



ESCUELA TÉCNICA SUPERIOR DE  
INGENIERÍA (ICAI)

# **Redes neuronales aplicadas a la detección de las anomalías de los diferentes componentes de un aerogenerador.**

Autor: Miguel González López

Director: Miguel Ángel Sanz Bobi

Director: Miguel Ángel Rodríguez López

Madrid Julio de 2018



Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título  
*"Redes Neuronales aplicadas a la detección de las anomalías  
de los diferentes componentes de un aerogenerador"*  
en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el  
curso académico *2017-2018* es de mi autoría, original e inédito y  
no ha sido presentado con anterioridad a otros efectos. El Proyecto no es  
plagio de otro, ni total ni parcialmente y la información que ha sido tomada  
de otros documentos está debidamente referenciada.

Fdo.: (Nombre del alumno)

*Agustín González López*

Fecha: ..... / ..... / .....

Autorizada la entrega del proyecto

EL DIRECTOR DEL PROYECTO

*J. J. Sim*

Fdo.: (Nombre del Director)

Fecha: *17* / *7* / *2018*

Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título

.....  
.....  
en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el  
curso académico ..... es de mi autoría, original e inédito y  
no ha sido presentado con anterioridad a otros efectos. El Proyecto no es  
plagio de otro, ni total ni parcialmente y la información que ha sido tomada  
de otros documentos está debidamente referenciada.

Fdo.: (Nombre del alumno)

Fecha: ...../...../.....

Autorizada la entrega del proyecto

EL DIRECTOR DEL PROYECTO

Fdo.: (Nombre del Director)

Fecha: 18 / 07 / 2018

Miguel A. Rodríguez López



**AUTORIZACIÓN PARA LA DIGITALIZACIÓN, DEPÓSITO Y DIVULGACIÓN EN RED DE PROYECTOS FIN DE GRADO, FIN DE MÁSTER, TESINAS O MEMORIAS DE BACHILLERATO**

**1ª. Declaración de la autoría y acreditación de la misma.**

El autor D. Miguel González López

DECLARA ser el titular de los derechos de propiedad intelectual de la obra: *"RN aplicadas a la detección de las anomalías de los difractores compuestos de un cristalógrafo"*, que ésta es una obra original, y que ostenta la condición de autor en el sentido que otorga la Ley de Propiedad Intelectual.

**2ª. Objeto y fines de la cesión.**

Con el fin de dar la máxima difusión a la obra citada a través del Repositorio institucional de la Universidad, el autor CEDE a la Universidad Pontificia Comillas, de forma gratuita y no exclusiva, por el máximo plazo legal y con ámbito universal, los derechos de digitalización, de archivo, de reproducción, de distribución y de comunicación pública, incluido el derecho de puesta a disposición electrónica, tal y como se describen en la Ley de Propiedad Intelectual. El derecho de transformación se cede a los únicos efectos de lo dispuesto en la letra a) del apartado siguiente.

**3ª. Condiciones de la cesión y acceso**

Sin perjuicio de la titularidad de la obra, que sigue correspondiendo a su autor, la cesión de derechos contemplada en esta licencia habilita para:

- a) Transformarla con el fin de adaptarla a cualquier tecnología que permita incorporarla a internet y hacerla accesible; incorporar metadatos para realizar el registro de la obra e incorporar "marcas de agua" o cualquier otro sistema de seguridad o de protección.
- b) Reproducirla en un soporte digital para su incorporación a una base de datos electrónica, incluyendo el derecho de reproducir y almacenar la obra en servidores, a los efectos de garantizar su seguridad, conservación y preservar el formato.
- c) Comunicarla, por defecto, a través de un archivo institucional abierto, accesible de modo libre y gratuito a través de internet.
- d) Cualquier otra forma de acceso (restringido, embargado, cerrado) deberá solicitarse expresamente y obedecer a causas justificadas.
- e) Asignar por defecto a estos trabajos una licencia Creative Commons.
- f) Asignar por defecto a estos trabajos un HANDLE (URL *persistente*).

**4ª. Derechos del autor.**

El autor, en tanto que titular de una obra tiene derecho a:

- a) Que la Universidad identifique claramente su nombre como autor de la misma
- b) Comunicar y dar publicidad a la obra en la versión que ceda y en otras posteriores a través de cualquier medio.
- c) Solicitar la retirada de la obra del repositorio por causa justificada.
- d) Recibir notificación fehaciente de cualquier reclamación que puedan formular terceras personas en relación con la obra y, en particular, de reclamaciones relativas a los derechos de propiedad intelectual sobre ella.

**5ª. Deberes del autor.**

El autor se compromete a:

- a) Garantizar que el compromiso que adquiere mediante el presente escrito no infringe ningún derecho de terceros, ya sean de propiedad industrial, intelectual o cualquier otro.
- b) Garantizar que el contenido de las obras no atenta contra los derechos al honor, a la intimidad y a la imagen de terceros.
- c) Asumir toda reclamación o responsabilidad, incluyendo las indemnizaciones por daños, que pudieran ejercitarse contra la Universidad por terceros que vieran infringidos sus derechos

intereses a causa de la cesión.

- d) Asumir la responsabilidad en el caso de que las instituciones fueran condenadas por infracción de derechos derivada de las obras objeto de la cesión.

**6º. Fines y funcionamiento del Repositorio Institucional.**

La obra se pondrá a disposición de los usuarios para que hagan de ella un uso justo y respetuoso con los derechos del autor, según lo permitido por la legislación aplicable, y con fines de estudio, investigación, o cualquier otro fin lícito. Con dicha finalidad, la Universidad asume los siguientes deberes y se reserva las siguientes facultades:

- La Universidad informará a los usuarios del archivo sobre los usos permitidos, y no garantiza ni asume responsabilidad alguna por otras formas en que los usuarios hagan un uso posterior de las obras no conforme con la legislación vigente. El uso posterior, más allá de la copia privada, requerirá que se cite la fuente y se reconozca la autoría, que no se obtenga beneficio comercial, y que no se realicen obras derivadas.
- La Universidad no revisará el contenido de las obras, que en todo caso permanecerá bajo la responsabilidad exclusiva del autor y no estará obligada a ejercitar acciones legales en nombre del autor en el supuesto de infracciones a derechos de propiedad intelectual derivados del depósito y archivo de las obras. El autor renuncia a cualquier reclamación frente a la Universidad por las formas no ajustadas a la legislación vigente en que los usuarios hagan uso de las obras.
- La Universidad adoptará las medidas necesarias para la preservación de la obra en un futuro.
- La Universidad se reserva la facultad de retirar la obra, previa notificación al autor, en supuestos suficientemente justificados, o en caso de reclamaciones de terceros.

Madrid, a ..... de ..... de .....

ACEPTA

*Miguel González López*

Fdo.....



Motivos para solicitar el acceso restringido, cerrado o embargado del trabajo en el Repositorio Institucional:

# REDES NEURONALES APLICADAS A LA DETECCIÓN DE LAS ANOMALÍAS DE LOS DIFERENTES COMPONENTES DE UN AEROGENERADOR.

**Autor: Miguel González López.**

Director: Miguel Ángel Sanz Bobi.

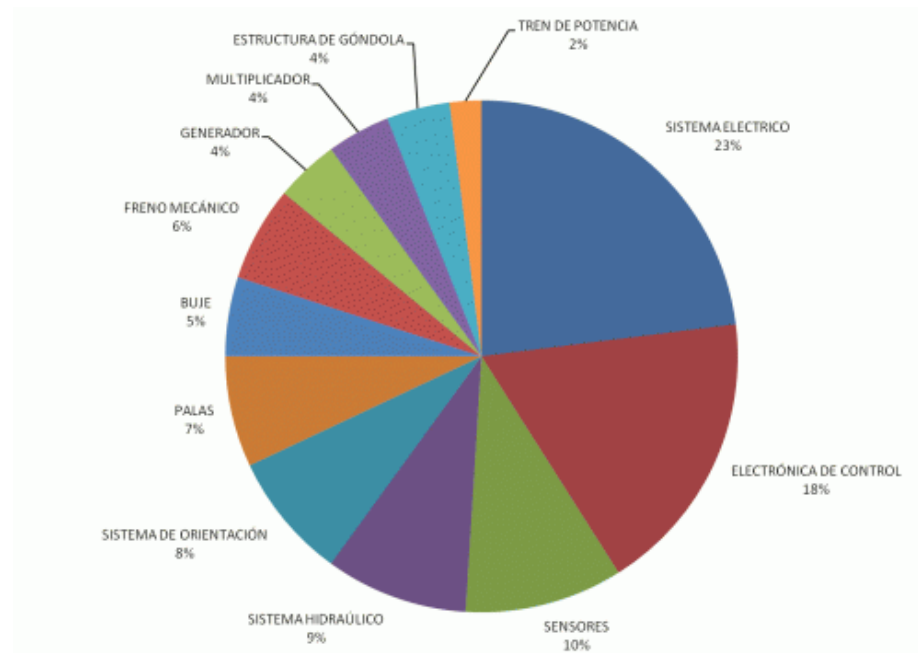
Director: Miguel Ángel Rodríguez López.

## Resumen del proyecto:

La escena energética actual ha cambiado profundamente desde hace décadas. Los combustibles fósiles, principales responsables del cambio climático, se consumen cada vez menos y son las energías renovables: hidroeléctrica, eólica o solar, las que cada vez tienen mayor explotación.

Uno de los principales problemas de las energías renovables son su falta de continuidad. Los combustibles fósiles se pueden consumir de forma continua una vez extraídos, mientras que las energías renovables dependen del sol, del agua o del viento. Otro gran problema es el importante esfuerzo económico que implica levantar grandes estructuras como presas o campos eólicos y solares, así como su mantenimiento. Los generadores eléctricos usados, los materiales e incluso las propias estructuras son muy caras. Además, que cualquier fallo o avería de cualquier componente supone una doble pérdida: mantenimiento o sustitución del componente averiado y pérdida de producción durante el tiempo de reparación.

Este proyecto se va a centrar en resolver el problema de la continuidad y del coste de no producción cuando la máquina esté parada. De entre las tres principales energías renovables va a focalizar su utilidad en los aerogeneradores. Los primeros aerogeneradores aparecen en el siglo XIX con la aparición de la electricidad siguiendo los modelos de los antiguos modelos de viento. No es hasta las décadas de los 70 y 80 cuando las primeras crisis del petróleo obligan a los diferentes gobiernos a estudiar las renovables. Como hemos mencionado, la energía eólica es de las energías renovables más longevas. El envejecimiento de los parques eólicos está provocando que en la actualidad la mayor parte del coste de estos parques eólicos sea su mantenimiento.



**Fig 1. Fallos más comunes en un aerogenerador. [1]**

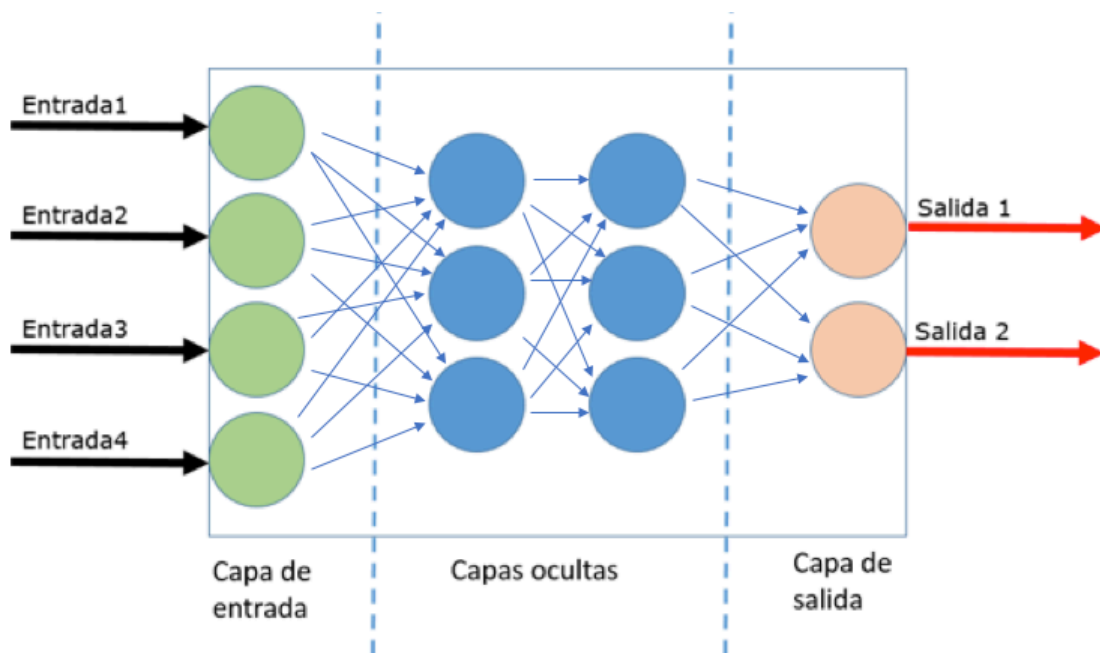
Como muestra la figura 1, ante la creciente necesidad de mantenimiento de los parques eólicos es necesario tener en cuenta dos factores: los fallos más comunes y el tiempo de impacto de parada (se detecta el error, se repara, y se vuelve a arrancar).

