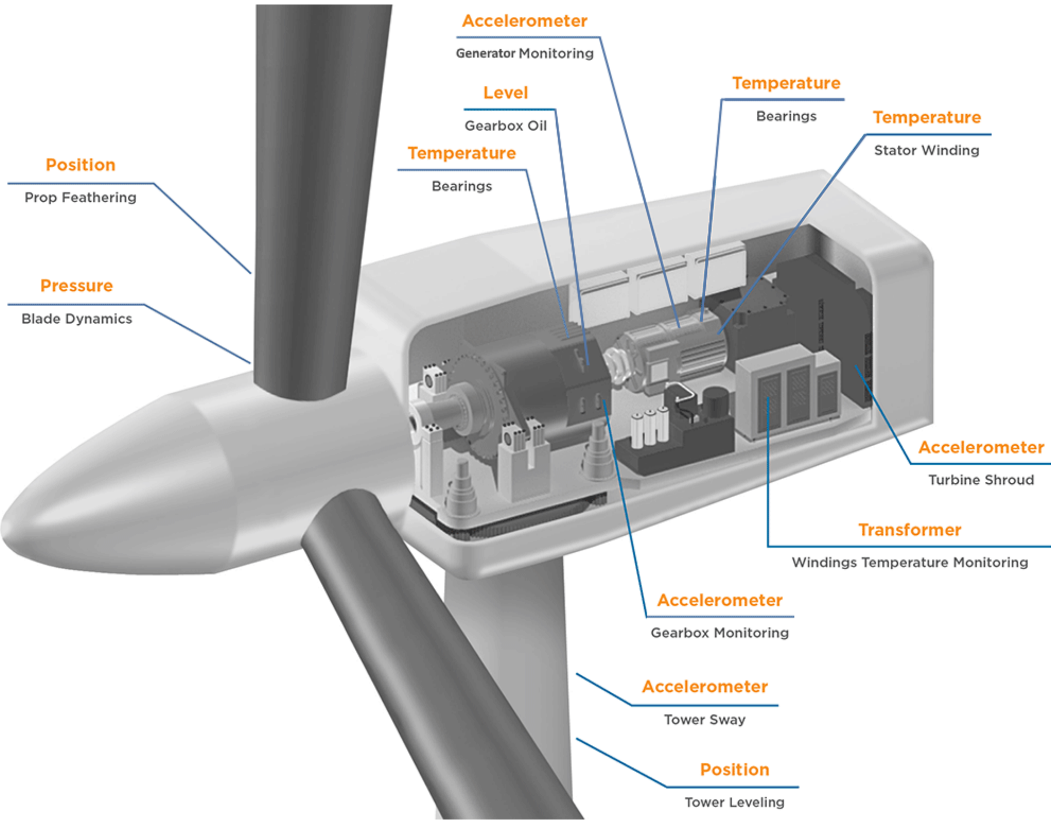


Figure 8. Wind Turbine system and sensors. Adapted from TE connectivity (http://www.te.com/).

4.2. Methodology Overview

Self organizing maps (SOM), introduced by T. Kohonen [10], is a type of unsupervised ANN mainly employed in feature reduction and data visualization. This neural network has been used in many different kinds of applications, ranging from speech processing (the original field in which Kohonen presented it, [29]), seismic data analysis [30], image processing [31], genetic data [32], etc.

The SOM uses an unsupervised algorithm based on competitive learning, in which the output neurons compete with each other to be activated, with the result that only one is activated at a given time. The result is that the neurons are forced to organize themselves in a specific manner which generates the map. Usually, the nodes of the network are organized in a regular 2D space, in which each unit (neuron) in the input layer is connected to all neurons in the output layer. Each connection has a weight and, this weight will be adjusted during the process with the aim of mapping input patterns to the output 2D structure by preserving the topology. This means that points that are near each other in the multiple dimensional input space will be mapped to nearby map units in the 2D SOM map. Therefore, SOM can be used as a cluster analyzing tool of high-dimensional data. Also, SOM has the ability to generalize, which means that the network can recognize or characterize entries it has never seen before. A new input vector is assimilated with the unit on the map to which it is mapped to.

Self Organizing Maps have been used in the condition monitoring area on several occasions. Some works [13,33–36] use the map generated with all turbines to explore how the data is distributed by performing an analysis in the unified distance matrix (U-matrix), which is a way to visualize the distances between neurons. Other works go one step further by applying clustering on the U-matrix to find patterns on the map [13,36–39].

In our case, we go beyond the classical approach proposed in the literature by adding a second step of the analysis in which the SOM is subdivided into sub-maps local to each one of the turbines (see details in Section 4.5 to find the behavioral patterns shown by every single turbine). A further regrouping of these patterns in a final step (see Section 4.6) leads to a global grouping of these patterns

The interest of this approach is to get an in-depth comprehension of which turbines are in better or less operational mode and helps to decide which specific turbines have to be inspected.

For this purpose, understanding of the meaning of both global or local clusters become critical.

The results of the clustering methods in general, including SOM, require some further processing to understand which are the meaning of the discovered clusters and to properly conceptualize them [12]. Classically, U-Matrix visualizations are used to interpret SOM results, and also projections of observed variables onto the SOM map. Another visual output derived from the SOM map are the heatmaps of the variables which is done individually for every single variable and provide a way of identifying the areas of the SOM map associated higher and lower values of the variable. However, it is difficult to get a global perspective, as these tools analyze every single variable separately.

Being a real application that needs to provide support to a real strategic decision in the company, getting a global overview of what the patterns are telling us regarding turbines' health is of vital importance. Thus, specific interpretation-oriented tools are introduced to support the understanding of the patterns discovered by the SOM (see Section 4.7). A crucial step in this unsupervised data-mining process is to transform the results of the SOM into understandable knowledge for providing effective decision-making support [40].

The Figure 9 contains a diagram of the whole proposed process, also in the following sub-Sections, specific details on each one of the steps of the proposed process are provided.

4.3. Software

The software selected to generate the model and analyze the data is **R** version 3.4.3. The library needed to generate the SOM maps is *Kohonen package* by Ron Wehrens and Johannes Kruisselbrink [41]. To create the clusters the base package hclust by Fionn Murtagh and Pedro Contreras [42]. The CPGs and TLPs are generated with by the KLASSv18 proprietary software by Karina Gibert [43] which is a data mining software, specifically designed to introduce expert knowledge and semantics into clustering processes of heterogeneous data and contains a specific module of interpretation oriented tools. Finally, the Simplex method implemented into the turbine's centroids pair computation is provided by the linprog package by Arne Henningsen [44]. To reproduce and repeat the results the same dataset must be used and also the same random seed, in our case, we defined the number 1 as the random seed (*set.seed(1)* in **R**).

4.4. Selection of the Optimal SOM Size

The size of the map depends to a large extent on the dimension of the input data. According to [13], the initial map size should contain $n = \lceil 5\sqrt{R} \rceil$ neurons, where R is the number of registers and the result is finally rounded up (ceiling). However, [33] indicates that it is possible to obtain the optimal dimensions through an exploratory way by using the *U-Matrix* [45].

In this work, we will use some metrics to set the map size. These will allow us to automate this step. It is known that larger map sizes produce over-fitting because fewer records are associated with each neuron, and hence each neuron specializes in a particular record, which is not the goal of the SOM [46]. However, a too small map size would not end up collecting the particular behavioral records that are not associated with isolated neurons, considered as *outliers*. That is why a trade-off has been sought using the metrics described in [10,47]. These metrics are the Topographic error (*TE*) and the Quantization error (*QE*), the first one increases as the map gets bigger meanwhile the second one decreases. With these two metrics behaving in opposite ways, a balanced size could be achieved between the two of the different maps generated.

The *TE* is calculated by analyzing each input register and considering the 1st and 2d best matching unit (BMU). If they are not adjacent, the *TE* is increased by 1, otherwise is kept at its previous value. When all the registers have been analyzed, the total value is divided by the total number of registers obtaining its mean. The *TE* value increases if the map becomes larger, due to the increase in the number of units and therefore the decreasing on the probability of having adjacent 1st and 2d BMUs.

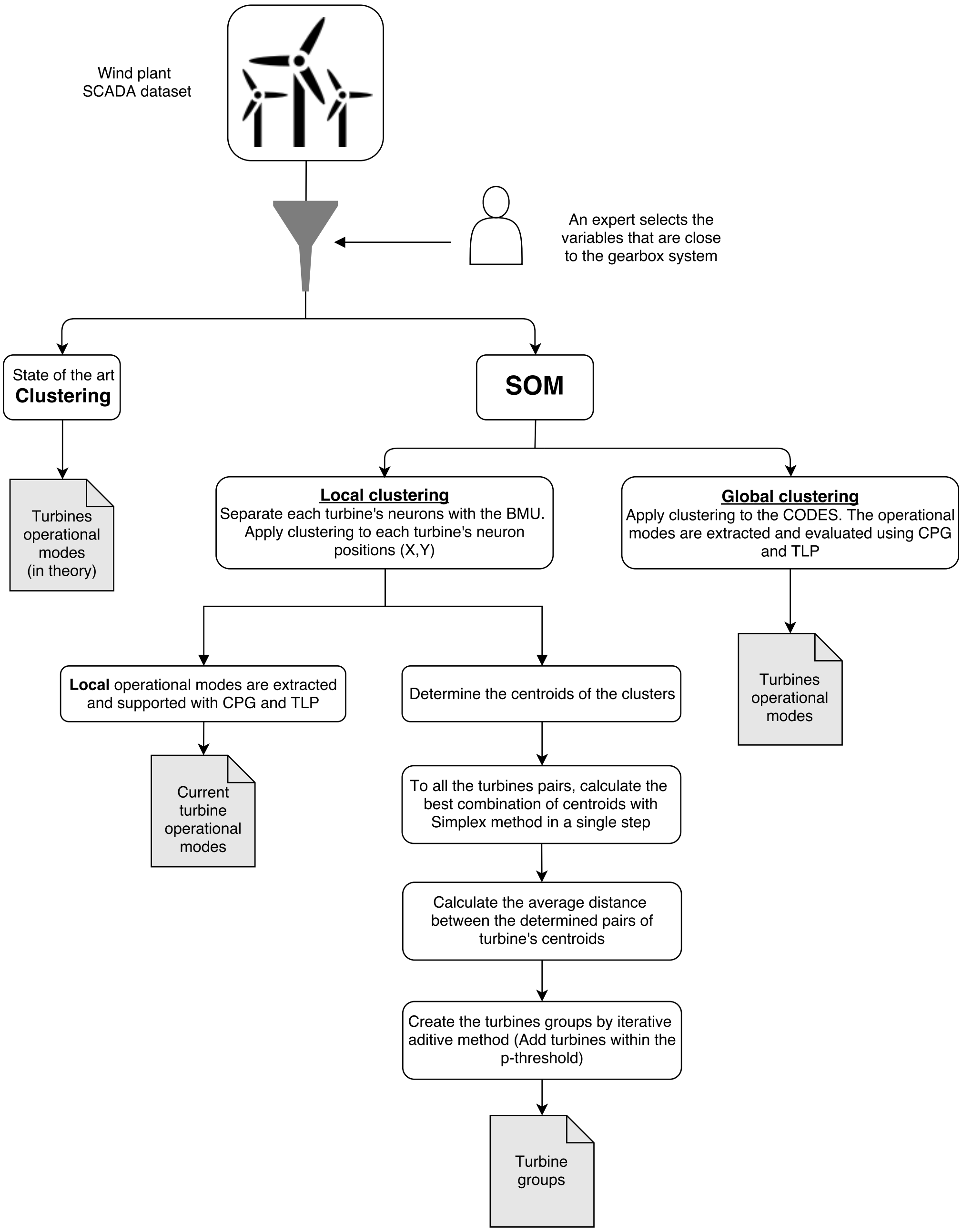

Wind plant
SCADA dataset
An expert selects the variables that are close to the gearbox system
State of the art **Clustering**
SOM
Turbines operational modes (in theory)
Local clustering Separate each turbine's neurons with the BMU. Apply clustering to each turbine's neuron positions (X,Y)
Global clustering Apply clustering to the CODES. The operational modes are extracted and evaluated using CPG and TLP
Local operational modes are extracted and supported with CPG and TLP
Determine the centroids of the clusters
Turbines operational modes
Current turbine operational modes
To all the turbines pairs, calculate the best combination of centroids with Simplex method in a single step
Calculate the average distance between the determined pairs of turbine's centroids
Create the turbines groups by iterative aditive method (Add turbines within the p-threshold)
Turbine groups

Figure 9. Overview of the proposed methodology, starting from the SCADA data and showing the results obtained at each step.

On the contrary, the *QE* decreases as the map become larger, since it quantifies the distance from each register to the assigned BMU. Hence, the larger the map is, the higher the resolution and the smaller all these distances because there is a higher chance that the register comes closer to its assigned BMU.

To find the best size for the SOM, the metrics mentioned above are first of all normalized in a range between 0 and 1. Then, the (normalized) values of the two metrics are obtained for several SOM versions with different sizes. Finally, the cut-off point between the two crossing curves of TE and QE will be used as the best size for the SOM. These metrics do not have a linear behavior, so the cut-off point substantially varies depending on the explored size of the SOM. Experimental tests showed that these metrics have an exponential behavior, one with negative decay while the other with positive decay. A linear shape may indicate over-fitting, and a different range of sizes should be tested, usually smaller than the previous ones.

4.5. Generating Sub-Maps by Turbine

In the previous Section, a method to derive an optimal size of the SOM for a given dataset is proposed, based on a trade-off between two metrics, evaluated in a range of different sizes.

In this Section, a proposal to subdivide the SOM map into sub-maps local to each turbine is presented.

First of all, for each turbine, a list of BMUs are obtained. This procedure is done by first selecting the registers from the original dataset that corresponds to the target turbine and then, the BMU of each of those registers is identified in the SOM results. The selected subset of BMUs provides a sub-map of the same size as the original, but with a subset of visible neurons (those activated by the registers of the target turbine) and invisible neurons (the neurons without registers of the target turbine). So, by comparing sub-maps of different turbines among them, it is possible to discover turbines sharing the same activation zones, which are candidates to be grouped. Identifying which turbines show common patterns of SOM activation is easy from a graphical point of view. However, to implement efficiently in a production phase the procedure in the daily activities of the company, this step has to be performed automatically.

The main challenge is that the specific BMUs activated by two similar turbines are not exactly the same, even if they are in close neighborhoods. Thus a local clustering of the BMUs activated by a single turbine and a centroids-based representation of these clusters will provide a synthetic view of the activation areas of a given turbine and will allow further comparisons to detect groups of similar turbines automatically. The clustering algorithm used in this work is based on the *Hierarchical clustering* [48].

4.6. Re-Grouping Turbines

Provided that in this particular context all turbines of a given wind farm are technologically similar, it has been seen that most of them show the same number of clusters *N*. This is very interesting because it enables pairwise comparisons between turbines in terms of Euclidean distances between their centroids-vector derived in Section 4.5.

However, the cluster identifier of a particular operational regime (like optimal production, for example) can change from one turbine to another one, since discovered clusters are automatically named by the algorithm. Thus, given a pair of turbines T and T′, cluster 1 in turbine T might point to a different scenario than cluster 1 in turbine T′. This means that even though the behavior of a certain turbine can be synthesized by a vector of *N* centroids, one per cluster, distances between pairs of turbines cannot be directly computed. The Simplex method [49] is introduced for this purpose, to find the permutation of centroids of turbine T′ that minimize the total distance to the centroids of turbine T ($d_{min}(T, T')$). The combination of centroids between turbines that generates $d_{min}(T, T')$ is the optimal one, and provides the correspondence between clusters in T and those in T′. The distance between the two turbines T and T′ is then defined as $\frac{d_{min}(T,T')}{N}$, that is, the average distance between pairwise centroids between T and T′.

Repeating this procedure with all distinct pairs of turbines a square (symmetric) distance matrix between turbines is obtained in Figure 4. Each row or column in Table 4 identifies a turbine.

Based on this distance matrix, a further grouping of similar turbines can be pursued. A density-based like clustering process is performed by setting a threshold *p-threshold* that determines the neighborhood of a certainly visited wind turbine and all other wind turbines inside this neighborhood are included in the same cluster. The *p-threshold* must be a positive real number from 0 to 1 which defines the proportion of the total distance range on the table. The process starts by finding the cell containing the smallest distance (*v-min*) in the table. The row that contains this cell identifies the first turbine T visited and its distances to the other turbines. Each column in the matrix represents another turbine (namely T′). All turbines T′ such that $d(T,T') < v\text{-}min + p\text{-}threshold$ will be added together in the cluster C_T. After the first group is set, the rows and columns identifying the turbines of it are eliminated from the distance matrix and the process is repeated to determine the next group, until all the turbines are clustered in some group.

Since the distance matrix is quadratic in the number of turbines, and this is not a huge dimensionality, this process can be repeated with several values of *p-threshold*, starting by a small value like (0.1) and increasing by steps for a posteriori evaluation of the preferred *p-threshold*. Higher values of *p-threshold* generate fewer groups which are more general. Lower values of *p-threshold* give more groups which are more specific.

In order to check the validity of the groups generated by this procedure, they will be compared with the failure probability generated by the experts in in situ inspections, as indicated in Section 4.1 (qualitative evaluation). Also, the maintenance and failure events of the turbines will be used by calculating the statistics of these indicators to check that the groups contain turbines with similar problems (quantitative evaluation).

4.7. Post-Processing the Results of Self-Organizing Maps for a Better Understanding of the Discovered Patterns

As mentioned before, a couple of tools are introduced as a post-processing of the SOM results and the hierarchical clustering processes used in this work. Both of them were designed with the aim of helping experts to conceptualize and label the resulting classes. Originally, CPGs and TLPs [50] were designed in the context of hierarchical clustering. In this paper, for the first time, they are used on clusters induced from a SOM network.

The CPG is based on a simple idea but resulted very powerful in previous real applications where clusters understanding was critical. It is based on placing in a single panel the conditional distributions of the variables with regards to the clusters. Columns correspond to variables and rows to clusters. Histograms or box-plots are displayed for numerical variables and bar-charts for qualitative ones [50]. It allows to identify particularities of classes in regards of specific variables. Basically, the inherent nature of the clustering is based on the idea that observations group in different clusters because, on the one hand, they can be distinguished by some characteristic behaving differently in one or other cluster and, on the other hand, they must share some distinctive commonalities with the other observations in the same cluster. The CPG permits a quick analysis to identify these distinctive commonalities.

One step forward in the level of abstraction of the interpretation-support tool is the TLP. TLP is a symbolic post-processing of the clustering results proved extremely useful and well-accepted by domain experts in several real applications [11,50]. TLP exploits the association between the traffic light colors and the main central trend of the variables in every class to help the expert to understand the clusters and to support the conceptualization. In fact, it can be visually built upon the image proposed by the CPG, or automatically computed in terms of overlapping measures among the conditional distributions of a variable in the several clusters. The main issue is that deciding whereas high values of the variable will be assigned red or green color is associated with the semantics of the variable itself, so bringing semantics into the picture of the interpretation process in a formal way. In this particular application, for example, producing high levels of power is better than low production, and that is why high levels will be associated with green and low with red color. In [12] an extension to *annotated-TLP* is presented, where the basic color of the cell is desaturated with a darker tone proportionally to the variability inside the class, so the expert is able to catch, from the picture, which are the cells which they can trust their decisions.

In this work, both CPG and TLP have been built to understand the patterns resulting from the hierarchical clustering of the SOM cell prototypes (BMU), as well as to understand the patterns resulting from the local analysis of each specific turbine when clustering their positions in the SOM map.The software KLASSv18 has been used for this purpose [43].

Acknowledgments: Financial support by the Agency for Management of University and Research Grants (AGAUR) of the Catalan Government to Alejandro Blanco-M.. is gratefully acknowledged.

Author Contributions: Alejandro Blanco-M. defined the main idea and structure of the work. Karina Gibert devised and evaluated the Class panel graphs and Traffic Lights Panel to support the conceptualization of the clusters results and made the comparison with the hierarchical basic clustering. Jordi Solé-Casals and Pere Marti-Puig made the experiments and the physical interpretation of clustering results. Jordi Cusidó provided the interface to the different Wind Plant's SCADA to download the data.

Conflicts of Interest: The authors declare no potential conflict of interest.

References

1. REN21 Secretariat. *Renewables 2016-Global Status Report*; Technical Report; Renewable Energy Policy Network for the 21st Century: Paris, France, 2016; ISBN 978-3-9818107-0-7.
2. European Comission. *Communication From the Commission to the European Parliament, The Council, the European Economic and Social Committee and the Committee of The Regions: Developing the European Dimension in Sport*; Technical Report 30.01.2013; Commission of the European Communities: Brussels, Belgium, 2011.
3. Eurostat. *Energy Balance Sheets 2011–2012*; Technical Report 9; Eurostats (European Union) publications Office: Luxembourg, Luxembourg, 2014.
4. Milborrow, D. *Operation and Maintenance Costs Compared and Revealed*; Haymarket Business Media; Wind Stats; London, UK 2006; Volume 19, pp. 1–87.
5. Besnard, F.; Bertling, L. An approach for condition-based maintenance optimization applied to wind turbine blades. *IEEE Trans. Sustain. Energy* **2010**, *1*, 77–83, doi:10.1109/TSTE.2010.2049452.
6. Aubrey, C. Supply Chain: The Race to meet Demand. *Wind Directions*; EWEA; Brussels, Belgium, 2007; pp. 27–34.
7. McMillan, D.; Ault, G.W. Quantification of Condition Monitoring Benefit for Offshore Wind Turbines. *Wind Eng.* **2007**, *31*, 267–285, doi:10.1260/030952407783123060.
8. Santos, P.; Villa, F.L.; Renones, A.; Bustillo, A.; Maudes, J. An SVM-Based Solution for Fault Detection in Wind Turbines. *Sensors* **2015**, *15*, 5627–5648.
9. Vestas R+D. *General Specification VESTAS V90 3.0 MW*; Technical Report; Vestas Wind Systems; Central Denmark Region: Aarhus, Denmark, 2004.
10. Kohonen, T. *Self-Organizing Maps*, 3rd ed.; Springer: Berlin/Heidelberg, Germany, 2001; Volume 30, p. 502.
11. Gibert, K.; Garcia-Rudolph, A.; Garcia-Molina, A.; Roig-Rovira, T.; Bernabeu, M.; Tormos, J. Response to TBI-neurorehabilitation through an AI& Stats hybrid KDD methodology. *Med. Arch.* **2008**, *62*, 132–135.
12. Gibert, K.; Conti, D. aTLP: A color-based model of uncertainty to evaluate the risk of decisions based on prototypes. *AI Commun.* **2015**, *28*, 113–126.
13. Vesanto, J.; Alhoniemi, E. Clustering of the self-organizing map. *IEEE Trans. Neural Netw.* **2000**, *11*, 586–600, doi:10.1109/72.846731.
14. Ward, J. Hierarchical grouping to optimize an objective function. *J. Am. Statist. Assoc.* **2012**, *58*, 236–244.
15. Calinski, T.; Harabasz, J. A dendrite method for cluster analysis. *Commun. Stat. Simul. Comput.* **1974.**, *3*, 1–27.
16. Davies, D.L.; Bouldin, D.W. A Cluster Separation Measure. *IEEE Trans. Pattern Anal. Mach. Intell.* **1979**, *PAMI-1*, 224–227, doi:10.1109/TPAMI.1979.4766909.
17. Rousseeuw, P.J. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.* **1987**, *20*, 53–65, doi:10.1016/0377-0427(87)90125-7.

18. International Electrotechnical Commission (IEC). *International Standard IEC CEI 61223-3-2*; International Electrotechnical Commission (IEC): Geneva, Switzerland, 2007.
19. OPC Fundation. *OPC Is the Interoperability Standard for the Secure and Reliable Exchange of Data in the Industrial Automation Space and in Other Industries*; OPC Fundation: Scottsdale, AZ, USA 2016.
20. Blanco, M.A.; Solé-Casals, J.; Marti-Puig, P.; Justicia, I.; Cardenas, J.J.; Cusido, J. Impact of target variable distribution type over the regression analysis in wind turbine data. In Proceedings of the 2017 International Work Conference on Bio-Inspired Intelligence, Intelligent Systems for Biodiversity Conservation, IWOBI 2017-Proceedings, Funchal, Portugal, 10–11 July 2017.
21. Battiti, R. Using mutual information for selecting features in supervised neural net learning. *IEEE Trans. Neural Netw.* **1994**, *5*, 537–550.
22. Cheng, H.; Qin, Z.; Feng, C.; Wang, Y.; Li, F. Conditional mutual information-based feature selection analyzing for synergy and redundancy. *ETRI J.* **2011**, *33*, 210–218.
23. Yang, H.H.; Moody, J.E. Data Visualization and Feature Selection: New Algorithms for Nongaussian Data. *NIPS Citeseer* **1999**, *99*, 687–693.
24. Peng, H.; Long, F.; Ding, C. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Trans. Pattern Anal. Mach. Intell.* **2005**, *27*, 1226–1238.
25. Meyer, P.E.; Bontempi, G. On the use of variable complementarity for feature selection in cancer classification. In *Applications of Evolutionary Computing*; Springer: Berlin, Germany, 2006; pp. 91–102.
26. Fleuret, F. Fast binary feature selection with conditional mutual information. *J. Mach. Learn. Res.* **2004**, *5*, 1531–1555.
27. Jakulin, A. Machine Learning Based on Attribute Interactions. Ph.D. Thesis, Univerza v Ljubljani, Ljubljana, Slovenia, 2005.
28. Gibert, K.; Sànchez-Marrè, M.; Izquierdo, J. A survey on pre-processing techniques: Relevant issues in the context of environmental data mining. *AI Commun.* **2016**, *29*, 627–663.
29. Kohonen, T. The "Neural" Phonetic Typewriter. *Computer* **1988**, *21*, 11–22, doi:10.1109/2.28.
30. Klose, C.D. Self-organizing maps for geoscientific data analysis: Geological interpretation of multidimensional geophysical data. *Comput. Geosci.* **2006**, *10*, 265–277.
31. Jiang, Y.; Zhou, Z.H. SOM ensemble-based image segmentation. *Neural Process. Lett.* **2004**, *20*, 171–178, doi:10.1007/s11063-004-2022-8.
32. Nikkilä, J.; Törönen, P.; Kaski, S.; Venna, J.; Castrén, E.; Wong, G. Analysis and visualization of gene expression data using Self-Organizing Maps. *Neural Netw.* **2002**, *15*, 953–966, doi:10.1016/S0893-6080(02)00070-9.
33. Huysmans, J.; Baesens, B.; Vanthienen, J.; Van Gestel, T. Failure prediction with self organizing maps. *Expert Syst. Appl.* **2006**, *30*, 479–487, doi:10.1016/j.eswa.2005.10.005.
34. Du, M.; He, Q. A SCADA Data based Anomaly Detection Method for Wind Turbines. In Proceedings of the 2016 China International Conference on Electricity Distribution (CICED 2016), Xi'an, China, 10–13 August 2016; Volume 7, pp. 10–13.
35. Zhao, L.; Pan, Z.; Shao, C.; Yang, Q. Application of SOM neural network in fault diagnosis of wind turbine. In Proceedings of the International Conference on Renewable Power Generation (RPG 2015), Beijing, China, 17–18 October 2016; pp. 2–5.
36. Hernández, L.; Baladrón, C.; Aguiar, J.M.; Carro, B.; Sánchez-Esguevillas, A. Classification and clustering of electricity demand patterns in industrial parks. *Energies* **2012**, *5*, 5215–5228, doi:10.3390/en5125215.
37. Yang, L.; Ouyang, Z.; Shi, Y. A modified clustering method based on self-organizing maps and its applications. *Procedia Comput. Sci.* **2012**, *9*, 1371–1379, doi:10.1016/j.procs.2012.04.151.
38. Kiang, M.Y. Extending the Kohonen self-organizing map networks for clustering analysis. *Comput. Stat. Data Anal.* **2001**, *38*, 161–180, doi:10.1016/S0167-9473(01)00040-8.
39. Gil, A.; Sanz-Bobi, M.A.; Rodríguez-López, M.A. Behavior Anomaly Indicators Based on Reference Patterns—Application to the Gearbox and Electrical Generator of a Wind Turbine. *Energies* **2018**, *11*, doi:10.3390/en11010087.
40. Gibert, K.; Rodríguez-Silva, G.; Annicchiarico, R. Post-processing: Bridging the gap between modelling and effective decision-support. The Profile Assessment Grid in Human Behaviour. *Math. Comput. Model.* **2013**, *57*, 1633–1639, doi:10.1016/j.mcm.2011.10.046.
41. Wehrens, R.; Buydens, L. Self- and Super-organising Maps in R: The kohonen package. *J. Stat. Softw.* **2007**, *21*, doi:10.18637/jss.v021.i05.

42. Murtagh, F.; Contreras, P. Methods of Hierarchical Clustering. *arXiv* **2011**, arXiv:1105.0121.
43. Gibert, K.; Nonell, R. Pre and Postprocessing in KLASS. In Proceedings of the iEMSs 4th Biennal Meeting: International Congress of Environmental Modeling and Software (DMTES'08 Workshop) iEMSs, Barcelona, Spain, July 2008; Volume III, pp. 1965–1966.
44. Henningsen, A. Linprog R Package. 2012. Available online: https://cran.r-project.org/web/packages/linprog/index.html (accessed on 03 March 2018).
45. Ultsch, A. U*-Matrix: A Tool to visualize Clusters in high dimensional Data. *Computer* **2003**, *52*, 1–10.
46. Lampinen, J.; Kostiainen, T. Overtraining and model selection with the self-organizing map. In Proceedings of the International Joint Conference on Neural Networks, 1999. IJCNN'99, Washington, DC, USA, 10–16 July 1999; Volume 3, pp. 1911–1915.
47. Khalilia, M.; Popescu, M. Topology preservation in fuzzy self-organizing maps. *Stud. Fuzziness Soft Comput.* **2014**, *312*, 105–114, doi:10.1007/978-3-319-03674-8_10.
48. Ben-dov, M.; Feldman, R. *Data Mining and Knowledge Discovery Handbook*; Springer: Berlin, Germany, 2010; pp. 321–352.
49. Nelder, J.A.; Mead, R. A Simplex Method for Function Minimization. *Comput. J.* **1965**, *7*, 308–31.
50. Gibert, K.; Conti, D.; Vrecko, D. Assisting the end-user in the interpretation of profiles for decision support. An application to wastewater treatment plants. *Environ. Eng. Manag. J.* **2012**, *11*, 931–944.

© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).

3.2. Effects of the Pre-processing Algorithms in Fault Diagnosis of Wind Turbines

Pere Marti-Puig, Alejandro Blanco-M, Juan José Cárdenas, Jordi Cusidó, Jordi Solé-Casals, Effects of the pre-processing algorithms in fault diagnosis of wind turbines, Environmental Modelling & Software, 2018, ISSN 1364-8152, https://doi.org/10.1016/j.envsoft.2018.05.002

(PREPRINT) Effects of the Pre-processing Algorithms in Fault Diagnosis of Wind Turbines

Pere Marti-Puig[a,*], Alejandro Blanco-M.[a,b], Juan José Cárdenas[b], Jordi Cusidó[b], Jordi Solé-Casals[a]

[a]*Data and Signal Processing Group, U Science Tech, University of Vic - Central University of Catalonia, c/ de la Laura 13, 08500 Vic, Catalonia, Spain*
[b]*Smartive-ITESTIT SL, Catalonia, Spain*

Abstract

The Wind sector has roughly 2200M€ of profit losses due to wind turbines failures and these failures doesn't contribute to the goal of reducing greenhouse gases emissions of many states. The 25-35% of the generation costs are operation and maintenance services. To lower this ratio, the wind turbine industry are backing on the Machine Learning techniques over SCADA data. This data can contain errors produced by missing data, miss calibrated sensors or human errors. Each kind of error must be handled carefully since extreme values are not always due to noise or data reading errors. This document evaluates the impact of removing extreme-outliers values applying widely used techniques like Quantile, Hampel and Extreme Studenized Deviation filters with the recommended cut-off values. Experimental results on real data show that removing outliers systematically is not a good practice, leading to an increment of the operation and maintenance costs of the park.

Keywords: Wind Farms, SCADA data, Pre-processing, Outliers, Fault Diagnosis, Renewable Energy

[*] Corresponding Author
Phone: +34 938815519
Fax: +34 938814307
Email: pere.marti@uvic.cat

1. Introduction

The reduction of greenhouse gases emissions and the independence of the fossil fuels are main goals of many states (Fabiani Appavou (2016)). That can be achieved by producing electricity using sustainable sources (Eurostat (2013)). For instance, the EU commission established a series of demanding climate and energy targets to be met by 2020 that includes the use of at least 20% of energy coming from renewable sources EU commission (2008). Wind power is the most growing renewable source (Fabiani Appavou (2016)), however the operation and maintenance of the wind turbines account for 25% to 35% of the generation costs (Milborrow (2003)). In order to increase the economic competitiveness with respect to fossil fuels and accelerate the transition towards ecologically sustainable systems, there is a need of more efficient management and this requires highly monitoring of wind turbines. Modern wind turbine records more than 200 analogous variables (Pedro Santos (2015)) at intervals of 5 to 10 minutes by means of their SCADA (Supervisory Control and Data Acquisition) system, therefore automatic supervision systems have to be developed in order to detect and fix possible failures.

The SCADA system collects data in different parts of the turbine, which are grouped into systems (Vestas R+D (2004)) providing information about: temperatures, electrical indicators, physical positions, speeds, vibration, etc. Therefore these systems generate a huge amount of data (Valeri Voev,Siemens A.G. (2014)), which has to be preprocessed and modeled in a feasible time (Justin Heinermann (2015)). Due to this, cloud-based platform services seem to be a good option to process SCADA data taking into account its processing scalability (Rodero-Merino (2011)) and high availability to build prediction engines for early diagnosis (called prognosis), improving the efficiency of the wind farms that is translated into a bigger profit or a reduction of generation costs (F. Besnard (2010)). These prognosis systems have to work in a wide set of wind turbines, with different models, manufacturers and SCADA-data configurations.

Raw data obtained from SCADA contains several kind of errors, which can be categorized as: missed data caused by communications failures, presence of extreme values due to sensors failures, data coming from poorly calibrated sensors or by replaced sensors which report outputs in a different range, errors in the SCADA system or even human errors (Gray (2011)). As a result, prognosis models reduce their performance and hence, operation and maintenance costs are increased. Therefore, previously to apply prognosis algorithms, a preprocessing step must be implemented. The importance of that step is often underestimated considering its great impact in the final results.

This study shows the importance of the outliers into prognosis models for wind turbines. We reveal that systematically removing outliers, habitually considered as noise or extreme values, is not a good strategy since important information of the system malfunction is removed, generating high accuracy rates in the training step but low accuracy rates in real-time testing step. This document is organized as follows: Section 1 contains the Introduction and objectives of the paper; Section 2 is devoted to describe the real data used in all this research, the pre-selection of variables done by an expert, the description of three filtering algorithms in order to detect and eliminate outliers and the strategy proposed to measure the effect of the filtering step. Results and discussions are presented in Section 3, while Section 4 contains the final conclusions of our research.

2. Materials and methods

This section covers the techniques that have been applied in order to identify and remove the outliers/extreme values. For each case we present a description of the method and the data work-flow over the algorithm. Each method will be applied separately over the same input dataset. In order to demonstrate the effect of the method, we execute an independent analysis for each case using all the available information on several wind turbines. Then, we analyze the results generated by the models taking into account the technique used when removing outliers/extreme values and comparing them with the results obtained with the original values (i.e.: without removing outliers/extreme values).

Each independent analysis starts from a dataset which is split in two parts. The datasets are split taking into account the time arrangement since taking random samples introduces non-available patterns in the *train* dataset which will affect the model estimation and the final results. Some works point out the benefits of removing outliers in order to improve final results in a machine learning system. That is the case of (A. Christya (2015), McClelland (2000), Patel (2011), Jason W. Osborne (2004), Denis Cousineau (2010)). But even if in many cases this procedure can be correct, to our knowledge, there is no study showing how to proceed in the case of SCADA data coming from wind turbines. Hence, we will investigate this effect in our research by evaluating the impact of removing outliers on real noisy dataset from the wind turbines with some of the most widely used univariate outliers removal algorithms (Marjan Bakker (2014)).

2.1. Data background

Data is produced by wind turbine's SCADA that follows the IEC 61400-25 format IEC (2006) which provides data with a structure of *logical devices* representing wind turbines and *logical nodes* representing physical wind turbine systems and subsystems. Data was gathered via an Open Platform Communications (OPC) OPC Fundation (2016) each 5 or 10 minutes which reports events, instant values and statistics indicators. Only the events and the statistics indicators are kept. Each sensor has its statistics indicators which commonly are *mean*, *min*, *max* and *standard deviation* values.

In our case, data are stored on a local database, which has been recording values from the wind plants over the years. Data is structured as a *table* with entries containing different instances of each sensor at each time interval. Events containing information about the system errors are stored in a different table since they are recorded in different format and contain the specific time in which the event raised. This events are entries which are categorized as *alarms* (failure states) and *warnings* (non-important events like machine stop by maintenance service, start or stop messages).

An example of the data format generated is shown in table 1, which contains registers (rows) and variables (columns) ordered by *date_time* variable, i.e. the insertion time on the database, when a notification of variable and value is submitted by the OPC.

date_time	power	bearing_temp	gen_1_speed	temp_oil_mult
2014-12-08 06:20:00	1701.17	29.40625	1291.84	36.39
2014-12-08 06:30:00	1583.11	28.14462	1055.23	22.08
2014-12-08 06:40:00	1664.03	28.03261	1132.16	23.43
2014-12-08 06:50:00	1722.47	29.8721	1312.66	22.68
2014-12-08 07:00:00	1647.91	29.0121	1231.78	21.82

Table 1: Example of the data analyzed (part of a real table)

Turbine Model	Num. of machines	Num. of years	Rows per year	Num. of variables	Num. of triggered alarms	Total registers evaluated
Fuhrlander fl2500	5	4	105.120	303	72.422	2.102.400
Vestas V90 ’wf1’	7	4	52.560	194	9.681	1.471.680
Vestas V90 ’wf2’	13	4	52.560	63	5.063	2.733.120
Siemens Izar 55/1300	26	1	52.560	24	369.218	1.366.560
Wfa H1	1	7	52.560	406	83.716	52.560
Total	52	20		992	540.100	7.726.320

Table 2: Data summary

2.2. Input data pre-selection

In order to study an specific type of alarm, an expert have to choose a subset of events based on the physical system or subsystem to be analyzed. Therefore, the expert will select events and variables from all the available variables and events, those containing information about the failures of the wind turbines (Vestas R+D (2004)). In our experiments we focus in the transmission system, and more specifically the *Main Bearing subsystem*, which supports the rotor of the wind turbine and is at the origin of many alarms.

Based on the selected subset of events, a contrast of hypothesis is generated in order to select the variables that are more related with the selected events. The null hypothesis H_0 is defined as *no statistical relevance on the change of a variable mean* on the day when alarm/failure event is present. The alternative hypothesis H_a defines that a variable presents a *statistically relevant difference in its mean value* when an alarm/failure even is present at that day. The interval of confidence is defined at 95% which determines a p-value of 0.05. Any variable which has a p-value smaller than 0.05 is considered as a possible input variable for the model. We then sort all the candidates to be input variables of the model, from smaller to bigger p-value, and select the first six variables in order to analyze them. We do not consider all the possible variables for computational reasons.

2.3. ESD 3σrule

Extreme Studentized Deviate test (ESD) is a statistical test allowing to detect outliers in a univariate data set having normally distributed population. ESD defines that any point being away more than t standard deviations from the mean is an extreme-outlier value. As shown in equation 1, any value falling outside the interval is considered an outlier:

$$(\mu - (t * \sigma)) < x_i < (\mu + (t * \sigma)) \tag{1}$$

Where:

$$\begin{aligned} x_i &: \text{is the } \; i \text{ entry from a single variable X} \\ \mu &: \text{is the mean of the current variable X} \\ t &: \text{is the number of standard deviations} \\ \sigma &: \text{is the standard deviation of a single variable X} \end{aligned}$$

The most common value for the threshold t is $t = 3$, which means that all points that deviates 3σ from the mean value will be rejected. Therefore, about 0.3% of the observed data will be considered as an outlier. This method is very sensitive to distributions that contains many outliers, so the threshold value at 3σ helps to minimize this effects, but the method will fail with data containing more than 10% of outliers, as indicated in (Pearson (2005))

In our case we have our data in a table with the format indicated in 1. Variables are in columns and instances in rows, so we will implement the ESD test as follows:

Algorithm 1 3 ESD outlier filter

```
procedure CLEANESD($variables$)
    $t \leftarrow 3$
    for all $variable$,$varID$ in $variables[:,:]$ do :
        $mean \leftarrow$ mean($variable[:]$)
        $\sigma \leftarrow$ sd($variable[:]$)
        for all $entry$,$entID$ in $variable[:]$ do :
            if $entry < mean - (t * \sigma)$ or $mean + (t * \sigma) < entry$ then
                $outlierList[varID, entID] \leftarrow entry$        ▷ save the outlier for analysis
                $entry \leftarrow NULL/NAN$        ▷ is labeled as an outlier, value removed
            end if
        end for
    end for
end procedure
```

In order to maintain the original structure of our data (number of rows and columns of the table) the outliers found by the test will be changed to NAN (Not a Number) or NULL values.

2.4. Adjusted box-plot rule

Another commonly used rule to detect outliers is based on the distance of the points being above of the third quartile or below of the first quartile. This quartiles values determines the acceptable range of the values following the next expression 2:

$$(Q_1 - (c * IQR)) < x_i < (Q_3 + (c * IQR)) \tag{2}$$

Where:

$$\begin{aligned} x_i &: \text{is the } \ i \text{ entry from a single variable X} \\ Q_1 &: \text{is the } \ first \text{ quartile of the current variable X} \\ Q_3 &: \text{is the } \ third \text{ quartile of the current variable X} \\ IQR &: \text{is the interquartile as in equation (3)} \\ c &: \text{is the number of interquartile range} \end{aligned}$$

$$IQR = (Q_3 - Q_1) \tag{3}$$

A common value for c is $c = 1.5$. This method is less sensitive to outliers than the ESD. However, is well suited for asymmetric distributions since it does not depend of a ”center” of the data (Pearson (2005)). On the contrary, it’s usually too aggressive since it declares as outliers many nominal observations determined as non-outliers by a human expert.

The simplified algorithm has been implemented as follows:

Algorithm 2 Quantile outlier filter

```
procedure CLEANQUANTILE($variables$)
    $c \leftarrow 1.5$
    $outlierList \leftarrow []$                                ▷ The outlier list is initialized
    for all $variable$,$varID$ in $variables[:,:]$ do :
        $Q1 \leftarrow$ quantile($variable[:], 25\%$)
        $Q3 \leftarrow$ quantile($variable[:], 75\%$)
        $IQR \leftarrow Q3 - Q1$
        for all $entry$,$entID$ in $variable[:]$ do :
            if $entry < (Q1 - c * IQR)$ or $(Q3 + c * IQR) < entry$ then
                $outlierList[varID, entID] \leftarrow entry$        ▷ save the outlier for analysis
                $entry \leftarrow NULL/NAN$             ▷ is labeled as an outlier, value removed
            end if
        end for
    end for
end procedure
```

2.5. Hampel identifier

The Hampel identifier is based on two robust measures of location and scale, the median and the median of the absolute deviations (MAD) from the median, respectively. Observations too far from the median of the data with respect to their MAD are declared to be outliers (Christophe Leys (2013)). Again, a proportion factor k will modulate how to calculate that distance. In our case, this factor is calculated using the inverse of the Gaussian cumulative distribution (Φ^{-1}) function calculated on the 75% confidence interval which takes the area until the quantile Q_3 (3/4):

$$k = 1/\left(\Phi^{-1}(3/4)\right) \approx 1.4826 \tag{4}$$

So having k, the accepted range for the detection procedure is as follows:

$$(\hat{X} - (k * MAD)) < x_i < (\hat{X} + (k * MAD)) \tag{5}$$

Where:

x_i : is the i entry from a single variable X
$\hat{X}$: is the median of single variable X
k : is the constant scale factor calculated as in equation (4)
MAD : is the median absolute deviation calculated as in equation (6)

$$MAD = median(|x_i - \hat{X}|) \tag{6}$$

Where:

x_i : is the i entry from a single variable X
$\hat{X}$: is the median of single variable X

The median and the median of the absolute deviations are more robust to the influence of outliers than the mean and standard deviation. This means that the Hampel identifier is more effective than the ESD identifier in outlier detection, although as a quantile based filter it can be too aggressive, declaring many points as outliers even if they really are not so. The simplified algorithm has been implemented as follows:

Algorithm 3 Hampel outlier filter

```
procedure cleanHampel($variables$)
    $k \leftarrow 1.4826$
    $outlierList \leftarrow []$                              ▷ The outlier list is initialized
    for all $variable$,$varID$ in $variables[:,:]$ do :
        $median \leftarrow$ median($variable[:]$)
        $MAD \leftarrow$ mad($variable[:]$)
        for all $entry$,$entID$ in $variable[:]$ do :
            if $entry < (median - k * MAD)$ or $(median + k * MAD) < entry$ then
                $outlierList[varID, entID] \leftarrow entry$       ▷ save the outlier for analysis
                $entry \leftarrow NULL/NAN$                 ▷ is marked as outlier,value removed
            end if
        end for
    end for
end procedure
```

2.6. Evaluation

The evaluation of the above mentioned methods will be carried out using the dataset of the wind farms indicated in the table 2. The dataset was divided in train and test dataset but preserving the temporal order. As pointed before, this is a crucial point in order to avoid the generation of new patterns that are not in fact present in the data. The filtering methods will be applied on the train datasets and the models will be tested on the (unknown) test dataset. All the experiments will be performed on one variable which corresponds to the temperature from the wind turbine gearbox.

In order to quantify the effect of the filtering step, we will use numerical indicators over the models results. One of the most effective method to evaluate the impact of such filters on machine learning algorithms is to implement a normality model based on Partial Least Squares (PLS) (Wold (2001)), which can be evaluated using a mean squared error (MSE). Therefore, we will compute the model using the same train dataset with and without outliers and then we will apply it to the test dataset. Apart from the MSE we will also use scatter plots of the real and estimated values and compute the best regression line that fits to it. Ideally, if there is a perfect relation between points, we will obtain a 45° gradient line.

3. Results

Starting from the same dataset, we run several test with and without filtering outliers and evaluate its effect using PLS normality models (Wold (2001)). The input variables where manually selected for each wind plant since each wind turbine model has it owns variable names.

For instance, for the first plant which is composed of Vestas V90 machines, an expert determined and reduced the input set of variables and selected the target variable for the model, which in this case is *gear_oil_temp_avg*. This variable has the distribution shown in figure 1 for the first turbine named *T13*, this will be indicated as *target* or *target variable* henceforth.

The following list shows the input variables ordered by most to less importance for the model. This variables are from the wind turbine systems (Vestas R+D (2004)). As you can see the maximum importance is between the *target variable* and *gear_bearing_temp_avg* which are from components that are physically close and connected by metal parts, which transfers the heat.

- gear_bearing_temp_avg: Temperature of bearing that holds the rotor with blades.
- power_avg: Average power generated
- wind_avg: Average wind speed
- hydraulic_oil_temp_avg: Temperature of the oil which cool the gearbox.
- blades_pitchangle_max: Angle of the Wind Turbine blades.
- blades_bladea_controlvoltage_min: Voltage of the motors which controls the angle of the blades.

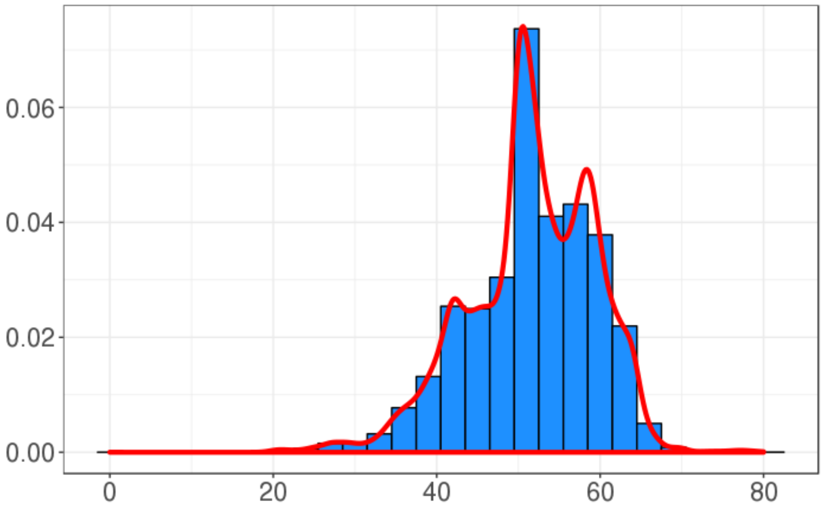
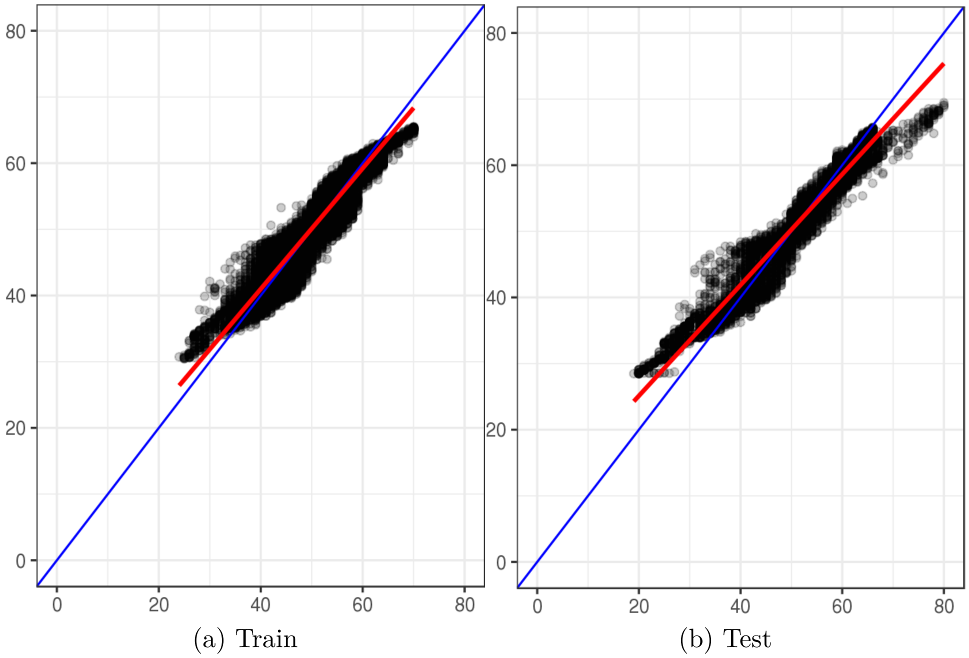


Figure 1: Histogram of target variable



(a) Train

(b) Test

Figure 2: Train and Test estimation vs. real value of target variable

Results for the model derived without filtering are shown in figure 2 for the train and test datasets. On the left we can see the result of the estimated model over the same train dataset (in black) and the perfect 1:1 relation (45°) in blue, as a reference. The red line is the regression line obtained using the real points and those generated by the model, which is slightly leaned with respect to the reference. In this example the obtained gradient has a value of 42.4° for the training dataset, which indicates that the model is not estimating all the values perfectly even on the same training dataset. Result measured with MSE gives a value of 2.0768. If we analyze now the test dataset of the right side of the figure, we observe that now the gradient is 40° with an MSE of 2.612 which is worst than the previous one. This is what we expected as we are now dealing with new (unknown) data.

Now we have to compare these results with the ones obtained after filtering the data with the proposed systems. The following subsections will present them individually.

3.1. ESD 3σrule

With the data being filtered by the ESD rule, many periods of alarm were labeled as outliers, identified as outliers and alarms on the figures 3 and 4, each one corresponding to a different variable. In all these figures, outliers (which are values outside the interval) are in blue color. The values which have been labeled as outliers by the algorithms but at the same time an alarm was reported by the wind turbine are in red color. Alarms are indicated in cyan color whereas non filtered data, which is the data at the input of the PLS model, is indicated in yellow. Two variables are detailed, corresponding to the variables presenting the highest amount of alarms identified as outliers. This will reduce the number of alarms feed to the machine learning model and therefore will reduce its prediction capability. The outliers detected by this algorithm represents the 2.1% of the training data, taking into account all the variables.

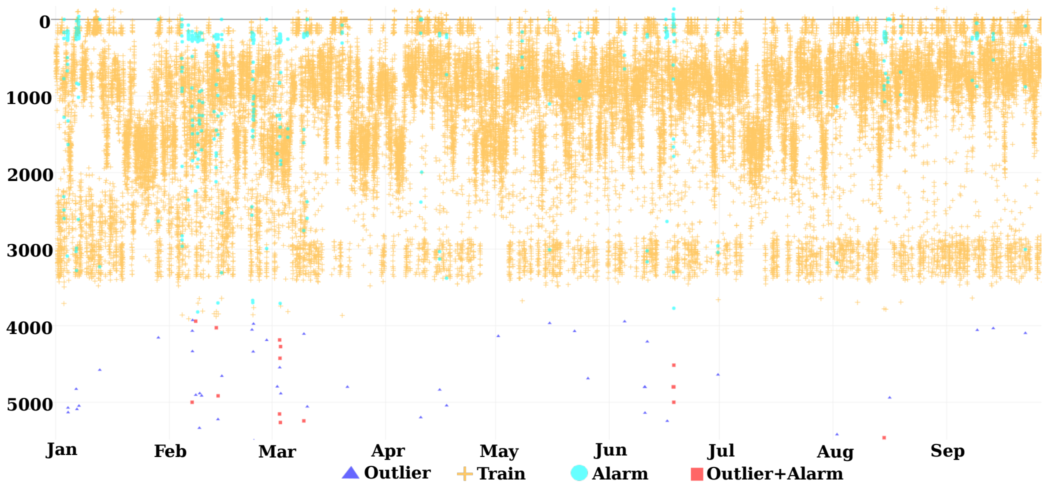


Figure 3: Marking of outliers, alarms on variable *blade control voltage* on filtered dataset

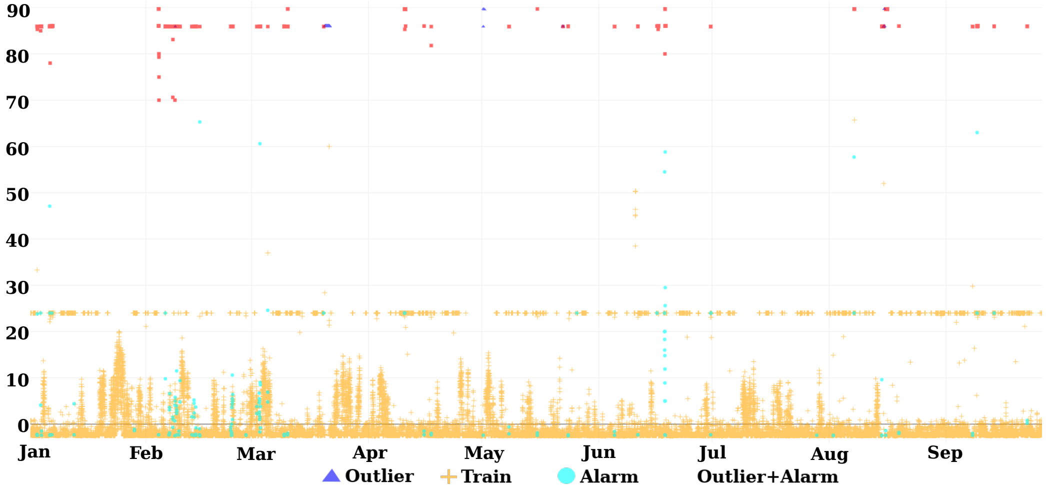
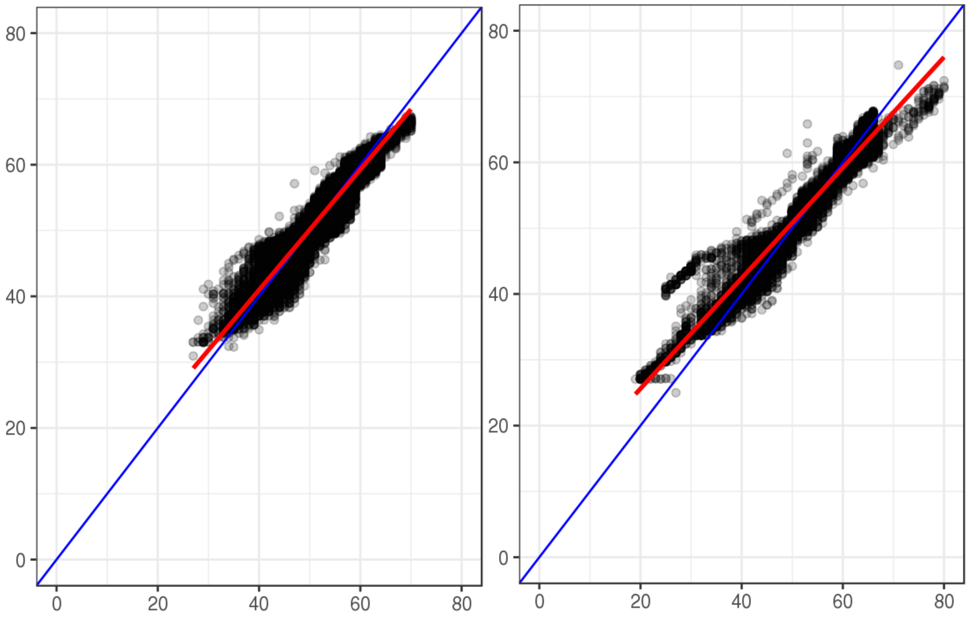


Figure 4: Marking of outliers, alarms on variable *blade pitch angle max* on filtered dataset

The impact on the model results are show in figure 5 which reveals an increase of the performance on the train dataset (left) filtering the outliers: MSE error decreases from 2.0768 to 1.963 and the slope increases from 42.4° to 42.5°: But on the other side, when testing the model with the test dataset (right), the MSE increased from 2.612 to 2.836 which is a sign of worst prediction capability. Concerning the slope of the regression line, even the gradient is almost the same, there is a new small region of new points far from the diagonal line indicating that the model is behaving worse.



(a) Train

(b) Test

Figure 5: Train and Test estimation vs. real value of target variable

3.2. Adjusted box-plot rule

Using the same procedure as in previous filtering strategy, we analyze now the effect of the Adjusted box-plot rule (quantile filter). In this case, when filtering the data, many periods of alarm were labeled as outliers as we can see in 6 and 7. Following the same color coding as in the previous case, outliers are in blue color, outliers that at the same time an alarm was reported are in red color, alarms are in cyan color whereas non filtered data is indicated in yellow color. We show two of the variables which presents the most amount of alarms identified as outliers. Again, filtering will reduce the number of alarms feed to the machine learning model and therefore will reduce its prediction capability. The outliers detected by this algorithm accounts for the 20.8% of the training data, taking into account all the variables.

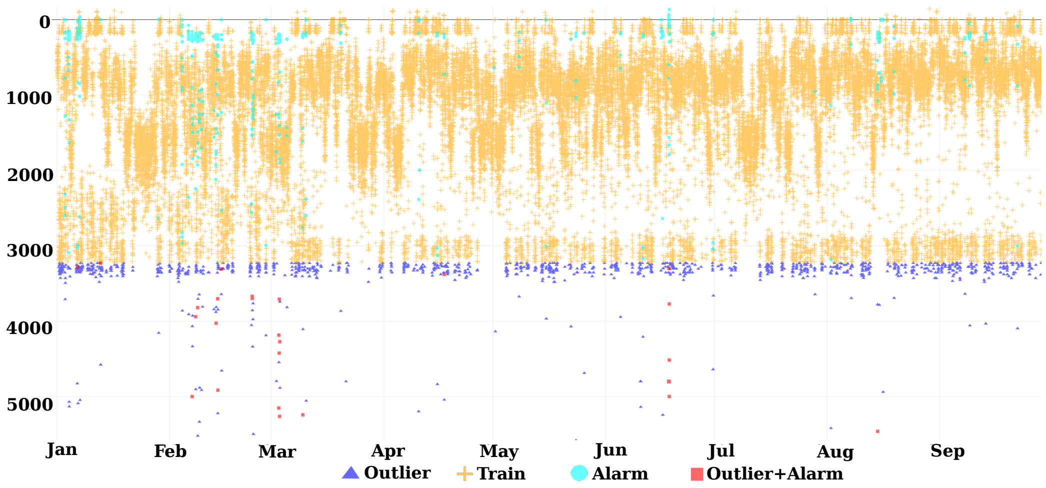


Figure 6: Marking of outliers, alarms on variable *blade control voltage* on filtered dataset

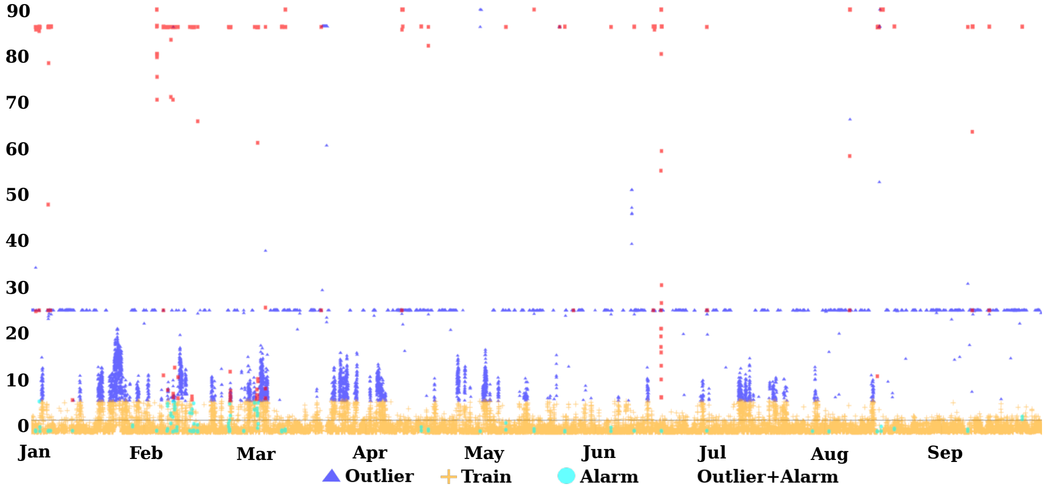
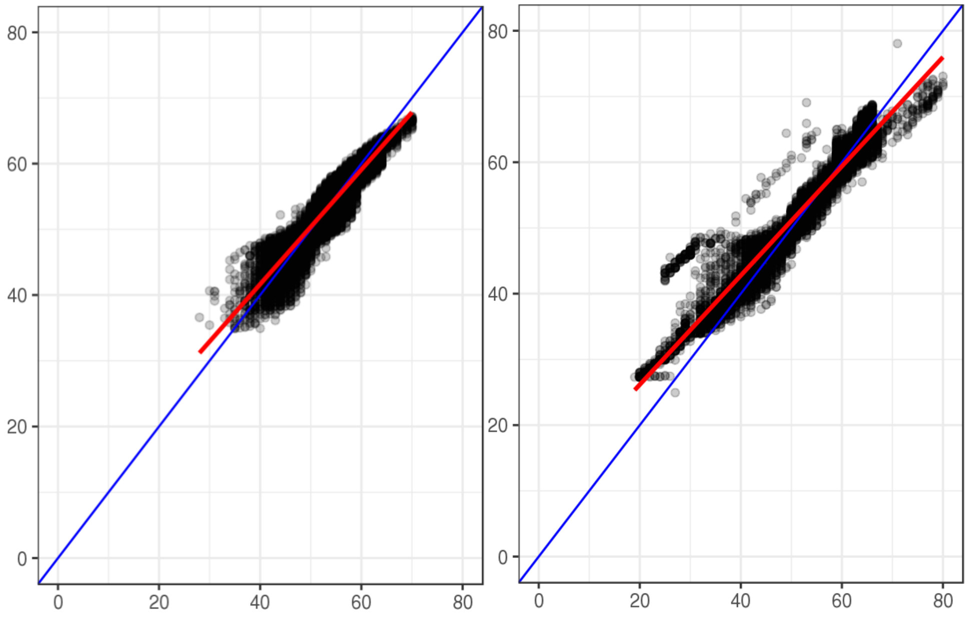


Figure 7: Marking of outliers, alarms on variable *blade pitch angle max* on filtered dataset

The impact on the model results are show in figure 8 which reveals an increase on the performance on the training dataset (right) with an MSE decreasing from 2.0768 to 1.893. This value is smaller than the one obtained with the ESD filter due the robustness of quartile to the outliers. On the contrary, results of the test dataset (left) reveals a higher increase of the MSE from 2.612 to 3.096, which means that the model generalization performance is worse than the ESD and the plot of estimation vs. real values indicates a decrease in the angle of the linear regression, from 40° to 39.7°. Some holes on the region between 50-60 can be observed due to the removal of possible input variables that generates the values of this area.



(a) Train (b) Test

Figure 8: Train and Test estimation vs. real value of target variable

3.3. Hampel identifier

Finally the third filtering system is analyzed in the same way as the previous ones. Figures with the results, using the same kind of representations, are shown in 9 and 10 for each variable. Again, we show two of the variables which presents the highest amount of alarms identified as outliers. The outliers detected by this algorithm represents the 32.2% of the training data, taking into account all the variables.

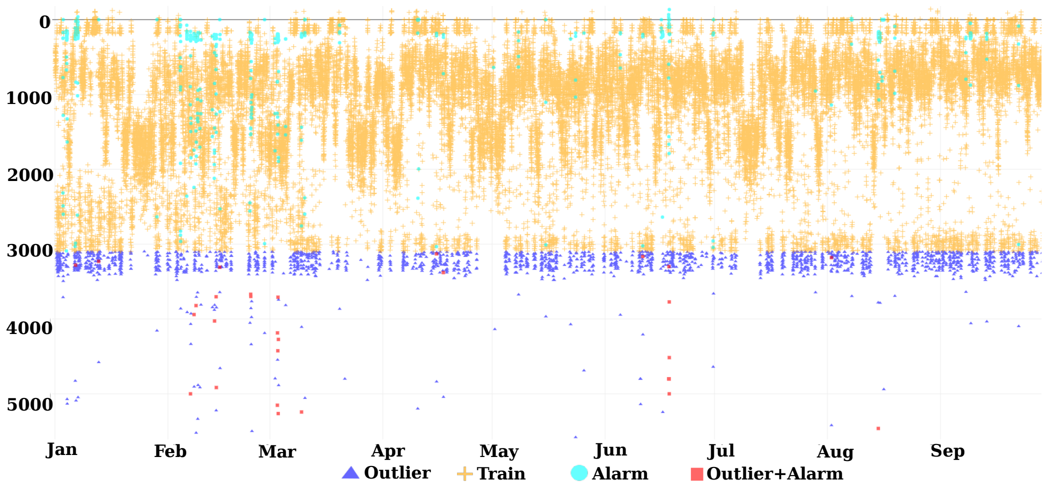


Figure 9: Outlier marking vs. alarms on variable *blade control voltage* on filtered dataset

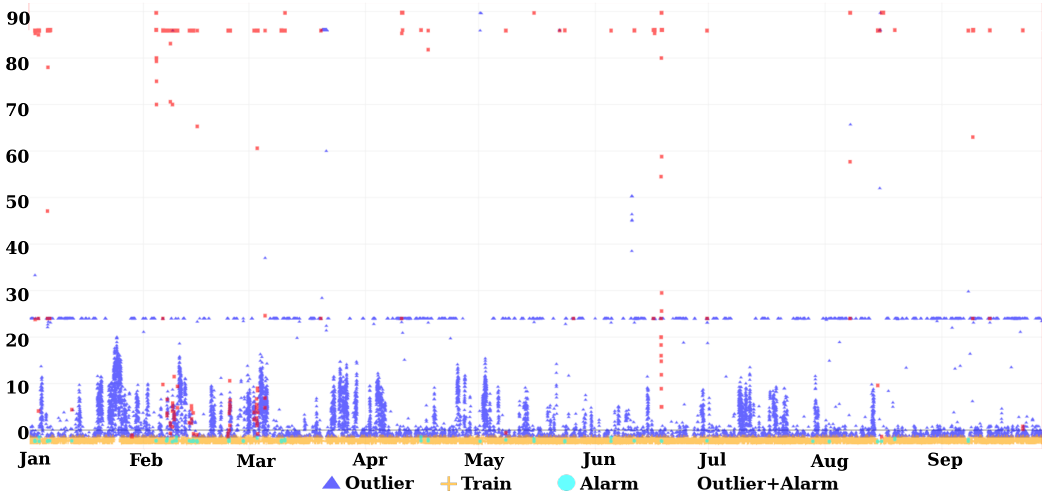
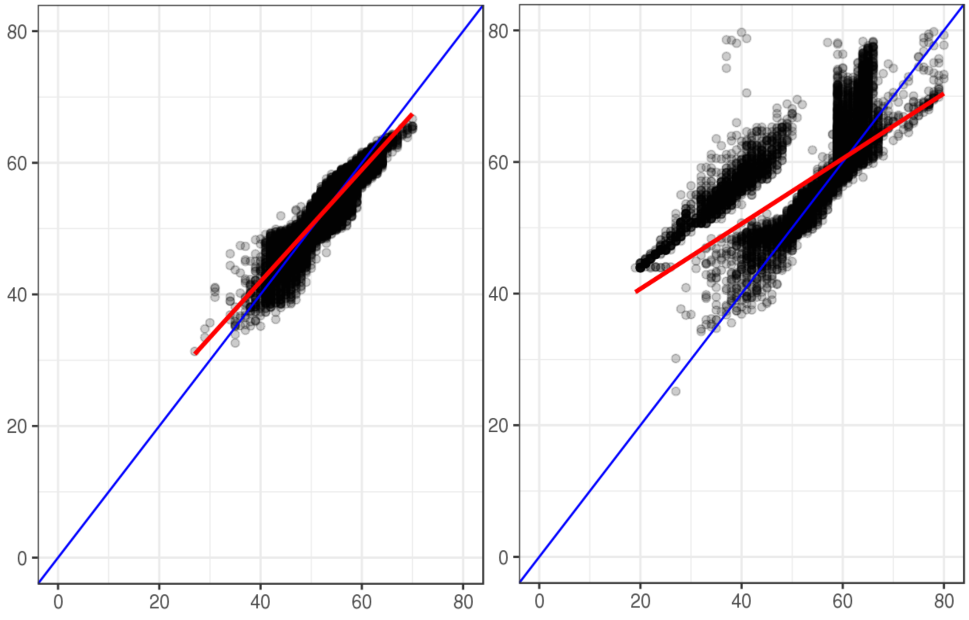


Figure 10: Outlier marking vs. alarms on variable *blade pitch angle* on filtered dataset

The impact on the model results are show in figure 11 which reveals, again, an increase on performance using the training dataset (left). The results reveals an even higher decrease of the MSE error from 2.0768 to 1.816 and 42.4° to 40.3° which is a better regression line for estimation vs real value. But the analysis of the test dataset (right) becomes the worst of all the filtering methods which generates a MSE of 12.6° which is 5 times worst than the estimation without filtering. The plot of the results reveals clear regions with problems, which are the regions of values that were removed for the filter and therefore we do not have these points of the input variable when estimating the target variable. The angle is about 17° which is the worst among the filtering systems analyzed and the linear regression line is clearly far from the theoretic one.



(a) Train

(b) Test

Figure 11: Train and Test estimation vs. real value of target variable

3.4. Results Summary

In table 3 we present a summary of some of the experiments performed on all the wind turbines of the wind parks detailed in 2. For the lack of space, we list here only some wind turbines of each park, and for the sake of clarity we present the MSE results on test dataset with the corresponding filter strategy normalized by the MSE result without filtering. If the filtering strategy is not helping when testing the PLS model then the quotient will be

	Machine id	3 σ	Quantile filter	Hampel filter
Model		MSE Ratio	MSE Ratio	MSE Ratio
Fuhrlander FL2500	80	1,002	1,002	*0,998*
	81	1,002	1,011	1,008
	82	*0,999*	*0,987*	1,002
	83	1,000	1,002	1,171
	84	*0,996*	1,458	44,838
Vestas V90 wfa1	67	*0,935*	1,090	5,274
	68	*0,780*	4,640	*0,753*
	69	*0,983*	1,319	1,868
	70	*0,983*	1,604	8,317
	71	*0,971*	1,162	8,253
	72	*0,996*	1,851	12,410
	73	*0,985*	1,168	6,892
	74	1,088	1,347	*0,912*
	75	1,046	*0,959*	5,505
	76	*0,992*	1,037	4,813
	77	*0,975*	1,267	5,801
	78	1,536	1,518	8,010
	79	1,085	1,185	4,826
Siemens Izar 55/1300	41	*0,961*	*0,882*	210,940
	42	*0,966*	*0,928*	307,942
	43	1,015	*0,905*	250,313
	44	*0,895*	*0,835*	242,414
	45	1,121	1,147	172,567
	46	1,057	1,022	218,819
	47	1,208	1,080	280,106
	48	1,158	1,133	157,796
Vestas V90 wfa2	112	*0,795*	1,033	1,239
	113	*0,971*	1,179	1,260
	114	1,193	1,247	1,418
	115	1,007	1,060	1,156
	116	*0,908*	1,019	1,057
	117	1,065	1,193	1,315

Table 3: Result summary

>1. On the contrary, if the filtering strategy is helpful when modeling the data, then the ratio will be smaller <1 (these cases are indicated in cursive font on the table 3).

As we can see, values are habitually >1 and this is the habitual case when using Adjusted box-plot rule or Hampel rule. It is important to note that when we use these two above mentioned filtering strategies, results are much worse than without filtering (i.e.: MSE quotients are ≫1). Only the ESD rule seems to be interesting in some cases, but even in these cases, corresponding to the quotient <1, the difference of MSE between filtering and non-filtering is small.

Analyzing in detail all the cases reported in table 3, in 17 over 32 cases the ESD filtering method is useful when testing the model, which roughly represents 53% of the cases. Even if that seems a high amount of cases, in all of them the quotient is ≈1 , indicating that the MSE is almost the same when using the filter compared to the original (non-filtered) case. For the quantile filter, only 6 over 32 cases reported a quotient smaller than one. It means that only about 19% of the cases improved results after filtering. Finally, for the Hampel filter only 3 cases over 32 reported a quotient higher than one, i.e.: 9% of the cases.

Computing all the filters analyzed, in 73% of the cases the filtering procedure increased the MSE. Therefore, filtering is not a good strategy by default, and only in a very few cases could slightly improve the results by decreasing MSE in the test dataset. According to our experiments, in the case of needing a filter, the best choice would be to use the ESD filter, as it is able to eliminate some outliers that are not relevant or related to alarms, as has been demonstrate in our experiments.

4. Conclusions

In this paper we explored several methods for outliers detection and compared their performance against the non-filtered data. Experimental results when deriving models using real data (from several wind farms and turbine models) and the following generalization on new data (test dataset) increases the error, measured trough MSE and regression line, when we use filters to eliminate outliers. This is due to that many outliers were failure states of the wind turbine, as indicated in section 3 with the variables affected by each filtering method. Filtered data performance could generate good results with cross-validation on the same train dataset, which is already filtered and the value of each variable is closer, hence easier to model, but the performance is reduced using new data in all the cases because of the poor generalization capability due the removed failure patterns that are present on the future datasets. In this case the performance of prognosis models over SCADA data performs best over new data with non-filtered train datasets. The effect of removing points labeled as outliers but that in fact contributes to identify alarm states can be observed in figures 3, 4, 6, 7, 9 and 10

In the light of these results, systematically filtering outliers from the data coming form SCADA wind turbines has to be reconsidered in order to derive better models which will lead to better prognosis of the wind turbines. This will have an effect on the management and maintenance costs which in turn will allow to increase the economic competitiveness of the wind energy with respect to fossil fuels and accelerate the transition towards ecologically sustainable systems.

Acknowledgement

Financial support by the Agency for Management of University and Research Grants (AGAUR) of the Catalan Government to Alejandro Blanco is gratefully acknowledged.

References

A. Christya, G. Meera Gandhib, S. V., march 2015. Cluster based outlier detection algorithm for healthcare data.

Christophe Leys, Olivier Klein, P. B., mar 2013. Detecting outliers: Do not use standard deviation around the mean, use absolute deviation around the median.

Denis Cousineau, S. C., dec 2010. Outliers detection and treatment: a review. Practical Assessment, Research & Evaluation (PARE),International Journal of Psychological Research 3 (1).

EU commission, 2008. Communication from the commission to the european parliament, the council, the european economic and social committee and the committee of the regions. Tech. rep., Commission of the european communities.

Eurostat, 2013. Energy balance sheets. Tech. rep., Eurostats (European Union).

F. Besnard, L. B., 2010. An approach for condition-based maintenance optimization applied to wind turbine blades. IEEE Transactions on Sustainable Energy 1 issue 2, 77 – 83.

Fabiani Appavou, Adam Brown, B. E., 2016. Renewables 2016 global status report. Tech. rep., Renewable energy policy network for the 21ST century.

Gray, C. S., 2011. A practical approach to the use of scada data for optimized wind turbine condition based maintenance.

IEC, dec 2006. International standard iec 61400-25-1.

Jason W. Osborne, A. O., mar 2004. The power of outliers (and why researchers should always check for them). Practical Assessment, Research & Evaluation (PARE) 9 (6).

Justin Heinermann, O. K., 2015. On heterogeneous machine learning ensembles for wind power prediction.

Marjan Bakker, J. W., 2014. Outlier removal, sum scores, and the inflation of the type i error rate in independent samples t tests: The power of alternatives and recommendations. Psychological Methods, American Psychological Association.

McClelland, G. H., 2000. Nasty data: Unruly, ill-mannered observations can ruin your analysis. Handbook of research methods in social and personality psychology. Cambridge: Cambridge University Press. 2, 393 – 411.

Milborrow, D., 2003. Operation and maintenance costs compared and revealed. Elsevier Science B.V.

OPC Fundation, oct 2016. What is opc.

Patel, V. R., sept 2011. Impact of outlier removal and normalization approach in modified k-means clustering algorithm.

Pearson, R. K., apr 2005. Mining Imperfect Data: Dealing with Contamination and Incomplete Records. SIAM: Society for Industrial and Applied Mathematics.

Pedro Santos, L. F. V., 2015. An svm-based solution for fault detection in wind turbines.

Rodero-Merino, L. M. V. L., 2011. Dynamically scaling applications in the cloud.

Valeri Voev,Siemens A.G., 2014. Siemens remote diagnostic services. Tech. rep., Siemens Wind Power.

Vestas R+D, 2004. General specification vestas v90 3.0mw. Tech. rep., Vestas Wind Systems.

Wold, S., oct 2001. Pls-regression: a basic tool of chemometrics. Chemometrics and Intelligent Laboratory Systems 58, 109–130.