El objetivo de este proyecto es reducir al máximo el tiempo de parada de un aerogenerador por fallos de sus componentes. Por eso el estudio trata de desarrollar un sistema para la detección de fallos en cada componente. El método elegido es la elaboración de modelos de comportamiento normal basado en redes neuronales. Estas redes, que son una forma de inteligencia artificial, aprenden a partir de un conjunto de datos reales de las partes del aerogenerador en condiciones normales de operación. Una vez se realicen los modelos se tienen las referencias esenciales de comportamiento para detectar anomalías. Finalmente, tras un estudio de modelado del error, se definirá el concepto de anomalía aplicada a estos sistemas, con el que se pretende garantizar un completo y preciso estudio de los diferentes componentes de un aerogenerador. En definitiva, no solo se busca predecir y simular sino también identificar el error y la anomalía en los diferentes componentes del aerogenerador, con vistas al diagnóstico de sus casos y su aplicación al correspondiente mantenimiento.

Como hemos mencionado, la base de este proyecto es la red neuronal, que en la actualidad es uno de los campos con mucha proyección en la inteligencia artificial. [2]. Son un tipo

de modelos simplificados de las redes neuronales que forman nuestro cerebro, de ahí su nombre y topología. Las características que hacen singulares a estos modelos de inteligencia artificial son las siguientes [4]:

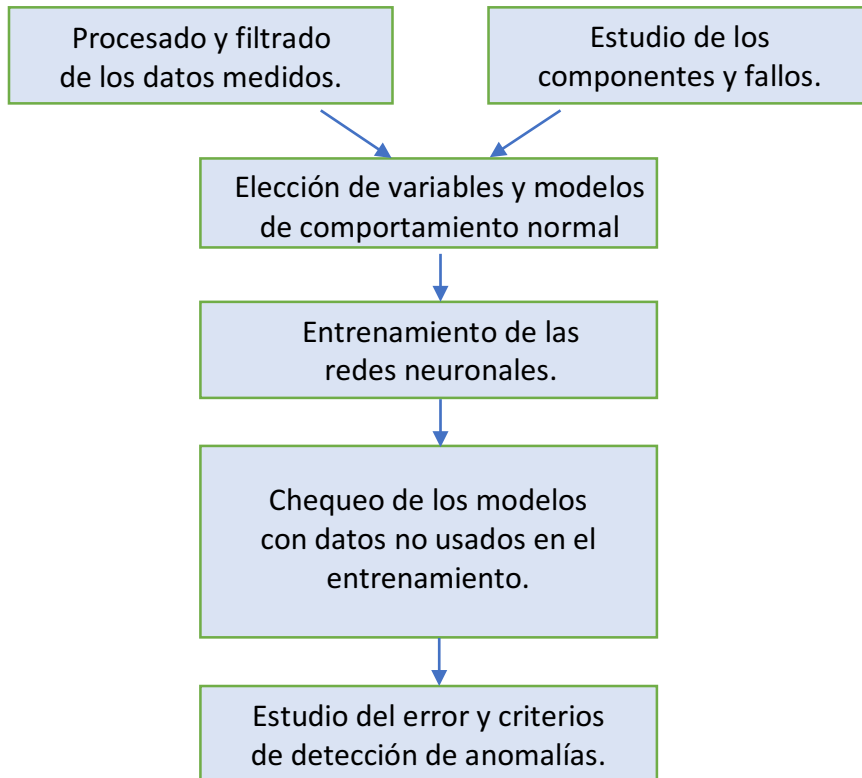
- Tienen **capacidad de aprendizaje**.
- **Velocidad** de procesamiento.
- La **robustez** de la red en conjunto.



*Fig 2. Esquema básico de un perceptrón multicapa.*

En la figura 2, observamos el perceptrón multicapa es el tipo de red neuronal que vamos a desarrollar en el proyecto. Podemos ver como en cada capa tenemos una serie de nodos por donde pasan las entradas multiplicándose por el peso que atribuimos a cada nodo. Finalmente, el resultado de las multiplicaciones y las sumas da las salidas simuladas.

Una vez elegida la tecnología con la que se va a llevar a cabo el proyecto, el proceso a seguir en el proyecto queda resumido en el siguiente esquema:



El procesado y filtrado de datos se realizó en Excel por su sencillez y el resto del proyecto en el entorno de Matlab y Simulink. La versatilidad de Matlab permite: primero; entrenar por separado cada una de las redes neuronales y segundo, desarrollar sus modelos normales de error. Después quedan integrados en un diagrama de bloques muy visual que permite ver con los Scopes tanto los resultados de la simulación como las anomalías en los componentes.

En paralelo al procesado de datos se estudiaron cualitativamente las partes de un aerogenerador:

- **Álabes**
- **Rotor**
- **Ángulo pitch**
- **Freno**
- **Eje de baja velocidad**
- **Multiplicadora**
- **Generador**
- **Controladores**
- **Anemómetro**

- **Veleta**
- **Góndola**
- **Eje de alta velocidad**
- **Unidad de guía**
- **Motor guía**
- **Torre**

Tras el estudio de la base de datos y de los componentes, que se detallará posteriormente, se han elegido seis variables a modelar mediante redes neuronales. En general, son variables que a priori son difíciles de medir, y que se obtienen a partir de medidas más sencillas como la temperatura ambiente, la temperatura de la góndola o la velocidad del viento. De esta forma se crean los modelos de comportamiento normal (redes neuronales), en los que se van a introducir nuevos datos, diferentes a los del entrenamiento, para ver si los componentes que simulan se comportan de la forma prevista:

- **Modelo de la Potencia.**
- **Modelo rodamiento NDE.**
- **Modelo rodamiento DE.**
- **Modelo rodamiento alta de la multiplicadora.**
- **Modelo Pitchangle.**
- **Modelo de los anillos SLD.**

## **Entrenamiento de una red neuronal**

Esta parte es la más metódica. Usando el propio editor de redes neuronales de Matlab, simplemente había que elegir la configuración que mejores resultados daba a cada variable simulada. Se probaron diferentes configuraciones de inputs y outputs, de capas y de distribuciones de los datos de entrenamiento.

Es evidente la importancia de los datos utilizados: las muestras no deben presentar errores, para cada unidad de tiempo utilizada, en ninguna de las variables medidas. De las 6 redes neuronales creadas la que peores resultados da, es aquella cuya muestra de datos era la más reducida, lo que revela la importancia de un entrenamiento completo.

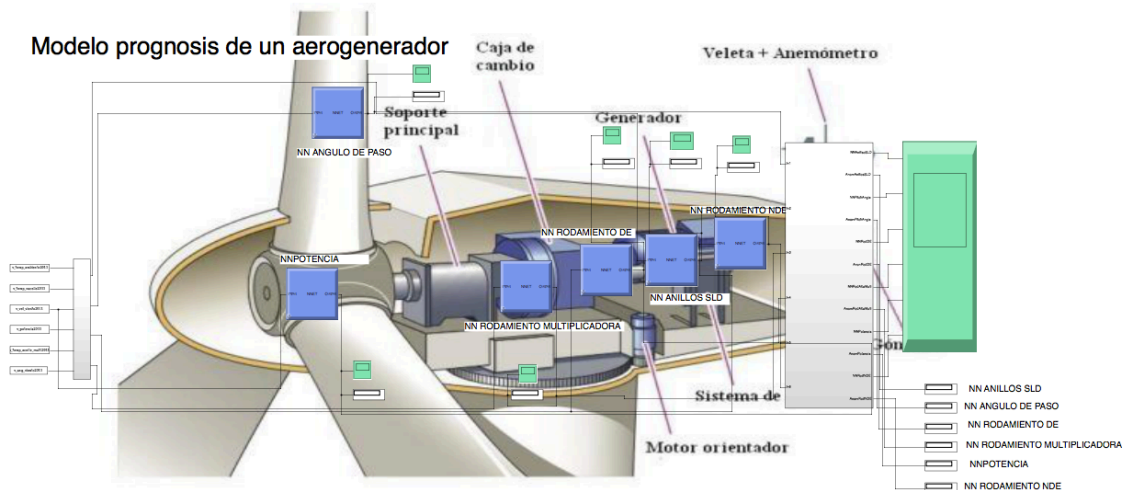
Observando las distribuciones de los errores en cada red neuronal por año podemos detectar una clara tendencia del error a incrementarse según pasa el tiempo. Esto es lógico, pues los comportamientos de los componentes empeoran según se deterioran. Comparando las evoluciones de las distribuciones de error encontramos aquellos componentes que mantienen el mismo error a lo largo de los años, lo que quiere decir, que tienen menos necesidad de ser cambiados o reparados que los otros (NN del rodamiento DE, y de alta de la multiplicadora). En el caso del Rodamiento NDE el aumento de la media del error de 0,125 a 5,019 (casi del 4000%) puede indicar una avería en el 2014, ya que en los años anteriores la media del error casi no se ve afectada.

Para detectar la existencia de errores se han usado distribuciones normales al considerarse que los errores de las redes neuronales se adaptaban a este tipo de modelos. Un estudio más a fondo de estos errores podría determinar mejores distribuciones, que se adaptaran con más precisión a la realidad.

Además, cabe destacar la detección de anomalías ha seguido un modelo bastante sencillo, definiendo como anomalía la existencia de un error fuera del intervalo del 95% de confianza tres veces seguidas. Esta sencillez invita a diversos estudios más en profundidad sobre otras formas más eficaces de determinar estas faltas en los componentes.

## Simulador.

Para poder integrar los seis modelos realizados eran necesarios numerosos bloques de llamada de variables, así como aquellos necesarios para detectar errores y anomalías. Para mayor claridad se ha intentado integrar de la mejor manera posible cada red neuronal en un diagrama de las partes de un aerogenerador.



## Resultados finales de cada red neuronal:

Componente	Anomalías por hora.	Anomalías por año.
Potencia	0,0161	90
Rodamiento NDE	0,0063	35
Rodamiento DE	0,0389	218
Rod. Alta Multi.	0,0059	33
Anillos SLD	0,0358	201
PitchAngle	0,0768	-

Observamos como las redes neuronales del rodamiento NDE y del rodamiento de Alta de la Multiplicadora son las redes que dan menos anomalías por año. La red neuronal del PitchAngle no tenía suficientes datos para ser simulada por lo que sus datos no arrojaban suficiente precisión. Las redes neuronales de los anillos SLD y del rodamiento DE tiene mayor cantidad de anomalías lo que puede indicar un envejecimiento de los componentes simulados. Esto puede desencadenar un fallo que es preciso monitorizar.

## **Conclusión.**

Este proyecto presenta una idea teórica y desarrollada informáticamente sobre un posible modelo de predicción y detección de fallos de un aerogenerador. Hemos observado como la utilización de la red neuronal, una tecnología rápida, robusta y sencilla de manejar, aplicada a una considerable muestra de datos permite crear un modelo que nos permite:

1. Simular las variables difíciles de medir: medidas de los componentes menos accesibles en el interior de la góndola, como son los rodamientos del aerogenerador o de la multiplicadora.
2. Crear unos modelos de comportamiento normal que nos permitan identificar las medidas anómalas.
3. Predecir y controlar el envejecimiento de los componentes más sensibles al fallo de un aerogenerador.

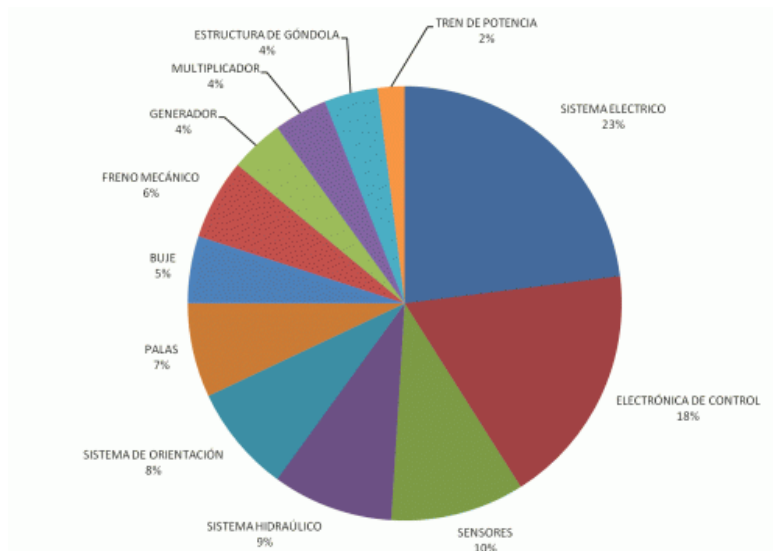
En definitiva, este modelo teórico desarrollado y llevado a la práctica nos permitiría lidiar con una de las causas de los principales problemas que presentan las energías renovables: la continuidad. Es verdad que seguiremos dependiendo de que se manifiesten en la naturaleza, pero evitaremos al máximo el error de nuestras máquinas.

## English version:

The energy scene nowadays has deeply changed during the last decades. Fossil fuels, contributors to global warming, are less used. Renewable energies like hydroelectric, wind or solar energy are expanding.

One of the biggest issues of renewable energies is their lack of continuity. Fossil fuels can be consumed once they are extracted but renewables rely totally on water, sun and wind. The second issue is the cost of paying for all the new infrastructure that renewables need to be harnessed in addition to their maintenance cost. Electric generators, materials are very expensive. Any kind of fault of their components implies a loss of money to repair the damaged component plus the time that machine is not producing energy.

This Project is going to focus on a possible solution to the lack of continuity and the cost of not producing energy while the machine is being repaired. Out of the three main renewable energies mentioned above, it is going to study the results in wind turbines. The first wind turbines are built in the late 19<sup>th</sup> century when electric energy was discovered. It is not until the 1970s and 1980s that the firsts gas crisis made different governments study renewables. As we can see above, wind energy is one of the oldest of the renewable energies. As wind farms age the cost of maintenance is highly increasing.



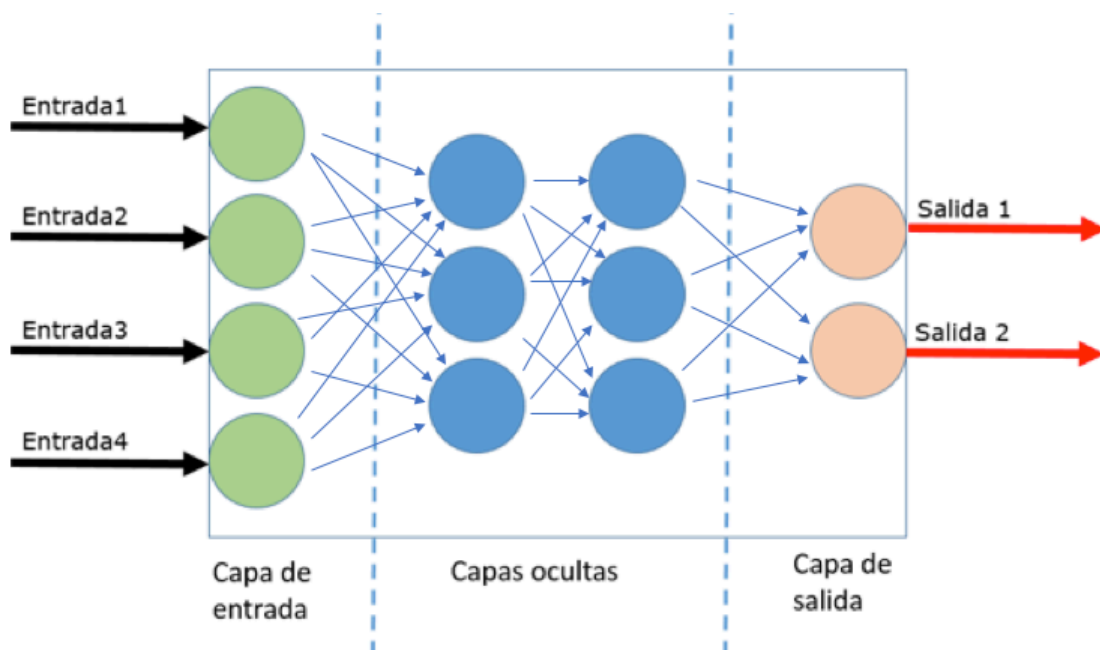
**Fig 1. Most common faults in a wind turbine. [1]**

Looking at figure 1, facing the increasing necessity of maintenance is important to take into account two key factors: most common faults in the components and the time impact (error detection, reparation and restart).

The objective of the Project is to optimize the time that the generator is not running due to failure in their components. That is why the project is going to develop a model to detect faults in wind generators. The chosen method is the development of the normal behaviour models based on Neural Networks. These networks, a form of artificial intelligence, can learn from real data of the different parts of the generator. Once the models are done, we will have the essential behavioural references to detect anomalies. Finally, after a study of the modelling of the errors, the concept of anomaly is defined, doing a complete and precise study of the components. Definitely, it not only predicts and simulates the results but the model tries to identify the error and the anomalies, focusing on diagnosing malfunctions and repairing them.

As mentioned above, the base of this Project is the Neural Network, one of the most important fields in artificial intelligence nowadays [2]. They are a simplified model of our brain structures. Their singular characteristics are [4]:

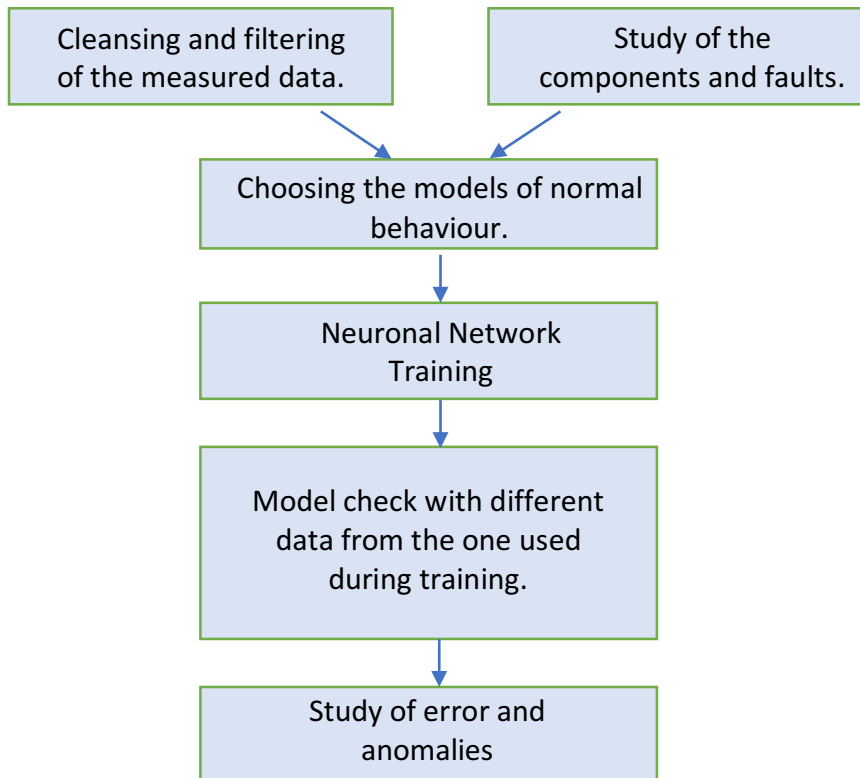
- Learning capability.
- Processing speed.
- Network Robustness.



*Fig 2. Multi-layer Perceptron basic scheme.*

As we can see in figure 2, the multi-layer perceptron is the type of Neural Network used in this project. We can see how in every layer we have a series of nodes where inputs multiply by their specific weight. Finally, the result of the multiplications and sums are the simulated outputs.

Once the technology was chosen, the process followed during the project is the following:



The cleansing and filtering of data was done on Excel and the rest of the Project in Matlab-Simulink. The versatility of Matlab allows first to train each NN separately. After all the NN are integrated in the same Simulink model where the scopes allow to see the result of the simulation and the anomalies.

In parallel to the cleansing and filtering of data, the wind turbine components qualitatively studied were the following:

- **Blades.**
- **Rotor.**
- **Pitch Angle.**
- **Break.**

- **Low-Speed Axis.**
- **Gearbox.**
- **Generator.**
- **Controllers.**
- **Anemeter.**
- **Vane.**
- **Capsule.**
- **High-Speed Axis.**
- **Guide unit.**
- **Guide motor.**
- **Tower.**

After the study of the data base and the data cleansing, six variables have been chosen to be modelled. They are hardly accessible inside the capsule and are simulated with easy-access variables like outside temperature, capsule temperature or wind speed. After this process, normal behaviour models (NN) are created where new data is going to be introduced to check if the simulated components are right.

- **Power model.**
- **NDE bearing model.**
- **DE bearing model.**
- **High-Speed bearing model.**
- **PitchAngle model.**
- **SLD rings model.**

### **Neuronal Network training**

This is the most repetitive process of the Project. Using the NN editor of Matlab, we had to simply choose the configuration with best results. We tried different input and output configuration as well as training distributions.

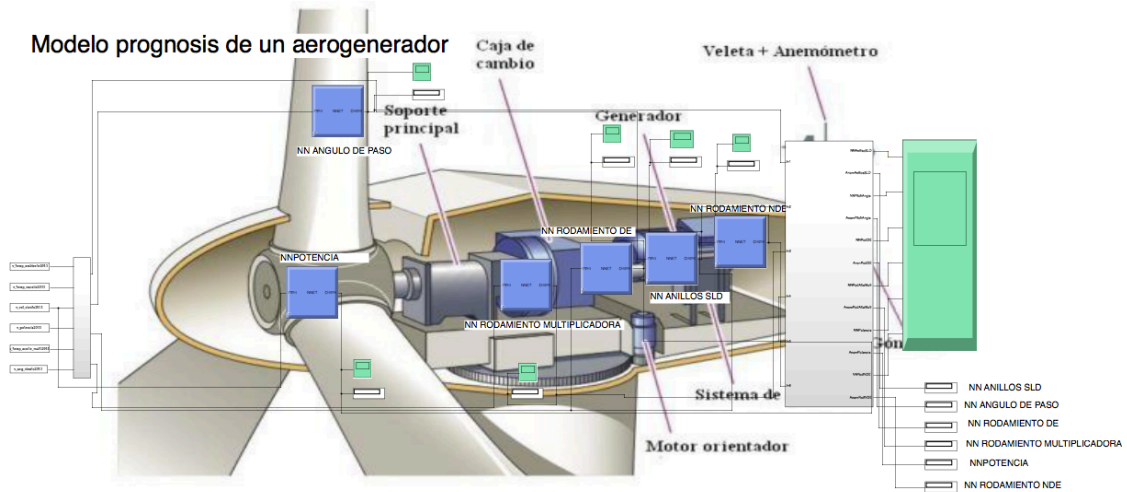
The process showed the importance of the used data. These variables should not present any errors. Out of the 6 NN models the one with worst results is the PitchAngle NN which has the less data. Studying the error distribution in each NN we can detect a clear tendency to increase the error each year. This is logical, because the behaviour of the components deteriorates with time. As an exception, we can see that comparing the error evolution during the years we can see how components (DE bearing and High-Speed bearing) that maintain the same error through time, which points to less faults and maintenance. In the NN of the NDE bearing the average error escalates almost by 4000% in the year 2014, this can indicate a fault.