3.3. Impact of target variable distribution type over the regression analysis in wind turbine data

Impact of target variable distribution type over the regression analysis in Wind Turbine data

Alejandro Blanco-M.
Data and Signal Processing Group
U Science Tech, University of Vic –
Central University of Catalonia
C. de la Laura, 13
08500 Vic, Catalonia (Spain)

Jordi Solé-Casals
and Pere Marti-Puig
Data and Signal Processing Group
U Science Tech, University of Vic –
Central University of Catalonia
C. de la Laura, 13
08500 Vic, Catalonia (Spain)

Juan José Cárdenas
Isaac Justicia,
and Jordi Cusidó
Smartive-ITESTIT SL
Catalonia (Spain)

Abstract—**The Wind sector has roughly 2200M euros of profit losses due to wind turbine failures and these failures do not contribute to the goal of reducing greenhouse gas emissions of many states. The 25-35% of the generation costs are operation and maintenance services. To lower this ratio, the wind turbine industry is backing on the Machine Learning techniques over SCADA data. Signal trending analysis supported on linear regression models presents the problem of how to carefully choose the right target variable, which reproduces as close as possible the behavior of a failure from a component. This document evaluates the impact of that choice by comparing as target different variables with discrete-non normal distribution, commonly selected by feature selection methods, versus variables that are continuous over time with a near normal distribution. Experimental results on real data show the use of continuous target variables selected by human expert on the field give better results than the use of targets obtained through feature selection algorithm.**

Index Terms—**Renewable Energy,Wind Farms,feature selection,regression models,SCADA Fault Diagnosis**

I. Introduction

The reduction of greenhouse gas emissions and the independence from fossil fuels are the main goals of many states [1]. That can be achieved by producing electricity using sustainable sources [2]. For instance, the EU commission established a series of demanding climate and energy targets to be met by 2020 that includes the use of at least 20% of energy coming from renewable sources [3]. Wind power is the most growing renewable source [1], however operation and maintenance of wind turbines account for 25% to 35% of the generation costs [4]. In order to increase the economic competitiveness with respect to fossil fuels and accelerate the transition towards ecologically sustainable systems, there is a need for more efficient management and this requires intense monitoring of wind turbines. A Modern wind turbine records more than 200 analogous variables [5] at intervals of 5 to 10 minutes by means of their SCADA (Supervisory Control and Data Acquisition) system, therefore automatic supervision systems have to be developed in order to detect and fix possible failures.

The SCADA system collects data in different parts of the turbine, which are grouped into systems [6] providing information about: temperatures, electrical indicators, physical positions, speeds, vibration, etc. Therefore these systems generate a huge amount of data [7], which has to be preprocessed and modeled in a feasible time [8].

In order to determine when a Wind Turbine is going to failure, a prognosis model is implemented. The model analyses the variations of a target variable over time, known as the signal trending analysis.

This study shows the importance of selecting the optimal target variable, i.e., the variable under the scope of the prognosis model that generates a reference signal with the expected values and then compared to the real signal. The optimal selection of this variable is made by human expert knowledge on the field and the result of the prognosis model is then compared with feature selection algorithms.

This document is organized as follows: Section I contains the introduction and objectives of the paper; subsection I-A and I-B are devoted to describe the real data used in all this research, the pre-selection of input variables and the alarms to identify the unhealthy Wind Turbine states. Section II presents the target selection methods to be evaluated and the description of the process. Results and discussions are presented in Section III, while Section IV contains the final conclusions of our research.

A. Data background

Data is produced by the wind turbine's SCADA that follows the IEC 61400-25 format [9] which provides data with a structure of *logical devices* representing wind turbines and *logical nodes* representing physical wind turbine systems and subsystems. Data was collected via an Open Platform Communications (OPC) [10] each 5 or 10 minutes which reports events, instant values and statistics indicators. Only the events and the statistics indicators are kept. Each sensor has its statistics indicators which commonly are *the average*, *minimum*, *maximum* and *standard deviation* values.

In our case, the data is stored on a local database, which has been recording values from the wind plants over the years. Data is structured in a *table* with entries containing different instances of each sensor at each time interval. Events

978-1-5386-0850-0/17/$31.00 ©2017 IEEE

TABLE I: Example of the data analyzed (part of a real table)

date_time	power	bearing_temp	gen_1_speed	temp_oil_mult
2014-12-08 06:20	1701.17	29.40625	1291.84	36.39
2014-12-08 06:30	1583.11	28.14462	1055.23	22.08
2014-12-08 06:40	1664.03	28.03261	1132.16	23.43
2014-12-08 06:50	1722.47	29.8721	1312.66	22.68
2014-12-08 07:00	1647.91	29.0121	1231.78	21.82

containing information about the system errors are stored in a different table since they are recorded in different format and contain the specific time in which the event ocurred. These events are entries which are categorized as *alarms* (failure states) and *warnings* (non-important events like machine stop by maintenance service, start or stop messages). An example of the data format generated is shown in table I, which contains registers (rows) and variables (columns) ordered by *date_time* variable, i.e. the insertion time on the database, when a notification of the variable and the value were submitted by the OPC.

The experiment described on this document uses three to seven different input variables from the whole set, pre-selected by an expert in the field, i.e. power, wind_speed, temperatures... Each manufacturer has a different name but the system which represents is the same. As for instance in table II, which are variables related with Generation system, from now on ***Ivars***.

B. Alarm event selection

The starting point of model prognosis is to select which alarm events have a relation with the physical system to be monitored. The alarms have a code, a description and in some case the value which triggered it.

Based on this information and the structure of the Wind Turbine physical system [6], the human expert chooses a subset of alarms which indicates failures on a specific system. This subset will be used in order to find the variables more related to the alarms state in case of the feature selection analysis, and will be used as evaluation of the model performance about the capability to detect a real alarm.

In our experiments we focus in the transmission and generation systems, in which the variables involved have a slower dynamic change rate. More specifically the *Main Bearing subsystem*, which supports the rotor of the wind turbine and is at the origin of many alarms, and the generator which has a big impact on the final energy production. Since the alarms are different for each manufacturer, the expert has to select a subset of alarms for each Wind Turbine separately, henceforward ***Acodes***.

II. Materials and methods

This section covers the techniques that have been applied in our experiments in order to identify the target and input variables that will be evaluated with a regression model response expecting a big deviation on the periods in which there are active alarms and minimal deviation from the periods where the Wind Turbine is healthy. For each case we present a description of the method and the data work-flow over the algorithm which will be applied separately over the same input dataset.

In order to be able to compare the results as much as possible, we have fixed the input variables set (***Ivars***) at subsection I-B for each Wind Turbine manufacturer and for all the methods. The feature selection methods are applied individually in order to choose the optimal target variable (***Tvar***). Each feature selection algorithm uses a subset of predefined alarms events (***Acodes***) to determine from all available variables, which one is the most relevant regarding this event that will be considered as the regression model target.

The results of each ***Tvar*** selection output will be evaluated using a regression model based on PLS [11] graphically by plotting the response signal and comparison plot, also numerically by calculating the Mean absolute percentage error (MAPE) [12], which is a value without units that indicates the accuracy of the model in percentage. The MAPE is useful to compare different data results obtained from different target variables, as happens in our case.

A. Human expert target selection

According to the alarm subset, the expert chooses the target variable of the regression model supporting its decision on the variable name that follows the IEC format [9] and the visual plotting support of each variable versus the alarm events ***Acodes***. His own experience determines which value is deviating before and after the alarm occurrence.

This analysis is done in a semi-automatic way for each manufacturer, since the name and range of the variable change for each manufacturer.

B. Recursive feature elimination for target selection

This method is based on a simple backwards selection, which recursively selects the best feature using the model importance score at each iteration.

This algorithm is already implemented in some toolboxes, like in this case the Caret wrapper package [13] which includes a set of feature selection algorithms.

From all the available models types, in this case, the recursive selection is supported by a random forests algorithm and evaluates the variable importance using a 10-fold cross-validation strategy. At each step, the less important variables are discarded.

To calculate the variable importance in random forests, the prediction accuracy is measured with the original data. Then, each variable is thrown from the data and the random forest is executed again obtaining a new prediction accuracy. The normalized and averaged accuracy difference of all trees is the importance of the current variable.

C. Random Forest feature selection for target selection

Another algorithm based on Random Forest, which is the top-down search method for relevant features, is implemented in the Boruta package [14].

TABLE II: Example of Input Variable selected in columns for each manufacturer.

Fuhrlander fl2500	Vestas V90 'wf1'	Vestas V90 'wf2'	Siemens Izar 55/1300	Wfa H1
wgdc_avg_TriGri_PwrAt	avg_hydraulic_oil_temp	hydraulic_oil_temp_avg	GeneratorSpeed	wrot_avg_PtTmpCoolBl2
wtrm_avg_Brg_OilPres	max_blades_pitch_angle	blades_pitchangle_max	ActualPowerWecProduction	wnac_avg_WdSpd1
wtrm_avg_Gbx_OilPres	min_blades_blade_A_control_voltage	blades_bladea_controlvoltage_min	GearTemperature	wtur_avg_W
wtrm_avg_Brg_OilPresIn	max_ambient_wind_dir_relative	ambient_winddir_relative_maxium	HydraulicOilTemperature	
wnac_max_NacTmp	avg_grid_production_power	power_avg,wind_avg	NacelleTemperature	
wgen_avg_RtrSpd_WP20350	avg_ambient_wind_speed	gear_bearing_temp_avg		
	avg_gear_bearing_temp			

TABLE III: Data summary

Turbine Model	Num. of machines	Num. of years	Rows per year	Num. of variables	Num. of triggered alarms	Total registers evaluated
Fuhrlander fl2500	5	4	105.120	303	72.422	2.102.400
Vestas V90 'wf1'	7	4	52.560	194	9.681	1.471.680
Vestas V90 'wf2'	13	4	52.560	63	5.063	2.733.120
Siemens Izar 55/1300	26	1	52.560	24	369.218	1.366.560
Wfa H1	1	7	52.560	406	83.716	52.560
Total	52	20		992	540.100	7.726.320

As same strategy of Caret package on subsection II-B, Boruta implementation [14] relies on measuring the importance of each variable to the model. But it has an added mechanism to improve the feature selection rate by creating a shadow copy of each feature. These copies are shuffled individually (the entries are randomly sorted) in order to remove any correlation with the original.

Then when the variable importance algorithm is running for each variable as explained on II-B, at the same time that the less important feature is eliminated at each step, the importance of all other variables are compared to the importance of the shadow copies, which are calculated in the same manner as the original ones. If some of the other variables that haven't been eliminated have less importance than the shadow copy then theses are marked with "not acceptable" flag in order to remove them since have the same importance than a random sorted copy.

D. Hypothesis Testing target selection

Based on the selected subset of events ***Acodes***, a contrast of hypothesis is generated in order to identify the variables that are more related with the selected events. The null hypothesis H_0 which defines the *no statistical relevance about the change of a variable mean* when a day contains alarm/failure and in other side the alternative hypothesis H_a defines that a variable presents a *statistically relevant difference in its average value* when an alarm/failure event is present at that day. The interval of confidence is defined at 95% which determines a p-value of 0.05. Any variable which has a p-value smaller than 0.05 is considered as a possible input variable for the model. We then sort all the candidates to be considered as input variables of the model, from the smallest to the biggest p-value, and we select the first six variables in order to analyze them.

E. CMIM feature selection for target selection

The conditional mutual information maximization (CMIM) is an algorithm which searches for the smallest subset of features that contains the maximum possible information [15].

To evaluate which set of features gives the highest amount of information, the algorithm is fed by many entries of all possible input variables and evaluated to a target variable, in this case the alarm events that have been selected previously as described in subsection I-B. This algorithm aims to minimize the entropy of each subset S $S \subset T$ from the available features T versus a target variable. The subset that has the lowest entropy, in others words, a variable is a good predictor when the subset has the lowest entropy, in other words, the probability that the variable gets approximately the same value when the alarm is active. The procedure of Maximization evaluates all the possible variables subsets in order to find, for different numbers of features, the optimal feature combinations.

III. Results

In this section, the experimental results are presented for different manufacturers comparing the human selected variables to the others of methods. The data range evaluated in this section contains no alarms because we expect to follow the real signal as close as possible with the less deviation from the diagonal of the difference plot between real and model output value.

The summarized results for each Wind Turbine manufacturer is presented on table IV which contains the results of MAPE, measured in average, from all the Wind Turbine of same manufacturer.

The current analyzed machine is the Siemens Izar 55/1300, which has problems with the temperature of the generator subsystem.

A. Human expert

The human expert chose as reference the target ***Tvar*** *TemperatureGenerator1* as shown in figure 1, which shows the histogram of the temperature from the generator system of the Wind Turbine [9]. As a general trend, it tends to behave like a normal distribution. The results of the model for this case are visualized on sub-figures (b) and (c) which shows the real signal (blue) and the response of the model (light blue) and is able to follow the real over the time. On the sub-figure (c), a comparison of real and model estimated value is done point to point, which reveals that the points are close to the diagonal. The red line is the best fit line for the points with a linear regression equation.

B. *Random Forest feature selection*

The first method to be compared is the Random forest based on Boruta method on figure 2, which selectes the *BladesPosition* variable, that contains the *Pitch* of the Wind Turbine blades. By the knowledge on the Wind Turbine domain, these kind of variable are very discrete since the control mechanism has a discrete table of positions. In this case, for example, a large portion of time, the variable stays from 0° to 30° , which are acceptable values, with some extreme cases at 90° when the Wind Turbine control try to stop. As it can be seen in sub-figure (b) and (c), the model is poor since it generally tends to give a bigger value than the real value. It is clear that the regression line in sub-figure (c) is far from good.

C. *Recursive feature elimination*

The second method based on recursive feature elimination from Caret package gives the same results as the previous method since it chooses the same variable.

D. *Hypothesis Testing*

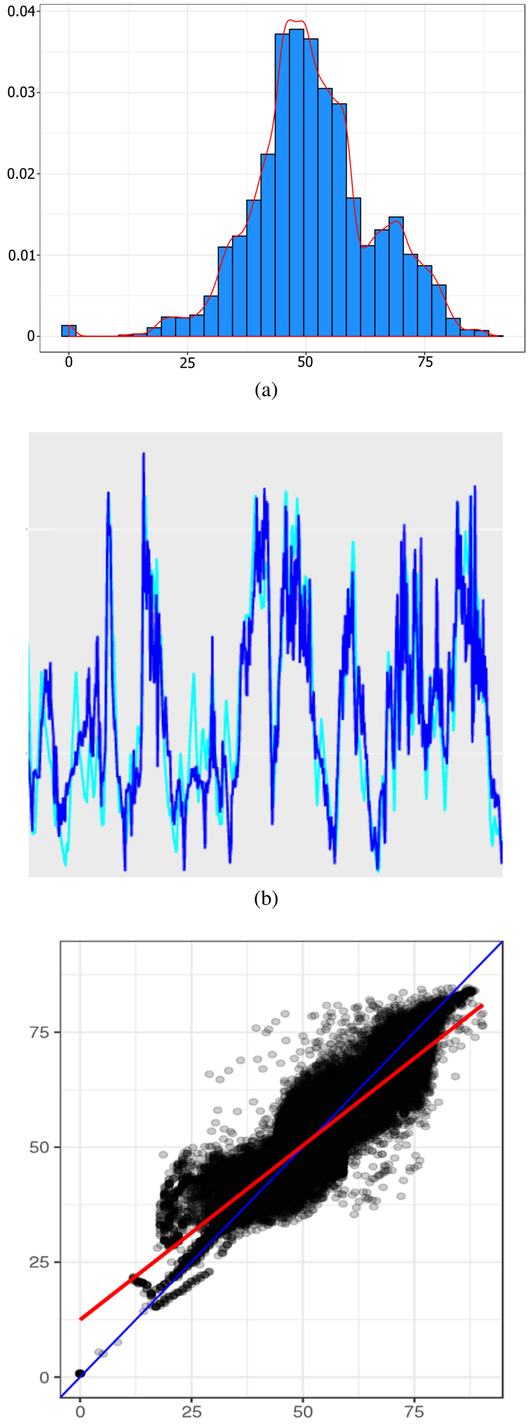
As a third method, for this Wind Turbine, it is the Hypothesis testing method. In the same way as the first two methods, again the ***Tvar*** variable *BladesPosition* is selected, which is not a good target for the regression model.

E. *CMIM feature selection*

The last method, CMIM is presented in figure 3, which chooses another type of variable, specifically the *RotationSpeedMin* that measures the revolutions per minute of the Wind Turbine Rotor. This variable has a big impact on the transmission system since it determines the possible power to generate, so the CMIM chooses a variable which is considered by the Human expert as important, but not as same as the *TemperatureGenerator1*. On the sub-figures (c), a cluster of points seems deviated at x=0, this could be because the *RotationSpeedMin* histogram has a big separation between 1 and 10, so the model tends to predict bigger values than the real ones as seen on (b). Although, the result is much better than the ones obtained with the two previous methods.

IV. Conclusion

In this paper we explored several methods for the selection of the best ***Tvar*** target variable for a linear regression model to measure its performance compared to the human expert criteria. The methods presented in this paper, which generated different variable combinations, have been implemented using a linear regression model. We observed that automatic feature selection algorithms tend to choose variables which are more discrete.



(a)

(b)

(c)

Fig. 1: Human selected variables results for Siemens Izar 55. In (a) the histogram of variable TemperatureGenerator1. In (b) the model output with dark blue real signal, light blue model estimated signal. In (c) The model at Y axis vs. real value at X axis

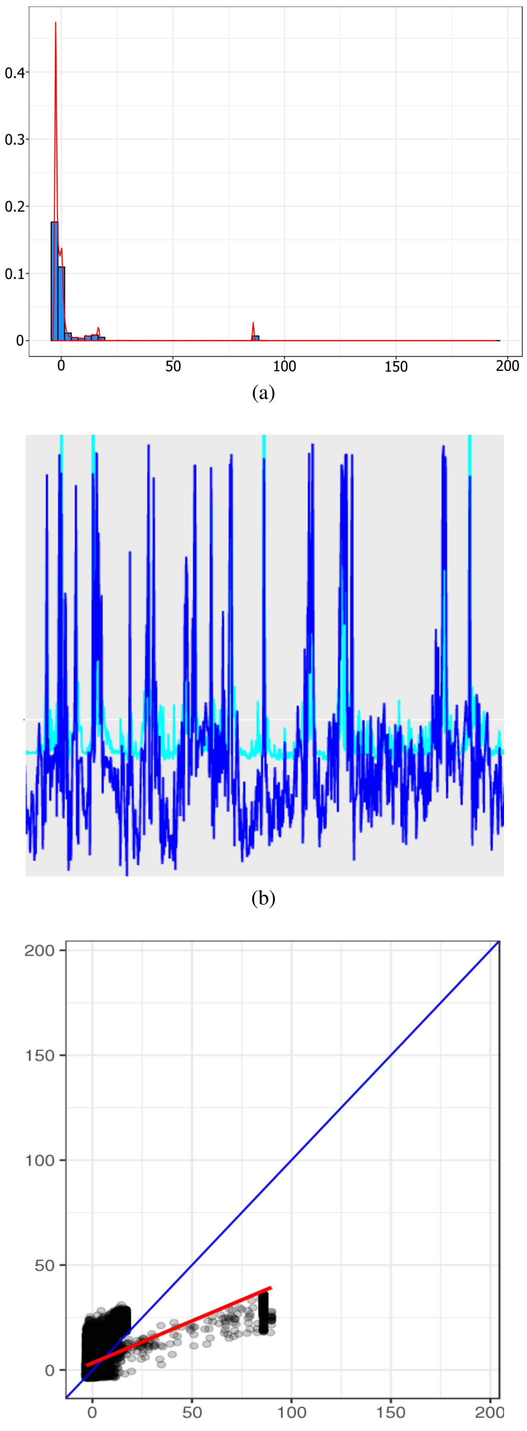


Fig. 2: Random Forest selected variable results for Siemens Izar 55. In (a) the histogram of variable BladesPosition. In (b) the model output with dark blue real signal, light blue model estimated signal. In (c) The model at Y axis vs. real value at X axis

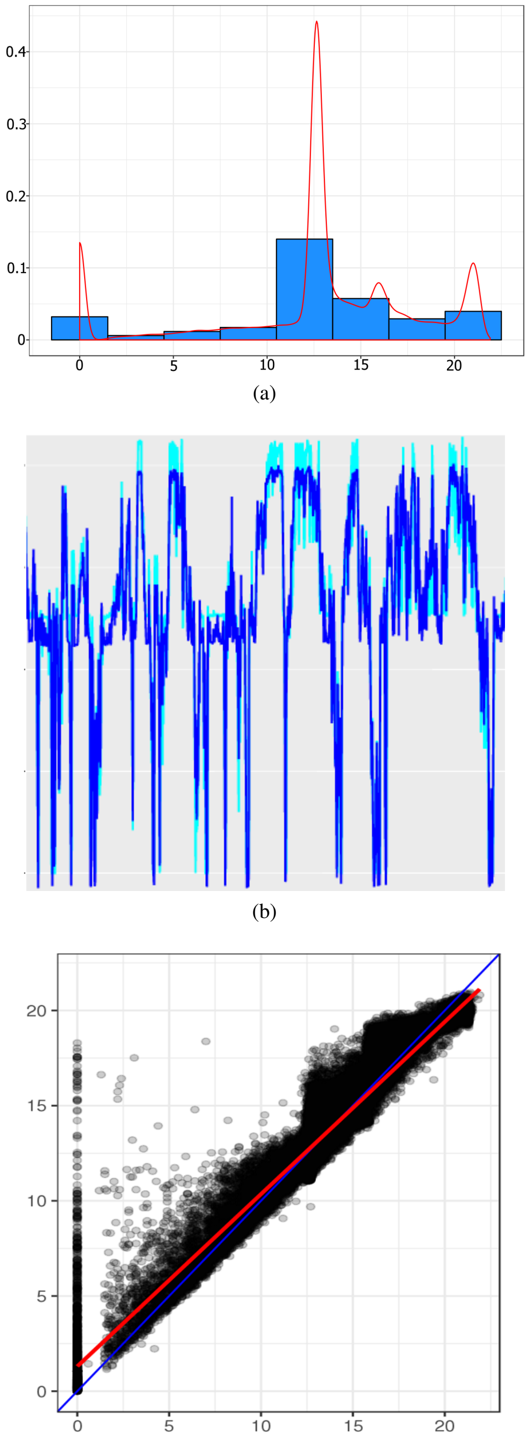


Fig. 3: Conditional Mutual Information Maximization (CMIM) results for Siemens Izar 55. In (a) the histogram of variable BladesPosition. In (b) the model output with dark blue real signal, light blue model estimated signal. In (c) The model at Y axis vs. real value at X axis

These variables are not suitable for linear regression models, except for the Conditional Mutual Information Maximization which chooses better variables. These methods have been applied to several manufacturers, as detailed in Table V.

Experimental results show that the human expert knowledge in the Wind Turbine field is the best strategy when selecting the target variable to build a linear model to obtain the smallest error. As a summary of results, we present in table IV a comparison for the case with the lowest MAPE for each method and Wind Turbine for the Human Expert, the Random Forest, the Caret, the CMIM and the Hypothesis Testing methods. Notice that the smaller the MAPE value is, the better the model is. We can calculate the average (column-wise) in order to know which one are (in mean) the best methods for all the turbine models. As we can see in the table, the Human expert selection averaged 21.55% , the Random Forest 37067120.32% , the Recursive feature elimination 37067601.12%, the Conditional Mutual Information Minimization (CMIM) 15582.34% and finally Hypothesis Testing method 37063591.47%. There is clearly a great dispersion on these values, mainly due to the fact that for specific turbine models, some methods are working very poorly, therefore dramatically increasing the error of the model. Mainly, Siemens Izar model poses many problems to three of the methods, while Fuhlander model and Wfa H1 models are also problematic for two of them. On the contrary, we obtain a few cases in which automatic feature selection methods are the best ones. This happens, for example, for the Random Forest method working on the Fuhrlander wind turbines or the CMIM method working on the Vestas turbines.

Our work on several wind turbines and providers have allowed us to reveal that automatic feature selection methods have to be taken carefully when building linear regression models. Human expertise has to be considered in coordination with automatic algorithms in order to ensure that the best target variable for the type of alarm is considered. This strategy will provide better models, therefore allowing a better management of the wind farm and a reduction of maintenance costs.

Acknowledgment

Financial support by the Agency for Management of University and Research Grants (AGAUR) of the Catalan Government to Alejandro Blanco-M. is gratefully acknowledged.

References

[1] B. E. Fabiani Appavou Adam Brown, “Renewables 2016 Global status report,” Renewable energy policy network for the 21ST century, Tech. Rep., 2016.

[2] Eurostat, “Energy balance sheets,” Eurostats (European Union), Tech. Rep., 2013.

[3] E. commission, “Communication from the commission to the european parliament, the council, the european economic and social committee and the committee of the regions,” Commission of the european communities, Tech. Rep., 2008.

[4] D. Milborrow, *Operation and maintenance costs compared and revealed*. Elsevier Science B.V, 2003.

[5] L. F. V. Pedro Santos, “An SVM-Based Solution for Fault Detection in Wind Turbines,” 2015.

[6] Vestas R+D, “General Specification VESTAS V90 3.0MW,” Vestas Wind Systems, Tech. Rep., 2004.

[7] Valeri VoevSiemens A.G., “Siemens Remote Diagnostic Services,” Siemens Wind Power, Tech. Rep., 2014.

[8] O. K. Justin Heinermann, “On Heterogeneous Machine Learning Ensembles for Wind Power Prediction,” 2015.

[9] IEC, “Part 25-1: Communications for monitoring and control of wind power plants Overall description of principles and models,” *IEC 61400-25-1 First Edition 2006-12*, dec 2006. [Online]. Available: https://webstore.iec.ch/preview/info_iec61400-25-1%7Bed1.0%7Den.pdf

[10] OPC Fundation, “What is OPC,” oct 2016. [Online]. Available: https://opcfoundation.org/about/what-is-opc/

[11] S. S. M. Wold and L. Eriksson, “PLS-Regression: A Basic Tool of Chemometrics,” *Chemometrics and Intelligent Laboratory Systems*, vol. 58, pp. 109–130, oct 2001.

[12] C. Tofallis, “A Better Measure of Relative Prediction Accuracy for Model Selection and Model Estimation,” *Journal of the Operational Research Society*, vol. 66, no. 8, pp. 1352–1362, 2015. [Online]. Available: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2635088

[13] M. Kuhn, *Classification and Regression Training*, 2016. [Online]. Available: https://cran.r-project.org/web/packages/caret/caret.pdf

[14] W. R. R. Miron B. Kursa, “Feature Selection with the Boruta Package,” *Journal of Statistical Software*, vol. 36, no. 11, 2010. [Online]. Available: https://www.jstatsoft.org/article/view/v036i11

[15] M. Brown, Gavin and Pocock, Adam and Zhao, Ming-Jie and Lujan, “Conditional likelihood maximisation: a unifying framework for information theoretic feature selection,” *The Journal of Machine Learning Research*, vol. 13, no. 1, pp. 27–66, 2012.

TABLE IV: MAPE best results of each methods for each manufacturer

Turbine Model	Human Expert	Random Forest (Boruta)	Recursive feature elimination (Caret)	Conditional Mutual Informat. Max. (CMIM)	Hypothesis Testing
Fuhrlander fl2500	51.47%	23.61%	23.61%	8042.04%	4330.63%
Vestas V90 ’wf1’	2.8%	40.1%	128.38%	91.56%	91.56%
Vestas V90 ’wf2’	11.26%	22004.7%	22004.7%	10.45	26.94%
Siemens Izar 55/1300	10.62%	185313488.5%	185313488.5%	7.6%	185313488.5%
Wfa H1	31.6%	41.67%	2360.4%	69760.06%	19.73%
Average	21.55%	37067120.316%	37067601.118%	15582.342%	37063591.472%

TABLE V: Results Summary of Feature Selection methods

Turbine Model	Human Expert	Random Forest (Boruta)	Recursive feature elimination (Caret)	Conditional Mutual Informat. Max. (CMIM)	Hypothesis Testing
Fuhrlander fl2500	wtrm_avg_TrmTmp_Brg1	wtrm_min_Brg_OilPres	wtrm_min_Brg_OilPres	wgdc_avg_TriGri_PwrAt	wtrm_avg_Brg_OilPres
Vestas V90 ’wf1’	avg_generator _slip_ring_temp	first_alarm_parameter _2_in_10min_frame	first_alarm_parameter _2_in_10min_frame	avg_generator_phase _1_temp	stdv_ambient_wind_dir_relative
Vestas V90 ’wf2’	gear_oil_temp_avg	hourcounters_average _turbineok_avg	grid_production _reactivepower_min	hydraulic_oil_pressure_min	hourcounters_average_run_avg
Siemens Izar 55/1300	TemperatureGenerator1	BladesPosition	BladesPosition	RotationSpeedMin	BladesPosition
Wfa H1	wrot_avg_PtTmpMotBl1	wrot_min_RotPosInd	wrot_sdv_PtAngSpBl1	wtrf_sdv_TrfTmpTrfTur	wcnv_sdv_Hz

3.4. Sensors Selection in Wind Turbine Prognosis

SENSORS SELECTION IN WIND TURBINE PROGNOSIS

Alejandro Blanco-M [*,1,2], **Pere Marti-Puig**[*,1], **Juan José Cárdenas**[2], **Jordi Cusidó**[2], **and Jordi Solé-Casals**[1]

[*]**These authors contributed equally to this work.**

[1]**Data and Signal Processing Group, University of Vic - Central University of Catalonia, c/ de la Laura 13, 08500 Vic, Catalonia, Spain**

[2]**Smartive-ITESTIT SL, Carretera BV-1274, Km1, 08225 Terrassa, Catalonia, Spain**

Corresponding author:
Jordi Solé-Casals[1]

Email address: jordi.sole@uvic.cat

ABSTRACT

Annually, the wind power sector suffers losses of profits due to wind turbine failures and operating and maintenance costs. Wind farm operators employ Machine Learning techniques to manage available SCADA data for improving prognosis models of specific system failures. The involved procedures therein require the selection of incoming data from the most important and reduced sensors set. Due to the high number of sensors, exhaustive-search algorithms are unfeasible. In this work we propose a quasi-optimal (QO) algorithm to select the best sensor set, which is compared with several well-known feature selection (FS) algorithms. The prognosis model evaluation is performed using a *k*-NN classification algorithm working with sensor selections obtained trough different FS algorithms as well as the QS method. Experiments are performed on sensor data from five Fuhrländer wind turbines taken along an entire year. Prognosis tasks have been focused on the gearbox and the transmission systems, two of the most expensive wind turbine systems. In the case study, the prognostic methods using the QO algorithm always worked slightly better than the others.

1 INTRODUCTION

Each year wind sector has profit losses due to wind turbines failures that can be about 200 M€ in Spain, 700 M€ in Europe and 2 200 M€ in the rest of the world. Additionally, if operation costs are taken into account, these losses can be tripled. Owing to the volume of losses and the actual economic situation in the sector, without any bonuses to the generation and with generation selling prices policy restricted by the new energy law, operations related to maintenance and operation improvement are a key for wind farms operators, maintenance companies, financial institutions, insurance companies and investors.

One of the main tasks in Operation and Maintenance (O&M) process is to find out the possible causes of failures. This process is crucial to reduce the time of repair or detect more critic faults in earlier stages. Methodologies and tools that can support this type of operation can benefit wind farm owners to increase availability and production and reduce costs.

The operating and environmental conditions of virtually all wind turbines (WT) in operation today are recorded by a SCADA ([10]). The SCADA system collects information on all sensors placed in a turbine, storing all the data in a database that is commonly located in the park. These sensors produce high-frequency information in the form of variables which are grouped into systems [4], [5], [22], [19]. The number of sensors (variables) collected by the SCADA and the number of systems varies depending on the model and the manufacturer. The SCADA system also obtains the exact date of appearance and information on faults that have occurred on the machine, and this is why the park operators use it as a solution to the problem of monitoring and prediction of failures applying Machine Learning techniques. This system has a high potential of information to support this specific task due to the data availability, more when the data for fault diagnosis is already available requiring no further implementation or hardware installation in the wind turbine. However, storing all the data becomes a problem of space and processing [13], [2], [21], so knowing which variables are the most important when creating a Machine Learning model would allow us to select a subset of data, generating the necessary space to collect data of higher quality and resolution. In general, to overcome this problem, the SCADA systems installed at the wind turbines plants collect information and generates four statistical indicators(min, max, avg, std.) at intervals of 5 to 10 minutes for each sensor.

A wind turbine is a complex machine composed of different systems as indicated on the IEC [15]. Figure 1 shows an example of various sensors (or variables) integrated into these systems. In total, a wind turbine could generate more than two hundred signals mixing higher frequencies sensors (+60Hz) like accelerometers, and current frequencies with lower frequencies (1Hz), e.g., temperatures, wind speeds, rpms, etc.

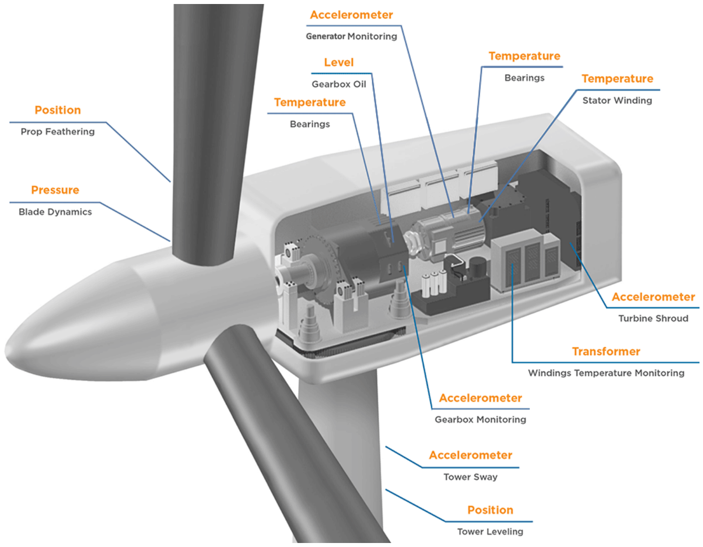


Figure 1. Example of Wind Turbine sensors types. *Adapted from TE connectivity*

Therefore, when analyzing a fault in a particular system, it is essential to reduce the number of input variables by discarding information of sensors not related to these failures, allowing to reduce the complexity of the models and increasing the necessary space to obtain data from the selected sensors with higher resolution. To achieve this objective, several Machine Learning techniques can be used aimed at finding the smallest set of variables related to a failure.

To explore an extensive set of variables in a reasonable time, in this work we present a thorough study of automatic input selection algorithms for wind turbine failure prediction and propose an exhaustive-search-based quasi-optimal algorithm (QO), which has been used as a reference for the automatic algorithms. This will allow us to consider all variables of the analyzed subsystem and select automatically the smallest set of relevant variables, which in turn will simplify the models and permit a graphical representation of their time evolution.

2 MATERIAL AND METHODS

2.1 Automatic feature selection algorithms

When working with classification systems, using more variables doesn't implies better performance [3]. This is because the existence of redundant or little-correlated variables with the class to be predicted. The use of feature (variable) selection algorithms avoids exploring all possible combinations of them, generating smaller subsets with the most relevant and not redundant variables [1], [24]. One way to do this consists of applying a criterion allowing to obtain a score for each variable (feature) by employing information theory measures. These methods are iterative, obtaining the variables with the highest score for the original available set of variables.

For measuring the relevance, many methods use the iterative Mutual Information (MI) evaluation $I(X_k,Y)$, where X_k is the current variable under analysis and Y is the target class vector [6]. At each iteration, MI is computed for all the remaining variables and the best one is kept. However, in order to reduce redundancy, some methods rely on maximizing the complementary information among all variables selected in each iteration.

The feature selection algorithms that have been used in this work are detailed in table 1, which contains the list of acronyms, names, references and if the method employs a second term to avoid redundancy in features (*CondRed*) or has some way to capture the inter-class correlation (*Rel/Red*) that improves the classification performance (as it is observed in some data-sets). A detailed description of all these algorithms can be found in [8].

Table 1. Information-based criteria used in theses experiments. *CondRed*: Detection of redundancy between variables. *Rel/Red*: Balance of redundancy and relevancy.

Criterion	*Full name*	*Authors*	*CondRed*	*Rel/Red*
MIFS	Mutual Information Feature Selection	Battiti	yes	no
CMI	Conditional Mutual Information	**?**	yes	yes
JMI	Joint Mutual Information	Yang and Moody	yes	yes
mRMR	Min-Redundancy Max-Relevance	Peng et al.	yes	no
DISR	Double Input Symmetrical Relevance	Meyer and Bontempi	yes	yes
CMIM	Conditional Mutual Info Maximisation	Fleuret	yes	yes
ICAP	Interaction Capping	Jakulin	yes	yes

2.2 Proposed Exhaustive-search-based quasi-optimal algorithm

In this section, we will present a quasi-optimal (QO) algorithm for feature selection, in order to establish a reference or *gold standard* for the rest of experiments performed using automatic feature selection algorithms. Optimal feature selection implies to test all possible combinations and select the one that gives us the best classification rate. Unfortunately, this is only possible when the number of features is sufficiently small, due to the exponentially growing of possible combinations when increasing the number of features. This effect is known as the *curse of dimensionality*. Indeed, the number of combinations of n features taking k at a time (without repetition) is equal to the binomial coefficient.

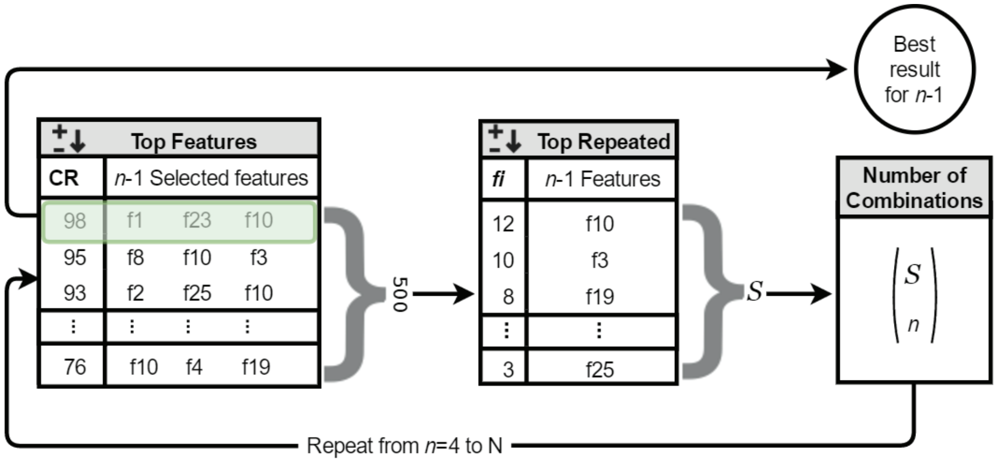


Figure 2. Proposed exhaustive-search-based quasi-optimal algorithm.

In our specific case, each sub-system has 4 variables (minimum value, maximum value, average value, standard deviation) which gives us 36 features (4 variables x 9 sub-systems) coming from the gearbox, transmission and nacelle wind sensors systems of wind turbines (see Table 2 for the exact list of variables). This implies, for example, that we have 7 140 combinations of three features, 58 905 combinations of four features and 376 992 combinations of five features. The worst case, when taking 18 features, gives a total of 9 075 135 300 combinations.

Therefore, we will calculate all the possible combinations of 1, 2 and 3 features and will implement a QO strategy for 4, 5, and 6 features. In all the cases, the criteria for selecting the best combination will be based on the classification rate obtained with the *k*-NN classifier. The following strategy (see Figure 2 for a block diagram) gives the details on how we implement the QO feature selection for more than 3 features. Lets suppose that we want to determine the best combination of *n* characteristics. Then:

1. Calculate the frequency of selection of the characteristics for the case *n-1* using the best 500 results.
2. Sort the features according to its frequency.
3. Select the subset of *S* features with the highest frequency.
4. Calculate all possible combinations of these *S* features taking *n* at a time (without repetition).
5. Select the best combination based on the classification rate obtained with the *k*-NN classifier.

For the case *n=4* we will use the best 20 frequent features (*S=20*) of the case *n=3*, which will generate a total of 4 845 combinations of *4* characteristics. For the case *n=5* we will use the best 15 features (*S=15*) of the case *n=4*, which will generate a total of 3 003 combinations of *5* characteristics. Finally, for the case *n=6* we will use the best 15 features (*S=15*) of the case *n=5*, which will generate a total of 5 005 combinations of *6* characteristics.

The advantage of optimal feature selection is that we are testing all possible combinations (interactions) between features. The disadvantage is the impossibility to implement a large number of combinations when the number of features is huge and we want to consider a substantial number of features in each group. The QO strategy presented above give us an approximation to the optimal feature selection, but still, we are probably missing some combinations which could be better, and even if we diminish the number of combinations we still have a considerable amount of cases to test with the classification algorithm. Moreover, we are interested in a fast algorithm for automatic feature selection, which can deal with all the 36 features and rank them accordingly to its importance for the classification problem. Therefore, the aim is to substitute the QO feature selection by one automatic feature selection algorithm without losing performance and allowing us to exploit all the available characteristics.

2.3 Study case

In the following section, we will detail the data-set used in the experiments and the classification system employed. The general scheme of experiments is depicted in Figure 3.

2.3.1 Data-set description

The collected data-set used in this work covers an entire year (2014) of a wind farm with five Fuhrländer wind turbines in Catalonia. The original set of more than 200 variables comes in a 5-minutes format for analogous variables and as a record of events for digital data (alarms) from the wind farm's SCADA. Among all these features, a subset of them related to wind turbine gearbox and transmission system was used in the experiments. The events are labeled as 0 for normal functioning, 1 for warning and 2 for alarm. The difference between warning and alarm is in the state of the wind turbine, on working for the warning state but stopped for the alarm state. Given into account that a warning is a sign that something wrong may occur, we will integrate warnings and alarms together and will focus on the improvement of classification events between working and failure (warning or alarm) condition.

2.3.2 Classification system

One of the simplest and oldest methods for classification is the *k* nearest neighbors (*k*-NN) classifier. It classifies an unknown observation to the class of majority among its *k* nearest neighbors observations, as measured by a distance metric, in the training data [9]. Despite its simplicity, *k*-NN gives competitive results and in some cases even outperforms other sophisticated learning algorithms. However, *k*-NN is affected by non-informative features in the data, often the case with high dimensional data. Attempts have been made to improve the performance of nearest neighbors classifier by ensemble techniques.

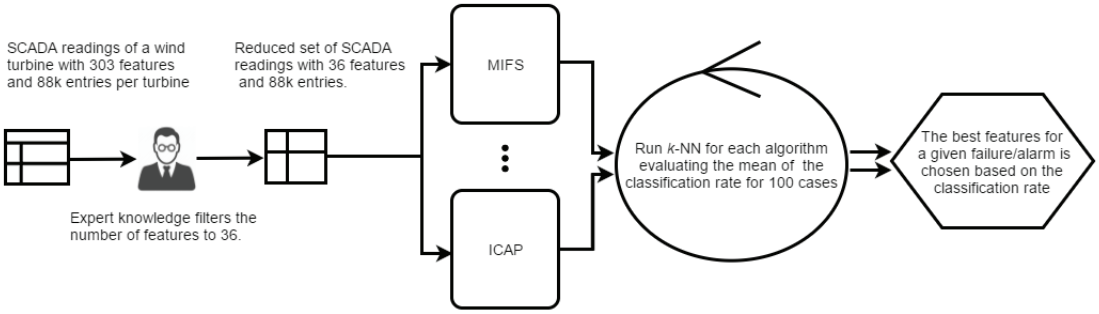

SCADA readings of a wind turbine with 303 features and 88k entries per turbine
Reduced set of SCADA readings with 36 features and 88k entries.
Expert knowledge filters the number of features to 36.
MIFS
ICAP
Run k-NN for each algorithm evaluating the mean of the classification rate for 100 cases
The best features for a given failure/alarm is chosen based on the classification rate

Figure 3. General scheme of the experiments

Some related work on an ensemble of *k*-NN classifiers can be found in [11] [25] [14] [20]. Analyzing a significant amount of data often consumes extensive computational resources and execution time. However, sometimes all data features do not equally contribute to the final results. Thus, it is plausible to identify the major contributing features and use them as representatives of the data. Other features with low contribution can be eliminated to reduce the time/resource consumption in data analysis.

In general, cases using *k*-NN classification *k=1* is often not the best choice as noise can easily degrade the classification accuracy. With the increase of *k*, multiple nearest neighbors helps improve the classification accuracy. However, if *k* is very large, the classification accuracy of *k*-NN tends to decrease as the nearest and farthest neighbors are assigned equal weights in the decision-making process. To sum up, the classification accuracy of the *k*-NN algorithm experiences a *rise–peak–drop* process and in practical situations, it is essential to determine the optimal *k* value. We will discuss the used value in section 3.

To measure the performance of our system we will use the Classification Rate (CR) as the percentage of well-classified instances divided by the total number of instances. To have statistically consistent results, we will calculate the CR for 100 different cases obtained by randomly splitting the database 100 times into two subsets: the first for deriving the model (training subset) and the second to test it (test subset). Since almost all the time the wind turbines (WT) are in normal state, the database is clearly biased and presents a high amount of instances of this class. Therefore we will balance the training set by keeping the same number of instances for each class. As the splitting process is random, all the instances will be used at the end of all 100 experiments. This strategy allows us to derive balanced classification systems with almost symmetrical confusion matrices.

3 EXPERIMENTAL RESULTS AND DISCUSSION

All the experiments (see Figure 3) will use the data-set presented in section 2.3.1, which contains 36 features and each target has a label indicating a healthy state, warning state or alarm state. A warning will be considered equivalent to alarms, therefore we face a binary classification problem. The selection of the best features to be used as input to the classification system was implemented as detailed in section 2.1. We have performed several experiments using all the WT and a range of features from 1 to 6 obtained through several feature selection algorithms. In panel (a) of figure 4 we plot the CR against the number of features for the quasi-optimal algorithm and all the WT. Results are excellent in all the WT, reaching above 85% of CR when the number of features is 3 or higher. Adding new features increases the CR slightly, but for more than 4 features the change is almost imperceptible. Numerical results of these experiments are detailed in Table 3. All results are obtained with *k=1*.

3.1 Quasi-optimal versus automatic feature selection

Next step is to look for a feature selection algorithms able to obtain similar results with a few number of features. Results for those feature selection algorithms are presented in panels (b) to (f) of Figure 4. Each panel corresponds to one WT and contains the result obtained for the quasi-optimal method (as a reference, dashed line) and the results obtained with all the others algorithms for this WT. Experimentally we observe that some WT are easy to model (see for example WT4) while others are more challenging (see for example WT5). Numerical results for all the experiments are detailed in Table 4. When comparing results obtained by the quasi-optimal exploratory method and the automatic feature selection methods we

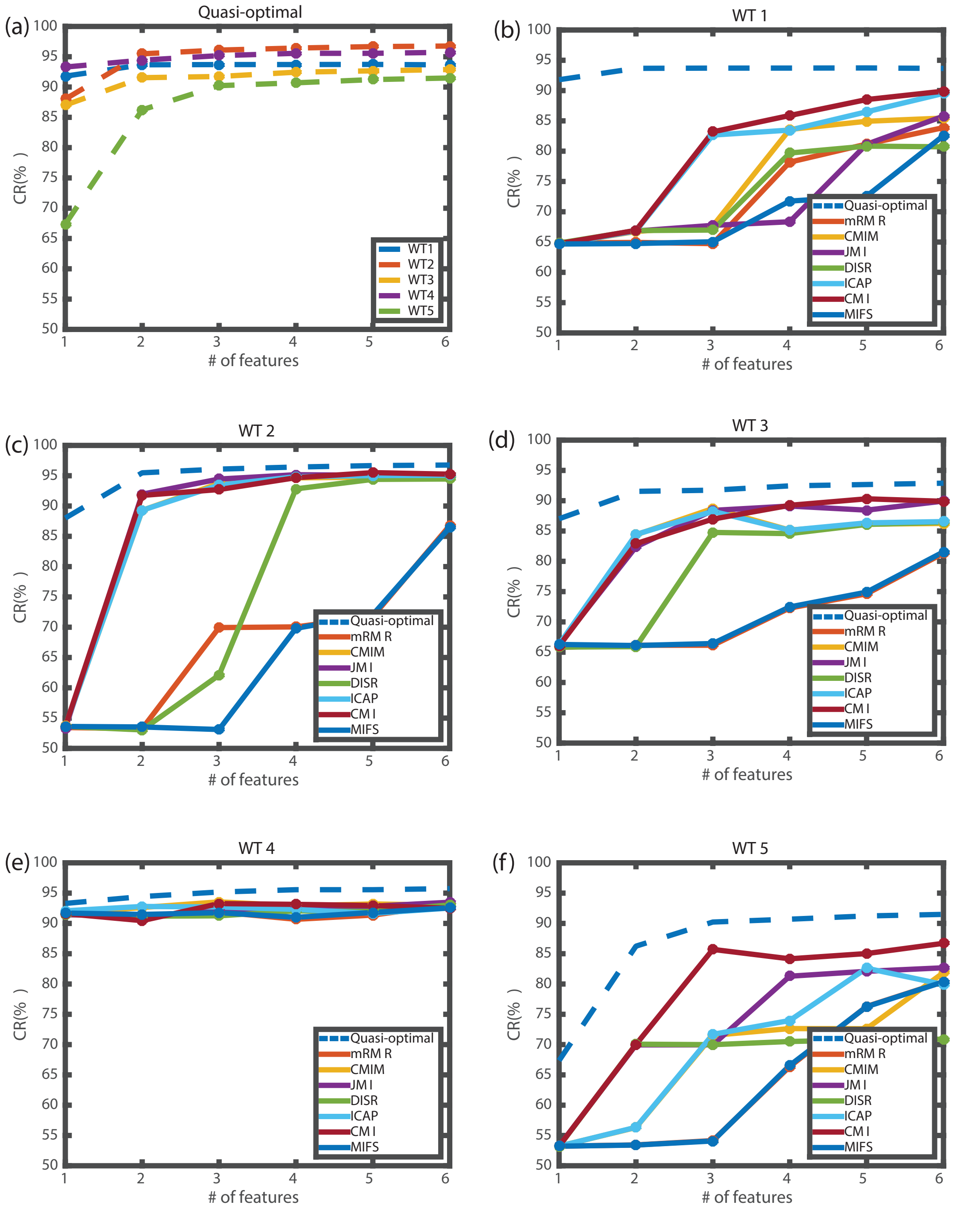


Figure 4. Evolution of the CR(%) against the number of features. (a): Quasi-optimal feature selection case, all WT. (b) to (f): Specific results for each WT and all the automatic feature selection algorithms analyzed. The dashed line in each panel corresponds to the quasi-optimal result for that specific WT.

observe that QO results are always the best ones, as expected, but several automatic methods also obtain very good solutions. Among all the automatic algorithms, CMI emerges as stable along all the WT and getting (almost) always an excellent result, comparable to that obtained with the quasi-optimal method for a number of features equal or higher than 4.

Exploring all possible combinations of features allow us to determine which number of features is the best one. We can see that CR saturates for 6 features, therefore the system will not increase its performance by adding new features. It is essential to keep the number of features as low as possible to develop less complex classification systems. Besides, if systems are less complicated it will be easier to train the models, and the risk of overfitting will be lower. Finally, using a small number of features can allow us to graphically represent the information, especially if we have up to 3 features, which is of great importance as a tool in the front-end of real applications for the managers of the wind farms. Hence, CMI with 3 or 4 features is an excellent choice in our experiment, with CR comparable to the quasi-optimal one for all WT.

3.2 Effect of the number of neighbors considered

To analyze the effect of the number of neighbors in the *k*-NN algorithm, we also performed experiments exploring all the cases for *k=1* to *k=50* in all the algorithms, using the best combination of features for each case. When analyzing the quasi-optimal case, *k=1* is the best choice for all the WT. But when we use any of the automatic feature selection algorithms, if the number of features is small then the number of neighbors affects the CR and habitually *k=1* is not the best option. Nevertheless, even increasing the number of neighbors, the CR obtained is lower than the QO case for the number of features analyzed. If the number of features increases, and therefore also the CR increases, *k=1* becomes again, the best choice and CR tends to the QO case. The advantage of increasing the neighbors is compensated by increasing the number of features. This effect can be observed in Figure 5: On the left column, we present two examples (WT1 and WT3) of the evolution of the CR as a function of *k*, for the quasi-optimal set of features from 1 to 6. On the right column, we can see the same WT but now using features obtained with the best feature selection algorithm among all the analyzed algorithms. Note that increasing the number of neighbors is only useful for the CMI algorithm when the number of features used is small (1 or 2), but does not help to increase the CR when the number of features is higher. For the quasi-optimal feature selection case, *k=1* is (almost) always the best option regardless of the number of features. Therefore, changing the number of neighbors has only impact when using 1 or 2 features in the CMI algorithm and degrades CR when the number of features is high or when we use QO method.

4 CONCLUSIONS

In this paper we explore several methods for automatic feature selection for wind turbine failure prediction. These features come from several sensors that monitors the turbine status. Due to the large number of available variables, Machine Learning algorithms have to be used to select the best subset among them. Therefore we compare the performance of several feature selection algorithms against the quasi-optimal feature selection performed by examining all possible combinations of features as explained in section 2.3.2. Experimental results using the 36 sensor variables listed in Table 2 show that CMI algorithm obtains good CR for all the WT with up to six features and only one neighbor. Therefore, we can speed the system by using this algorithm instead of exhaustive-search-based quasi-optimal strategy. The advantages are its low computational cost and fast speed calculation to find the best subset of features for wind turbine failure prediction. Although our studies confirm that a selected set of three to six more discriminant variables are required to obtain the best prognosis performance, that selection is somewhat difficult to be represented. This is why sets of three selected variables, admitting a 3D Cartesian plot, becomes interesting. In this scenario, time evolution can be included generating plot animations. These dynamic representations provide powerful and intuitive insights about the behavior of variables 21 days before failure and become a useful tool to improve the models used for prognostic. In future works we will explore these dynamic representations of three features in order to visualize interactions between them, aiming to simplify and facilitate optimal management of wind parks.

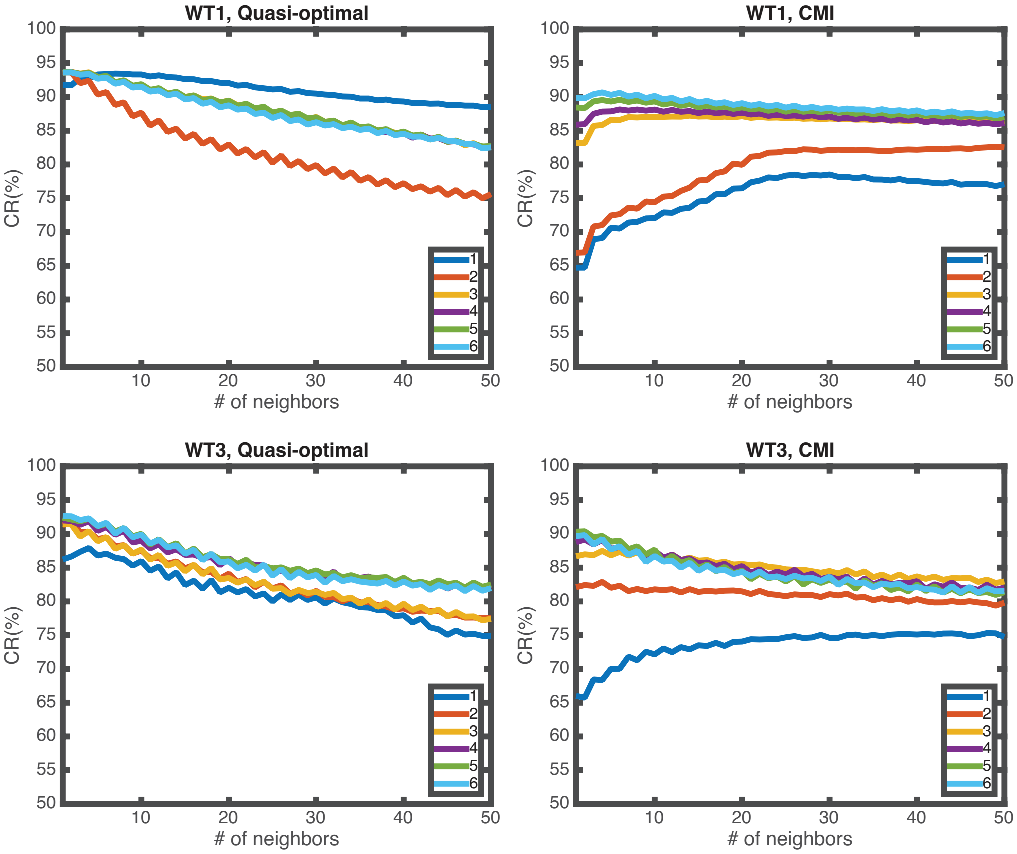


Figure 5. Effect of the number of neighbors for WT1 and WT3. Each colored curve corresponds to a specific number of features, from 1 to 6. Only the Optima-feature selection case and the CMI case are reported here.

Table 2. Variable code to variable name.

Group	*Variable code*	*Variable Name*	*Description*
A	1	WGDC.TrfGri.PwrAt.cVal.avgVal	Active power
	2	WGDC.TrfGri.PwrAt.cVal.minVal	
	3	WGDC.TrfGri.PwrAt.cVal.maxVal	
	4	WGDC.TrfGri.PwrAt.cVal.sdvVal	
B	1	WTRM.TrmTmp.Brg1.avgVal	Main bearing 1 Temperature
	2	WTRM.TrmTmp.Brg1.minVal	
	3	WTRM.TrmTmp.Brg1.maxVal	
	4	WTRM.TrmTmp.Brg1.sdvVal	
C	1	WTRM.TrmTmp.Brg2.avgVal	Main bearing 2 Temperature
	2	WTRM.TrmTmp.Brg2.minVal	
	3	WTRM.TrmTmp.Brg2.maxVal	
	4	WTRM.TrmTmp.Brg2.sdvVal	
D	1	WTRM.Brg.OilPres.avgVal	Main bearing oil pressure
	2	WTRM.Brg.OilPres.minVal	
	3	WTRM.Brg.OilPres.maxVal	
	4	WTRM.Brg.OilPres.sdvVal	
E	1	WTRM.Gbx.OilPres.avgVal	Gearbox oil pressure
	2	WTRM.Gbx.OilPres.minVal	
	3	WTRM.Gbx.OilPres.maxVal	
	4	WTRM.Gbx.OilPres.sdvVal	
F	1	WTRM.Brg.OilPresIn.avgVal	Main bearing oil pressure
	2	WTRM.Brg.OilPresIn.minVal	
	3	WTRM.Brg.OilPresIn.maxVal	
	4	WTRM.Brg.OilPresIn.sdvVal	
G	1	WNAC.WSpd1.avgVal	Wind Speed sensor 1
	2	WNAC.WSpd1.minVal	
	3	WNAC.WSpd1.maxVal	
	4	WNAC.WSpd1.sdvVal	
H	1	WNAC.Wdir1.avgVal	Wind direction sensor 1
	2	WNAC.Wdir1.minVal	
	3	WNAC.Wdir1.maxVal	
	4	WNAC.Wdir1.sdvVal	
I	1	WNAC.Wdir2.avgVal	Wind director sensor 2
	2	WNAC.Wdir2.minVal	
	3	WNAC.Wdir2.maxVal	
	4	WNAC.Wdir2.sdvVal	

Table 3. Numerical results for the CR(%) and list of best features for the quasi-optimal feature selection case. Results are grouped in sub-tables for each WT and each row of each sub-table corresponds to the number of features from 1 to 6. The column indicating the selected features uses the variables codes detailed in table 2.

	CR(%)	1F	CR(%)	2F	CR(%)	3F	CR(%)	4F	CR(%)	5F	CR(%)	6F
WT1	91.79	*A1*	93.67	*A2 E3*	93.71	*A2 B2 B3*	93.71	*A1 B1 B2 B3*	93.73	*A3 A4 B1 B3 B4*	93.66	*A1 A2 A3 B1 B2 B3*
	91.78	*A3*	93.66	*A1 E3*	93.70	*A3 B4 E2*	93.70	*A1 A3 B4 E3*	93.69	*A1 A2 A3 B4 E3*	93.64	*A1 A4 B1 B2 B3 B4*
	91.71	*A2*	93.65	*A3 B1*	93.70	*A2 B1 B3*	93.68	*A1 A2 A3 E3*	93.68	*A1 A3 A4 B4 E3*	93.61	*A1 A2 A3 A4 B4 E2*
	81.70	*B3*	93.64	*A3 E3*	93.69	*A1 A3 E3*	93.68	*A3 A4 B2 B3*	93.67	*A1 A4 B1 B3 B4*	93.61	*A1 A3 A4 B1 B2 B3*
	81.63	*B2*	93.62	*A2 B3*	93.69	*A1 B1 B2*	93.67	*A2 A4 B2 B3*	93.65	*A3 B1 B2 B3 B4*	93.60	*A1 A2 A3 A4 B1 B2*
WT2	88.01	*B3*	95.48	*A3 C2*	96.10	*A2 C2 D1*	96.43	*B1 C2 D1 G3*	96.67	*A3 B1 C2 D2 G3*	96.77	*A2 A3 B3 C2 D2 H1*
	87.87	*B1*	95.46	*A1 C2*	96.05	*A3 C2 D1*	96.42	*A3 C2 D1 G3*	96.62	*A2 B2 C2 D1 G3*	96.74	*A1 A3 B1 C2 D2 H1*
	87.85	*B2*	95.31	*A2 C2*	95.99	*A3 C2 D2*	96.38	*A2 C2 D1 H1*	96.56	*A1 A3 C2 D2 H1*	96.73	*A1 A2 B3 C2 D1 G3*
	85.83	*C2*	95.20	*B2 C2*	95.89	*A1 C2 D1*	96.38	*B1 C2 D2 G3*	96.55	*A2 A3 C2 D1 H1*	96.73	*A2 A3 B1 C2 D2 G3*
	85.60	*E1*	94.99	*B3 C2*	95.77	*A2 C2 D2*	96.38	*A1 C2 D2 G3*	96.55	*A3 B3 C2 D1 G1*	96.73	*A1 A2 B1 C2 D1 H1*
WT3	87.02	*C3*	91.54	*A2 E3*	91.74	*A3 B1 E3*	92.45	*A3 C1 D3 E3*	92.67	*B3 C1 C3 D2 E3*	92.89	*B3 C1 C3 D2 E1 E3*
	86.90	*C2*	91.44	*A1* E3	91.73	*B1 C3 E3*	92.36	*A1 C1 D3 E3*	92.66	*B3 C1 C3 D2 E1*	92.85	*B1 C1 C3 D2 E1 E3*
	79.33	*B1*	91.37	*A3* E3	91.67	*A2 B3 E3*	92.23	*B1 C1 D1 E3*	92.61	*A3 C1 D2 E1 E3*	92.82	*A2 C1 C3 D2 E1 E3*
	78.95	*B2*	91.10	*B2* E3	91.65	*A3 A4 E3*	92.18	*B3 C1 D3 E3*	92.58	*B1 C1 C3 D2 E3*	92.80	*A3 C1 C3 D2 E1 E3*
	78.79	*B3*	91.01	*B1* E3	91.62	*B3 C3 E3*	92.17	*B2 C1 D2 E3*	92.58	*B2 C1 C3 D2 E1*	92.78	*B1 B4 C1 C3 D2 E3*
WT4	93.30	*C2*	94.44	*C2 D2*	95.18	*B1 C2 D2*	95.56	*B1 C2 D2 E2*	95.56	*B1 B2 C2 D2 H3*	95.74	*B1 C2 D2 D3 E2 H3*
	92.27	*C3*	94.32	*D1 E2*	95.14	*C2 D2 H3*	95.47	*B1 C2 D2 H3*	95.54	*B3 C2 D2 E2 H3*	95.59	*A4 B1 C2 D2 D3 H3*
	91.46	*C1*	94.32	*D2 E2*	94.97	*B3 C2 D2*	95.37	*B1 B4 C2 D2*	95.42	*B1 B3 C2 D2 D3*	95.55	*B2 B3 B4 C2 D2 E2*
	91.29	*D2*	94.22	*C2 D1*	94.94	*C2 D1 H3*	95.30	*B1 B3 C2 D2*	95.42	*B1 B4 C2 D2 E2*	95.55	*B1 B2 C2 D1 D2 E2*
	90.98	*D3*	93.74	*B3 C2*	94.92	*D1 E2 H3*	95.29	*B2 C2 D2 H3*	95.40	*B1 C2 D2 D3 H3*	95.47	*A4 B3 C2 D2 E2 H3*
WT5	67.37	*A2*	86.25	*A1 E2*	90.23	*A3 C3 E2*	90.70	*A2 C3 E2 E3*	91.23	*A1 B2 C3 E2 E3*	91.49	*A1 B3 C1 C3 E3 G1*
	67.28	*A3*	86.08	*A3 E2*	90.12	*A2 C3 E2*	90.64	*A3 C3 E2 E3*	91.22	*A3 B2 C3 E2 E3*	91.47	*A2 B3 C1 C3 E3 G1*
	67.21	*A1*	86.05	*A2 E2*	90.12	*A1 C3 E2*	90.63	*A1 C3 E2 E3*	91.22	*A2 B3 C3 E2 E3*	91.46	*A2 B1 C1 E2 E3 G1*
	66.31	*B3*	85.96	*A3 E3*	90.01	*A2 C2 E3*	90.62	*A1 B1 C3 E2*	91.22	*A1 B1 C3 E2 E3*	91.42	*A3 B3 C1 C3 E3 G1*
	66.27	*B2*	85.92	*A3 E1*	89.98	*A2 C3 E3*	90.59	*A1 B3 C3 E2*	91.22	*A1 B3 C3 E2 E3*	91.42	*A2 B2 C1 C3 E2 E3*

Table 4. Numerical results for the CR(%) and list of best features for the automatic feature selection algorithms analyzed and each WT. Results are grouped in sub-tables for each algorithm, and each row of each sub-table corresponds to wind turbines (WT1 to WT5). The column indicating the selected features uses the variables codes detailed in table 2.

	CR(%)	**1F**	**CR(%)**	**2F**	**CR(%)**	**3F**	**CR(%)**	**4F**	**CR(%)**	**5F**	**CR(%)**	**6F**
cmi	64.73	*E1*	66.93	*E1 E4*	83.19	*E1 E4 F1*	85.89	*E1 E4 F1 H1*	88.52	*A1 E1 E4 F1 H1*	89.9	*A1 C4 E1 E4 F1 H1*
	53.58	*E4*	91.76	*C2 E4*	92.72	*C2 E4 H1*	94.68	*A2 C2 E4 H1*	95.51	*A2 C2 D3 E4 H1*	95.26	*A2 C2 D3 E2 E4 H1*
	66.03	*D3*	82.92	*B1 D3*	86.97	*B1 C2 D3*	89.24	*B1 C2 D3 G3*	90.31	*B1 C2 D3 E3 G3*	89.90	*B1 C2 D3 E3 F4 G3*
	91.62	*D2*	90.45	*D2 F3*	93.27	*D2 E2 F3*	93.15	*D2 E2 E3 F3*	92.95	*A1 D2 E2 E3 F3*	92.50	*A1 D2 E2 E3 F3 H4*
	53.24	*E2*	70.03	*C3 E2*	85.71	*C3 E2 H3*	84.16	*C3 E2 F4 H3*	85.03	*C3 E2 F4 H1 H3*	86.72	*A1 C3 E2 F4 H1 H3*
cmim	64.68	*E1*	66.74	*E1 E4*	67.66	*E1 E2 E4*	83.59	*C1 E1 E2 E4*	84.94	*C1 C2 E1 E2 E4*	85.46	*C1 C2 E1 E2 E3 E4*
	53.68	*E4*	89.29	*D1 E4*	93.73	*A1 D1 E4*	94.64	*A1 D1 E2 E4*	94.78	*A1 D1 E2 E3 E4*	95.14	*A1 D1 E1 E2 E3 E4*
	66.02	*D3*	84.37	*C3 D3*	88.71	*B1 C3 D3*	85.15	*B1 C3 D3 H3*	86.13	*B1 C3 D3 F1 H3*	86.25	*A1 B1 C3 D3 F1 H3*
	91.60	*D2*	92.63	*D2 E3*	93.55	*D2 E2 E3*	92.91	*A1 D2 E2 E3*	93.21	*A1 D2 E2 E3 F4*	93.03	*A1 D2 E2 E3 F3 F4*
	53.24	*E2*	56.31	*E2 E3*	71.53	*E2 E3 F4*	72.64	*E1 E2 E3 F4*	72.62	*E1 E2 E3 E4 F4*	81.87	*C1 E1 E2 E3 E4 F4*
disr	64.84	*E1*	66.9	*E1 E4*	66.98	*B4 E1 E4*	79.69	*B4 C4 E1 E4*	80.83	*B4 C4 E1 E2 E4*	80.72	*A4 B4 C4 E1 E2 E4*
	53.62	*E4*	53.05	*A4 E4*	62.10	*A4 C4 E4*	92.83	*A4 C2 C4 E4*	94.40	*A1 A4 C2 C4 E4*	94.46	*A1 A4 C1 C2 C4 E4*
	65.84	*D3*	65.91	*A4 D3*	84.76	*A4 C3 D3*	84.57	*A4 C3 D1 D3*	86.08	*A4 C1 C3 D1 D3*	86.43	*A4 C1 C3 D1 D2 D3*
	91.52	*D2*	91.19	*A4 D2*	91.25	*A4 D1 D2*	92.07	*A4 D1 D2 D3*	91.96	*A4 B4 D1 D2 D3*	93.05	*A4 B4 D1 D2 D3 E3*
	53.19	*E2*	70.07	*C3 E2*	69.99	*C3 E1 E2*	70.51	*C3 E1 E2 E3*	70.80	*C2 C3 E1 E2 E3*	70.89	*C2 C3 C4 E1 E2 E3*
icap	64.64	*E1*	66.84	*E1 E4*	82.66	*C1 E1 E4*	83.48	*C1 E1 E3 E4*	86.50	*C1 E1 E3 E4 G1*	89.53	*A1 C1 E1 E3 E4 G1*
	53.65	*E4*	89.30	*D1 E4*	93.45	*A1 D1 E4*	94.84	*A1 D1 E2 E4*	95.02	*A1 D1 E1 E2 E4*	95.08	*A1 D1 E1 E2 E3 E4*
	66.28	*D3*	84.43	*C3 D3*	88.25	*B1 C3 D3*	85.13	*B1 C3 D3 H3*	86.34	*B1 C3 D3 F1 H3*	86.55	*A1 B1 C3 D3 F1 H3*
	92.08	*D2*	92.80	*D2 E3*	92.71	*A1 D2 E3*	92.31	*A1 D2 E3 F4*	91.65	*A1 D2 E3 F3 F4*	92.54	*A1 D2 E3 F3 F4 H1*
	53.23	*E2*	56.35	*E2 E3*	71.69	*E2 E3 F4*	73.97	*C4 E2 E3 F4*	82.60	*C1 C4 E2 E3 F4*	79.92	*C1 C4 E2 E3 F2 F4*
jmi	64.67	*E1*	66.82	*E1 E4*	67.75	*E1 E2 E4*	68.35	*E1 E2 E3 E4*	81.13	*C4 E1 E2 E3 E4*	85.78	*C2 C4 E1 E2 E3 E4*
	53.30	*E4*	91.96	*C2 E4*	94.45	*A1 C2 E4*	95.17	*A1 C2 D1 E4*	95.07	*A1 A2 C2 D1 E4*	94.99	*A1 A2 C2 D1 E2 E4*
	66.26	*D3*	82.39	*B1 D3*	88.40	*B1 C3 D3*	89.12	*B1 C3 D2 D3*	88.44	*B1 C3 D1 D2 D3*	89.94	*B1 C1 C3 D1 D2 D3*
	91.43	*D2*	91.30	*D2 F3*	92.02	*D2 D3 F3*	92.73	*D2 D3 E3 F3*	92.84	*D1 D2 D3 E3 F3*	93.49	*D1 D2 D3 E2 E3 F3*
	53.28	*E2*	69.95	*C3 E2*	69.96	*C3 E1 E2*	81.29	*C3 E1 E2 F4*	82.09	*C3 E1 E2 E3 F4*	82.68	*C2 C3 E1 E2 E3 F4*
mifs	64.68	*E1*	64.76	*B4 E1*	65.05	*A4 B4 E1*	71.76	*A4 B4 D4 E1*	72.57	*A4 B4 D4 E1 G4*	82.56	*A4 B4 C4 D4 E1 G4*
	53.62	*E4*	53.54	*A4 E4*	53.11	*A4 B4 E4*	69.82	*A4 B4 E4 G4*	72.06	*A4 B4 E4 F4 G4*	86.47	*A4 B4 D4 E4 F4 G4*
	66.27	*D3*	66.10	*B4 D3*	66.43	*A4 B4 D3*	72.47	*A4 B4 C4 D3*	74.91	*A4 B4 C4 D3 G4*	81.55	*A4 B4 C4 D3 G1 G4*
	91.71	*D2*	91.48	*A4 D2*	91.77	*A4 B4 D2*	91	*A4 B4 D2 G4*	91.81	*A4 B4 C4 D2 G4*	92.56	*A4 B4 C4 D2 G3 G4*
	53.23	*E2*	53.42	*A4 E2*	54.09	*A4 B4 E2*	66.56	*A4 B4 E2 G4*	76.26	*A4 B4 E2 G4 H2*	80.36	*A4 B4 C4 E2 G4 H2*
mrmr	64.83	*E1*	64.94	*B4 E1*	64.74	*A4 B4 E1*	78.19	*A4 B4 C4 E1*	81.16	*A4 B4 C4 D4 E1*	83.89	*A4 B4 C4 D4 E1 H1*
	53.44	*E4*	53.26	*A4 E4*	69.94	*A4 E4 G4*	70.05	*A4 B4 E4 G4*	71.76	*A4 B4 E4 F4 G4*	86.77	*A4 B4 D4 E4 F4 G4*
	65.81	*D3*	66.14	*B4 D3*	66.14	*A4 B4 D3*	72.29	*A4 B4 C4 D3*	74.63	*A4 B4 C4 D3 G4*	81.33	*A4 B4 C4 D3 G1 G4*
	91.45	*D2*	91.32	*A4 D2*	91.88	*A4 B4 D2*	90.68	*A4 B4 D2 G4*	91.37	*A4 B4 C4 D2 G4*	93.12	*A4 B4 C4 D2 G3 G4*
	53.24	*E2*	53.44	*A4 E2*	54.16	*A4 B4 E2*	66.37	*A4 B4 E2 G4*	76.29	*A4 B4 E2 G4 H2*	80.40	*A4 B4 C4 E2 G4 H2*

ACKNOWLEDGEMENTS

Financial support by the Agency for Management of University and Research Grants (AGAUR) of the Catalan Government to Alejandro Blanco-M. is gratefully acknowledged.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

REFERENCES

[1] Feature selection based on mutual information: Criteria of maxdependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1226–1238, 2005. doi: 10.1109/TPAMI.2005.159.

[2] Lossless compression of wind plant data. *IEEE Transactions on Sustainable Energy*, 2012(3): 598–606, 2012. doi: 10.1109/TSTE.2012.2195039.

[3] The effectiveness of feature selection method in solar power prediction. *Journal of Renewable Energy*, 2013(952613):9, 2013. doi: 10.1155/2013/952613.

[4] Vestas v90-3mw wind turbine gearbox health assessment using a vibration-based condition monitoring system. *Shock and Vibration*, 2016(6423587):18, 2016. doi: 10.1155/2016/6423587.

[5] High frequent scada-based thrust load modeling of wind turbines. *Wind Energy Science, EAWE*, 2017. doi: 10.5194/wes-2017-46.

[6] Mutual information based analysis for the distribution of financial contagion in stock markets. *Discrete Dynamics in Nature and Society*, 2017(3218042):13, 2017. doi: 10.1155/2017/3218042.

[7] Roberto Battiti. Using mutual information for selecting features in supervised neural net learning. *Neural Networks, IEEE Transactions on*, 5(4):537–550, 1994.

[8] Gavin Brown, Adam Pocock, Ming-Jie Zhao, and Mikel Luján. Conditional likelihood maximisation: a unifying framework for information theoretic feature selection. *The Journal of Machine Learning Research*, 13(1):27–66, 2012.

[9] Thomas Cover and Peter Hart. Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13(1):21–27, 1967.

[10] Edwin Wright David Bailey. *Practical SCADA for Industry*. Elsevier Science B.V, 2003.

[11] Carlotta Domeniconi and Bojun Yan. Nearest neighbor ensemble. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 1, pages 228–231. IEEE, 2004.

[12] François Fleuret. Fast binary feature selection with conditional mutual information. *The Journal of Machine Learning Research*, 5:1531–1555, 2004.

[13] GEsoftware. The case for an industrial big data platform. Technical report, General Electric (GE), 2013.

[14] Peter Hall and Richard J Samworth. Properties of bagged nearest neighbour classifiers. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(3):363–379, 2005.

[15] IEC. International standard iec 61400-25-1. dec 2006.

[16] Aleks Jakulin. *Machine learning based on attribute interactions*. PhD thesis, Univerza v Ljubljani, 2005.

[17] Patrick E Meyer and Gianluca Bontempi. On the use of variable complementarity for feature selection in cancer classification. In *Applications of Evolutionary Computing*, pages 91–102. Springer, 2006.

[18] Hanchuan Peng, Fuhui Long, and Chris Ding. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(8):1226–1238, 2005.

[19] Vestas R+D. General specification vestas v90 3.0mw. Technical report, Vestas Wind Systems, 2004.

[20] Richard J Samworth et al. Optimal weighted nearest neighbour classifiers. *The Annals of Statistics*, 40(5):2733–2763, 2012.

[21] Vestas&IBM. Turning climate into capital with big data. Technical report, IBM, 2011.

[22] Michael Wilkinson. Use of higher frequency scada data for turbine performance optimisation. Technical report, DNV GL, EWEA, April 2016.

[23] Howard Hua Yang and John E Moody. Data visualization and feature selection: New algorithms for nongaussian data. In *NIPS*, volume 99, pages 687–693. Citeseer, 1999.

[24] Lei Yu and Huan Liu. Efficient feature selection via analysis of relevance and redundancy. *Journal of Machine Learning Research*, 5:1205–1224, 2004.

[25] Zhi-Hua Zhou and Yang Yu. Adapt bagging to nearest neighbor classifiers. *Journal of Computer Science and Technology*, 20(1):48–54, 2005.