To detect errors, we have used normal error distributions. A deeper error study could determine better distributions, that could adapt with more precision to reality.

Moreover, it is important to say that the anomaly detection method is very simple, considering the existence of an error outside of the 95% confidence interval three times in a row an anomaly. Deeper studies can determine more precise forms of finding these anomalies.

## Simulator.

In order to be able to integrate the six models many variable call blocks were needed, the same as output blocks. For better clarity, the NN blocks are as close as possible to the component that they simulate in the wind turbine diagram.



## Resultados finales de cada red neuronal:

Component	Anomalías per hour	Anomalías per year
Power	0,0161	90
NDE bearing	0,0063	35
DE bearing	0,0389	218
High-Speed bearing	0,0059	33
SLD Rings	0,0358	201
PitchAngle	0,0768	-

Looking at the table we can see how the NN of the NDE bearing and the High-Speed bearing present less anomalies. The PitchAngle NN did not have enough real data to be simulated properly. The NNs of the SLD Rings and the DE bearing have more anomalies that can point to the aging of these components. This can trigger faults that have to be monitored.

## **Conclusion.**

This Project presents a theoretical and simulated model to predict and detect faults in a wind turbine. We have seen how the use of Neural Network based models can achieve great results with enough real data that allows:

1. Simulation of variables that are hard to measure: measure the least accessible components of the inside of the capsule, like the inside bearings.
2. Creation of normal behaviour models to identify anomalies.
3. Predict and control the aging of the most fault-sensible components.

Definitely, this theoretical method if developed and tested a real wind turbine can help face one of the most important issues that renewable energies present: continuity. We will still depend on nature but we will avoid our machines failures.



## Índice

- 1. Estado del arte.**
    - 1.1 Objetivos y desarrollo del proyecto.**
    - 1.2 Introducción. Fallos en un aerogenerador**
    - 1.3 Redes Neuronales**
    - 1.4 Partes de un aerogenerador**
    - 1.5 Modelos a realizar**
  - 2. Filtrado, selección y retardo de datos**
  - 3. Modelos a realizar**
    - 3.1 Red Neuronal Potencia.**
    - 3.2 Red Neuronal Rodamiento DE.**
    - 3.3 Red Neuronal Rodamiento NDE.**
    - 3.4 Red Neuronal Rodamiento Alta Multiplicadora.**
    - 3.5 Red Neuronal Anillos SLD.**
    - 3.6 Red Neuronal Ángulo de paso de las palas.**
  - 4. Creación de modelos de error para la detección de anomalías.**
    - 4.1 Definición y determinación de anomalías.**
    - 4.2 Gráficas normales de error.**
  - 5. Modelo conjunto.**
    - 5.1 Análisis de la información que proporciona el modelo.**
    - 5.2 Análisis de resultados.**
    - 5.3 Conclusión.**
  - 6. Agradecimientos.**
- ANEXO. Código.**



## **1. Estado del arte**

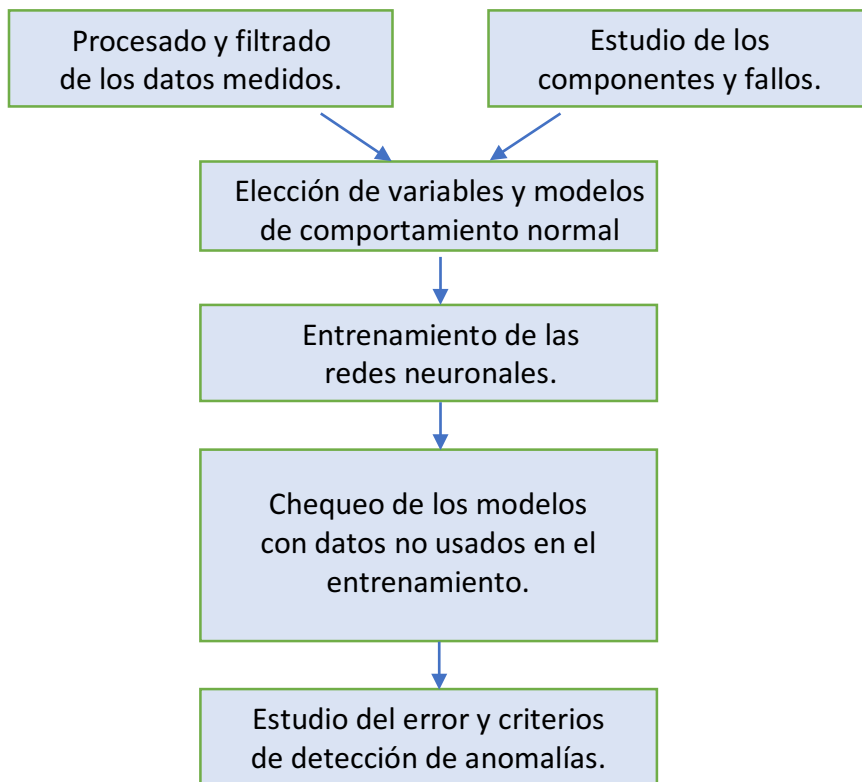
### **1.1 Objetivos y desarrollo del proyecto**

La escena energética actual ha cambiado profundamente desde hace décadas. Los combustibles fósiles, principales responsables del cambio climático, se consumen cada vez menos y son las energías renovables: hidroeléctrica, eólica o solar, las que cada vez tienen mayor explotación.

Uno de los principales problemas de las energías renovables son su falta de continuidad. Los combustibles fósiles se pueden consumir de forma continua una vez extraídos, mientras que las energías renovables dependen de sol, el agua o del viento. Otro gran problema es el importante esfuerzo económico que implica levantar grandes estructuras como presas o campos eólicos y solares, así como su mantenimiento. Los generadores eléctricos usados, los materiales e incluso las propias estructuras son muy caras. Además, que cualquier fallo o avería de cualquier componente supone una doble pérdida: mantenimiento o sustitución del componente averiado y pérdida de producción durante el tiempo de reparación.

El objetivo de este proyecto es el estudio de la detección de fallos en los aerogeneradores. El método elegido es la elaboración de modelos de comportamiento normal basado en redes neuronales. Las redes neuronales aprenden a partir de un conjunto de datos reales de las partes del aerogenerador a simular, en condiciones normales de operación. Una vez se realicen los modelos se tienen las referencias esenciales de comportamiento para detectar anomalías. Finalmente, tras un estudio de modelado del error, se definirá el concepto de anomalía aplicada a estos modelos, con el que se pretende garantizar un completo y preciso estudio de los diferentes componentes de un aerogenerador. En definitiva, no solo se busca predecir y simular sino también identificar el error y la anomalía en los diferentes componentes del aerogenerador, con vistas al diagnóstico de sus casos y su aplicación al correspondiente mantenimiento.

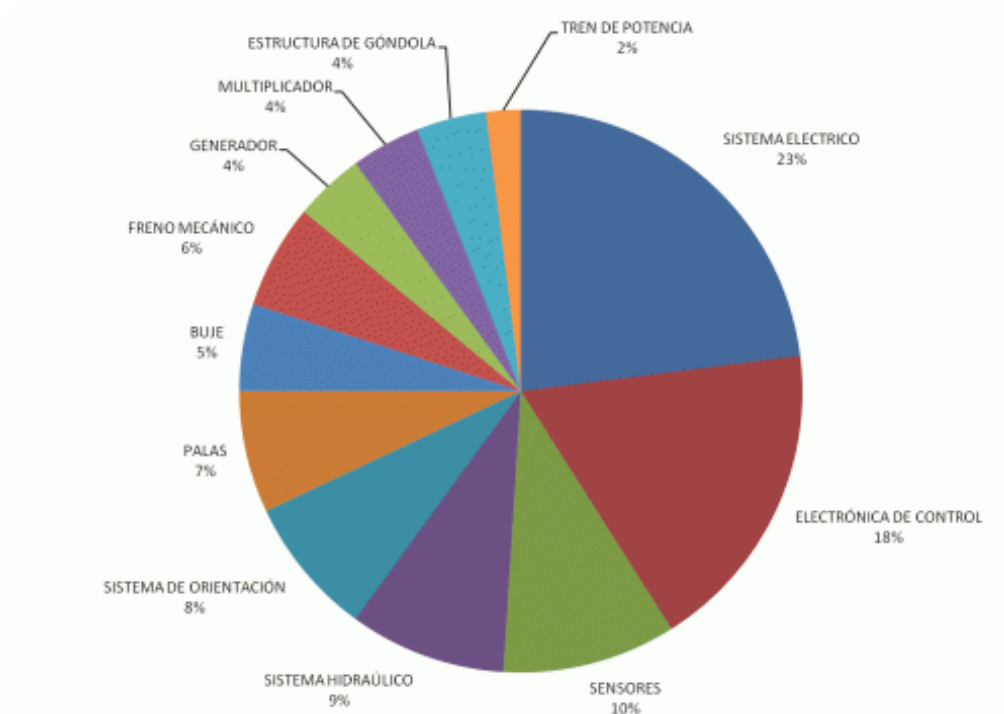
El proceso seguido en el proyecto queda resumido en el siguiente esquema:



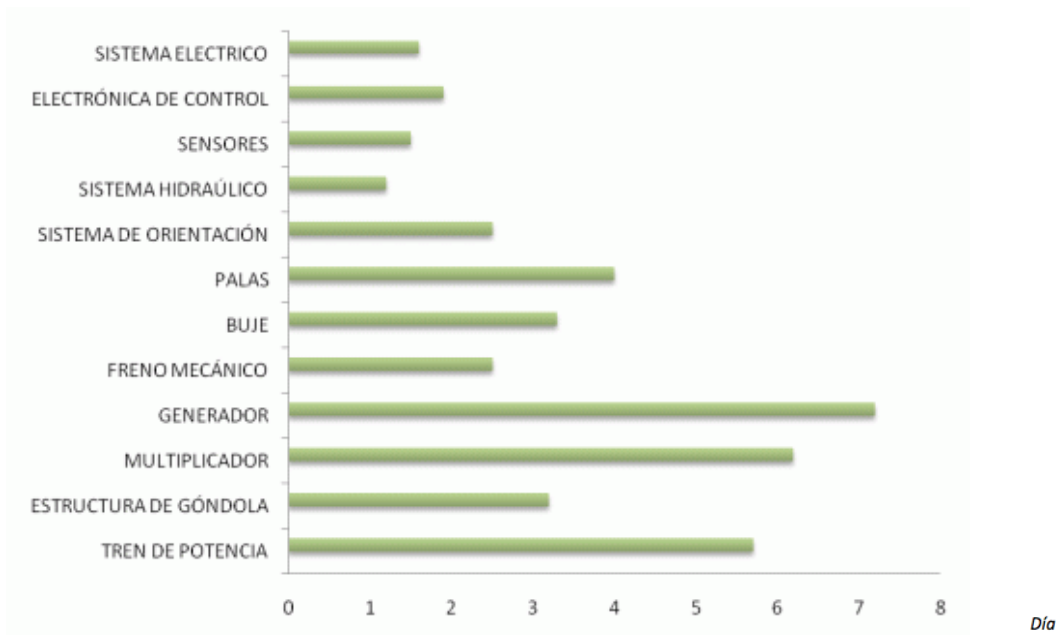
El procesado y filtrado de datos se realizó en Excel por su sencillez y el resto del proyecto en el entorno de Matlab y Simulink. La versatilidad de Matlab permite primero entrenar por separado cada una de las redes neuronales y desarrollar sus modelos normales de error. Después quedan integrados en un diagrama de bloques muy visual que permite ver con los Scopes tanto los resultados de la simulación como las anomalías en los componentes.