CAPÍTULO 4

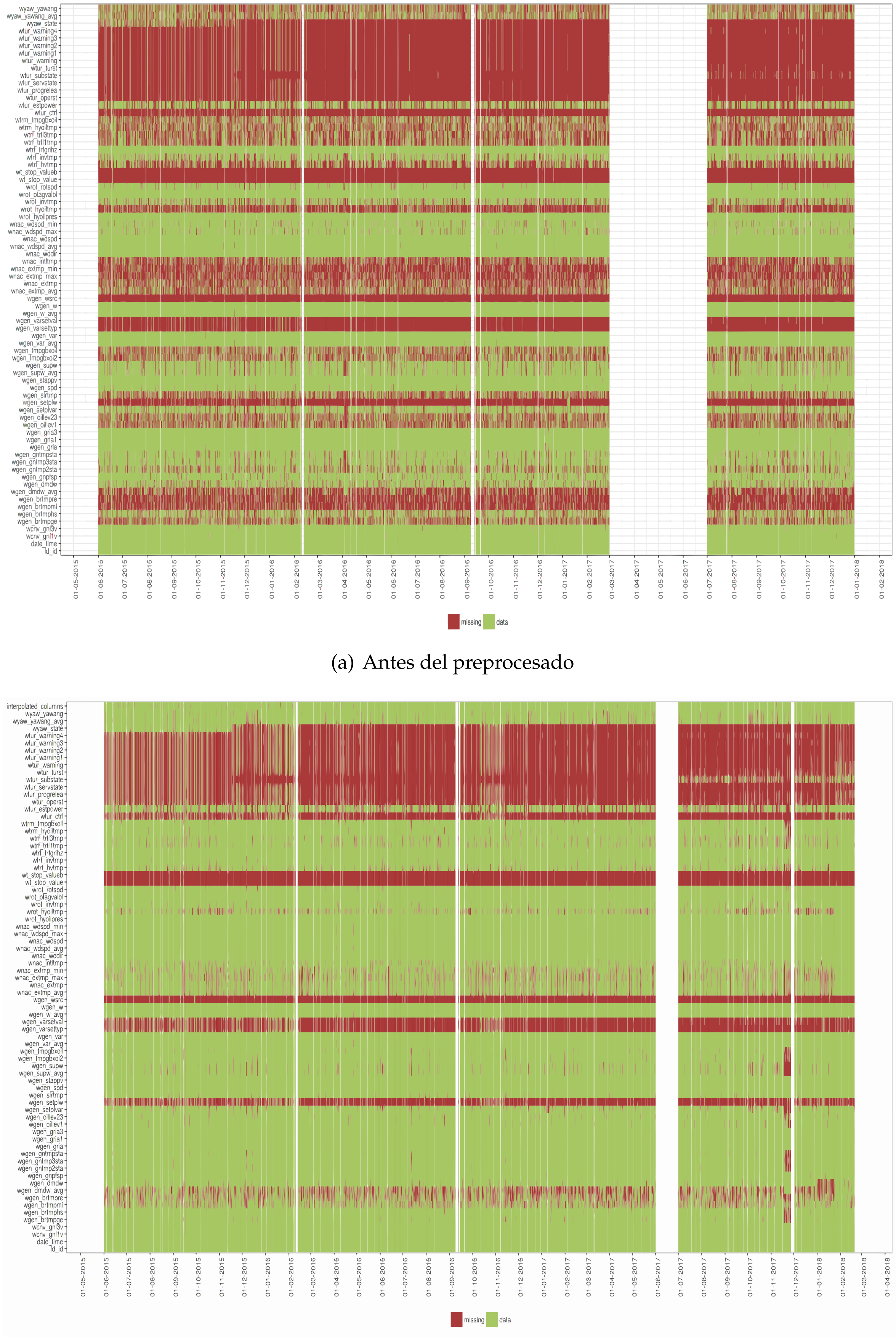
RESUMEN DEL TRABAJO REALIZADO DENTRO DE LA TESIS

4.1. Preprocesado

El procesado de los datos es un paso importante, sobre todo cuando se tratan de datos capturados en entornos industriales donde la pérdida de información e introducción de errores humanos, resulta una fuente común de ruido que afecta a la calidad de los datos [24].

4.1.1. Casos particulares

En algunos casos se han tenido que implementar métodos de reconstrucción/imputación de datos facilitados por el gestor del parque eólico, como con los datos de EDP Renováveis (EDPR) que solo almacena datos nuevos si un sensor (variable) ha cambiado respecto al registro anterior en alguno de sus 4 indicadores estadísticos (min,max,media,desviación). Esto significa que los datos sin tratar vienen con muchos huecos (como en la Figura 4.1(a)) pero que aplicando una técnica de preprocesado acaba resultando como la Figura 4.1(b). En este caso se ha utilizado por copia/interpolación, mediante la media del último registro siempre y cuando no supere 12 registros de diferencia (2 horas en frecuencia de datos 10 minútales). Para poder discriminar entre un hueco por error de comunicación o porque no ha cambiado, se utiliza la variable viento (*wnac_wdspd*), ya que esta siempre se actualiza. Esta forma de guardar datos, es por el ahorro y compresión del espacio en las distintas bases de datos de los mantenedores de parques eólicos, ya que generan GB de información en poco tiempo [54].



(a) Antes del preprocesado

(b) Después del preprocesado

Figura 4.1: *Heatmap* antes y después del preprocesado. Eje vertical muestra las variables, horizontal la fecha. Huecos en rojo, datos en verde.

4.1.2. Eliminación de valores extremos

Después de haber transformado los datos, se aplican metodologías de eliminación de ruido basadas en métodos univariable y multivariables. Dentro de los univariables tenemos los cubiertos en la publicación de la Sección 3.2, que son métodos comúnmente utilizados basados en test estadísticos como el *Extreme Studentized Deviate* (ESD), filtro de cuartiles y basado en medianas como el *Hampel Filter* que utiliza las desviaciones absolutas de medias (MAD, Median Absolute Deviations). Estos filtros presentan distintos niveles de sensibilidad frente a los valores extremos (*outliers*), siendo efectivos frente a datos sintéticos [12] con configuraciones comunes. Sin embargo, estos filtros han de ser ajustados de una forma particular para los Datasets de las turbinas eólicas, pudiendo evitar que ocurra lo descrito en la publicación de la Sección 3.2.

4.1.3. Técnicas de muestreo de los datos

Los Datasets generados a partir de los datos de turbinas de viento, se caracterizan por ser extremadamente desbalanceados [14]. Debido a esto, podemos encontrar casos en que la clase mayoritaria se compone del 97 % de los datos (registros no marcados como fallo) frente al 3 % de los datos (registros marcados como fallo). Esto produce que los métodos de clasificación tengan una alta *especificidad* y una baja *sensibilidad* [49], produciendo una *exactitud* (Accuracy) alta, similar al porcentaje que ocupa la clase mayoritaria en el Dataset.

Para atenuar este efecto se aplican las técnicas descritas en las siguientes secciones.

Oversampling por copia

En el *oversampling* se repiten tantas muestras tomadas de forma aleatoria (con reposición) de la clase minoritaria, la cual presenta un estado de fallo, hasta igualar en una proporción a la clase mayoritaria. Este método provoca en muchas ocasiones *overfitting*, obteniendo un modelo que tiene poca capacidad de generalización [29]. En otros casos es inefectivo para técnicas de clusterización, ya que genera los mismos puntos una y otra vez, no aportando más información a algoritmos como el kNN.

Undersampling por selección aleatoria

En undersampling se eliminan casos de la clase mayoritaria, por lo general sampleando una cantidad reducida de forma aleatoria sin reposición, hasta igualar en una cierta proporción a la clase minoritaria. El problema de este método es que, en el caso de las turbinas donde se ve condicionado por la estacionalidad y el viento, el muestreo puede no contener muestras de distintas estaciones o modos de funcionamiento de la turbina.

TOMEK + SMOTE + RBM

Este ensamble se compone de tres métodos encontrados en el estado del arte, el cual tras diversas verificaciones con Datasets de turbinas eólicas han dado buen resultado. Existen distintos trabajos publicados acerca de clasificación en otros campos, donde se ha validado la eficacia de concatenar dos de ellos (SMOTE + RBM [65]). En este caso, se han concatenado tres de forma secuencial como se describe en las siguientes líneas:

El primer método aplicado es el *TOMEK* [53], que crea en un primer paso los *"T-link"* buscando para cada individuo el vecino más cercano, y en un segundo paso va eliminando cada pareja de *"T-link"*, donde cada elemento pertenece a una clase distinta del elemento que pertenezca a la mayoritaria. Con este método se hace *undersampling* de forma indirecta, aunque sólo eliminando los puntos de la clase mayoritaria que están cerca de la minoritaria, dejando mayores diferencias entre ambas clases.

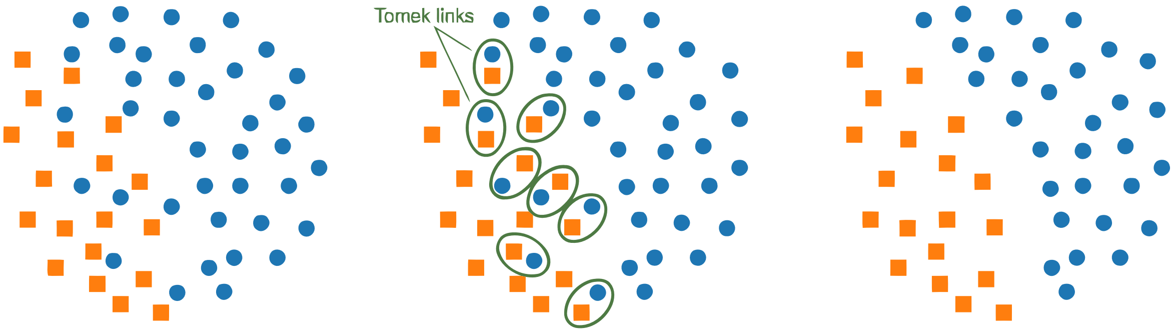


Figura 4.2: Ejemplo gráfico de la aplicación del método TOMEK

El segundo método, *SMOTE* (Synthetic Minority Oversampling Technique) [9], se aplica únicamente a los individuos de la clase minoritaria. Este método toma individuo a individuo, busca los **k** vecinos mas cercanos y selecciona a uno de forma aleatoria. Dado esto, se calcula la distancia entre el individuo y el vecino seleccionado y se multiplica por un factor de 0 a 1 de forma aleatoria para cada una de sus variables. El método se puede repetir hasta conseguir la cantidad de individuos sintéticos deseado.

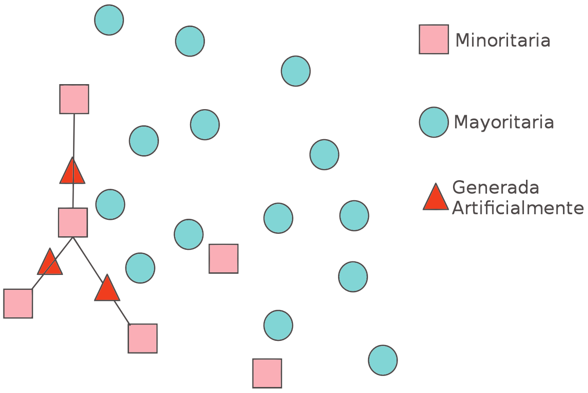


Figura 4.3: Ejemplo gráfico de la aplicación del método SMOTE

El tercer método, RBM (Restricted Boltzmann Machine) [28], se utiliza para adecuar los nuevos individuos creados por el *SMOTE*, dándole un comportamiento más cercano a los propios datos reales desde los que han sido creados. El RBM es un tipo de red ANN, que conecta todos los nodos entre sí siendo capaz de capturar la relación entre variables de entrada. El RBM se compone de neuronas visibles (tantas como variables de entrada) y neuronas ocultas (de cantidad variable, independientes a la de entrada), las neuronas visibles y ocultas están conectadas entre sí por los pesos y las funciones de activación. La red funciona en un entrenamiento de dos pasos (*Forward* y *Backward*), en el que se ajustan los pesos. Para poder utilizarla, se deben hacer ambos pasos pero sin ajustar los pesos, haciendo que el resultado de salida sea adecuado por la relación capturada en el entrenamiento. En este caso, la red se preentrena con todos los individuos de la clase minoritaria originales y en una segunda fase, se presentan los individuos generados de forma sintética por el *SMOTE*, que al hacer un ciclo completo de *Forward* y *Backward*, el modelo de RBM transforma los datos de forma que las relaciones entre las variables se adecuan a las de los datos reales.

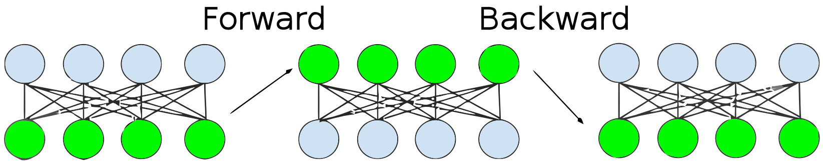


Figura 4.4: Pasos funcionamiento Restricted Boltzmann Machine (RBM)

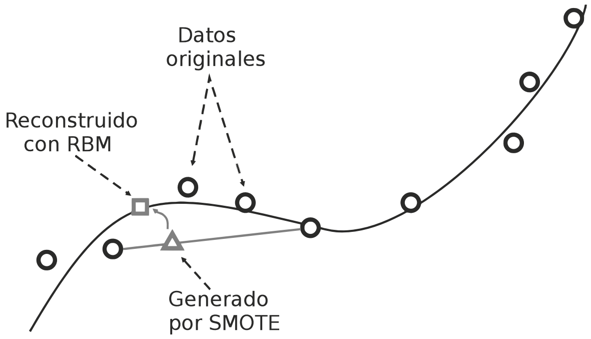
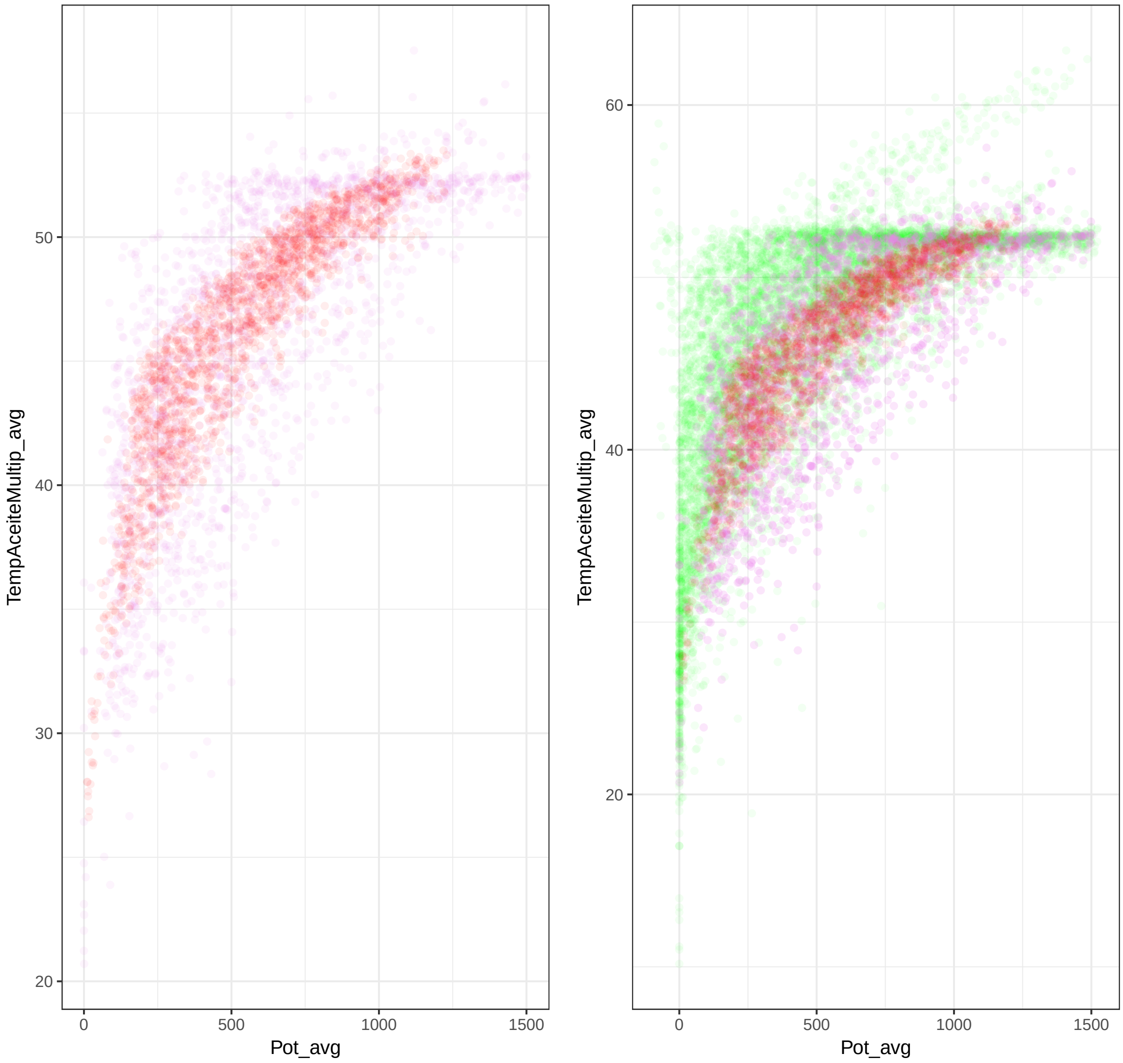


Figura 4.5: Muestra de la aplicación de RBM sobre los resultados de SMOTE

A continuación se muestran los resultados en el gráfico 4.6 de dos variables seleccionadas del Dataset de turbinas de ACCIONA, mostrando los puntos originales y los generados por la secuencia de los tres métodos. Como se puede observar, los datos generados de forma sintética se asemejan a los registros de la clase minoritaria, añadiendo peso a la sección central de la curva, de manera que se encuentran lo suficientemente separados de la región verde de la subfigura 4.6(b).



(a) Fallos originales (violeta), generados (rojo).

(b) Sin fallos (verde), fallos originales (violeta) y los generados (rojo).

Figura 4.6: Resultado aplicación TOMEK+SMOTE+RBM representado en base a dos variables. (a) se muestran solamente los casos de registros con fallo. (b) se muestran todos los registros.

4.2. Reducción de dimensiones: selección/transformación de variables

4.2.1. Selección de variables

Una vez se tienen los datos limpios, se procede a separar las variables que aportan información de las que no. Para ello, se han utilizado diez métodos de selección de variables identificados en el estado del arte, como apropiados para los datos que se están tratando. En la publicación de congreso copiada en la Sección 3.3, se puede observar el resultado de la aplicación de cuatro de estos métodos y el impacto que generan en los resultados. En la publicación en revisión de la Sección 3.4 se muestran siete de estos métodos en comparación a una selección quasi-óptima y el resultado de evaluación utilizando un clasificador kNN.

La metodología detallada a continuación, es una ampliación de los métodos tratados en la publicación con mejoras añadidas a posteriori. En ella se eliminan las variables con una correlación superior al 98 % en los resultados de selección obtenidos, para cada uno de los métodos por separado. Además, como resultado final se efectúa una fusión de las puntuaciones generadas por los distintos algoritmos con el objetivo de extraer los predictores, que obtengan valores más altos en los distintos métodos, siendo de esta manera más sólidos.

En la Tabla 4.1, se puede encontrar una descripción general de los métodos de selección de variables implementados en este trabajo. Todos los métodos se ejecutan sobre el mismo conjunto de datos y etiquetas, generando una puntuación. Una vez ejecutado cada uno de los métodos, se normalizan las puntuaciones en un rango de 0-1, siendo 1 la mayor puntuación. Se computa la media y desviación estándar de cada variable, así como la puntuación del percentil a la que se quiere cortar *p-corte* (variables por encima del percentil que se quieren seleccionar). Las variables en las que su desviación estándar supera 0.5, son descartadas de la selección, el resto de variables se filtran por la puntuación *p-corte*, y son las que quedan para realizar el modelo. De esta forma, se consigue un acuerdo entre los distintos métodos de selección, obteniendo variables en las que su puntuación presenta un nivel alto en los distintos métodos, pudiendo ser indicativo de ser un buen predictor.

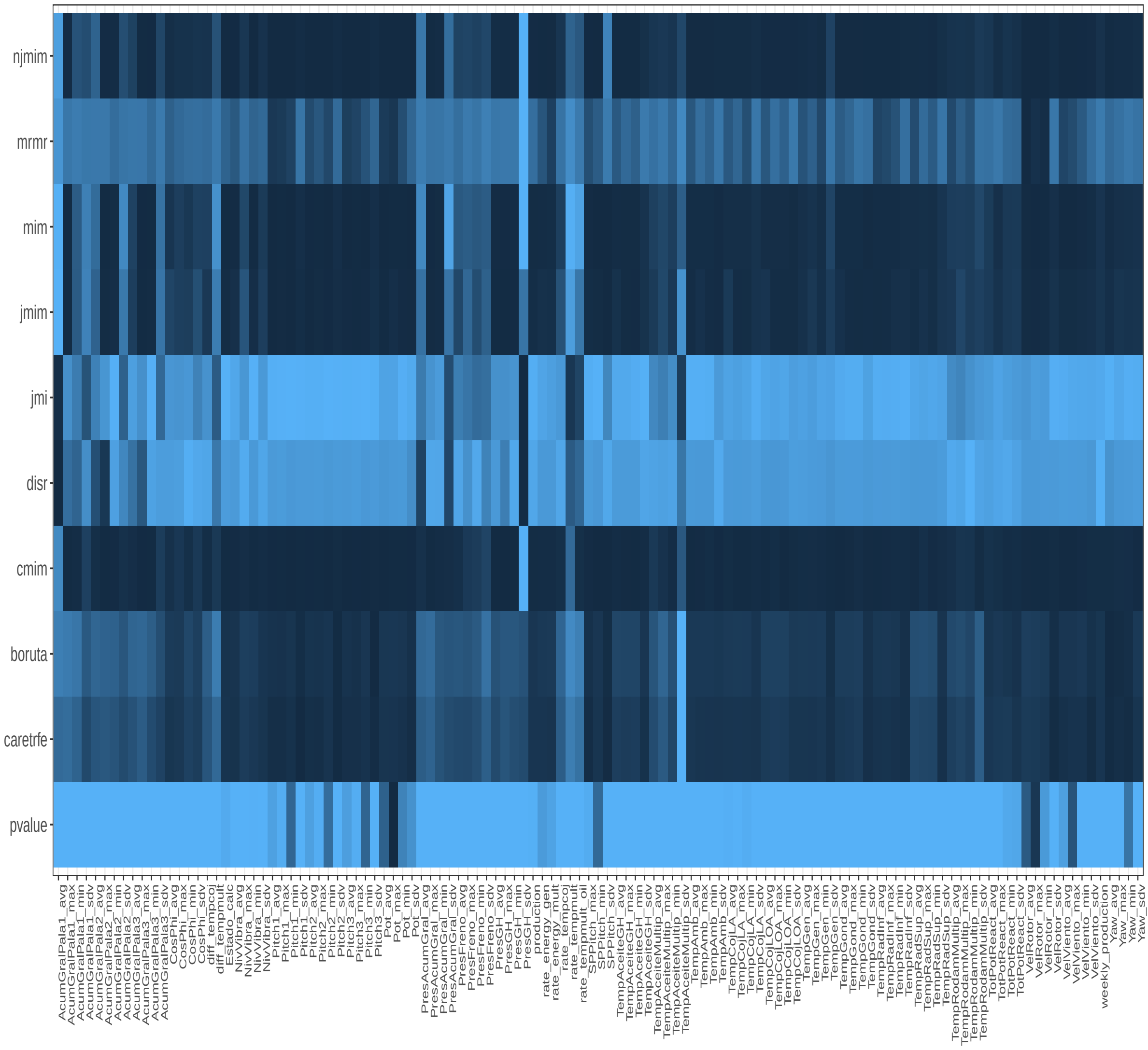


Figura 4.7: Heatmap resultado de los 10 métodos de selección de variables, en color más claro las que tienen mayor importancia para cada método.

Tabla 4.1: Algoritmos de selección de variables utilizados en los experimentos realizados.

Nombre corto	*Nombre completo*	*Autor*
pvalue	T-test de valores medios por clase (0.05<p-value)	Jaeger et al.
boruta	Boruta	Kursa et al.
rfe	Random forest recursive Feature Elimination	Granitto et al.
jmi	Joint Mutual Information	Yang and Moody
jmim	Joint Mutual Information Maximisation	Bennasar et al.
njmim	Normalised Joint Mutual Information Maximisation	Bennasar et al.
mrmr	Min-Redundancy Max-Relevance	Peng et al.
disr	Double Input Symmetrical Relevance	Meyer and Bontempi
cmim	Conditional Mutual Info Maximisation	François Fleuret
mim	Mutual Information Maximization	Lewis

4.2.2. Transformación: PCA

Con el objetivo de hacer una composición de variables, se ha explorado la correlación y comportamiento entre variables seleccionadas de la sección anterior mediante el PCA. En la Figura 4.8 se puede observar un caso, donde se han podido crear variables como *rate_tempmult* en las que se dividía la potencia generada (Pot_avg) entre la temperatura del aceite de la multiplicadora (TempAceite-Mult_avg) y se corregía por la temperatura ambiente, (TemAmb_avg) al encontrar que estas variables aportaban bastante información entre sí, ya que hay suficiente separación según el mapa *biplot* sobre las componentes PC1 y PC2. Estas variables representan otras que tienen una alta correlación, porque apuntan a la misma dirección y prácticamente superpuestas, como por ejemplo VelViento_avg y Pot_avg. Este criterio de selección y composición se elabora con el apoyo de una persona experta, de manera que se le muestra el mapa de las variables proyectadas sobre las componentes principales. Esta persona debe seleccionar, dependiendo el sistema a analizar, las variables que estén lo suficientemente separadas entre sí, mediante el ángulo que forman sus vectores proyectados y la relación con el sistema a predecir del fallo. Es por esto, que la aplicación de este método, requiere de un contraste con un modelo físico de la turbina eólica.

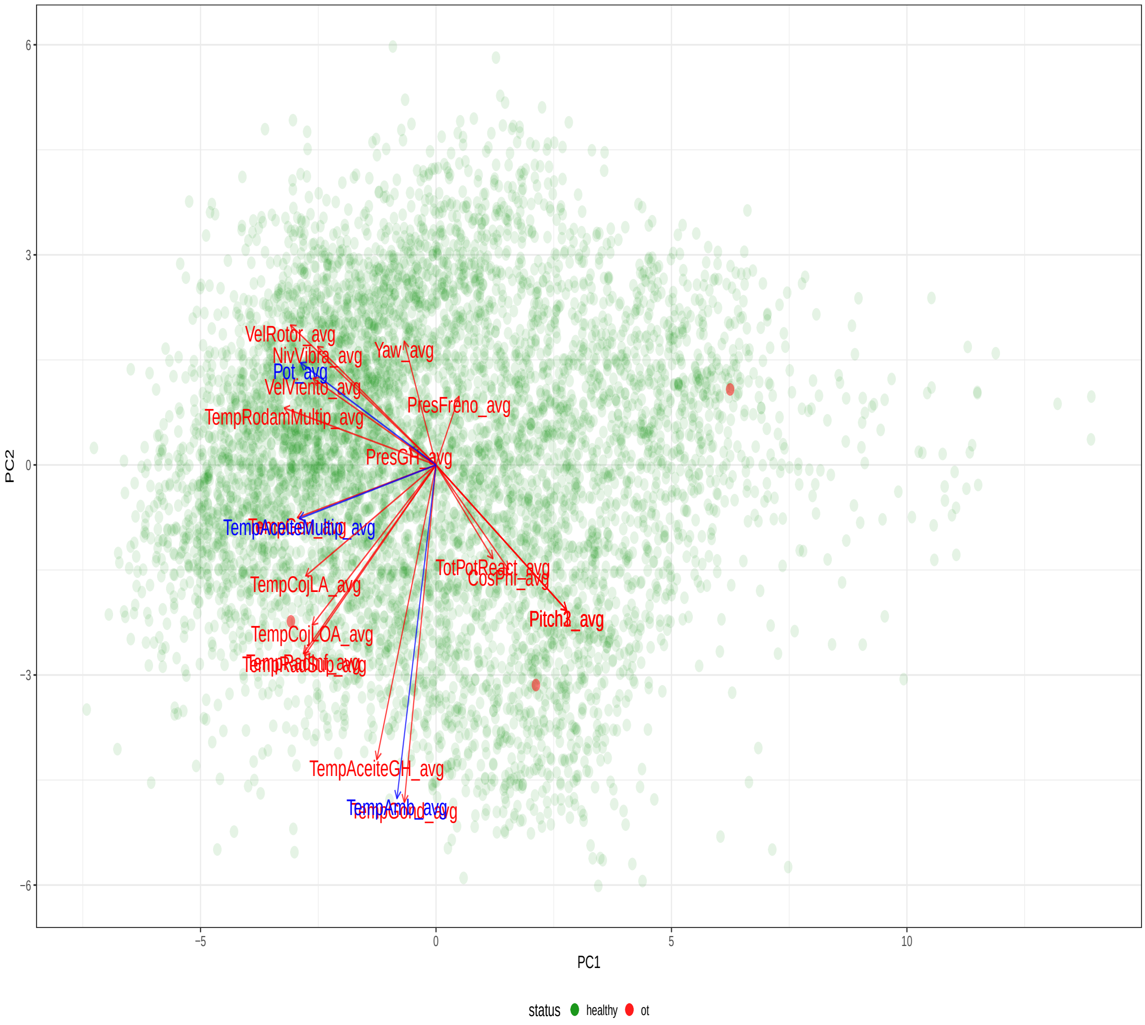


Figura 4.8: PCA con las componentes principales PC1 y PC2 para el parque Moncayuelo. En azul las variables seleccionadas (Pot_avg, TemAmb_avg, TempAceite-Mult_avg) para construir una nueva variable artificial.

4.3. Elaboración del modelo

En esta sección se describe tras haber filtrado, tratado y seleccionado los datos, que tipos de modelo se han llevado a cabo, tanto supervisado como no supervisado. En la sección, también se pueden encontrar resultados de otros métodos que se han implementado y validado al final de la tesis, pero en los que por falta de tiempo no se ha podido terminar la publicación.

Durante el desarrollo de la tesis se ha trabajado con modelos de clasificación, pero también se han utilizado modelos de regresión basados en PLS y métodos no supervisados como el SOM. Por eso las siguientes líneas han sido separadas entre estos dos grupos.

4.3.1. Supervisado

Para poder realizar un modelo de clasificación que tenga un resultado aplicable industrialmente, se han tenido que tener unas consideraciones y requerimientos que se han ido encontrado a medida que se desarrollaba la tesis:

No basta con clasificación, se requiere predicción: En la práctica no tiene sentido identificar que un registro pertenece a un estado de fallo, ya que el fallo ocurre en ese mismo instante. Es necesario que el fallo sea avisado con antelación, por lo tanto, a la hora de marcar los registros de fallo en el Dataset , hay que hacer un traslado de la etiqueta de fallo a un cierto tiempo hacia atrás como se muestra en la Figura 4.9.

El orden temporal importa: Esta consideración requiere que cuando se divide el Dataset entre *Train* y *Test*, no se puede tomar el Dataset completo y hacer un sampleo *random* sobre este. El motivo es que en el Dataset de *Train* se pueden introducir muestras del "futuro"sampleadas, después de una reparación o cambio, siendo esto, lo que queremos predecir. En distintos trabajos [41, 59] del estado del arte, utilizando un clasificador/predictor de fallos se comete este error, consiguiendo buenos resultados, pero que difícilmente son posibles en un entorno real, puesto a que no se disponen de datos del futuro sobre los que se puedan samplear, a la hora de hacer la predicción.

El ajuste del modelo no solo va dado por el mejor punto de la ROC: En sistemas de *Machine Learning* en aplicaciones industriales, tienen que generar confianza, es decir, generar la mínima cantidad de falsos positivos como sea posible, porque en el caso de las turbinas eólicas un aviso de falso positivo, implica parar la turbina para que pueda entrar un operador de parque. Este operador, en ocasiones tiene que desplazarse a zonas donde remotas o de difícil acceso, gastando bastante recursos. Por otro lado, se desea que la *precisión* sea alta para que el aviso de fallo dado por el modelo, sea lo suficientemente confiable para los usuarios del sistema.

Las clases de buen estado/fallo no están separadas de forma clara: Las turbinas con que las que se ha trabajado en esta tesis, no necesariamente están en el mejor estado, es decir, una turbina a la que no le han cambiado un sistema como la multiplicadora, no significa que esta se encuentre en buenas condiciones aunque se marcase como tal, o que otros sistemas que la componen no presenten defectos. Esto hace que los registros de una turbina en buen estado, no se separen de forma significativa de una turbina con registros de fallo. Dado esto, se han encontrado varios casos donde una turbina a la que se le ha cambiado la multiplicadora funciona peor (en términos de temperatura y vibraciones), que una a la que nunca se la han cambiado. Cuando se hace el cambio del sistema se considera como buen estado, ya que se asume que se ha reparado el problema, cosa que no se cumple en todos los casos o en distintas ocasiones, como se puede apreciar en la Figura 4.10.

Las turbinas eólicas están compuestas de sistemas: Desde el punto de vista aplicado, no tiene utilidad decir que la turbina va a fallar, sino que lo adecuado es indicar el qué va a fallar, de manera que se crea un modelo para cada sistema, como por ejemplo un modelo para la multiplicadora y otro para el generador. Esto conlleva a un problema, el cual provoca que el modelo genere muchos falsos positivos. Un motivo es la introducción en el conjunto de entrenamiento, de una turbina con registros marcados como "buenos"hablando de fallos de multiplicadora, pero sin embargo tenga el generador deteriorado de forma notoria, haciendo que salten falsos positivos porque este sistema está altamente acoplado físicamente con el otro, traspasando temperaturas y vibraciones. Otro motivo es debido a que cuando se repara ese sistema, introduce en diversas ocasiones escalones importantes en variables seleccionadas por el método de selección de variables como se puede ver en la Figura 4.11.

Se hacen inspecciones y mantenimiento de forma regular: Cada vez que se hace una inspección o mantenimiento preventivo (cambiar aceites, revisar sensores...), se introduce unos modos de funcionamiento y valores que no son etiquetados como fallos. Estos datos tomados para generar una predicción, introducen un ruido que el modelo interpreta erróneamente como que va a haber fallo N días después, cuando es falso. En algunos Datasets de varios parques es posible eliminarlos, porque se dispone de esta información. Sin embargo, en los que no se dispone, se deben utilizar técnicas para detectarlos mediante el método aplicado en la publicación del SOM (Sección 3.1).

Las variables de ambiente cambian la predicción: Cuando se hace predicción de fallos, se toma el viento que posiblemente vaya a haber en una vista futura de unos 15 días a 1 mes. Existe un problema y es que el viento es muy dinámico, por lo que cuando se hace la predicción de fallo utilizando datos de predicciones de viento altas en una turbina que pueda estar dañada, genere un positivo, sin embargo, puede ocurrir que después no lo sea, porque en las fechas previstas para la predicción, la presencia del viento ha sido débil. Lo mismo puede suceder en forma opuesta, cuando se hace la predicción se estima con un viento bajo y el modelo no genera positivos.

Teniendo en cuenta todos los puntos indicados anteriormente, los modelos de clasificación han presentado un comportamiento difícil de ajustar con los datos que se han trabajado, ya que ni la turbina considerada como buena, es tan buena ni la dañada ha generado el patrón esperado en sus variables antes del fallo. Es por ello, que se tomó la decisión de hacer predicción con métodos más complejos, que permitan capturar patrones más complejos en los datos, pero más difíciles de ajustar debido a la gran cantidad de hiperparámetros.

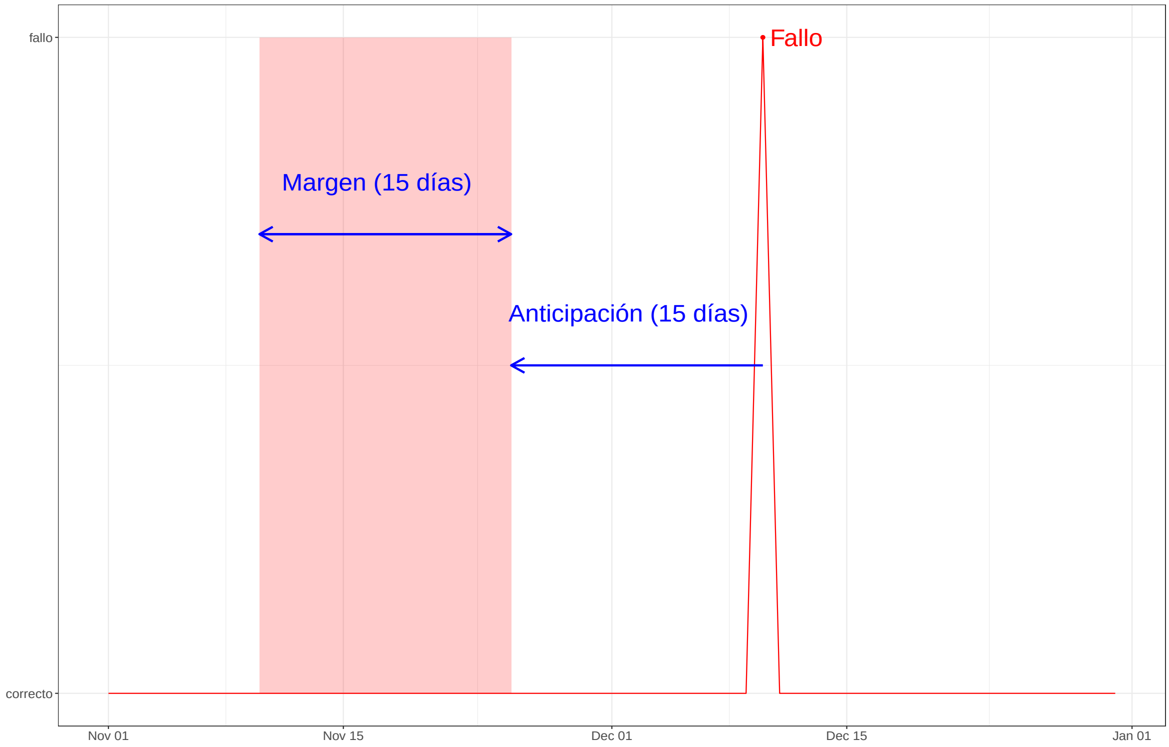
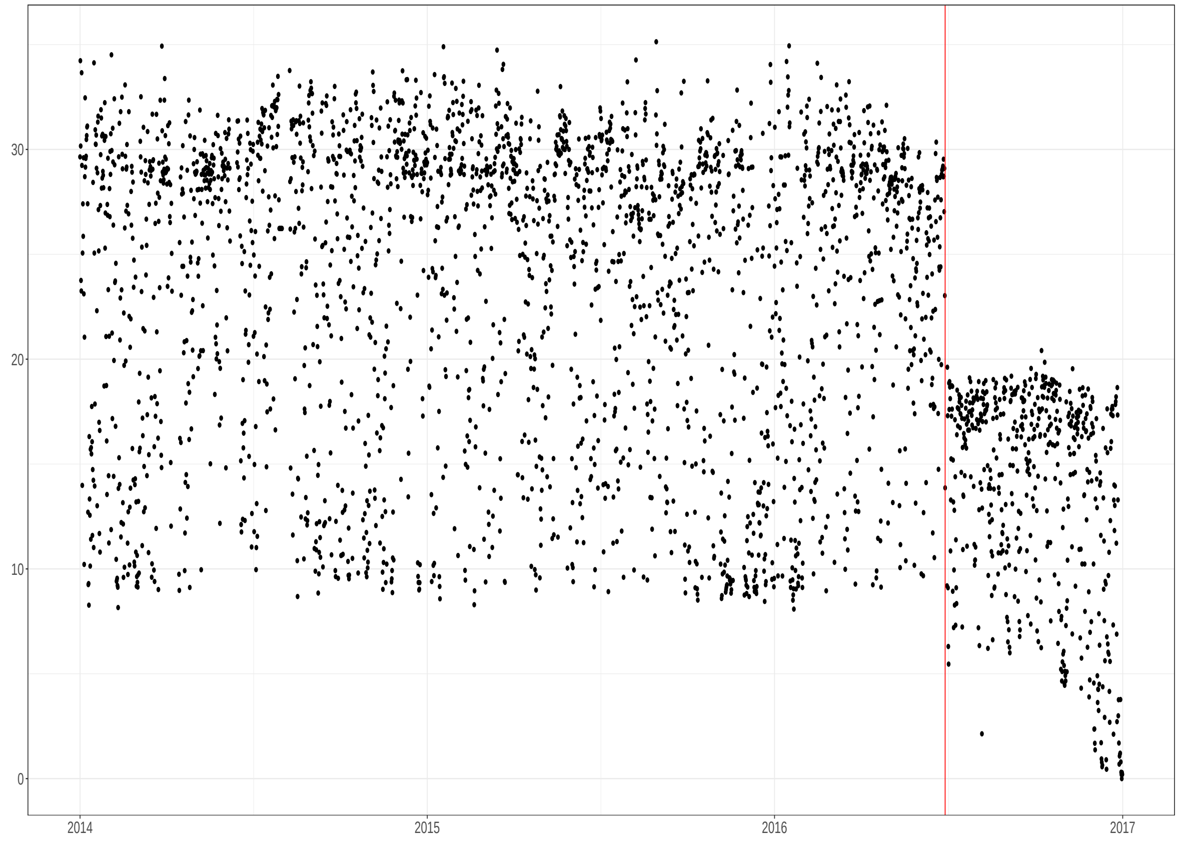
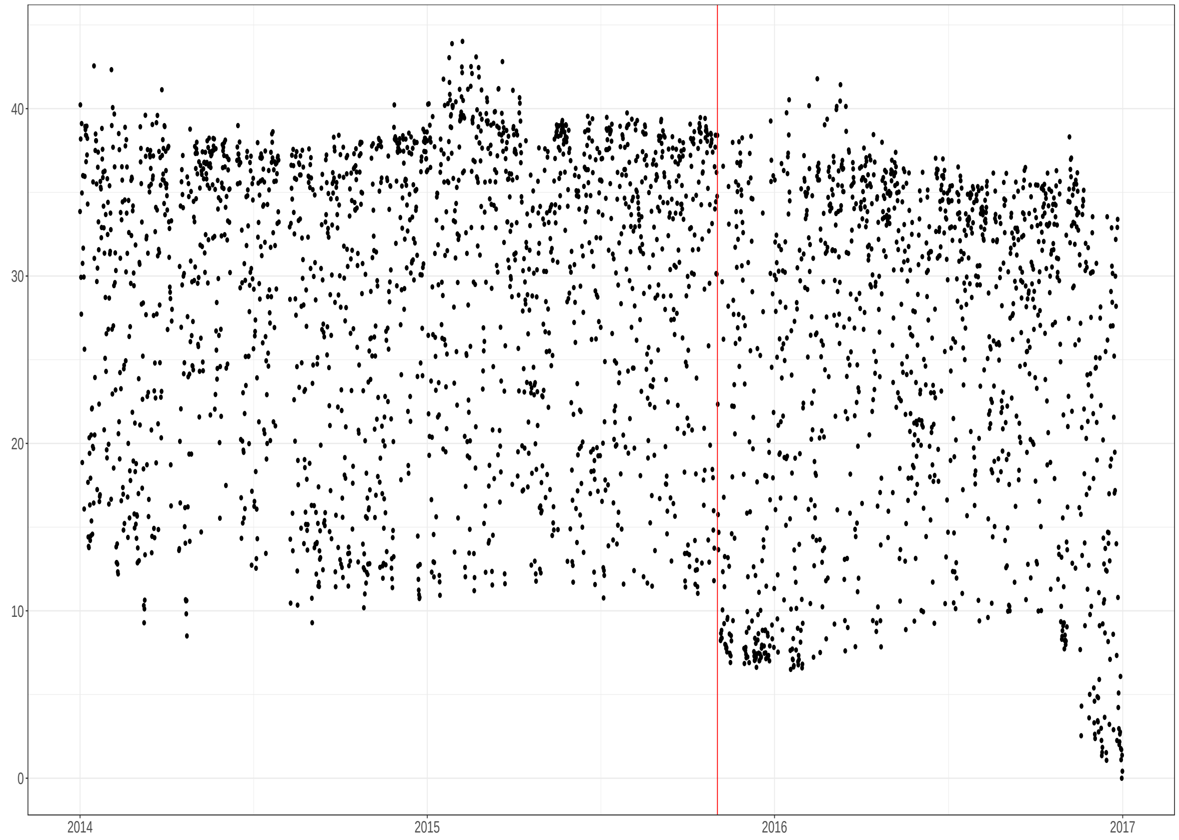


Figura 4.9: Creación de la prealarma dado un día de fallo *t*, para hacer predicción. Se marcan los registros desde *t*-30 (15 anticipación + 15 margen) a *t*-15 como positivo.