## 1.2 Fallos más comunes de un aerogenerador

Los primeros aerogeneradores [1] aparecen en el siglo XIX con la aparición de la electricidad siguiendo los modelos de los antiguos molinos de viento. No es hasta las décadas de los 70 y 80 cuando las primeras crisis del petróleo obligan a los diferentes gobiernos a estudiar las energías renovables. Como podemos observar, la energía eólica es de las energías renovables más longevas. El envejecimiento de los parques eólicos está provocando que en la actualidad la mayor parte de los gastos de estos parques sea su mantenimiento.

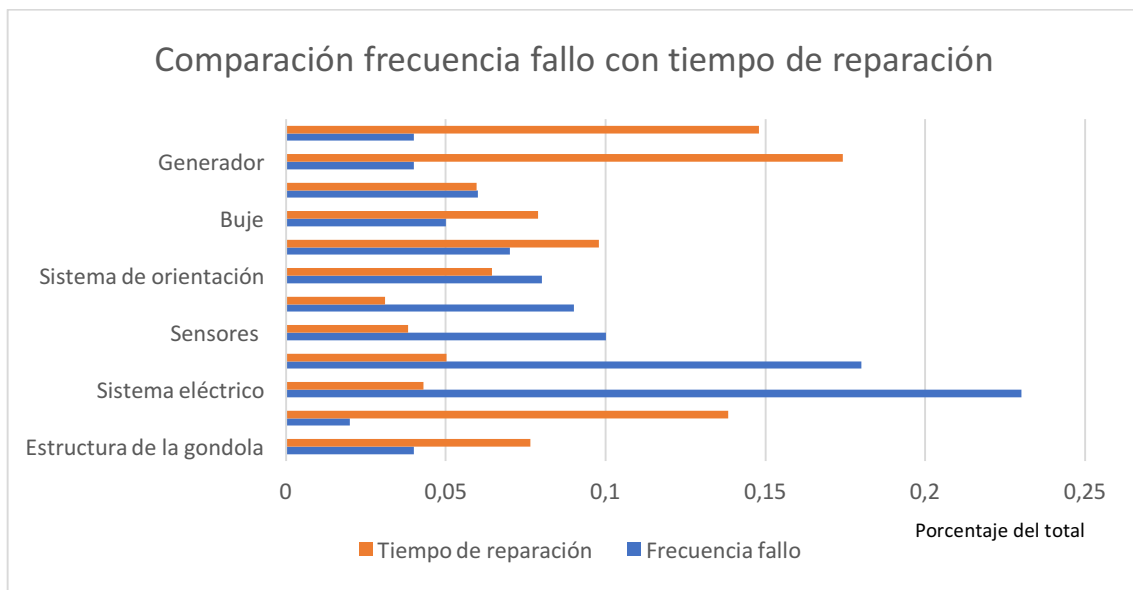


*Fig 1. Fallos más comunes en un aerogenerador. [1]*



**Fig 2. Días necesarios para la reparación de las diferentes partes. [1]**

Como podemos observar en las gráficas anteriores ante la creciente necesidad de mantenimiento de los parques eólicos es necesario tener en cuenta dos factores: los fallos más comunes y el tiempo de impacto de parada: se detecta el error, se repara, y se vuelve a arrancar.



**Fig 3. Tiempo de reparación vs frecuencia fallo.**

Es interesante tener en cuenta el gráfico de la figura 3 a la hora de planear la estrategia de mantenimiento de un parque eólico. También, es importante conocer las condiciones particulares de cada sitio ya que dependiendo de las condiciones meteorológicas habrá fallos más comunes o reparaciones más difíciles de realizar. De ahí la importancia de un buen sistema de prevención automático, como el que se puede realizar a partir de modelos de comportamiento normal basados en redes neuronales.

### 1.3 Partes de un aerogenerador.

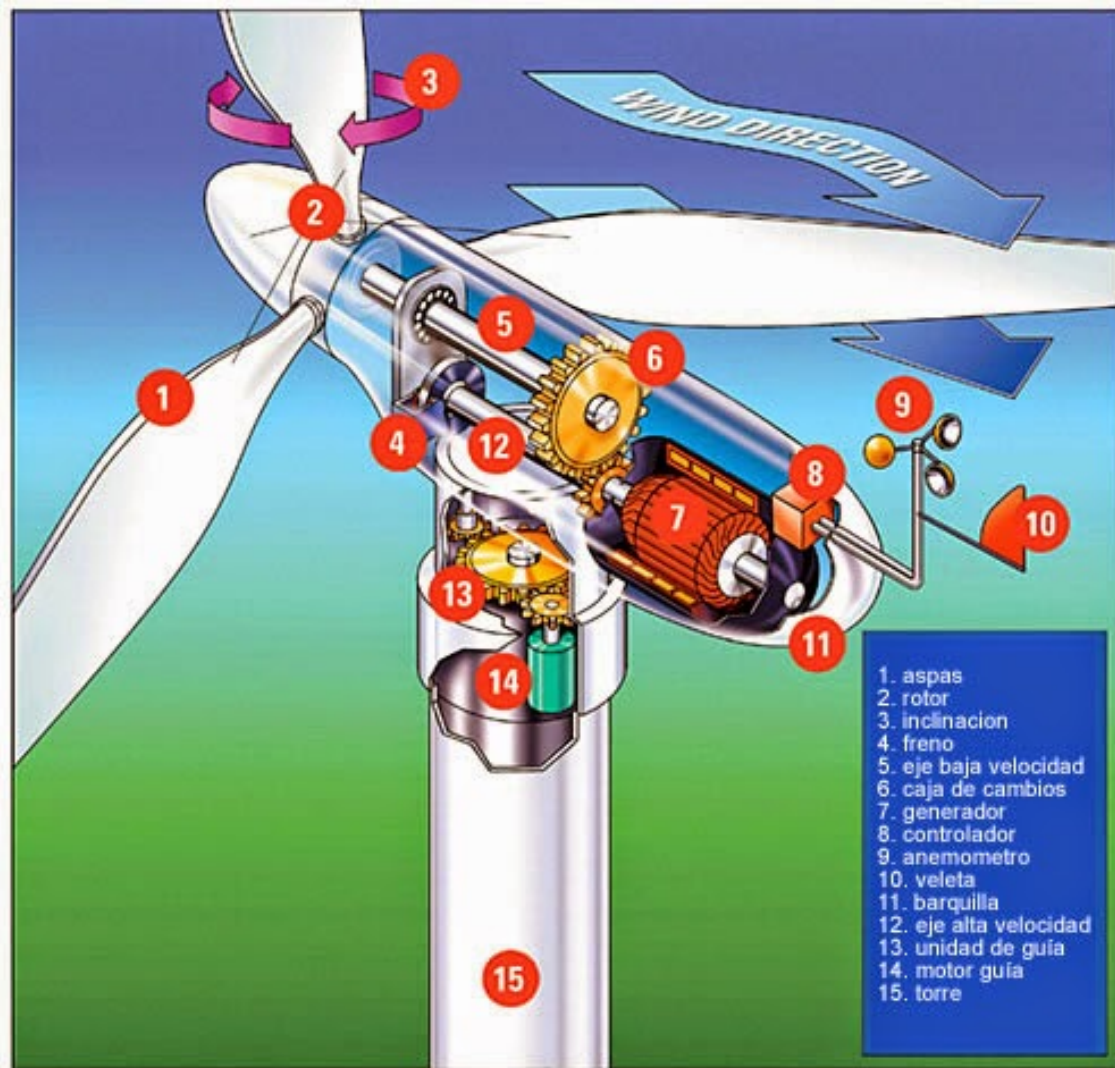


Fig 4. Partes de un aerogenerador [2]

#### 1.3.1 Álabes

Tienen un perfil aerodinámico estudiado para absorber la mayor cantidad de energía posible del viento. El número de palas es variable, aunque el modelo de tres palas es normalmente el más utilizado ya que permite mayor suavidad en el giro. Las palas están hechas de materiales altamente resistentes, pero también ligeros (composites), para poder girar con mayor facilidad.

#### 1.3.2 Rotor

Parte giratoria que acopla los álabes a la cápsula donde está alojado el generador. La mayoría son rotores horizontales.

### **1.3.3 Ángulo pitch**

El control de ángulo pitch es esencial para dar la máxima potencia ajustando el ángulo de paso de las palas según las condiciones del viento.

### **1.3.4 Freno**

Utilizado en caso de que la velocidad del viento sea superior a la que el aerogenerador pueda soportar para evitar averías. Tenemos dos sistemas de frenado [6]:

-Aerodinámico: los álabes del aerogenerador se colocan a 90 grados con eje longitudinal evitando que el viento los pueda empujar para que giren.

-Mecánico: generalmente se usa como apoyo del aerodinámico. Por ejemplo, se usa cuando la turbina está parada, como durante los trabajos de mantenimiento.

### **1.3.5 Eje de baja velocidad**

Es la primera parte de la cadena de transmisión de potencia y es el eje conectado a la turbina que gira a menor velocidad que la velocidad de giro del rotor del aerogenerador.

### **1.3.6 Multiplicadora**

Como hemos mencionado anteriormente, ya que el eje de la turbina y el generador eléctrico giran a velocidades diferentes, es necesario el uso de caja de cambios compuesta de engranajes con el mayor rendimiento posible. De esta forma se acoplan generador y rotor que giran a velocidades muy diferentes. Este elemento no está presente en todos los aerogeneradores como ya veremos más adelante.

### **1.3.7 Generador**

Los generadores [3] convierten la energía cinética en energía eléctrica que se inyecta a la red. Constan de dos partes: una móvil, que se llama rotor y otra estática, el estator. Ambas partes tienen uno o varios devanados por los que circula la corriente. Las corrientes en el devanado del rotor, que está en constante rotación, inducen campos electromagnéticos y, por tanto, corriente eléctrica en los devanados del estator.

La mayoría de aerogeneradores son de tres tipos:

Inducción de jaula de ardilla: precisan una multiplicadora debido a la gran diferencia de giro entre el generador y las palas. El estator está directamente conectado a la red. Son los más antiguos y se denominan de velocidad constante ya que el rotor apenas tolera variaciones de velocidad.

Generador de inducción de doble alimentación: el devanado del estator está directamente conectado a la red. Al devanado del rotor se le inyecta una corriente mediante electrónica de potencia. La frecuencia de la corriente inyectada en los devanados es variable permitiendo así la variación de velocidad en función de la velocidad del viento.

Generadores síncronos: las turbinas no tienen multiplicadora. Generador y red están desacoplados por electrónica de potencia. Los tamaños de los motores dependen de los fabricantes.

### **1.3.8 Controladores**

En función de la velocidad y dirección del viento, proporcionados por el anemómetro y la veleta, hacen girar al aerogenerador y determinan el ángulo de las palas. De esta forma y junto con la electrónica de potencia anteriormente comentada se busca encontrar la velocidad y la posición óptimas para la extracción de energía del viento.

### **1.3.9 Anemómetro**

Mide la velocidad del viento.

### **1.3.10 Veleta**

Mide la dirección del viento.

### **1.3.11 Góndola**

Estructura metálica que recubre a todo el conjunto de partes que componen el aerogenerador para evitar su deterioro.

### **1.3.12 Eje de alta velocidad**

En muchos casos de generación de energía eléctrica el eje conectado al aerogenerador gira a mayor velocidad que la velocidad de giro del rotor que gira con el viento.

### **1.3.13 Unidad de guía**

Encargada de permitir el giro de la góndola.

### **1.3.14 Motor guía**

Responsable del giro de la góndola.

### **1.3.15 Torre**

Eleva a la góndola a una altura suficiente para permitir el giro de las palas y para alcanzar velocidades de viento deseables, ya que cercano al suelo el viento es más lento y son más frecuentes las turbulencias. Soporta todo el peso y suele ser hueco para permitir el acceso. Entre los materiales más usados están el acero o el hormigón armado.

### 1.3 Modelos a realizar

Tras el estudio de la base de datos y la consiguiente limpieza de la muestra, que se detallará posteriormente, se decidió modelar el comportamiento normal de seis componentes esenciales en el funcionamiento del aerogenerador. Estos modelos son:

**1.3.1 Modelo de la Potencia:** Este es el modelo más sencillo de todos. Simula la potencia que el generador es capaz de producir a partir de la velocidad del viento incidente. Este modelo no proporciona una detección inmediata de una falta en el aerogenerador, pero siempre es conveniente tener una idea de la potencia que produce nuestra máquina en todo momento. Además, relacionar la potencia con la velocidad del viento, es especialmente interesante para el controlador encargado de determinar la velocidad de giro óptima de la máquina para la extracción de potencia. Puede utilizarse para asegurar el funcionamiento de los sensores de dicho control.