(a) Turbina 142, se aprecia un cambio destacable



(b) Turbina 136, no se aprecia un salto destacable

Figura 4.10: Variable *diff_tempmult* de dos turbinas del parque Moncayuelo. Se marca con una línea vertical roja el cambio de la multiplicadora.

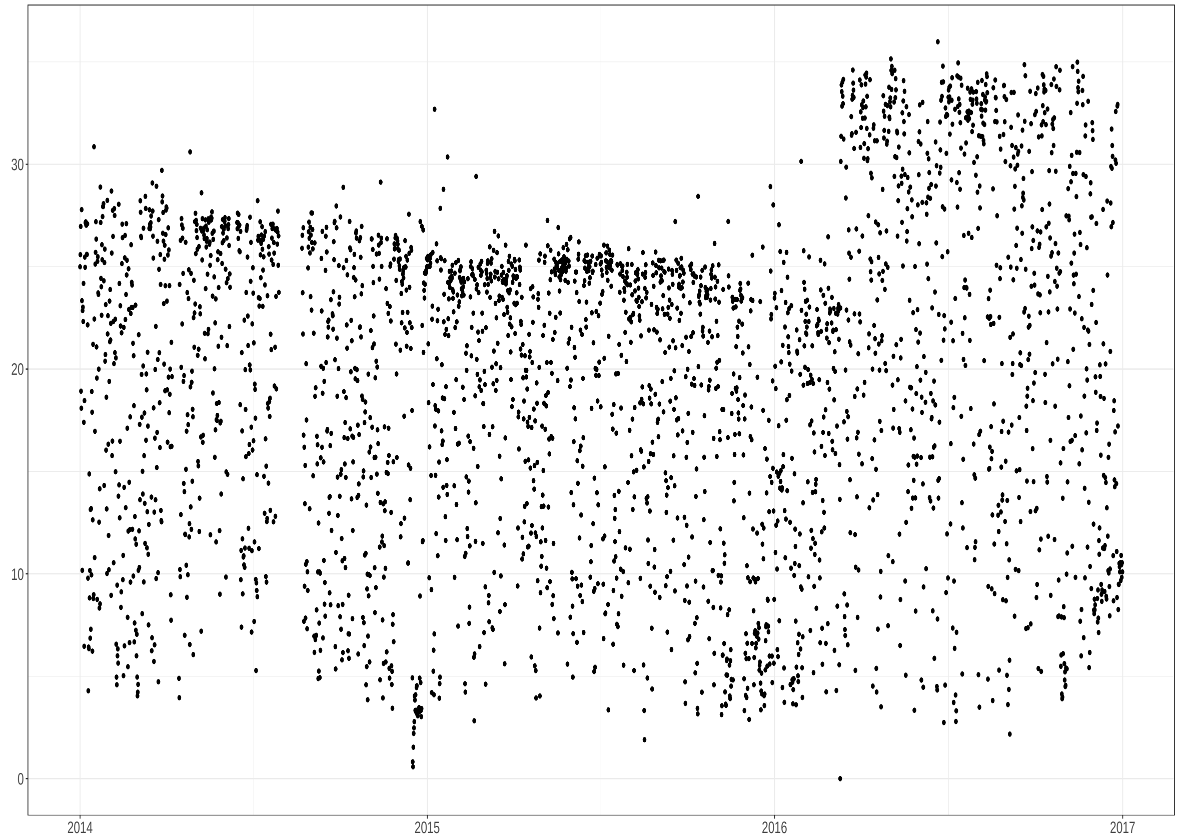
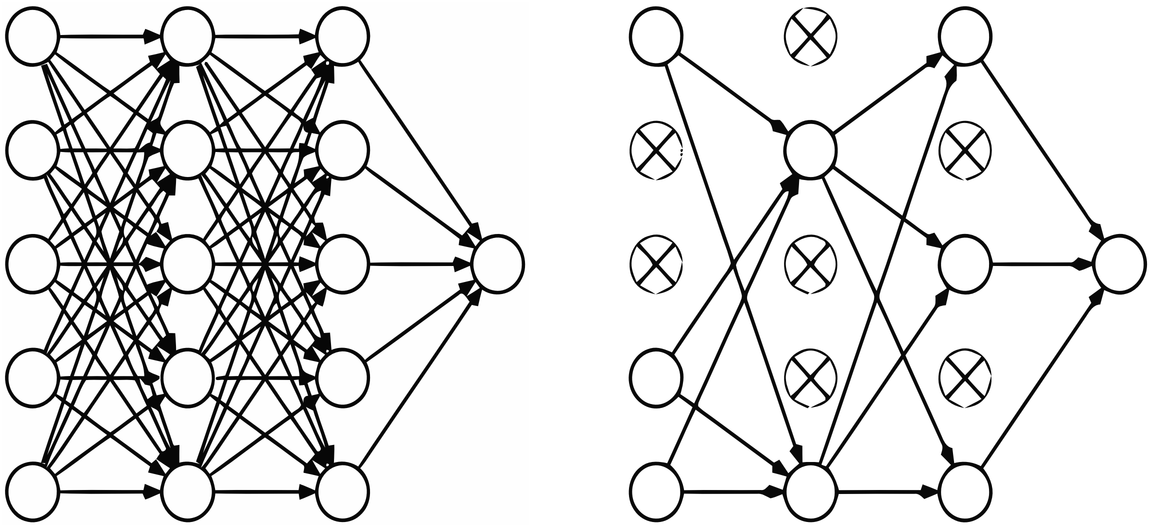


Figura 4.11: Variable *diff_tempmult* para turbina 144 del parque Moncayuelo. Sin cambios en la multiplicadora, salto apreciable.

Deep Learning: ANN

Se han trabajado en distintos modelos basados en redes ANN multicapa [37], poniendo una capa de regularización (*dropout* [52]) entre cada capa de la red como se ve en la Figura 4.12.

Estos modelos tienen múltiples hiperparámetros descritos en la documentación de la librería utilizada (Keras [11] + Tensorflow [1]). Siendo configurable desde el número de capas, la función de activación, tipo de inicialización, función de ajuste en entrenamiento y cientos de parámetros más [25].



(a) Sin dropout aplicado durante un *Epoch*

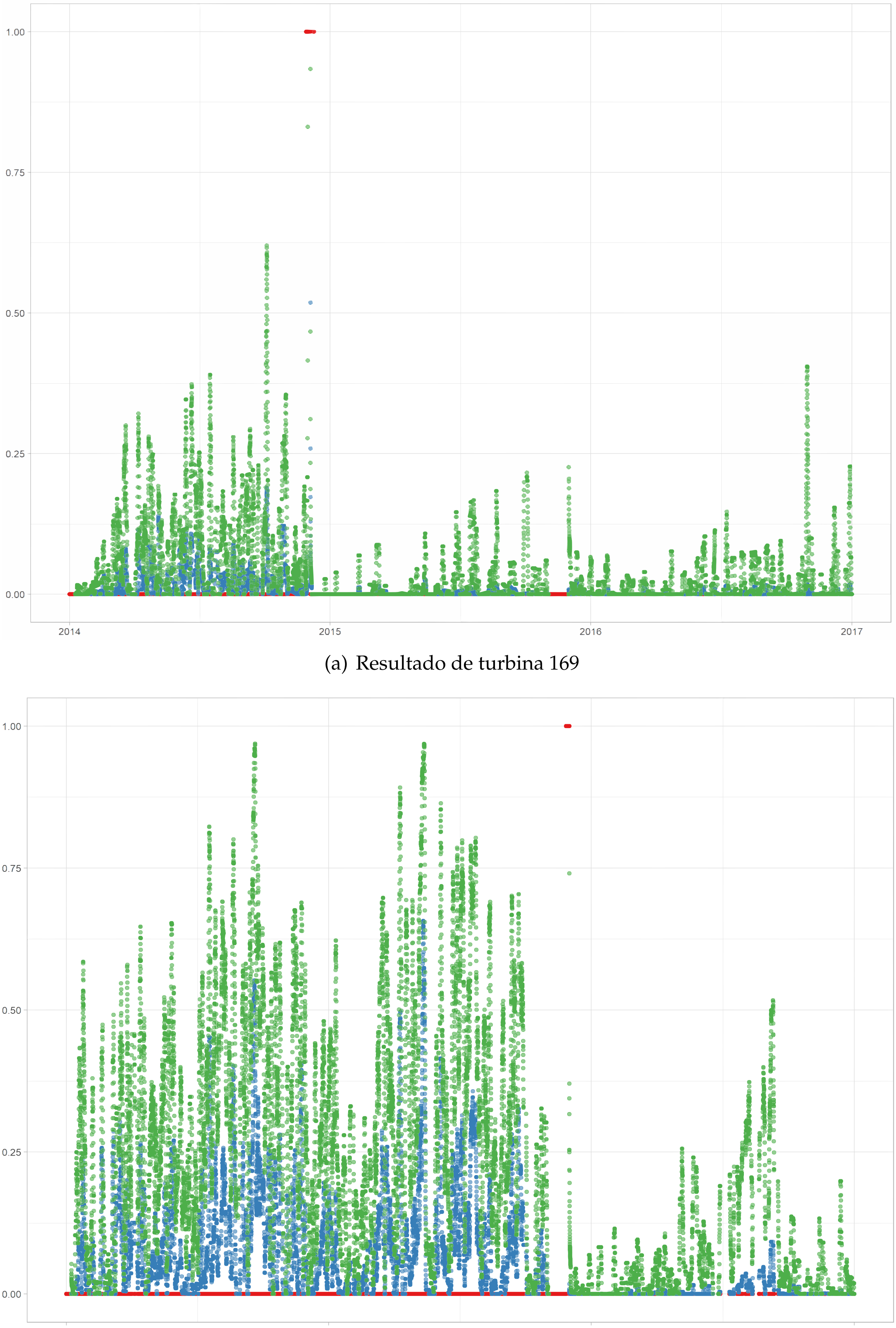
(b) Cuando se aplica dropout

Figura 4.12: Arquitectura red ANN deep cuando se le aplica *dropout* en cada *Epoch*.

Se han establecido distintas configuraciones según el tipo de sistema y parque a predecir, ya que van ligadas al Dataset y al filtrado que se les hace. En las siguientes líneas, se comenta el resultado de un modelo que ha sido configurado con 6 capas de 50-30-20-10-8-1 neuronas cada una, con una función de activación tipo Sigmoide y con una capa intercalada, entre capas, de regularización (*dropout*) establecida al 50 % . Además, el modelo se ha configurado para que la función de optimización (basada en *ADAM* [39]) dé un peso de 98 veces superior a la clase minoritaria "1"(indica fallo) respecto a la clase "0", buen estado, durante el entrenamiento. Se ha repetido para otro modelo pero con un peso de 5. En los resultados de las Figuras 4.13 y 4.14 se puede ver los puntos verdes representando al modelo con peso de 98 y a los puntos azules con un peso de 5. Se han obtenido casos, donde el modelo ha comenzado a dar positivo (superar una probabilidad de 0.5) con suficiente antelación como en la Figura 4.13, y por otro lado, casos donde el modelo solo hacía que dar falsos positivos como en la Figura 4.14. En la Tabla 4.2 se puede ver de forma resumida las métricas que indican la potencia de predicción a turbinas nunca vistas, teniendo en cuenta lo indicado al principio de esta sección. Se puede observar que en el modelo que tiene mayor sensibilidad (98) a los casos de fallo, se consigue mayor tasa de positivos reales (TP), pero la tasa de falsos positivos (FP) se dispara, esto se traduce en que el *Accuracy* (ACC) del modelo es inferior, aunque se gana en sensibilidad (SEN). Sin embargo, el Kappa [13] es más bajo porque la precisión de los casos de fallo es muy baja. En referencia al modelo que tiene menor sensibilidad (5), se observa una mayor *Accuracy* por arriesgarse menos a dar casos como positivos, la tasa de positivos reales cae a la mitad, por lo que la sensibilidad del modelo ante los casos de fallo es de un 47 %, pero con una precisión del 54 % versus al 8 % del modelo más sensible (98). Esto hace que el coeficiente Kappa sea mayor en este modelo, por lo que trasladaría mayor fiabilidad al usuario. Como se puede observar, el ajuste del modelo para un caso u otro es bastante crítico, debido a la dificultad del Dataset al que nos enfrentamos para fallos de generador, en este caso del parque Izco. En ambos casos la tasa de especificidad (SPEC) va a la par del *Accuracy*, indicado que sigue detectando una buena cantidad de casos de "no fallo". La tasa de verdaderos negativos (TN) no tiene cambios bruscos entre casos.

Tabla 4.2: Métricas que resumen el resultado de media, del modelo D-ANN en las 50 turbinas del parque de Izco con una anticipación de 15 días.

Sensibilidad del modelo	**TP**	**FP**	**TN**	**FN**	**ACC**	**PRES**	**SEN**	**SPEC**	**KAPPA**
98	137.902	1598.39	15909.098	18.024	90.85	0.08	0.884	0.91	0.132
5	72.585	60.439	17447.049	83.341	99.186	0.5456	0.4655	0.9965	0.498



(a) Resultado de turbina 169

(b) Resultado de turbina 173

Figura 4.13: Resultado de dos turbinas del parque Izco. Eje vertical (probabilidad de fallo), eje horizontal (tiempo). Punto rojo indica fallo, azul y verde resultados de clasificador.

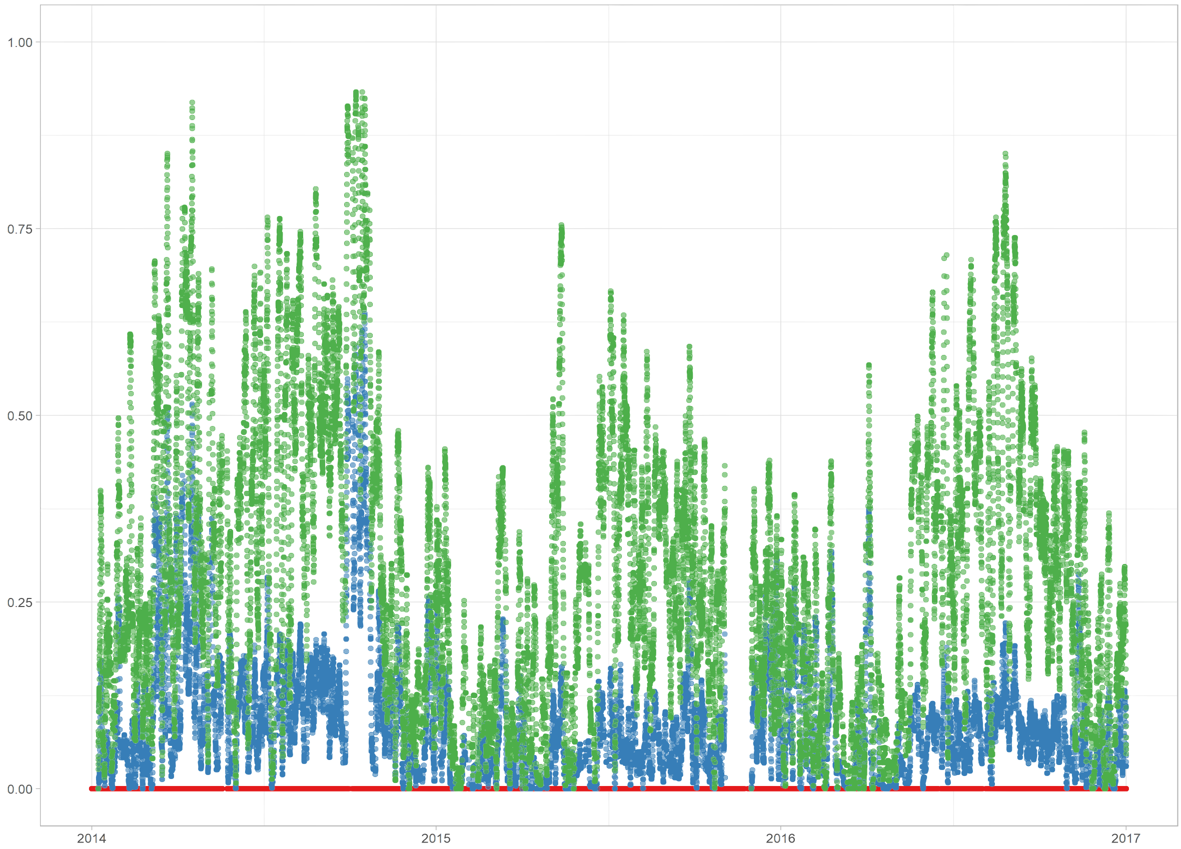


Figura 4.14: Predicción para turbina 191 parque Izco. Eje vertical muestra la probabilidad de fallo, horizontal el tiempo. Puntos azul y verde resultados de clasificador, siendo verde más sensible.

Deep Learning: LSTM

Tras haber trabajado durante varias iteraciones con redes ANN multicapa, se pudo observar que los resultados a partir de los Datasets con distinta cantidad y tipo de ruido, provoca que el resultado de una red ANN se comporte de forma sensible teniendo cambios grandes en su probabilidad de un registro temporal a otro que le sigue. Por ese motivo, se ha buscado algoritmos de clasificación que no solo capture las relaciones entre las variables de entrada, sino además su evolución temporal. Dentro de las técnicas novedosas de clasificación basadas en *Deep Learning*, se encuentran las redes LSTM (Long Short Term Memory). Estas redes tienen como característica tener interconectadas una cierta cantidad de neuronas en el tiempo capturando la relación de las variables de entrada además de su evolución temporal para una cierta ventana, formando un cubo 3D de conexionado entre neuronas como se puede observar en la Figura 4.16. Estas redes se han empleado con éxito para series temporales sobre una única variable o más concretamente en el campo de esta tesis, para hacer predicción de potencia a generar por una turbina [61]. No se ha encontrado mucha información sobre su uso en predicción de fallos de turbinas eólicas, por lo que se ha efectuado una primera aproximación que se describen en las siguientes líneas. Puesto que la arquitectura de las redes LSTM pueden configurase de distinta forma como se puede ver en la Figura 4.17, en este caso se ha optado por diseñar una del tipo *many to one*. Este tipo de arquitectura implica que para cada registro original que se tenga en el Dataset con el estado etiquetado como fallo "1.º no fallo "0", se deben concatenar **N** registros (la ventana a observar) hacia atrás que determinen el actual estado. Por esa razón, los datos han de tener una transformación previa antes de ejecutar este método 4.15, generando una matriz de tres dimensiones donde la primera dimensión son el número de registros originales, la segunda dimensión se encuentra la ventana de los **N** registros concatenados hacia atrás desde el registro original actual y finalmente la tercera dimensión contiene las distintas variables.

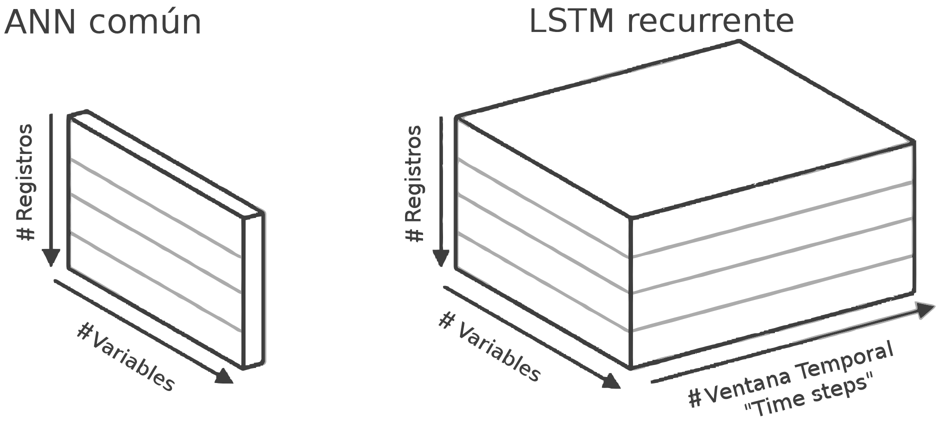


Figura 4.15: Comparación del formato común de tabla utilizado en ANN frente a la estructura que requiere una red LSTM recurrente.

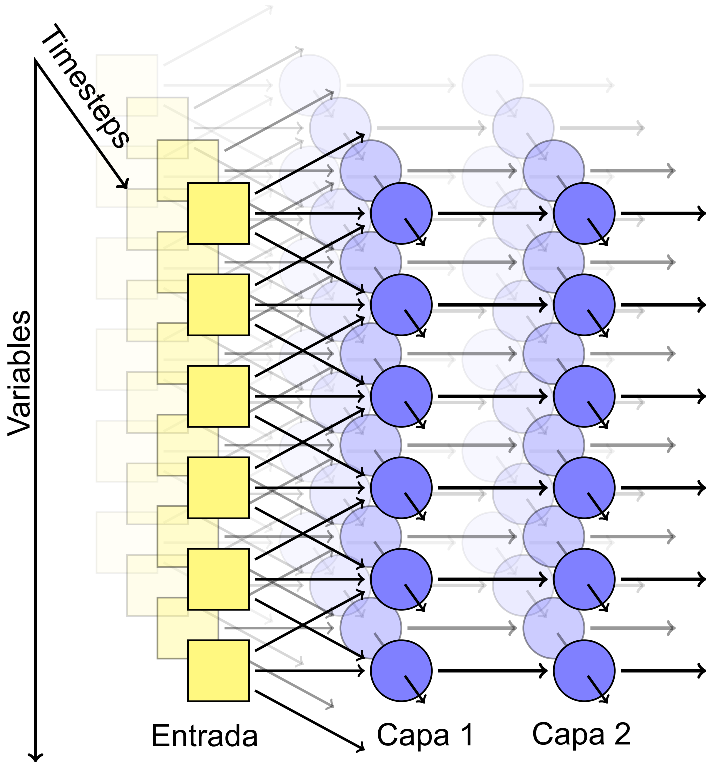


Figura 4.16: Estructura equivalente de una red recurrente, al ser desenrollada.

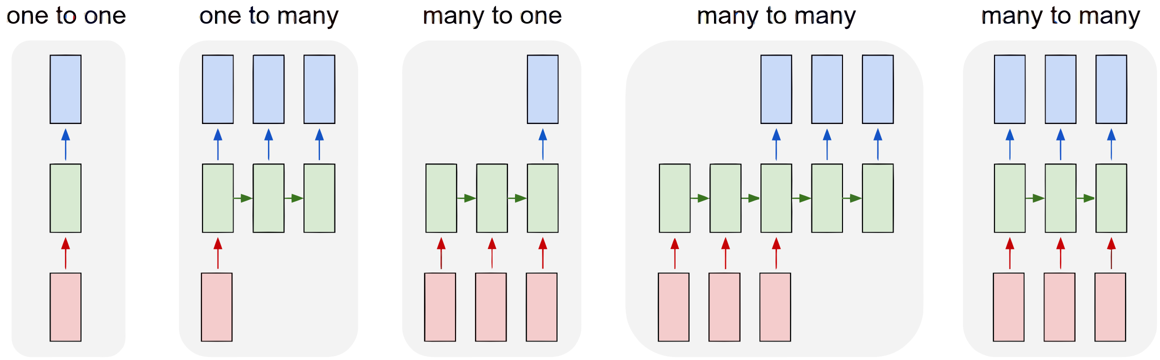
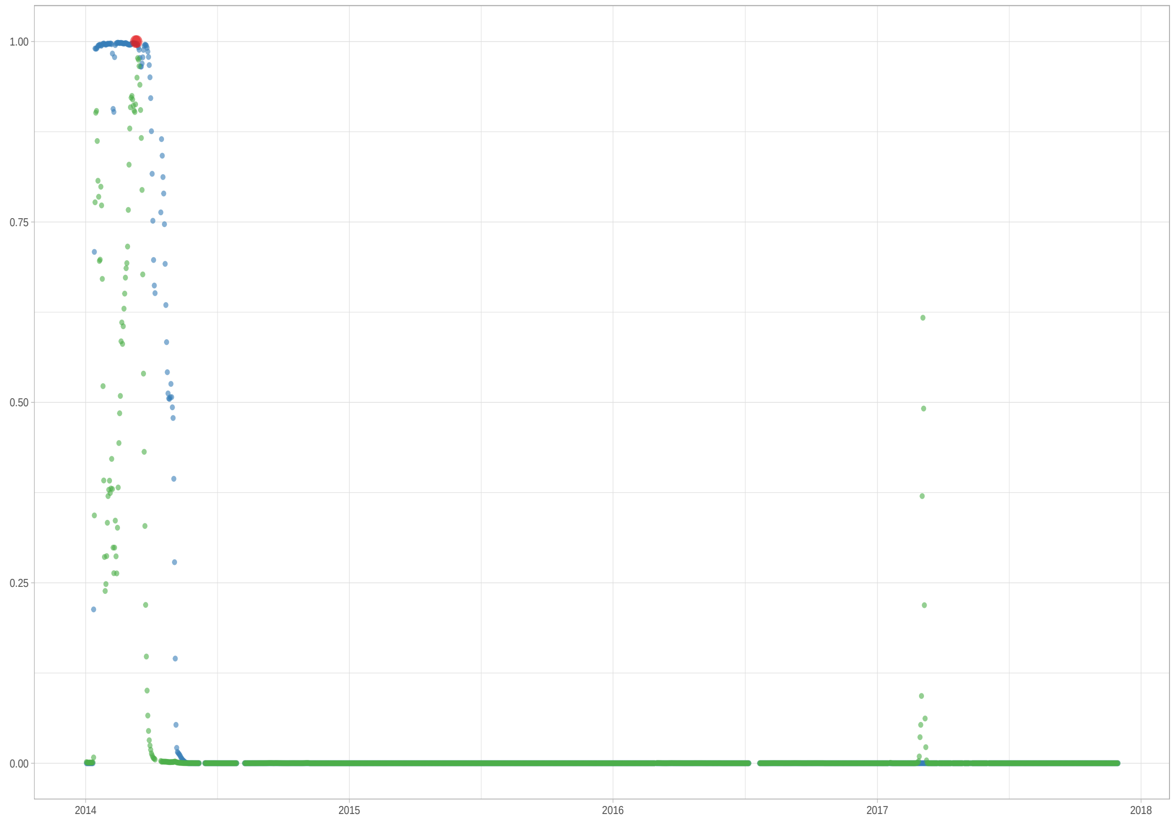


Figura 4.17: Tipos de Arquitecturas de redes LSTM. Se ha utilizado *many to one* ya que se quiere capturar la evolución antes de un fallo.

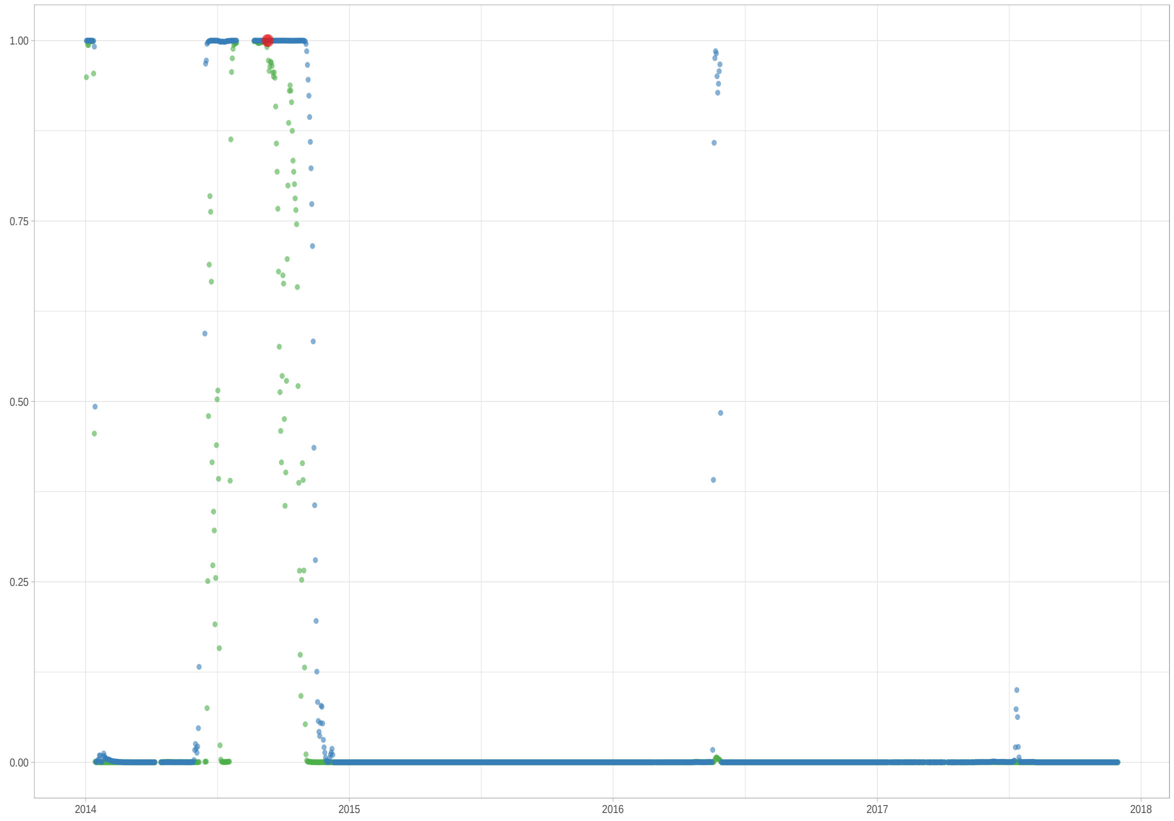
Esta red también tiene los mismos parámetros de configuración que la red ANN gracias a utilizar el framework Keras, con una particularidad, es que además hay que definir un *dropout* entre capas temporales para que se aplique una regularización también a la evolución temporal. Con esto se tiene mayor capacidad de generalización y sea disminuye el *overfit*.

A continuación se muestran los resultados de esta técnica con una configuración de 4 capas con número de neuronas por capa (10,10,10,1), la profundidad temporal, *timesteps*, es de 90 registros. La función de activación *tanh* y un *dropout* del 30 % entre las distintas capas incluyendo entre las temporales. La Figura 4.18 muestra el resultado de predicción que se produce con suficiente antelación, tiene una subida de probabilidad suave y estable a un valor de **1** antes del fallo (punto rojo) y bajada progresiva después de este. La serie de color verde tiene menor sensibilidad (un ratio de 1.5), mientras que la azul tiene mayor sensibilidad (3). Sin embargo, como se muestra en la Figura 4.19(a), existen varios casos con falsos positivos (FP), en ocasiones debido a mantenimiento efectuado a esa turbina o por saltos en las variables que fueron seleccionadas en la fase de *Feature Selection*. También existen casos de turbinas donde ha ocurrido un fallo, como en la Figura 4.19(b), donde ninguno de los dos modelos con distinta sensibilidad han detectado un patrón de fallo. Explorando las variables seleccionadas en este caso, se puede ver que no muestran un patrón de cambio antes y después del fallo (indicado con una barra vertical) como se ve en la Figura 4.20.

En la Tabla 4.3 se muestra un resumen de los resultados para fallos de multiplicadora en el parque de Moncayuelo compuesto por 32 turbinas. El modelo con menor sensibilidad (ratio de 5) obtiene mejor *Accuracy* pero tiene una sensibilidad del 54 % ante los casos de alarma, alcanzando una precisión de solo el 10 %. Sin embargo, el modelo con mayor sensibilidad (41), incrementa la tasa de positivos (TP) en más del doble, a la vez que incrementa la tasa de falsos positivos (FP), obteniendo un *Accuracy* menor pero aumentando de forma notable su sensibilidad (SEN) a un 86 % y ligeramente su precisión a un 13 %, generando un coeficiente kappa ligeramente mayor. Este parque es difícil de modelar, pues tiene una cierta antigüedad y el sistema a predecir (multiplicadora) presentaba desgaste en casi todas las turbinas, inclusive las utilizadas como buenas en el entrenamiento.

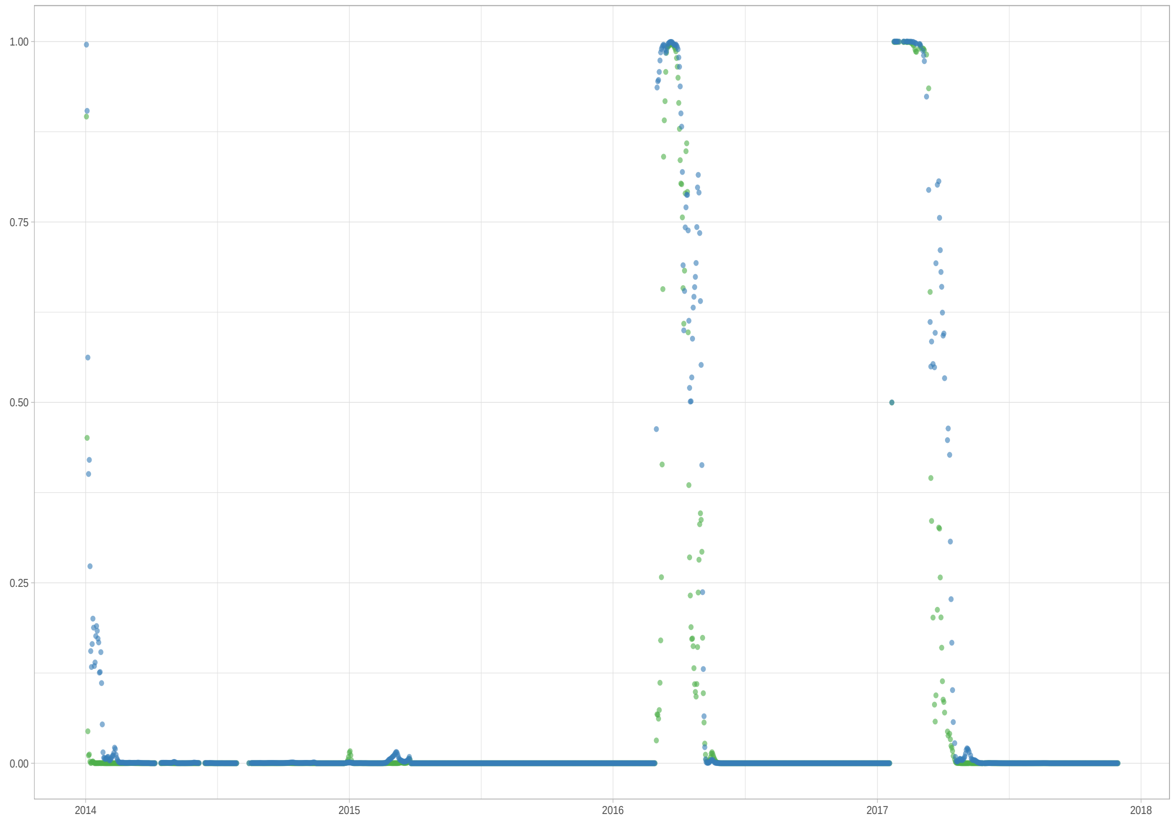


(a) Resultado de turbina 135

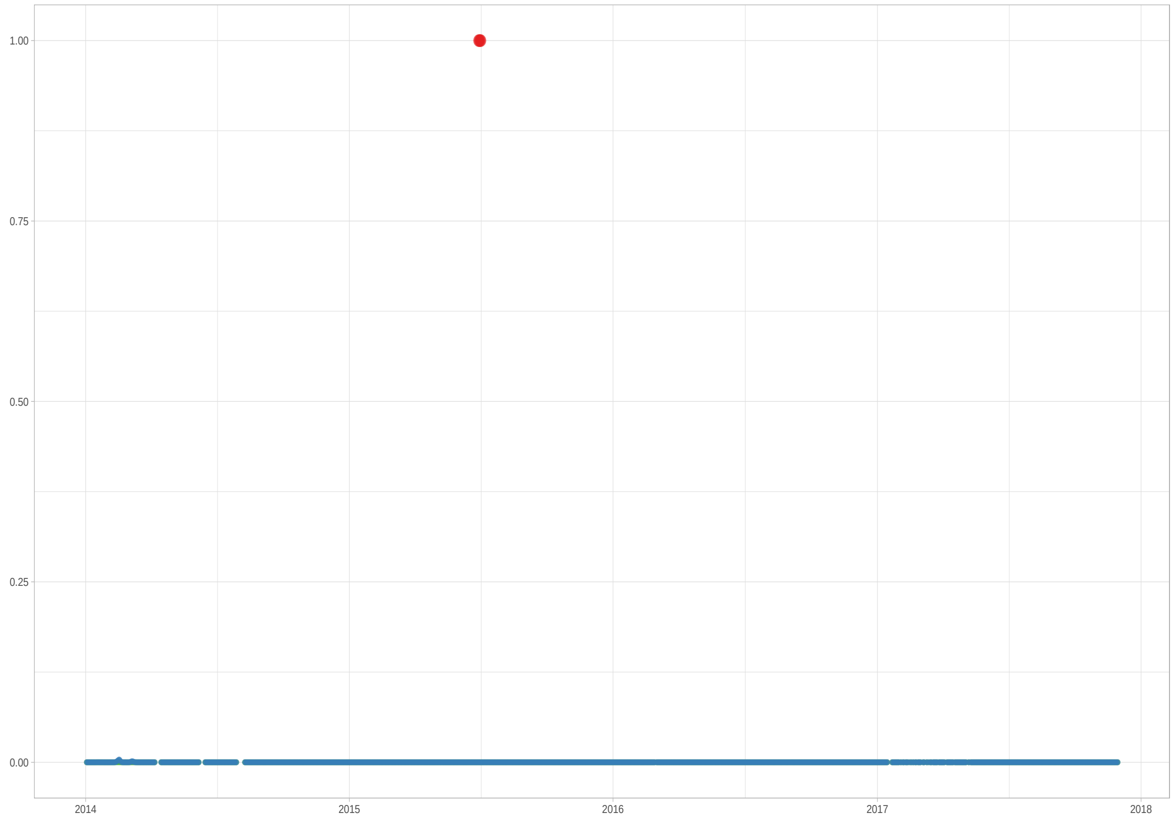


(b) Resultado de turbina 139

Figura 4.18: Clasificación en dos turbinas (parque Moncayuelo). Probabilidad de fallo en eje vertical, el horizontal es el tiempo. En rojo se indica fallo, verde y azul (+sensible) resultados del LSTM.