**1.3.2 Modelo rodamiento NDE:** se trata de uno de los rodamientos que unen el rotor al estator del aerogenerador Al igual que los anillos, el rodamiento NDE está sometido a mucha fricción y es interesante conocer su temperatura de funcionamiento para prevenir sus averías y poder sustituirlo a tiempo.

**1.3.3 Modelo rodamiento DE:** al igual que el modelo anterior, se busca simular el otro rodamiento que une el rotor con el estator del aerogenerador. Se trata de dos modelos de previsión de faltas directos.

**1.3.4 Modelo rodamiento alta multiplicadora:** se trata del rodamiento que une el rotor de las aspas de la turbina con el eje unido a la multiplicadora. Al igual que los modelos de los rodamientos del eje que une la multiplicadora con el aerogenerador, es un modelo de previsión de faltas directo. Este rodamiento también se encuentra afectado por mucho estrés mecánico y por altas temperaturas. Es conveniente conocer su temperatura para poder prevenir faltas y sustituirlo previamente.

**1.3.5 Modelo Pitchangle:** este modelo, más sofisticado que el modelo de la potencia anterior ya que tiene más de una entrada, permite conocer el ángulo que deben tener las aspas de cara a la dirección del viento incidente. Es especialmente importante para la optimización de la extracción de la energía. Al igual que el modelo anterior no es un sensor directo de faltas en el propio aerogenerador, sino que permite saber si los sensores funcionan correctamente para un mejor control y optimización de la potencia extraída.

**1.3.6 Modelo de los anillos SLD:** [5] son las escobillas del generador mediante las cuales las bobinas del estator se conectan a las bobinas en movimiento del rotor. Las escobillas experimentan una fricción constante y son uno de los elementos con una mayor inclinación a sufrir averías o deterioro del generador. Es por esto, que resulta especialmente interesante conocer su temperatura para predecir su comportamiento. A diferencia de los modelos anteriores, este es el primer modelo que sirve para una directa prevención de faltas en las partes del aerogenerador mencionadas.

Estos modelos han sido elegidos por ser los que presentan las anomalías más frecuentes en los aerogeneradores. Otros casos que se tuvieron en cuenta, pero finalmente no se modelaron debido a la falta de datos fueron:

- Temperatura de los arrollamientos del aerogenerador.
- Temperatura de los arrollamientos del transformador de salida.
- Temperatura del circuito de refrigeración de la multiplicadora.
- Sistema hidráulico.

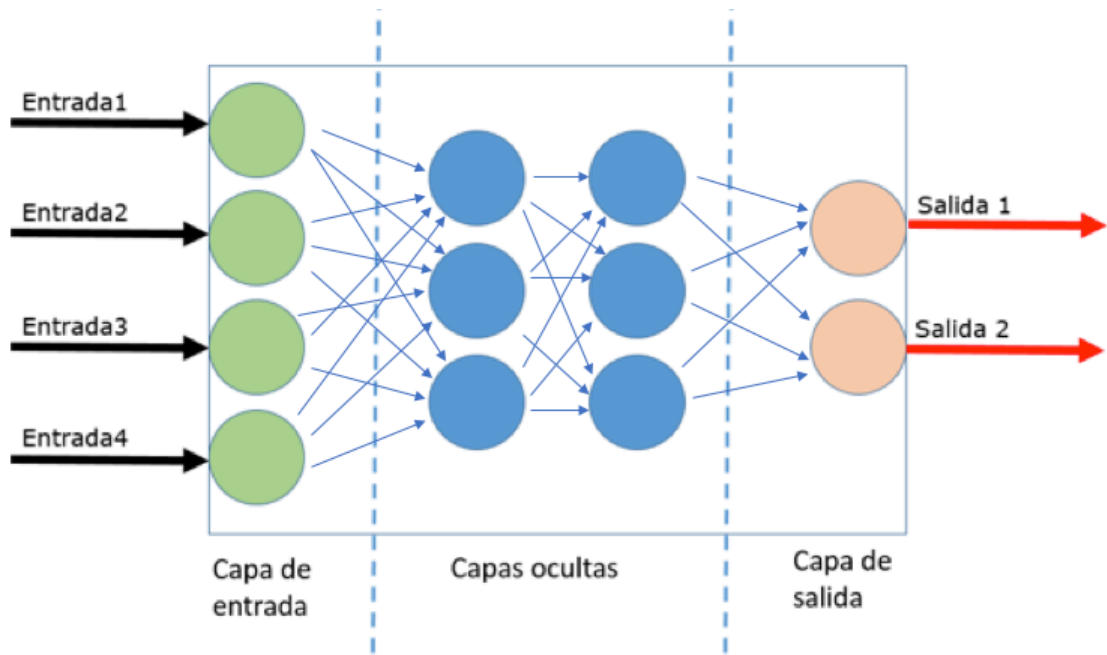
Como hemos mencionado anteriormente, el método elegido es la elaboración de los modelos de comportamiento normal expuestos en este apartado, basados en redes neuronales. Esta tecnología tiene una gran importancia en el mundo de la inteligencia artificial. A continuación, trataremos de dar una breve introducción a las redes neuronales.

## 1.4 Redes Neuronales

Una vez vistas las diferentes partes de un aerogenerador, los diferentes fallos que pueden ocurrir y la gravedad de estos fallos se llega fácilmente a la conclusión de la utilidad de un sistema preciso y completo de detección de faltas. En este proyecto, los modelos que caracterizan el comportamiento de los diferentes elementos del aerogenerador han sido realizados mediante redes neuronales en el entorno de Matlab y Simulink.

En la actualidad, uno de los campos más utilizados en la inteligencia artificial son las redes neuronales [2]. Son un tipo de modelos simplificados de las redes neuronales que forman nuestro cerebro, de ahí su nombre y topología. Las características que hacen singulares a estos modelos de inteligencia artificial son las siguientes [4]:

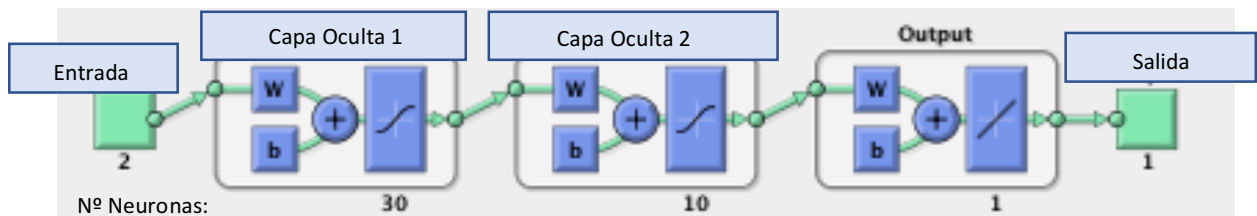
- Tienen **capacidad de aprendizaje**. Al contrario que otros programas informáticos que son codificaciones de las funciones que queremos que realicen los ordenadores, las redes neuronales son entrenadas a partir de la experiencia. En el caso de este trabajo, las redes neuronales, que simulan las diferentes partes del aerogenerador, son entrenadas con los datos históricos de un generador real. Los modelos son entrenados con una parte de los datos y, después, se valora su comportamiento con los datos que no se han utilizado en el entrenamiento.
- **Velocidad** de procesamiento de nuevas series de datos una vez concluido el entrenamiento. Una vez que se han ajustado los pesos de cada “neurona”, las nuevas series de datos se pueden simular con mucha celeridad.
- La **robustez** de la red en conjunto, ya que, si una parte de esta queda dañada, el resto de neuronas mantienen casi intactos los resultados.



**Fig 5. Esquema básico de un perceptrón multicapa.**

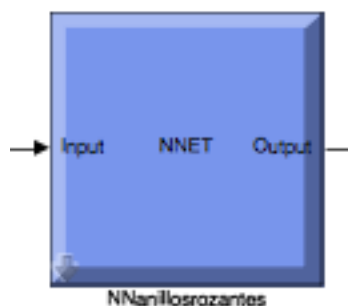
En la figura 5 podemos observar la sencillez y el funcionamiento de una red neuronal. Dependiendo del número de neuronas y las capas ocultas internas las redes neuronales pueden variar su tamaño. Las redes neuronales del tipo perceptrón multicapa son aproximadores funcionales universales. Esto les permite

En el entorno de Matlab y Simulink los esquemas que siguen las redes neuronales son los siguientes:



**Fig 6. Red Neuronal en el entorno de Matlab.**

Podemos ver que el esquema se asemeja mucho al anteriormente explicado. Podemos ver las entradas (en este caso dos), las diferentes capas (1 y 2) y los pesos empleados (que son los que se van ajustando con el entrenamiento:  $w$ ,  $b$ ) y, finalmente, una salida que también tiene sus pesos.



**Fig 7. Bloque red neuronal en el entorno de Simulink**

## **2 Filtrado, selección de datos y retardo de datos.**

### **2.1 Filtrado:**

Los datos usados en este proyecto [5], corresponden a un generador tripala horizontal de 2 MW de potencia. Tenemos disponibles 32 variables con 291.312 datos por variable, lo que hacen un total de 9.321.984 datos. Son datos del 15 de diciembre de 2008 al 7 de septiembre de 2014.

En primer lugar, una vez recibidos los datos se procedió a un primer filtrado para ver los errores en las diferentes variables, entre los que destacaban medidas imposibles como potencias negativas.

### **2.2 Selección de datos:**

Previo a construir cualquier modelo de comportamiento normal, es importante decidir las variables de entrada que mejor puedan explicar la variable deseada. El proceso seguido en este proyecto para la selección de las variables de cada red neuronal tiene dos partes. Primero, un razonamiento sobre aquellas variables con mayor posibilidad de tener una intensa relación con la variable a simular. Un ejemplo, en el modelo de la potencia, una de las variables que tiene más posibilidades de tener una intensa relación con la potencia es la velocidad del viento. Y segundo, con la ayuda de la función “relieff” de Matlab, comprobar matemáticamente que las suposiciones hechas anteriormente son correctas. Además, esta función permite distinguir de entre las variables que tienen relación con la salida, cuáles son las que tienen relaciones más fuertes.

### **2.3 Retardo:**

Se planteó la hipótesis de que hubiera casos en los que una variable tuviera más relación con otra con una diferencia temporal. Por ejemplo, si la temperatura exterior era en un momento de 25 °C afectaría con media hora o más tiempo de retraso a la temperatura de los rodamientos. Para determinar esto en cada red neuronal se usó la función “crosscorr” de Matlab, pero se llegó a la conclusión de que los retrasos de tiempo eran inferiores a los diez minutos que hay entre medida y medida o frecuencia de muestreo.