(a) Resultado de turbina 146



(b) Resultado de turbina 152

Figura 4.19: Clasificación en dos turbinas (parque Moncayuelo). Probabilidad de fallo en eje vertical, el horizontal es el tiempo. En rojo se indica fallo, verde y azul (+sensible) resultados del LSTM.

Tabla 4.3: Métricas que resumen el resultado de media, del modelo LSTM en las 32 turbinas del parque de Moncayuelo con una anticipación de 15 días.

Sensibilidad del modelo	**TP**	**FP**	**TN**	**FN**	**ACC**	**PRES**	**SEN**	**SPEC**	**KAPPA**
41	238.02	1602.1	19132.4	37.4	92.2	0.13	0.86	0.93	0.21
5	92.098	781.24	24139.463	78.3	96.57	0.105	0.54	0.97	0.167

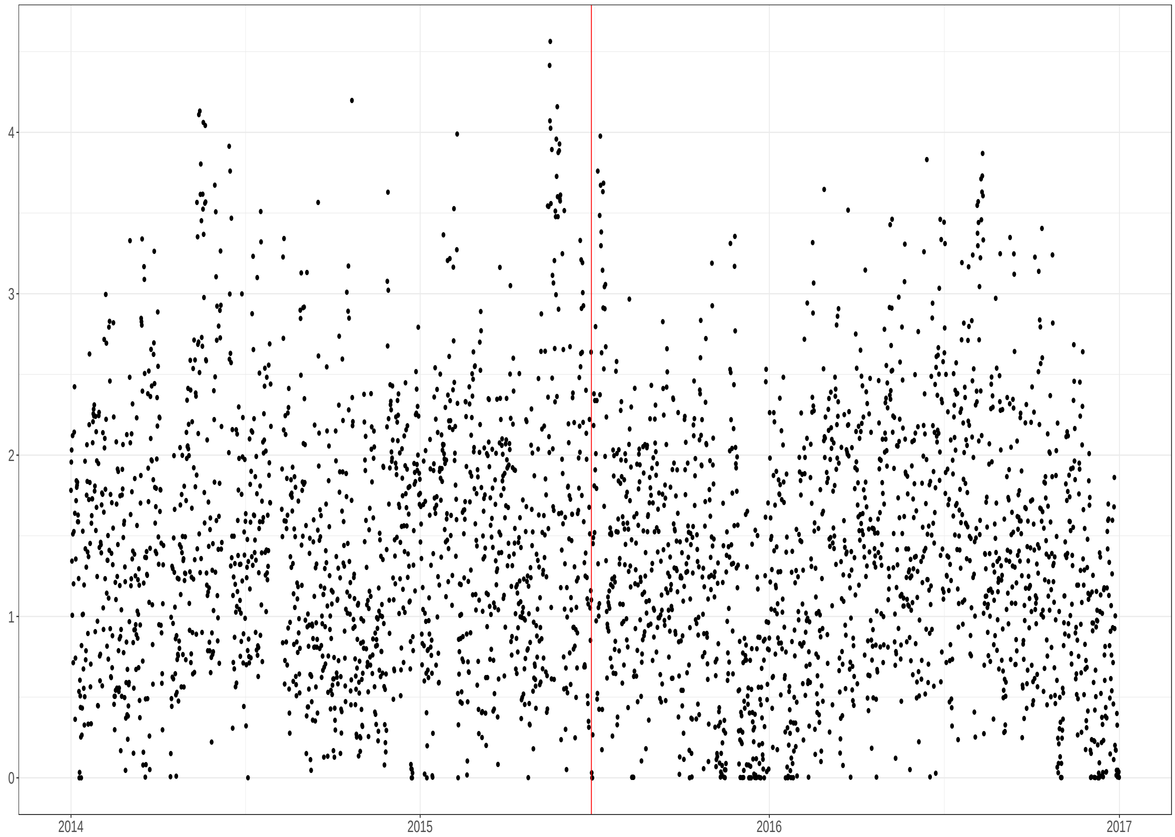


Figura 4.20: Variable *NivVibra_sdv* para turbina 152 del parque Moncayuelo. Cambio de multiplicadora.

4.3.2. No supervisado

Debido a los inconvenientes encontrados a la hora de hacer el etiquetado de los datos, paralelamente a los métodos supervisados, se han trabajado en métodos no supervisados, con el objetivo de extraer información extra que permitiera etiquetar de una forma alternativa los datos. En esta área del *Machine Learning*, se ha centrado esfuerzos en el método SOM, por los buenos resultados que se podían ver en el estado del arte en el campo de las turbinas eólicas, en cuanto a extracción de información. El enfoque aplicado es totalmente distinto ya que no se buscaba zonas de fallo, sino obtener los distintos modos en que opera las turbinas. El motivo es que se observó que los modelos de clasificación y regresión presentan más falsos positivos cuando están analizando datos de la zona baja de la curva de potencia de la turbina. Aplicando esta metodología es posible etiquetar estos estados y crear un modelo de clasificación por estado.

SOM

Con el objetivo de encontrar diversos comportamientos en un conjunto de turbinas de un parque que se desconoce a priori, se seleccionó la técnica SOM, por las ventajas descritas en distintos trabajos ([2, 7, 22, 4]). Estos trabajos señalaban la posibilidad de encontrar patrones extraños, posibles candidatos de anomalías en los datos. Aunque la aplicación por lo general se efectúa de forma individual, turbina por turbina en el tiempo. Sin embargo, el enfoque dado en esta tesis como se puede encontrar en la publicación de la Sección 3.1, es distinto. Este enfoque, toma la información de todas las turbinas indistintamente del estado de salud que se encuentren, con el objetivo de encontrar unos grupos de turbinas que funcionan de forma similar y poderlas categorizar con un post-análisis. Además, en el mismo trabajo se muestra cómo es posible a partir de la fusión de datos de todas las turbinas, generar zonas del mapa con distintos modos de operación gracias a juntar datos de un parque completo con el mismo tipo de turbina.

4.4. Conclusiones

En esta tesis se han abarcado distintas aproximaciones al mismo objetivo, detectar patrones de fallos. Durante el desarrollo se han ido encontrando diversos obstáculos debido a la naturaleza ruidosa y de baja calidad de los datos de las turbinas eólicas. Entre estos orígenes de ruido podemos encontrar el originado por operaciones de mantenimiento en la turbina. En ocasiones se tiene acceso a los historiales de mantenimiento, para poder descartar el rango de los registros que han sido afectados. Otros parques que no se disponía de registros de mantenimiento, han tenido que ser identificados con métodos como el SOM. Los distintos Datasets procesados poseían características como baja cantidad de muestras de fallo, dificultando el proceso de filtrado y etiquetado de los casos. Las exigencias han marcado los requerimientos a la hora de efectuar las predicciones de fallo, debido al carácter industrial de esta tesis. Requerimientos como la anticipación temporal o cómo se ha tenido que hacer la división entre los grupos de entrenamiento y test. Requerimientos que no se tienen cuando trabajas con un Dataset sintético o de entorno de laboratorio. Todos estos condicionantes han requerido crear una metodología a medida para su aplicación e implementación en una plataforma automatizada. Es por eso por lo que en cada paso y resultado validado durante la tesis, se ha implementado el algoritmo resultante en la plataforma desarrollada en la empresa, siendo integrado junto a otros módulos que también hacen uso de estos mismos resultados.

La primera fase de filtrado y etiquetado de fallos ha cobrado gran importancia por los resultados encontrados. Se ha generado una publicación en la que se ha podido verificar como el filtrado de forma sistemática empeoraba los resultados de un clasificador/regresor que fuese aplicado a posteriori. Los resultados de los métodos de marcado de valores extremos, no estaban siendo verificados. No se revisaban los registros eliminados mediante la comprobación de a que clase pertenecían, ni tampoco si estos formaban parte de una zona temporal donde la turbina estaba siendo sometida a un mantenimiento. Los registros que fueron marcados como extremos en los resultados, eran en su mayoría dónde la turbina estaba teniendo un comportamiento de fallo, siendo comprobado mediante diversas notificaciones de alarmas. Esto también producía que de cara a un clasificador que generase modelos para estos fallos, el modelo contara con poca capacidad de generalización ya que contaría con muy pocas muestras de casos de fallos. Para mejorar estos resultados actualmente se utiliza un método de rangos manuales ,en términos de valores absolutos y relativos a otras variables, definidos por una persona experta tal y como se indica en una de los artículos publicados.

Los Datasets de las turbinas eólicas se caracterizan por su gran desbalance de casos, dónde encontramos el 97 % de una clase y un 3 % de otra. Se han trabajado en métodos para mitigar esta característica, mediante una selección de técnicas de *undersampling* de clase mayoritaria como TOMEK y de *oversampling* de la clase minoritaria como SMOTE y finalmente suavizada con una técnica de captura de relación entre variables como RBM. Con esto se ha conseguido generar suficientes casos de la muestra minoritaria que mejorase este desbalance, mejorando la capacidad de predicción y generalización del modelo resultante.

Al efectuar la selección de variables teniendo las clases de los distintos estados de las turbinas, se debe proceder de forma cauta y crítica. Esto es porque según los resultados obtenidos en este apartado, es necesario la validación por parte de una persona experta en el dominio de los molinos de viento. Esto es debido a que se veía una clara tendencia a seleccionar variables con comportamientos discretos que no venían condicionadas al evento de fallo, sino a otros cambios externos como por ejemplo casos de mantenimiento. También se ha podido comprobar como haciendo un análisis exploratorio para cada variable seleccionada antes y después de las reparaciones en las distintas turbinas, se dan casos que no se ven patrones de cambio en las variables. Esto significa que no se ve una reacción causa efecto en muchas de las seleccionadas. Esto indica que se requiere un paso de comprobación adicional por una persona experta a la hora de o bien elegir el resultado entre todos los métodos de selección o bien al seleccionar las turbinas de entrenamiento asegurándose que vayan ligadas a un cambio en las variables seleccionadas.

Para poder hacer el análisis de las variables que cambian, la persona experta necesita ayuda de un método que resuma la información de las más de 200 variables que se tienen disponibles. Por eso en el proceso indicado en el trabajo publicado con el uso del SOM (Sección 3.1) y la aplicación de dos etapas de clusterización, es posible obtener los modos en que las turbinas se comportan , pudiendo crear zonas en un espacio bidimensional en las que categorizar esos puntos como en mantenimiento, producción, etc. Con esto la persona experta puede ver de forma resumida el impacto de cada variable en cada zona del mapa utilizando los histogramas de los clústers encontrados, también llamados modos de operación. Estos mismos clústers pueden ser utilizados para que un clasificador se enfoque cuando la turbina solamente esté produciendo, minimizando los falsos positivos cuando esta es parada. Estos modos de operación también se pueden emplear a la hora de hacer selección de variables ya que se elimina el ruido inducido en algunas de estas cuando la turbina arranca o para, consiguiendo la colección de variables que mejor describen un fallo y eliminando un factor importante que afecta en algunos casos, la estacionalidad.

Finalmente se pueden ver resultados prometedores utilizando métodos de *Deep Learning* como las redes recurrentes LSTM. Se considera el resultado positivo porque se ha visto una disminución considerable en los falsos positivos y un comportamiento analógico en distintos casos que la predicción es acertada. Esto ha generado más confianza y expectativas en seguir trabajando con esta metodología. Según los diversos experimentos realizados, esta metodología requiere inversión de muchos recursos al ser un diseño abierto con muchas configuraciones y posibilidades de configuración que no pueden ser determinadas de antemano sino que se ha de ir buscando y ajustando de forma exhaustiva y exploratoria.

4.5. Futuras líneas de investigación

De este trabajo se puede extraer distintas líneas de investigación, aunque se ha identificado una línea con bastante sentido práctico. Esta línea implicaría el uso de los clústers derivados de la aplicación de la técnica del SOM y generar un modelo para cada uno de ellos. Evaluar si un modelo aplicado sobre todos los datos en conjunto da peor resultado que construir tantos submodelos como clústers identificados por el SOM. Los datos de test deben ser primero identificados por el mismo modelo SOM para utilizar el submodelo que le corresponda según el clúster al que pertenece.

La segunda línea sería identificar como cambia el mapa de SOM en el tiempo para una turbina dada, mediante la distancia a la BMU (Best Matching Unit), por si hay alguna relación entre el deterioro de una turbina en concreto a cómo se proyecta en el mapa bidimensional de neuronas del SOM.

Por último, una línea bastante importante es buscar una nueva metodología para el filtrado de casos extremos que se pueda aplicar de forma automática sin la necesidad de una persona experta que defina los intervalos. Esta nueva metodología no puede eliminar datos que contengan estados de alarma como se ha mostrado en una de las publicaciones.

BIBLIOGRAFÍA

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, and Michael Isard. TensorFlow: A System for Large-Scale Machine Learning. *Osdi*, 16:265–283, 2016. ISSN 0270-6474. doi: 10.1038/nn.3331.

[2] A. Astel, S. Tsakovski, P. Barbieri, and V. Simeonov. Comparison of self-organizing maps classification approach with cluster and principal components analysis for large environmental data sets. *Water Research*, 41(19):4566–4578, 2007. ISSN 00431354. doi: 10.1016/j.watres.2007.06.030.

[3] C. Aubrey. Supply Chain: The Race to meet Demand. *Wind Directions*, (January/February):27–34, 2007.

[4] Fernando Bação, Victor Lobo, and Marco Painho. Self-organizing Maps as Substitutes for K-Means Clustering. pages 476–483, 2005. ISSN 03029743. doi: 10.1007/11428862_65.

[5] Mohamed Bennasar, Yulia Hicks, and Rossitza Setchi. Feature selection using Joint Mutual Information Maximisation. *Expert Systems with Applications*, 42 (22):8520–8532, 2015. ISSN 09574174. doi: 10.1016/j.eswa.2015.07.007.

[6] François Besnard and Lina Bertling. An approach for condition-based maintenance optimization applied to wind turbine blades. *IEEE Transactions on Sustainable Energy*, 1(2):77–83, 2010. ISSN 19493029. doi: 10.1109/TSTE.2010.2049452.

[7] Cenk Budayan, Irem Dikmen, and M. Talat Birgonul. Comparing the performance of traditional cluster analysis, self-organizing maps and fuzzy C-means method for strategic grouping. *Expert Systems with Applications*, 36(9):11772–11781, 2009. ISSN 09574174. doi: 10.1016/j.eswa.2009.04.022.

[8] Daniel J. Burke and Mark J. O'Malley. A study of principal component analysis applied to spatially distributed wind power. *IEEE Transactions on Power Systems*, 26(4):2084–2092, 2011. ISSN 08858950. doi: 10.1109/TPWRS.2011.2120632.

[9] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, 2002. ISSN 10769757. doi: 10.1613/jair.953.

[10] Fangzhou Cheng, Jun Wang, Liyan Qu, and Wei Qiao. Rotor-Current-Based Fault Diagnosis for DFIG Wind Turbine Drivetrain Gearboxes Using Frequency Analysis and a Deep Classifier. *IEEE Transactions on Industry Applications*, 54(2): 1062–1071, 2018. ISSN 00939994. doi: 10.1109/TIA.2017.2773426.

[11] François Chollet. Keras: Deep learning library for theano and tensorflow.(2015), 2015.

[12] Philippe Bernard Christophe Leys Olivier Klein. Detecting outliers: Do not use standard deviations around the mean, do use the median absolute deviation around the median | Olivier Klein and Christophe Ley - Academia.edu. *J Exp Soc Psychol*, In Press, mar 2013.

[13] Jacob Cohen. Weighted kappa: Nominal scale agreement provision for scaled disagreement or partial credit. *Psychological Bulletin*, 70(4):213–220, 1968. ISSN 00332909. doi: 10.1037/h0026256.

[14] Niamh Conroy, J. P. Deane, and Brian P. O Gallachoir. Wind turbine availability: Should it be time or energy based? - A case study in Ireland. *Renewable Energy*, 36(11):2967–2971, 2011. ISSN 09601481. doi: 10.1016/j.renene.2011.03.044.

[15] Mian Du, Shichong Ma, and Qing He. A SCADA data based anomaly detection method for wind turbines. *2016 China International Conference on Electricity Distribution (CICED)*, 7(CP0429):1–6, 2016. ISSN 2161749X. doi: 10.1109/CICED.2016.7576060.

[16] European Comission. COMMUNICATION FROM THE COMMISSION TO THE EUROPEAN PARLIAMENT, THE COUNCIL, THE EUROPEAN ECONOMIC AND SOCIAL COMMITTEE AND THE COMMITTEE OF THE RE-

GIONS: Developing the European Dimension in Sport. Technical Report 30.01.2013, Commission of the european communities, 2011.

[17] Eurostat. *Energy balance sheets 2011-2012*, volume 33. 2014. ISBN 9789279378584. doi: 10.2785/52802.

[18] Bärbel Epp Fabiani Appavou Adam Brown. Renewables 2016 Global status report. Technical report, Renewable energy policy network for the 21ST century, 2016.

[19] François Fleuret. Fast Binary Feature Selection with Conditional Mutual Information. *Journal of Machine Learning Research*, 5:1531–1555, 2004.

[20] Fausto Pedro García Márquez, Andrew Mark Tobias, Jesús María Pinar Pérez, and Mayorkinos Papaelias. Condition monitoring of wind turbines: Techniques and methods. *Renewable Energy*, 46:169–178, 2012. ISSN 09601481. doi: 10.1016/j.renene.2012.03.003.

[21] Angel Gil, Miguel A Sanz-Bobi, and Miguel A Rodríguez-López. Behavior Anomaly Indicators Based on Reference Patterns—Application to the Gearbox and Electrical Generator of a Wind Turbine. *Energies*, 11(1), 2018. ISSN 1996-1073. doi: 10.3390/en11010087.

[22] J. L. Giraudel and S. Lek. A comparison of self-organizing map algorithm and some conventional statistical methods for ecological community ordination. *Ecological Modelling*, 146(1-3):329–339, 2001. ISSN 03043800. doi: 10.1016/S0304-3800(01)00324-6.

[23] Pablo M. Granitto, Cesare Furlanello, Franco Biasioli, and Flavia Gasperi. Recursive feature elimination with random forest for PTR-MS analysis of agroindustrial products. *Chemometrics and Intelligent Laboratory Systems*, 83(2):83–90, 2006. ISSN 01697439. doi: 10.1016/j.chemolab.2006.01.007.

[24] Christopher S Gray, Franz Langmayr, Nikolaus Haselgruber, and Simon J

Watson. A Practical Approach to the Use of SCADA Data for Optimized Wind Turbine Condition Based Maintenance. *EWEC Offshore Event*, page 10, 2011.

[25] Antonio Gulli and Sujit Pal. *Deep Learning with Keras*. Packt Publishing Ltd, 2017.

[26] Luis Hernández, Carlos Baladrón, Javier M. Aguiar, Belén Carro, and Antonio Sánchez-Esguevillas. Classification and clustering of electricity demand patterns in industrial parks. *Energies*, 5(12):5215–5228, 2012. ISSN 19961073. doi: 10.3390/en5125215.

[27] Jürgen Herp, Mohammad H. Ramezani, Martin Bach-Andersen, Niels L. Pedersen, and Esmaeil S. Nadimi. Bayesian state prediction of wind turbine bearing failure. *Renewable Energy*, 116:164–172, 2018. ISSN 18790682. doi: 10.1016/j.renene.2017.02.069.

[28] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006. ISSN 00368075. doi: 10.1126/science.1127647.

[29] T Ryan Hoens and Nitesh V Chawla. Imbalanced Datasets: From Sampling to Classifiers. *Imbalanced Learning:Foundations, Algorithms, and Applications*, pages 43–59, 2013. doi: 10.1002/9781118646106.ch3.

[30] Johan Huysmans, Bart Baesens, Jan Vanthienen, and Tony Van Gestel. Failure prediction with self organizing maps. *Expert Systems with Applications*, 30(3): 479–487, 2006. ISSN 09574174. doi: 10.1016/j.eswa.2005.10.005.

[31] IEC. Part 25-1: Communications for monitoring and control of wind power plants – Overall description of principles and models. *IEC 61400-25-1 First Edition 2006-12*, dec 2006. doi: 10.1109/IEEESTD.2007.4288250.

[32] J Jaeger, R Sengupta, and W L Ruzzo. Improved gene selection for classification of microarrays. *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, 64:53–64, 2003. ISSN 2335-6936. doi: 10.1142/9789812776303_0006.

[33] Nathalie Japkowicz. Supervised versus unsupervised binary-learning by feed-forward neural networks. *Machine Learning*, 42(1-2):97–122, jan 2001. ISSN 08856125. doi: 10.1023/A:1007660820062.

[34] G. Jiang, H. He, P. Xie, and Y. Tang. Stacked multilevel-denoising autoencoders: A new representation learning approach for wind turbine gearbox fault diagnosis. *IEEE Transactions on Instrumentation and Measurement*, 66(9):2391–2402, 2017. ISSN 00189456. doi: 10.1109/TIM.2017.2698738.

[35] Guoqian Jiang, Ping Xie, Haibo He, and Jun Yan. Wind Turbine Fault Detection Using a Denoising Autoencoder with Temporal Information. *IEEE/ASME Transactions on Mechatronics*, 23(1):89–100, 2018. ISSN 10834435. doi: 10.1109/TMECH.2017.2759301.

[36] M. Y. Kiang. Extending the Kohonen self-organizing map networks for clustering analysis. *Computational Statistics and Data Analysis*, 38(2):161–180, 2001. ISSN 01679473. doi: 10.1016/S0167-9473(01)00040-8.

[37] Kwang Gi Kim. Book Review: Deep Learning. *Healthcare Informatics Research*, 22(4):351, 2016. ISSN 2093-3681. doi: 10.4258/hir.2016.22.4.351.

[38] Kyusung Kim, Girija Parthasarathy, Onder Uluyol, Wendy Foslien, Shuangwen Sheng, and Paul Fleming. Use of SCADA Data for Failure Detection in Wind Turbines. *ASME 2011 5th International Conference on Energy Sustainability, Parts A, B, and C*, pages 2071–2079, 2011. doi: 10.1115/ES2011-54243.

[39] D Kinga and J Ba Adam. A method for stochastic optimization. *International Conference on Learning Representations (ICLR)*, 2015.

[40] Miron B. Kursa, Aleksander Jankowski, and Witold R. Rudnicki. Boruta - A system for feature selection. *Fundamenta Informaticae*, 101(4):271–285, 2010. ISSN 01692968. doi: 10.3233/FI-2010-288.

[41] Andrew Kusiak and Wenyan Li. The prediction and diagnosis of wind turbine

faults. *Renewable Energy*, 36(1):16–23, 2011. ISSN 09601481. doi: 10.1016/j.renene.2010.05.014.

[42] David D Lewis. Feature Selection and Feature Extraction for Text Categorization. *In Proceedings of Speech and Natural Language Workshop*, pages 212–217, 1992.

[43] David McMillan and Graham W. Ault. Quantification of Condition Monitoring Benefit for Offshore Wind Turbines. *Wind Engineering*, 31(4):267–285, 2007. ISSN 0309-524X. doi: 10.1260/030952407783123060.

[44] Patrick E Meyer and Gianluca Bontempi. On the use of variable complementarity for feature selection in cancer classification. In *Applications of Evolutionary Computing*, pages 91–102. Springer, 2006.

[45] David Milborrow. Operation and maintenance costs compared and revealed. *WindStats*, 19(3):1–87, 2006.

[46] OPC Fundation. What is OPC. oct 2016.

[47] Hanchuan Peng, Fuhui Long, and Chris Ding. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27 (8):1226–1238, 2005.

[48] Francesc Pozo and Yolanda Vidal. Wind turbine fault detection through principal component analysis and statistical hypothesis testing. *Energies*, 9(1):3, 2016. ISSN 19961073. doi: 10.3390/en9010003.

[49] Takaya Saito and Marc Rehmsmeier. The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. *PLoS ONE*, 10(3), 2015. ISSN 19326203. doi: 10.1371/journal.pone.0118432.

[50] Pedro Santos, Luisa F. Villa, An??bal Re??ones, Andres Bustillo, and Jes??s

Maudes. An SVM-based solution for fault detection in wind turbines. *Sensors (Switzerland)*, 15(3):5627–5648, 2015. ISSN 14248220. doi: 10.3390/s150305627.

[51] Zhe Song, Zijun Zhang, Yu Jiang, and Jin Zhu. Wind turbine health state monitoring based on a Bayesian data-driven approach. *Renewable Energy*, 125:172–181, 2018. ISSN 18790682. doi: 10.1016/j.renene.2018.02.096.

[52] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014. ISSN 15337928. doi: 10.1214/12-AOS1000.

[53] Ivan Tomek. Two Modifications of CNN. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-6(11):769–772, 1976. ISSN 0018-9472. doi: 10.1109/TSMC.1976.4309452.

[54] Valeri VoevSiemens A.G. Siemens Remote Diagnostic Services. Technical report, Siemens Wind Power, 2014.

[55] Juha Vesanto and Esa Alhoniemi. Clustering of the self-organizing map. *IEEE Transactions on Neural Networks*, 11(3):586–600, 2000. ISSN 10459227. doi: 10.1109/72.846731.

[56] Vestas R+D. General Specification VESTAS V90 3.0MW. Technical report, Vestas Wind Systems, 2004.

[57] Long Wang, Zijun Zhang, Jia Xu, and Ruihua Liu. Wind Turbine Blade Breakage Monitoring with Deep Autoencoders. *IEEE Transactions on Smart Grid*, 2016. ISSN 19493053. doi: 10.1109/TSG.2016.2621135.

[58] Long Wang, Zijun Zhang, Huan Long, Jia Xu, and Ruihua Liu. Wind Turbine Gearbox Failure Identification with Deep Neural Networks. *IEEE Transactions on Industrial Informatics*, 13(3):1360–1368, 2017. ISSN 15513203. doi: 10.1109/TII.2016.2607179.

[59] Liu Wenyi, Wang Zhenfeng, Han Jiguang, and Wang Guangfeng. Wind turbine fault diagnosis method based on diagonal spectrum and clustering binary tree SVM. *Renewable Energy*, 50:1–6, 2013. ISSN 09601481. doi: 10.1016/j.renene.2012.06.013.

[60] Michael Wilkinson, Keir Harman, Thomas van Delft, and Brian Darnell. Comparison of methods for wind turbine condition monitoring with SCADA data. *IET Renewable Power Generation*, 8(4):390–397, 2014. ISSN 1752-1416. doi: 10.1049/iet-rpg.2013.0318.

[61] Wenzu Wu, Kunjin Chen, Ying Qiao, and Zongxiang Lu. Probabilistic short-term wind power forecasting based on deep neural networks. *2016 International Conference on Probabilistic Methods Applied to Power Systems (PMAPS)*, (October): 1–8, 2016. doi: 10.1109/PMAPS.2016.7764155.

[62] Howard Hua Yang and John E Moody. Data Visualization and Feature Selection: New Algorithms for Nongaussian Data. In *NIPS*, volume 99, pages 687–693. Citeseer, 1999.

[63] Le Yang, Zhongbin Ouyang, and Yong Shi. A modified clustering method based on self-organizing maps and its applications. In *Procedia Computer Science*, volume 9, pages 1371–1379, 2012. doi: 10.1016/j.procs.2012.04.151.

[64] Li Zhao, Zuowei Pan, Changsheng Shao, and Qianzhi Yang. Application of SOM neural network in fault diagnosis of wind turbine. In *Renewable Power Generation (RPG 2015), International Conference on*, pages 2–5, 2016. ISBN 978-1-78561-040-0. doi: 10.1049/cp.2015.0446.

[65] Maciej Zięba, Jakub M. Tomczak, and Adam Gonczarek. RBM-SMOTE: Restricted boltzmann machines for synthetic minority oversampling technique. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 9011:377–386, 2015. ISSN 16113349. doi: 10.1007/978-3-319-15702-3_37.

APÉNDICE A

PLATAFORMA DESARROLLADA EN SMARTIVE

A continuación se pueden ver unas capturas de pantallas, trozos de códigos desarrollados junto a otros compañeros como trabajo secundario durante la tesis para la presentación de los resultados y el análisis a nivel industrial. La plataforma se compone de una parte dedicada a *Machine Learning* estructurada como en la Figura A.1, implementada casi en su totalidad en R y C++ A.3 consumiendo los datos mostrados en la captura de la Figura A.2 . Para presentar los resultados y la información de los distintos parques, la plataforma se compone de un *Frontend* y *Backend*. El *Frontend* se puede ver en las Figuras A.4,A.5,A.6,A.7 estando implementado en IONIC que se basa en Angular.js A.8.

Es una interfaz web accesible desde `https://cast.smartive.eu` y `https://cm.smartive.eu`, mediante autorización previa de acceso (contactar a `info@smartive.eu`) para obtener un usuario demo.

Esta interfaz necesita un *Backend* A.9 desarrollado en Node.js, que efectúe el tratamiento y gestión necesaria de la información para que se pueda enviar para su visualización.

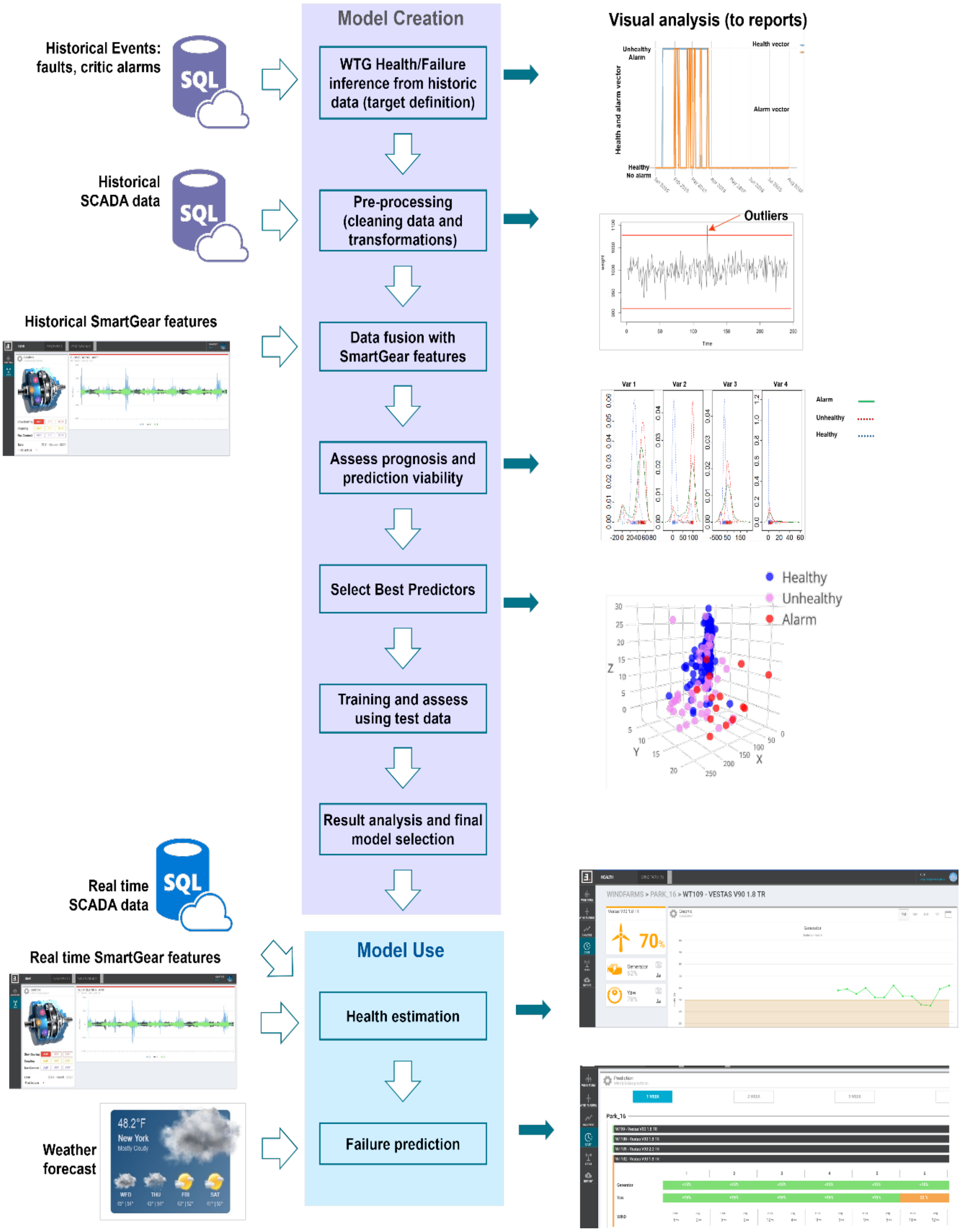
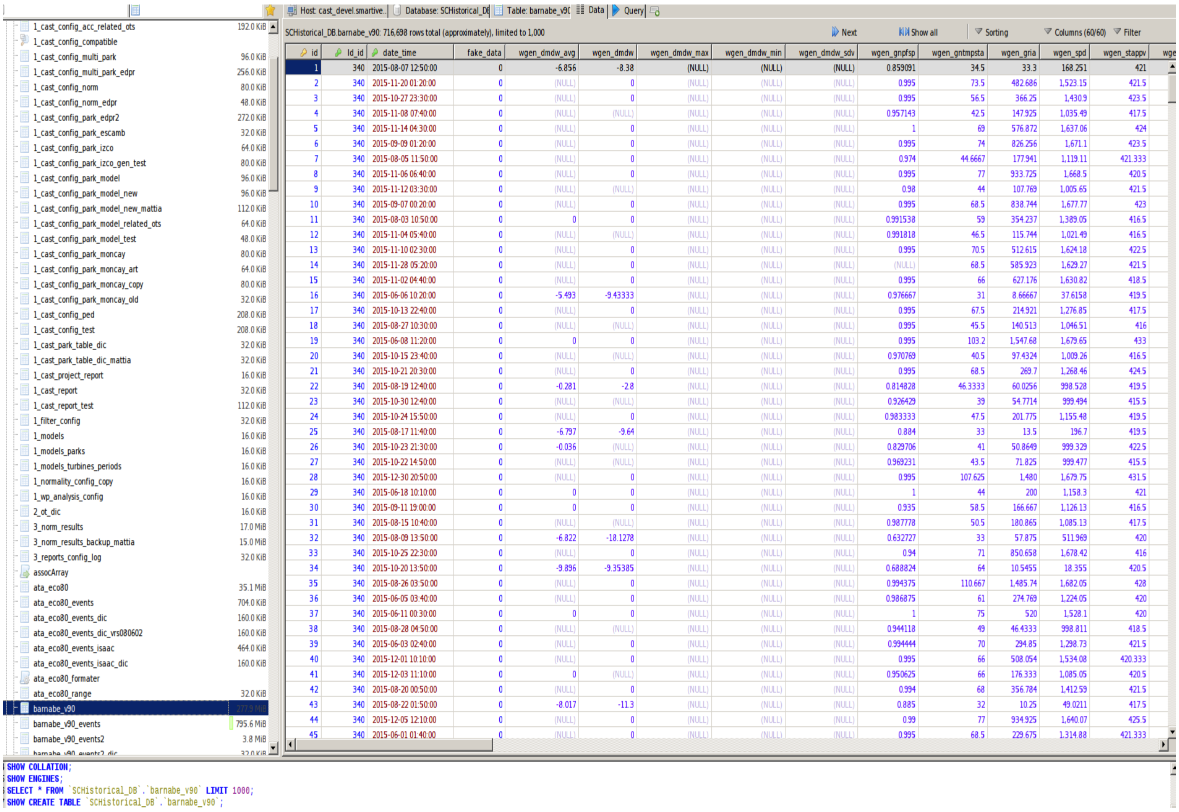

Historical Events: faults, critic alarms
SQL
Historical SCADA data
SQL
Historical SmartGear features
Model Creation
WTG Health/Failure inference from historic data (target definition)
Pre-processing (cleaning data and transformations)
Data fusion with SmartGear features
Assess prognosis and prediction viability
Select Best Predictors
Training and assess using test data
Result analysis and final model selection
Visual analysis (to reports)
Health and alarm vector
Unhealthy Alarm
Healthy No alarm
Health vector
Alarm vector
Outliers
Time
Var 1
Var 2
Var 3
Var 4
Alarm
Unhealthy
Healthy
Healthy
Unhealthy
Alarm
X
Y
Z
Real time SCADA data
SQL
Real time SmartGear features
Weather forecast
48.2°F
New York
Mostly Cloudy
WED
THU
FRI
SAT
Model Use
Health estimation
Failure prediction
WINDFARMS > PARK_16 > WT109 - VESTAS V90 1.8 TR
70%
Generator
Yaw
Prediction
Park_16