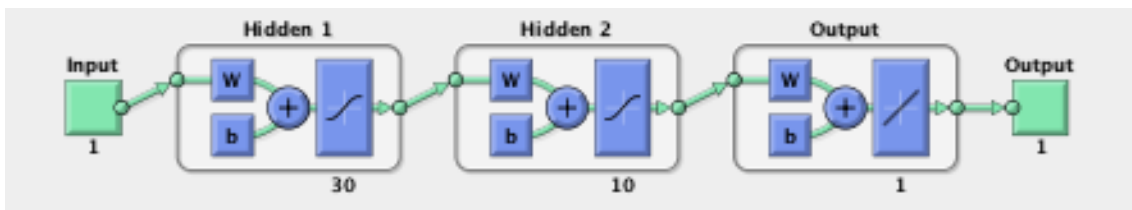


### 3 Estudio de los modelos realizados

#### 3.1 Modelo 1. Potencia esperada.

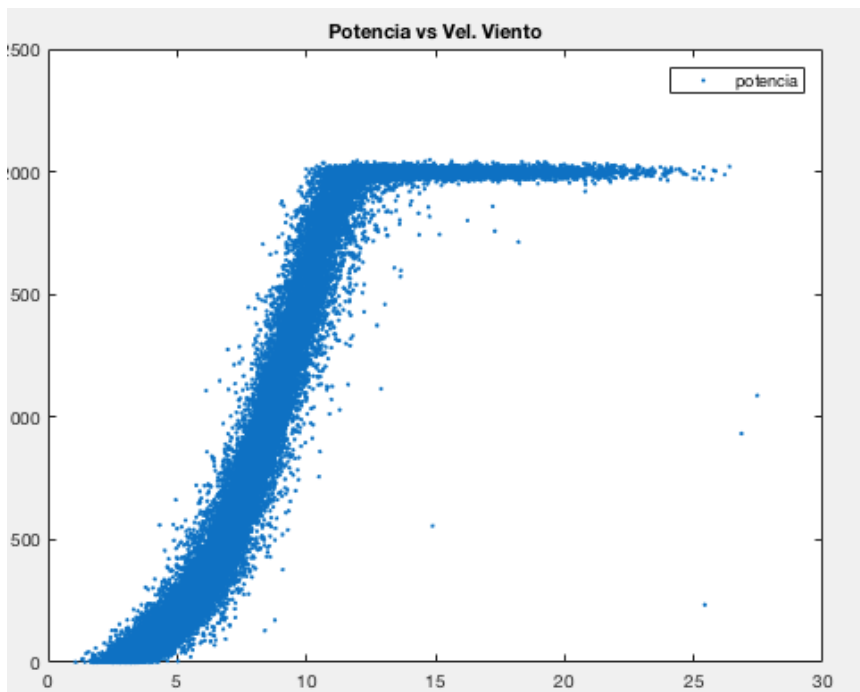
Se trata del modelo más sencillo de todos los realizados. Como hemos mencionado anteriormente, se trata de modelar la relación entre la potencia generada y la velocidad del viento.

Una vez filtrados y seleccionados los datos de entradas que mejor ajusten el modelo, se concluyó que con la velocidad del viento sería suficiente. Al tener solo un input es la red neuronal más rápida y sencilla de simular. Se trata de una red neuronal de doble capa. El esquema es el siguiente:



*Fig 8. Bloque red neuronal de la potencia en el entorno de Matlab.*

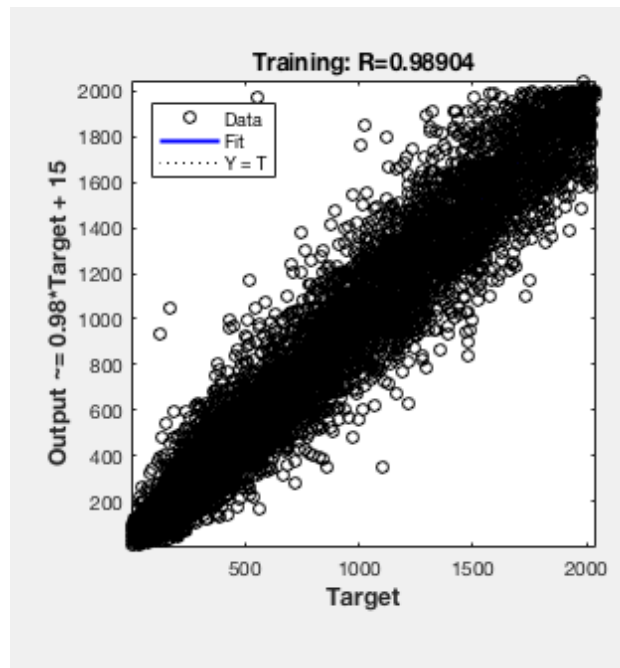
Como en esta red neuronal solo hay un input, antes del entrenamiento representamos los valores de la potencia en función de la velocidad del viento:



*Fig 9. Potencia vs Velocidad del Viento.*

Podemos observar claramente como la potencia sigue una recta tipo. Tenemos un límite para que el aerogenerador entre en funcionamiento, un límite de velocidad del viento a

partir del cual la potencia extraída no sigue aumentando y un límite a partir del que el aerogenerador no funciona por motivos de seguridad.



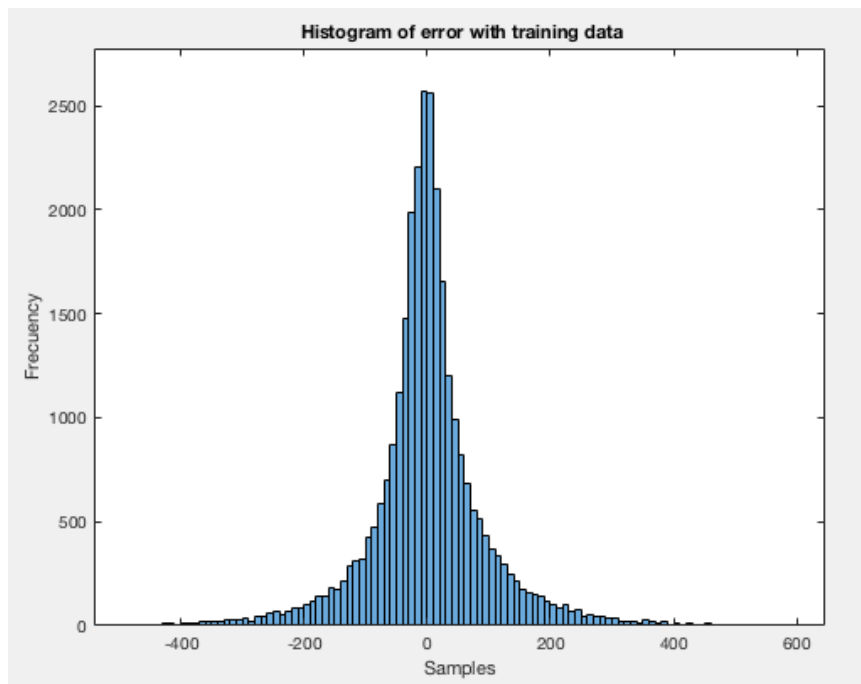
*Fig 10. Recta regresión RN Potencia.*

La recta de regresión con una  $R \approx 0,97$  nos muestra que la potencia tiene una intensa relación con la velocidad del viento y que el modelo es bueno.

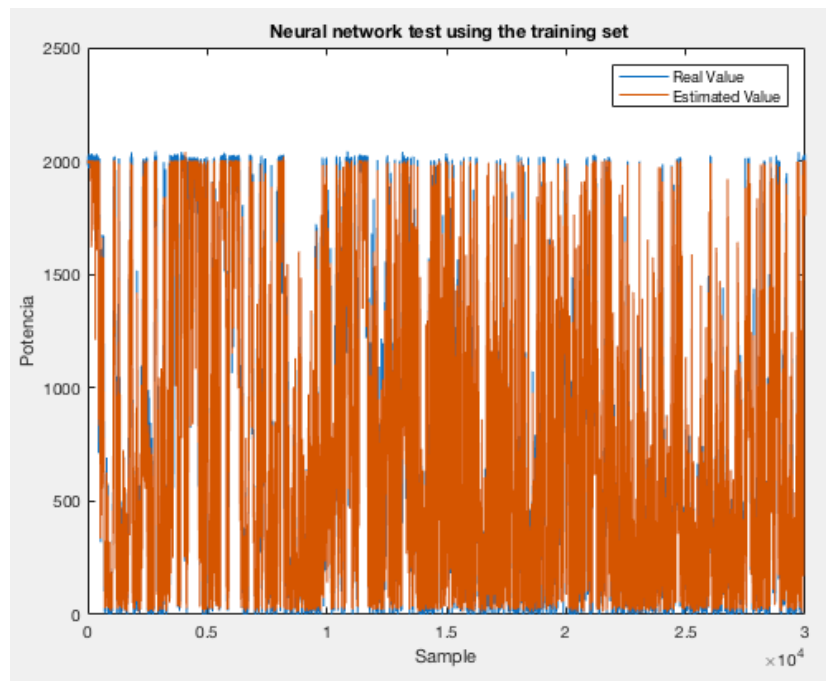
### **Estudio del error**

Como en todo modelo es importante conocer el error al que nos atenemos. Observamos los errores con los datos de entrenamiento (año 2009) y con los datos de los años siguientes. Para continuar con el estudio de la red neuronal también representamos los valores reales frente a los simulados para observar si existen comportamientos anormales.