Figura A.1: Estructura general plataforma de predicción de fallo de SMARTIVE



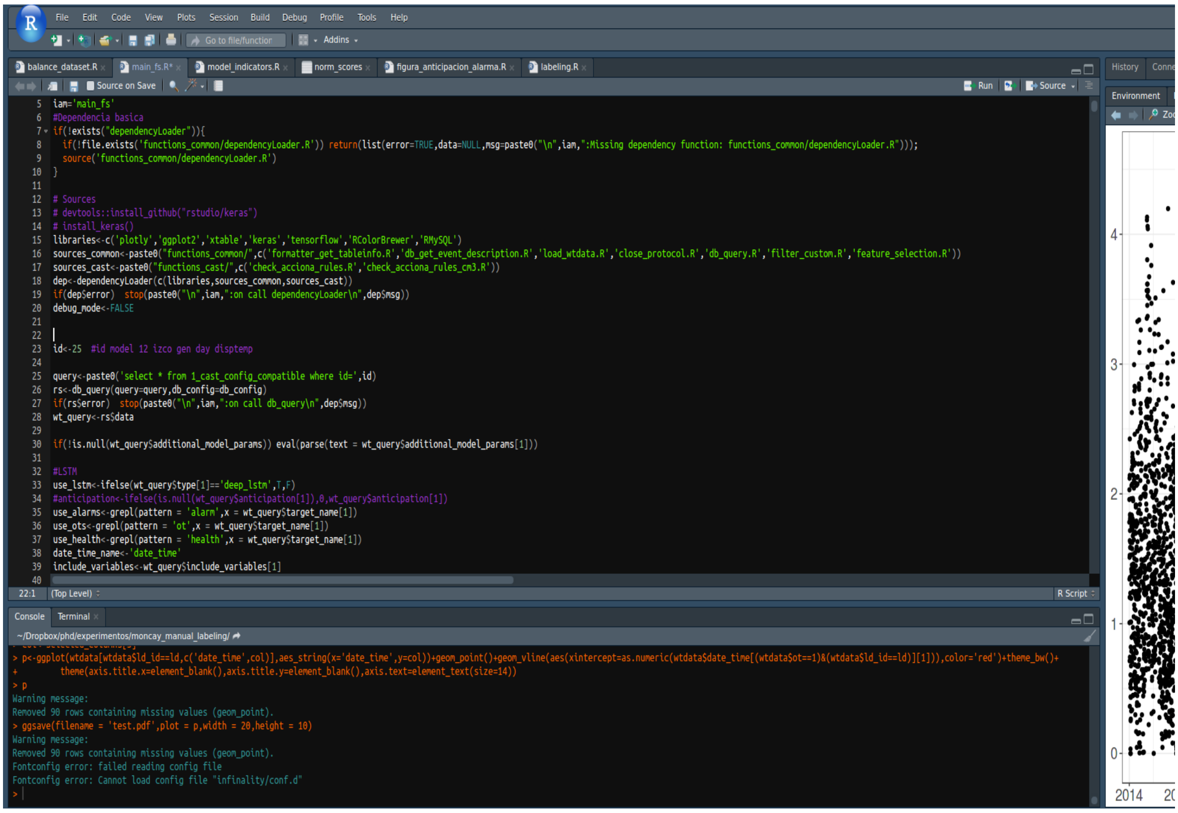
1_cast_config_acc_related_ots	192.0 KiB
1_cast_config_compatible	
1_cast_config_multi_park	96.0 KiB
1_cast_config_multi_park_edpr	256.0 KiB
1_cast_config_norm	80.0 KiB
1_cast_config_norm_edpr	48.0 KiB
1_cast_config_park_edpr2	272.0 KiB
1_cast_config_park_escamb	32.0 KiB
1_cast_config_park_izco	64.0 KiB
1_cast_config_park_izco_gen_test	80.0 KiB
1_cast_config_park_model	96.0 KiB
1_cast_config_park_model_new	96.0 KiB
1_cast_config_park_model_new_mattia	112.0 KiB
1_cast_config_park_model_related_ots	64.0 KiB
1_cast_config_park_model_test	48.0 KiB
1_cast_config_park_moncay	80.0 KiB
1_cast_config_park_moncay_art	64.0 KiB
1_cast_config_park_moncay_copy	80.0 KiB
1_cast_config_park_moncay_old	32.0 KiB
1_cast_config_ped	208.0 KiB
1_cast_config_test	208.0 KiB
1_cast_park_table_dic	32.0 KiB
1_cast_park_table_dic_mattia	32.0 KiB
1_cast_project_report	16.0 KiB
1_cast_report	32.0 KiB
1_cast_report_test	112.0 KiB
1_filter_config	32.0 KiB
1_models	16.0 KiB
1_models_parks	16.0 KiB
1_models_turbines_periods	16.0 KiB
1_normality_config_copy	16.0 KiB
1_wp_analysis_config	16.0 KiB
2_ot_dic	16.0 KiB
3_norm_results	17.0 MiB
3_norm_results_backup_mattia	15.0 MiB
3_reports_config_log	32.0 KiB
assocArray	
ata_eco80	35.1 MiB
ata_eco80_events	704.0 KiB
ata_eco80_events_dic	160.0 KiB
ata_eco80_events_dic_vrs080602	160.0 KiB
ata_eco80_events_isaac	464.0 KiB
ata_eco80_events_isaac_dic	160.0 KiB
ata_eco80_formater	
ata_eco80_range	32.0 KiB
barnabe_v90	277.9 MiB
barnabe_v90_events	795.6 MiB
barnabe_v90_events2	3.8 MiB
barnabe_v90_events2_dic	32.0 KiB

SCHistorical_DB.barnabe_v90: 716,698 rows total (approximately), limited to 1,000

id	Id_id	date_time	fake_data	wgen_dmdw_avg	wgen_dmdw	wgen_dmdw_max	wgen_dmdw_min	wgen_dmdw_sdv	wgen_gnpfsp	wgen_gntmpsta	wgen_gria	wgen_spd	wgen_stappv
1	340	2015-08-07 12:50:00	0	-6.856	-8.38	(NULL)	(NULL)	(NULL)	0.859091	34.5	33.3	168.251	421
2	340	2015-11-20 01:20:00	0	(NULL)	0	(NULL)	(NULL)	(NULL)	0.995	73.5	482.686	1,523.15	421.5
3	340	2015-10-27 23:30:00	0	(NULL)	0	(NULL)	(NULL)	(NULL)	0.995	56.5	366.25	1,430.9	423.5
4	340	2015-11-08 07:40:00	0	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	0.957143	42.5	147.925	1,035.49	417.5
5	340	2015-11-14 04:30:00	0	(NULL)	0	(NULL)	(NULL)	(NULL)	1	69	576.872	1,637.06	424
6	340	2015-09-09 01:20:00	0	(NULL)	0	(NULL)	(NULL)	(NULL)	0.995	74	826.256	1,671.1	423.5
7	340	2015-08-05 11:50:00	0	(NULL)	0	(NULL)	(NULL)	(NULL)	0.974	44.6667	177.941	1,119.11	421.333
8	340	2015-11-06 06:40:00	0	(NULL)	0	(NULL)	(NULL)	(NULL)	0.995	77	933.725	1,668.5	420.5
9	340	2015-11-12 03:30:00	0	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	0.98	44	107.769	1,005.65	421.5
10	340	2015-09-07 00:20:00	0	(NULL)	0	(NULL)	(NULL)	(NULL)	0.995	68.5	838.744	1,677.77	423
11	340	2015-08-03 10:50:00	0	0	0	(NULL)	(NULL)	(NULL)	0.991538	59	354.237	1,389.05	416.5
12	340	2015-11-04 05:40:00	0	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	0.991818	46.5	115.744	1,021.49	416.5
13	340	2015-11-10 02:30:00	0	(NULL)	0	(NULL)	(NULL)	(NULL)	0.995	70.5	512.615	1,624.18	422.5
14	340	2015-11-28 05:20:00	0	(NULL)	0	(NULL)	(NULL)	(NULL)	(NULL)	68.5	585.923	1,629.27	421.5
15	340	2015-11-02 04:40:00	0	(NULL)	0	(NULL)	(NULL)	(NULL)	0.995	66	627.176	1,630.82	418.5
16	340	2015-06-06 10:20:00	0	-5.493	-9.43333	(NULL)	(NULL)	(NULL)	0.976667	31	8.66667	37.6158	419.5
17	340	2015-10-13 22:40:00	0	(NULL)	0	(NULL)	(NULL)	(NULL)	0.995	67.5	214.921	1,276.85	417.5
18	340	2015-08-27 10:30:00	0	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	0.995	45.5	140.513	1,046.51	416
19	340	2015-06-08 11:20:00	0	0	0	(NULL)	(NULL)	(NULL)	0.995	103.2	1,547.68	1,679.65	433
20	340	2015-10-15 23:40:00	0	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	0.970769	40.5	97.4324	1,009.26	416.5
21	340	2015-10-21 20:30:00	0	(NULL)	0	(NULL)	(NULL)	(NULL)	0.995	68.5	269.7	1,268.46	424.5
22	340	2015-08-19 12:40:00	0	-0.281	-2.8	(NULL)	(NULL)	(NULL)	0.814828	46.3333	60.0256	998.528	419.5
23	340	2015-10-30 12:40:00	0	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	0.926429	39	54.7714	999.494	415.5
24	340	2015-10-24 15:50:00	0	(NULL)	0	(NULL)	(NULL)	(NULL)	0.983333	47.5	201.775	1,155.48	419.5
25	340	2015-08-17 11:40:00	0	-6.797	-9.64	(NULL)	(NULL)	(NULL)	0.884	33	13.5	196.7	419.5
26	340	2015-10-23 21:30:00	0	-0.036	(NULL)	(NULL)	(NULL)	(NULL)	0.829706	41	50.8649	999.329	422.5
27	340	2015-10-22 14:50:00	0	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	0.969231	43.5	71.825	999.477	415.5
28	340	2015-12-30 20:50:00	0	(NULL)	0	(NULL)	(NULL)	(NULL)	0.995	107.625	1,480	1,679.75	431.5
29	340	2015-06-18 10:10:00	0	0	0	(NULL)	(NULL)	(NULL)	1	44	200	1,158.3	421
30	340	2015-09-11 19:00:00	0	0	0	(NULL)	(NULL)	(NULL)	0.935	58.5	166.667	1,126.13	416.5
31	340	2015-08-15 10:40:00	0	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	0.987778	50.5	180.865	1,085.13	417.5
32	340	2015-08-09 13:50:00	0	-6.822	-18.1278	(NULL)	(NULL)	(NULL)	0.632727	33	57.875	511.969	420
33	340	2015-10-25 22:30:00	0	(NULL)	0	(NULL)	(NULL)	(NULL)	0.94	71	850.658	1,678.42	416
34	340	2015-10-20 13:50:00	0	-9.896	-9.35385	(NULL)	(NULL)	(NULL)	0.688824	64	10.5455	18.355	420.5
35	340	2015-08-26 03:50:00	0	(NULL)	0	(NULL)	(NULL)	(NULL)	0.994375	110.667	1,485.74	1,682.05	428
36	340	2015-06-05 03:40:00	0	(NULL)	0	(NULL)	(NULL)	(NULL)	0.986875	61	274.769	1,224.05	420
37	340	2015-06-11 00:30:00	0	0	0	(NULL)	(NULL)	(NULL)	1	75	520	1,528.1	420
38	340	2015-08-28 04:50:00	0	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	0.944118	49	46.4333	998.811	418.5
39	340	2015-06-03 02:40:00	0	(NULL)	0	(NULL)	(NULL)	(NULL)	0.994444	70	294.85	1,298.73	421.5
40	340	2015-12-01 10:10:00	0	(NULL)	0	(NULL)	(NULL)	(NULL)	0.995	66	508.054	1,534.08	420.333
41	340	2015-12-03 11:10:00	0	0	(NULL)	(NULL)	(NULL)	(NULL)	0.950625	66	176.333	1,085.05	420.5
42	340	2015-08-20 00:50:00	0	(NULL)	0	(NULL)	(NULL)	(NULL)	0.994	68	356.784	1,412.59	421.5
43	340	2015-08-22 01:50:00	0	-8.017	-11.3	(NULL)	(NULL)	(NULL)	0.885	32	10.25	49.0211	417.5
44	340	2015-12-05 12:10:00	0	(NULL)	0	(NULL)	(NULL)	(NULL)	0.99	77	934.925	1,640.07	425.5
45	340	2015-06-01 01:40:00	0	(NULL)	0	(NULL)	(NULL)	(NULL)	0.995	68.5	229.675	1,314.88	421.333

```
SHOW COLLATION;
SHOW ENGINES;
SELECT * FROM `SCHistorical_DB`.`barnabe_v90` LIMIT 1000;
SHOW CREATE TABLE `SCHistorical_DB`.`barnabe_v90`;
```

Figura A.2: Captura de la base de datos donde se almacenan los distintos parques



```r
iam='main_fs'
#Dependencia basica
if(!exists("dependencyLoader")){
  if(!file.exists('functions_common/dependencyLoader.R')) return(list(error=TRUE,data=NULL,msg=paste0("\n",iam,":Missing dependency function: functions_common/dependencyLoader.R")));
  source('functions_common/dependencyLoader.R')
}

# Sources
# devtools::install_github("rstudio/keras")
# install_keras()
libraries<-c('plotly','ggplot2','xtable','keras','tensorflow','RColorBrewer','RMySQL')
sources_common<-paste0("functions_common/",c('formatter_get_tableinfo.R','db_get_event_description.R','load_wtdata.R','close_protocol.R','db_query.R','filter_custom.R','feature_selection.R'))
sources_cast<-paste0("functions_cast/",c('check_acciona_rules.R','check_acciona_rules_cm3.R'))
dep<-dependencyLoader(c(libraries,sources_common,sources_cast))
if(dep$error)  stop(paste0("\n",iam,":on call dependencyLoader\n",dep$msg))
debug_mode<-FALSE


id<-25  #id model 12 izco gen day disptemp

query<-paste0('select * from 1_cast_config_compatible where id=',id)
rs<-db_query(query=query,db_config=db_config)
if(rs$error)  stop(paste0("\n",iam,":on call db_query\n",dep$msg))
wt_query<-rs$data

if(!is.null(wt_query$additional_model_params)) eval(parse(text = wt_query$additional_model_params[1]))

#LSTM
use_lstm<-ifelse(wt_query$type[1]=='deep_lstm',T,F)
#anticipation<-ifelse(is.null(wt_query$anticipation[1]),0,wt_query$anticipation[1])
use_alarms<-grepl(pattern = 'alarm',x = wt_query$target_name[1])
use_ots<-grepl(pattern = 'ot',x = wt_query$target_name[1])
use_health<-grepl(pattern = 'health',x = wt_query$target_name[1])
date_time_name<-'date_time'
include_variables<-wt_query$include_variables[1]
```

```
> p<-ggplot(wtdata[wtdata$ld_id==ld,c('date_time',col)],aes_string(x='date_time',y=col))+geom_point()+geom_vline(aes(xintercept=as.numeric(wtdata$date_time[(wtdata$ot==1)&(wtdata$ld_id==ld)][1])),color='red')+theme_bw()+
+         theme(axis.title.x=element_blank(),axis.title.y=element_blank(),axis.text=element_text(size=14))
> p
Warning message:
Removed 90 rows containing missing values (geom_point).
> ggsave(filename = 'test.pdf',plot = p,width = 20,height = 10)
Warning message:
Removed 90 rows containing missing values (geom_point).
Fontconfig error: failed reading config file
Fontconfig error: Cannot load config file "infinality/conf.d"
> |
```

Figura A.3: Captura donde se desarrolla el código de los modelos, mediante Rstudio en R y C++

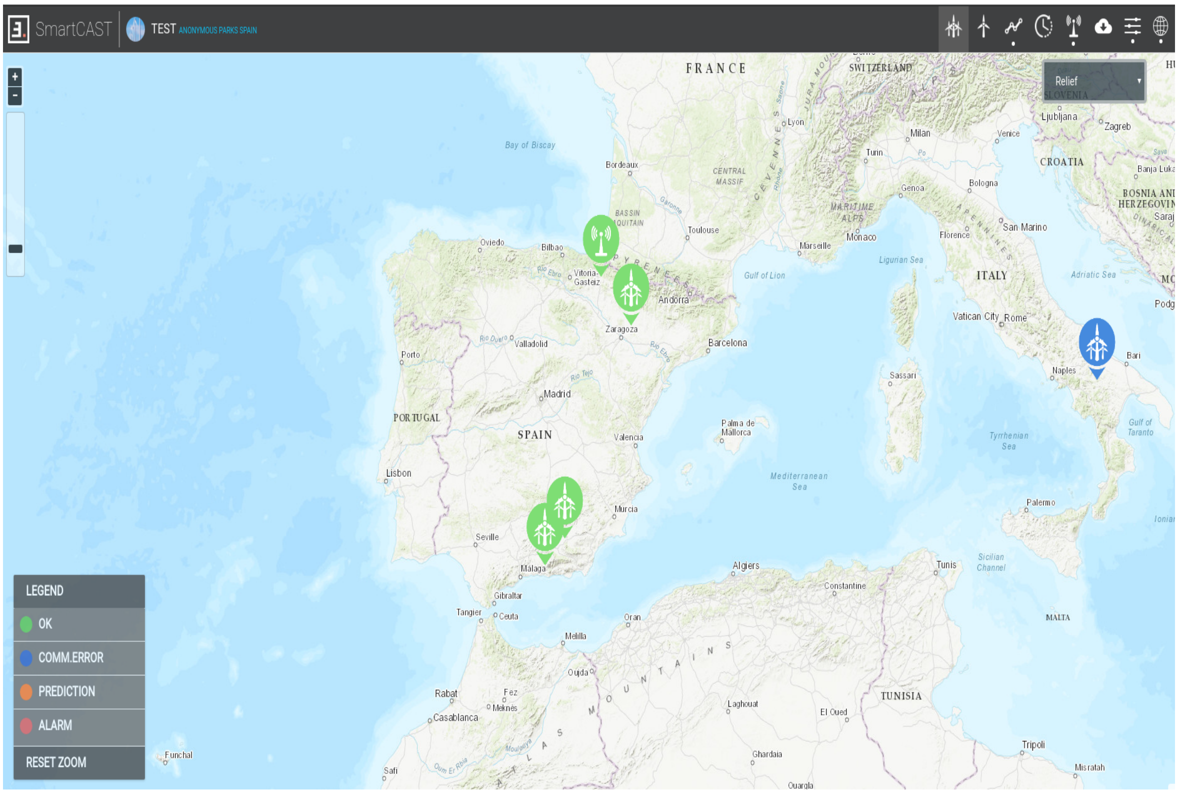
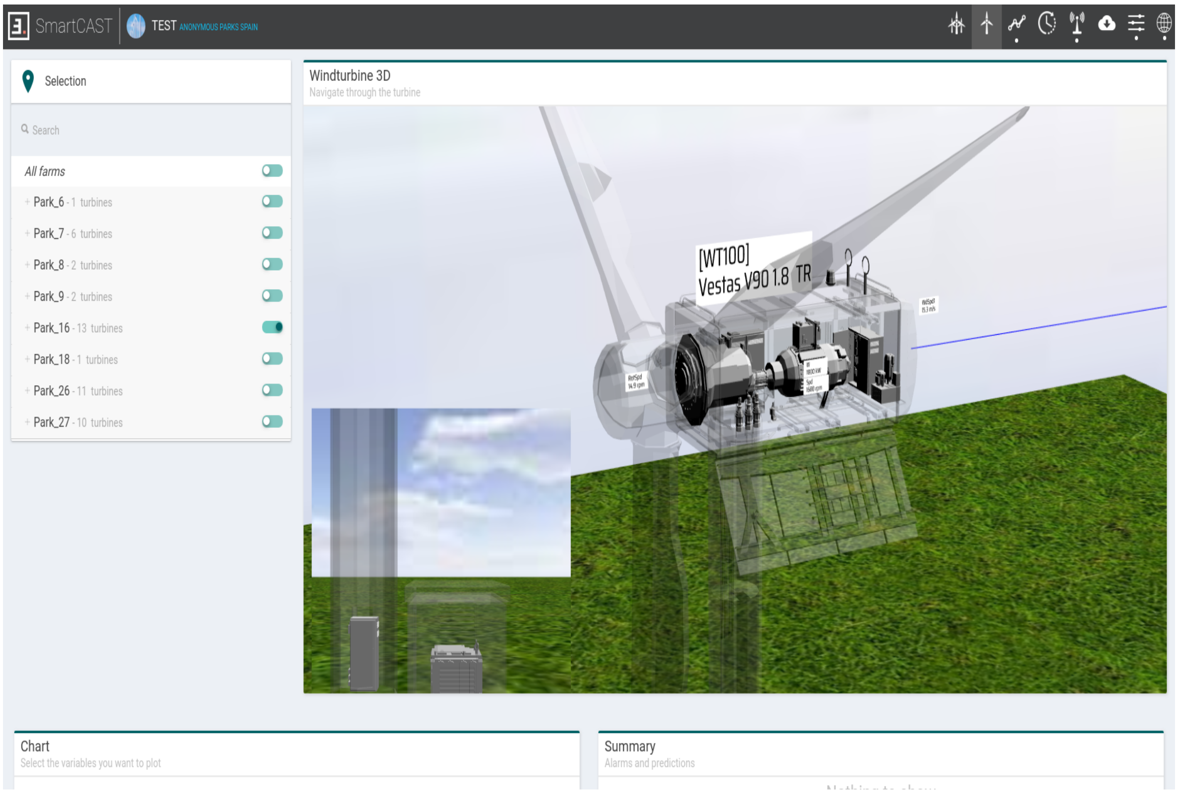


Figura A.4: Pantalla principal de la interfaz.Se muestran los parques y los resultados de las analíticas. Desarrollado en IONIC(Angular.js)



All farms
Park_6 - 1 turbines
Park_7 - 6 turbines
Park_8 - 2 turbines
Park_9 - 2 turbines
Park_16 - 13 turbines
Park_18 - 1 turbines
Park_26 - 11 turbines
Park_27 - 10 turbines

Figura A.5: Pantalla detallada de una turbina, con modelo 3D interactivo (con Webgl) desarrollado en Three.js

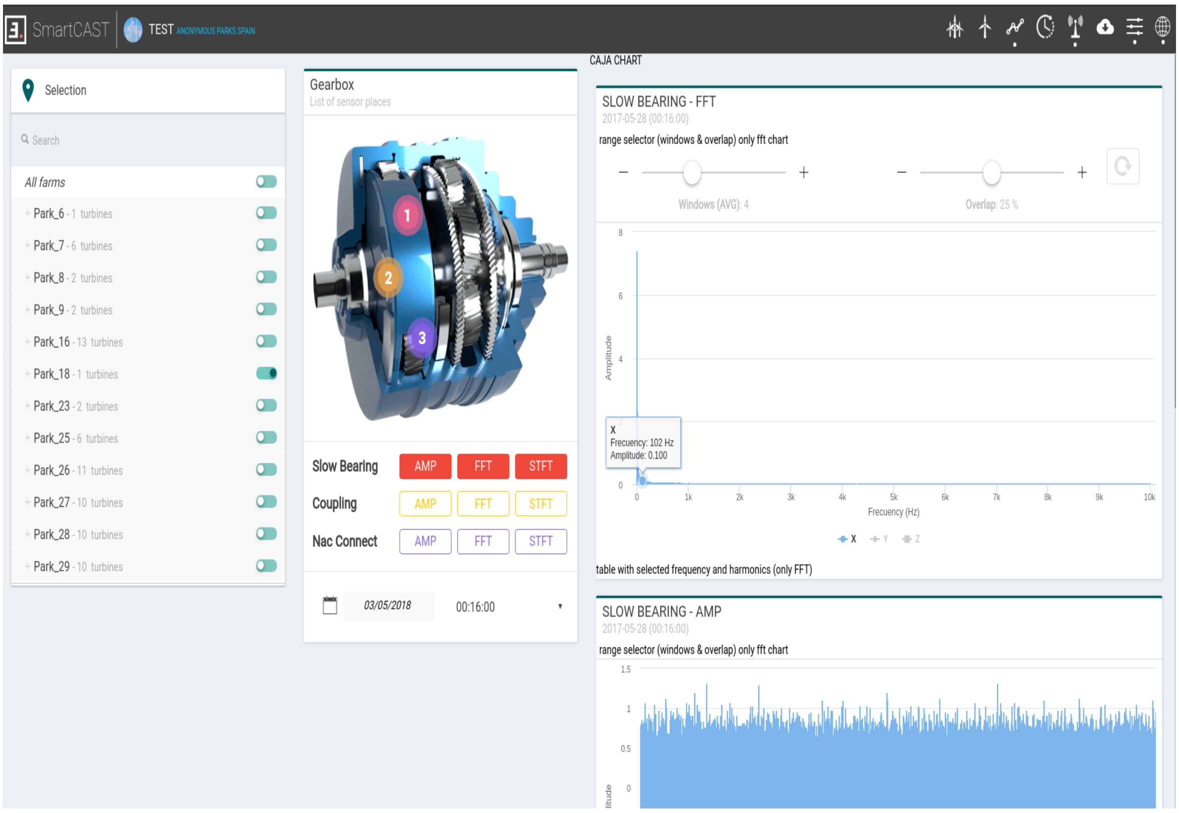


Figura A.6: Pantalla de analíticas por variable y sistema.

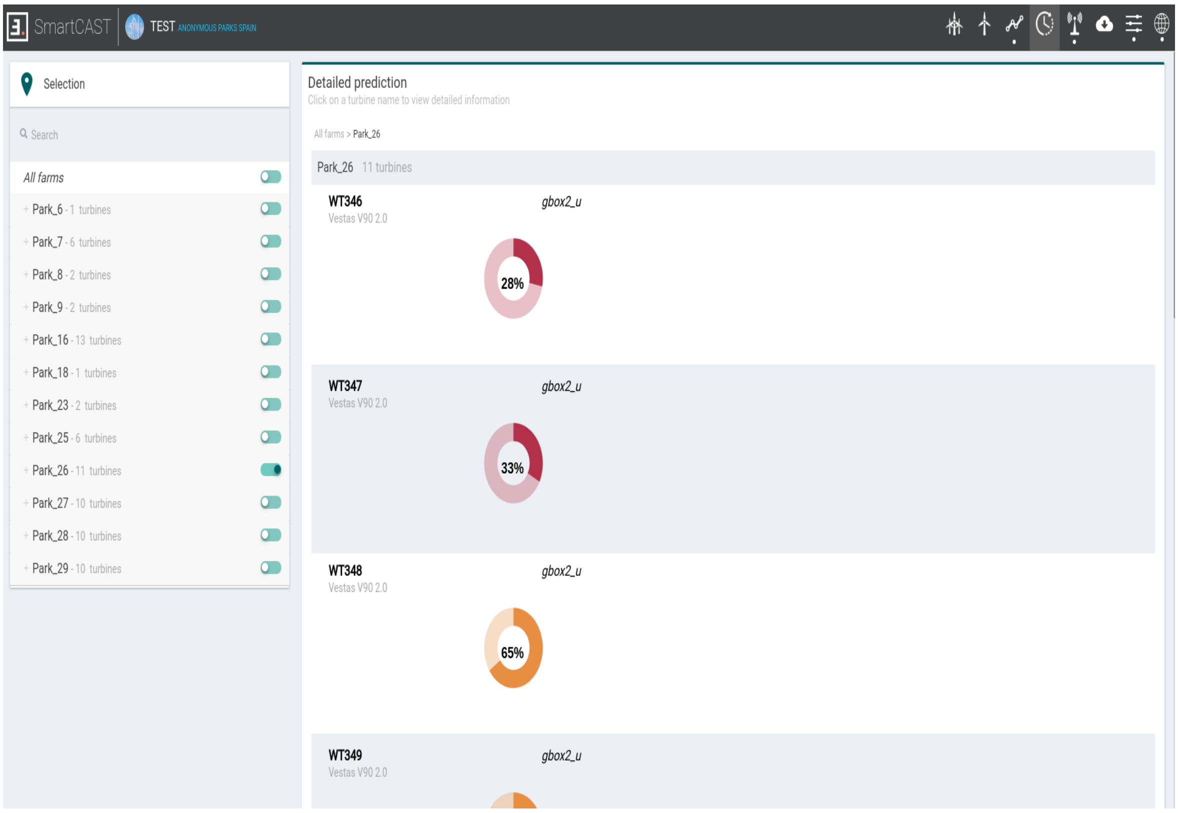
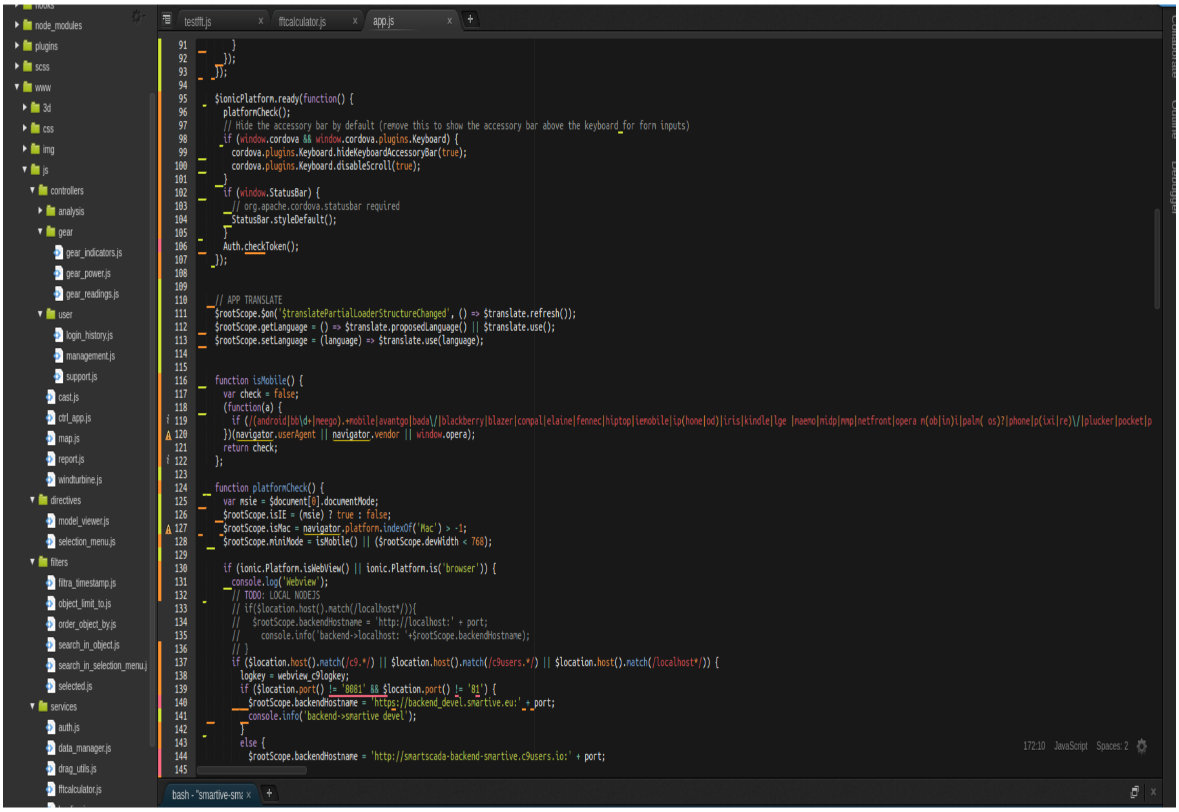
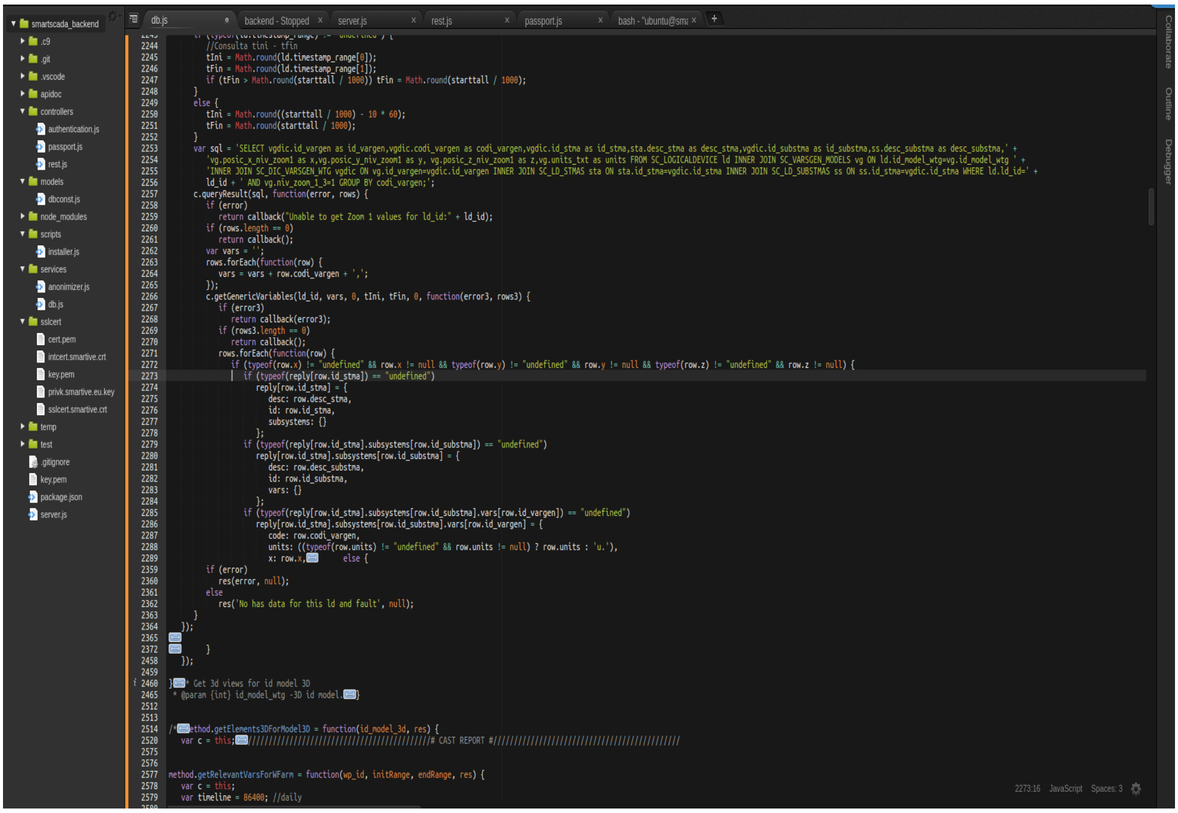


Figura A.7: Pantalla de resumen de predicción por sistema de una turbina.



```
      }
    });
  });

  $ionicPlatform.ready(function() {
    platformCheck();
    // Hide the accessory bar by default (remove this to show the accessory bar above the keyboard for form inputs)
    if (window.cordova && window.cordova.plugins.Keyboard) {
      cordova.plugins.Keyboard.hideKeyboardAccessoryBar(true);
      cordova.plugins.Keyboard.disableScroll(true);
    }
    if (window.StatusBar) {
      // org.apache.cordova.statusbar required
      StatusBar.styleDefault();
    }
    Auth.checkToken();
  });


  // APP TRANSLATE
  $rootScope.$on('$translatePartialLoaderStructureChanged', () => $translate.refresh());
  $rootScope.getLanguage = () => $translate.proposedLanguage() || $translate.use();
  $rootScope.setLanguage = (language) => $translate.use(language);


  function isMobile() {
    var check = false;
    (function(a) {
      if (/(android|bb\d+|meego).+mobile|avantgo|bada\/|blackberry|blazer|compal|elaine|fennec|hiptop|iemobile|ip(hone|od)|iris|kindle|lge |maemo|midp|mmp|netfront|opera m(ob|in)i|palm( os)?|phone|p(ixi|re)\/|plucker|pocket|p
    })(navigator.userAgent || navigator.vendor || window.opera);
    return check;
  };

  function platformCheck() {
    var msie = $document[0].documentMode;
    $rootScope.isIE = (msie) ? true : false;
    $rootScope.isMac = navigator.platform.indexOf('Mac') > -1;
    $rootScope.miniMode = isMobile() || ($rootScope.devWidth < 768);

    if (ionic.Platform.isWebView() || ionic.Platform.is('browser')) {
      console.log('Webview');
      // TODO: LOCAL NODEJS
      // if($location.host().match(/localhost*/)){
      //   $rootScope.backendHostname = 'http://localhost:' + port;
      //     console.info('backend->localhost: '+$rootScope.backendHostname);
      // }
      if ($location.host().match(/c9.*/) || $location.host().match(/c9users.*/) || $location.host().match(/localhost*/)) {
        logkey = webview_c9logkey;
        if ($location.port() != '8081' && $location.port() != '81') {
          $rootScope.backendHostname = 'https://backend_devel.smartive.eu:' + port;
          console.info('backend->smartive devel');
        }
        else {
          $rootScope.backendHostname = 'http://smartscada-backend-smartive.c9users.io:' + port;

```

Figura A.8: Captura de código fuente de la interfaz (Frontend).



```
         //Consulta tini - tfin
         tIni = Math.round(ld.timestamp_range[0]);
         tFin = Math.round(ld.timestamp_range[1]);
         if (tFin > Math.round(starttall / 1000)) tFin = Math.round(starttall / 1000);
      }
      else {
         tIni = Math.round((starttall / 1000) - 10 * 60);
         tFin = Math.round(starttall / 1000);
      }
      var sql = 'SELECT vgdic.id_vargen as id_vargen,vgdic.codi_vargen as codi_vargen,vgdic.id_stma as id_stma,sta.desc_stma as desc_stma,vgdic.id_substma as id_substma,ss.desc_substma as desc_substma,' +
         'vg.posic_x_niv_zoom1 as x,vg.posic_y_niv_zoom1 as y, vg.posic_z_niv_zoom1 as z,vg.units_txt as units FROM SC_LOGICALDEVICE ld INNER JOIN SC_VARSGEN_MODELS vg ON ld.id_model_wtg=vg.id_model_wtg ' +
         'INNER JOIN SC_DIC_VARSGEN_WTG vgdic ON vg.id_vargen=vgdic.id_vargen INNER JOIN SC_LD_STMAS sta ON sta.id_stma=vgdic.id_stma INNER JOIN SC_LD_SUBSTMAS ss ON ss.id_stma=vgdic.id_stma WHERE ld.ld_id=' +
         ld_id + ' AND vg.niv_zoom_1_3=1 GROUP BY codi_vargen;';
      c.queryResult(sql, function(error, rows) {
         if (error)
            return callback("Unable to get Zoom 1 values for ld_id:" + ld_id);
         if (rows.length == 0)
            return callback();
         var vars = '';
         rows.forEach(function(row) {
            vars = vars + row.codi_vargen + ',';
         });
         c.getGenericVariables(ld_id, vars, 0, tIni, tFin, 0, function(error3, rows3) {
            if (error3)
               return callback(error3);
            if (rows3.length == 0)
               return callback();
            rows.forEach(function(row) {
               if (typeof(row.x) != "undefined" && row.x != null && typeof(row.y) != "undefined" && row.y != null && typeof(row.z) != "undefined" && row.z != null) {
               |  if (typeof(reply[row.id_stma]) == "undefined")
                     reply[row.id_stma] = {
                        desc: row.desc_stma,
                        id: row.id_stma,
                        subsystems: {}
                     };
                  if (typeof(reply[row.id_stma].subsystems[row.id_substma]) == "undefined")
                     reply[row.id_stma].subsystems[row.id_substma] = {
                        desc: row.desc_substma,
                        id: row.id_substma,
                        vars: {}
                     };
                  if (typeof(reply[row.id_stma].subsystems[row.id_substma].vars[row.id_vargen]) == "undefined")
                     reply[row.id_stma].subsystems[row.id_substma].vars[row.id_vargen] = {
                        code: row.codi_vargen,
                        units: ((typeof(row.units) != "undefined" && row.units != null) ? row.units : 'u.'),
                        x: row.x,⟺      else {
      if (error)
         res(error, null);
      else
         res('No has data for this ld and fault', null);
    }
  });
⟺
⟺     }
  });

}⟺* Get 3d views for id model 3D
 * @param {int} id_model_wtg -3D id model.⟺}


/*⟺ethod.getElements3DForModel3D = function(id_model_3d, res) {
  var c = this;⟺//////////////////////////////////////////# CAST REPORT #//////////////////////////////////////////


method.getRelevantVarsForWFarm = function(wp_id, initRange, endRange, res) {
  var c = this;
  var timeline = 86400; //daily
```

Figura A.9: Captura del código que corre en el servidor (Backend) basado en Node.js.