**Datos 2009 (entrenamiento):**

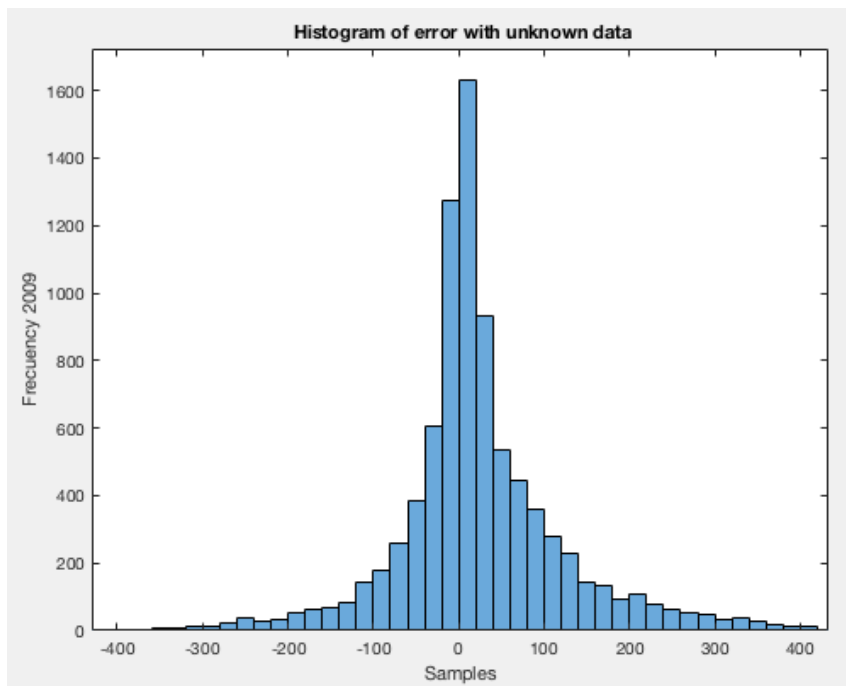


*Fig 11. Histograma error.*

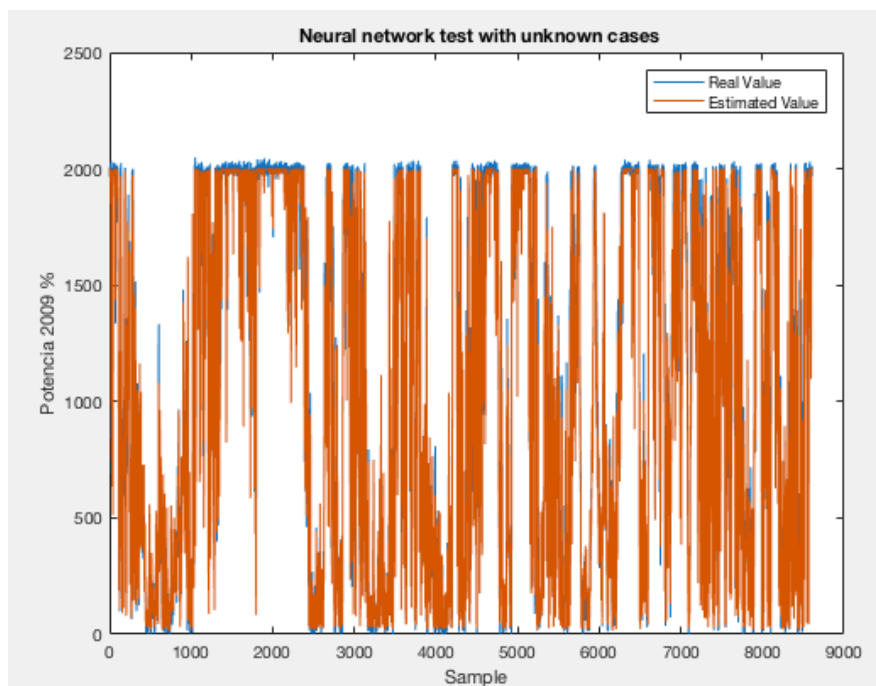


*Fig 12. Comparación datos reales/estimados.*

**Datos 2009 (simulación):**

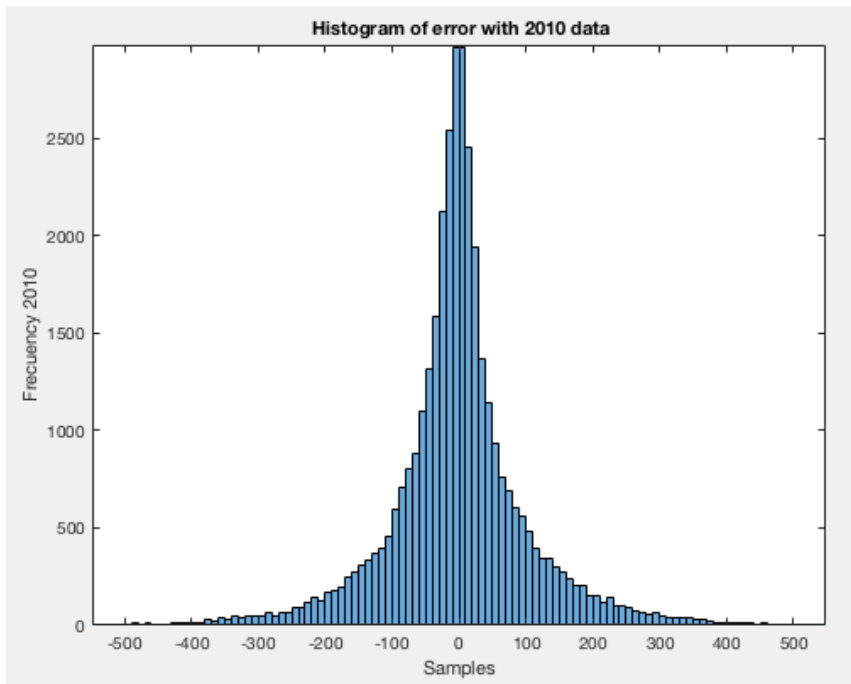


*Fig 13. Histograma error.*

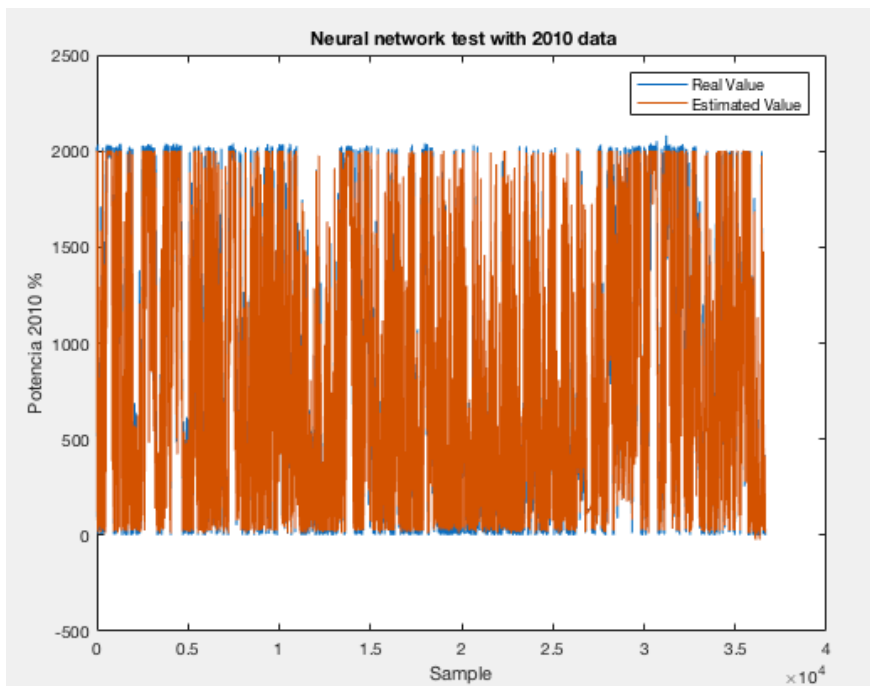


*Fig 14. Comparación datos reales/estimados.*

**Datos 2010:**

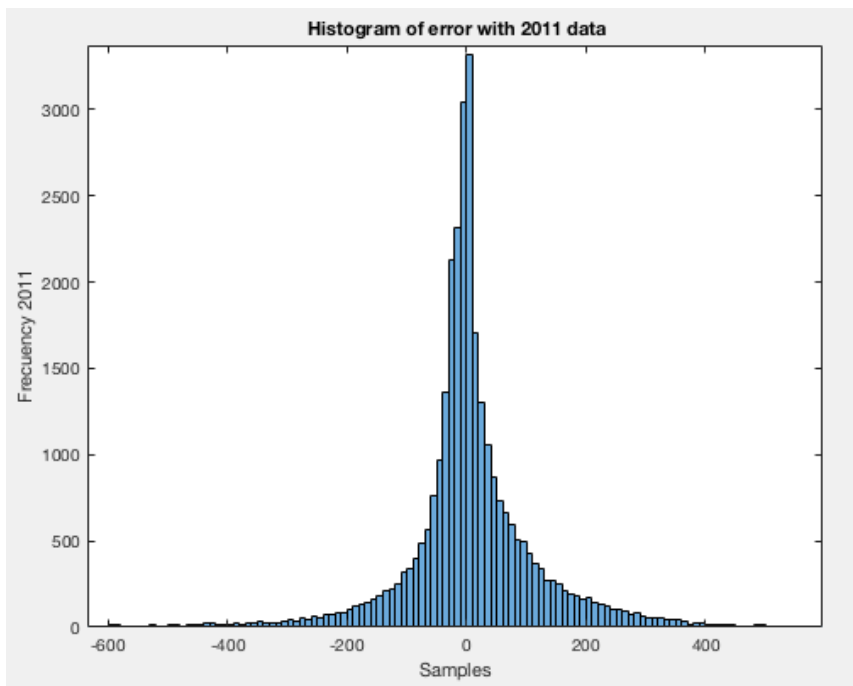


*Fig 15. Histograma error.*

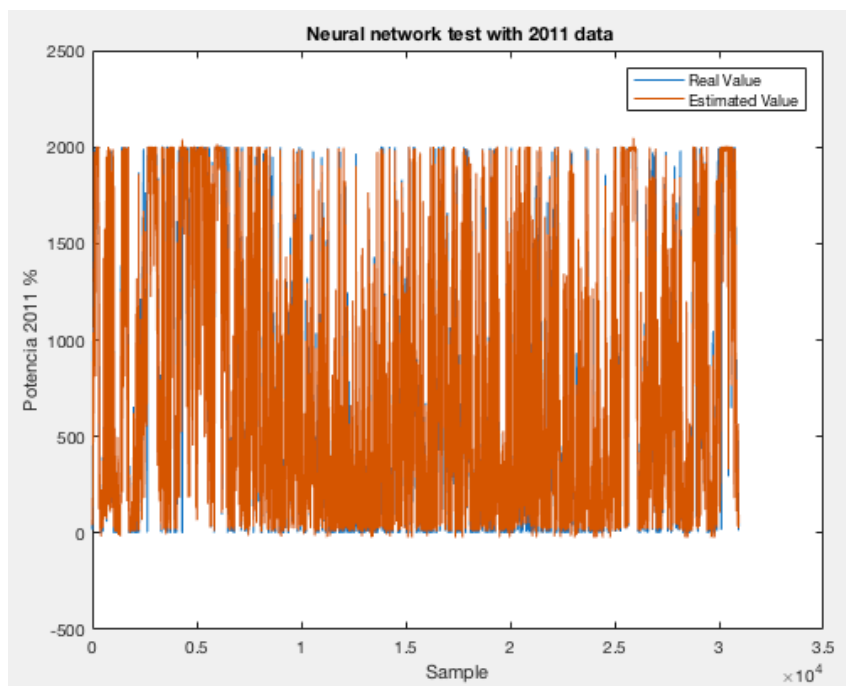


*Fig 16. Comparación datos reales/estimados.*

**Datos 2011:**

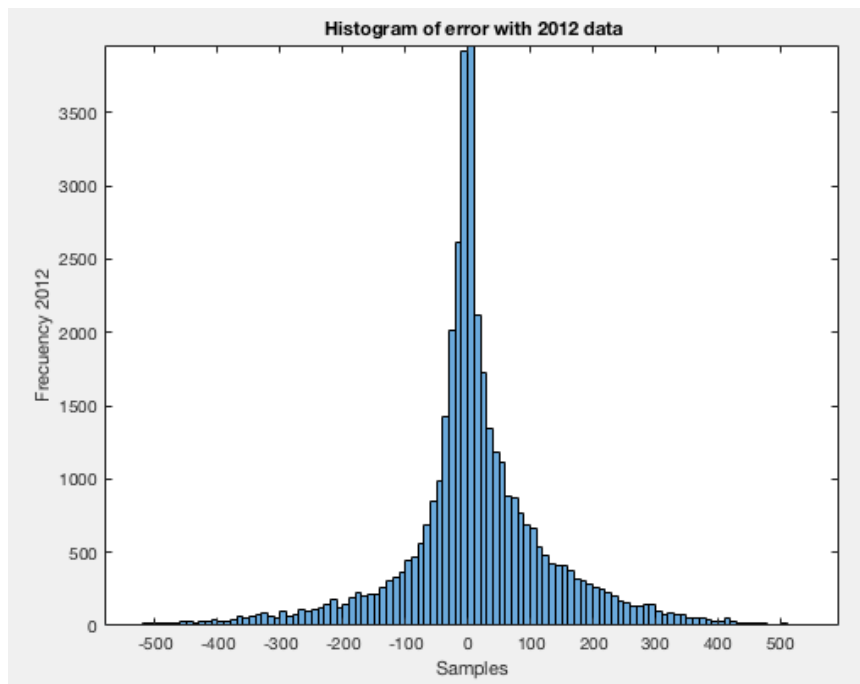


*Fig 17. Histograma error.*

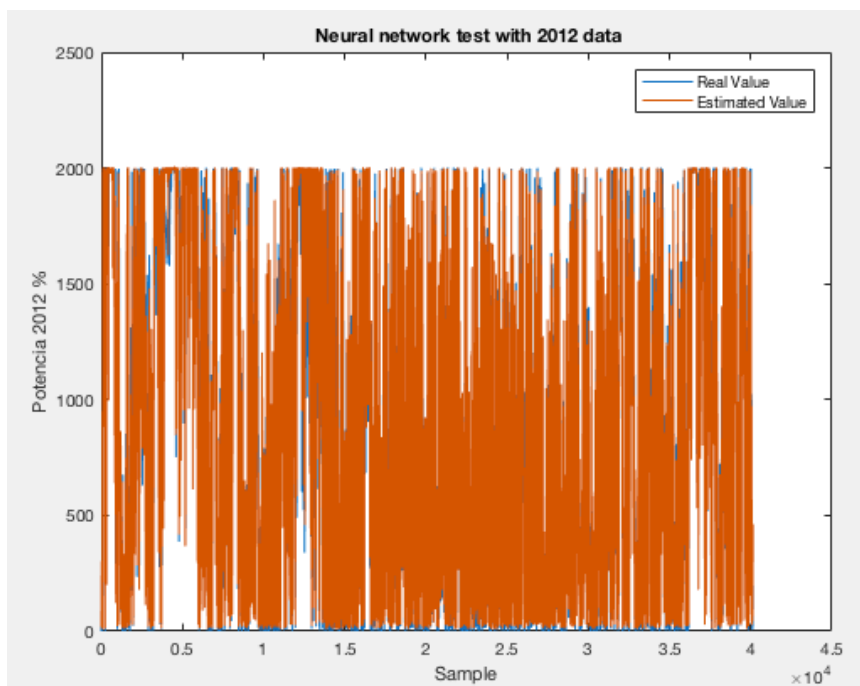


*Fig 18. Comparación datos reales/estimados*

**Datos 2012:**

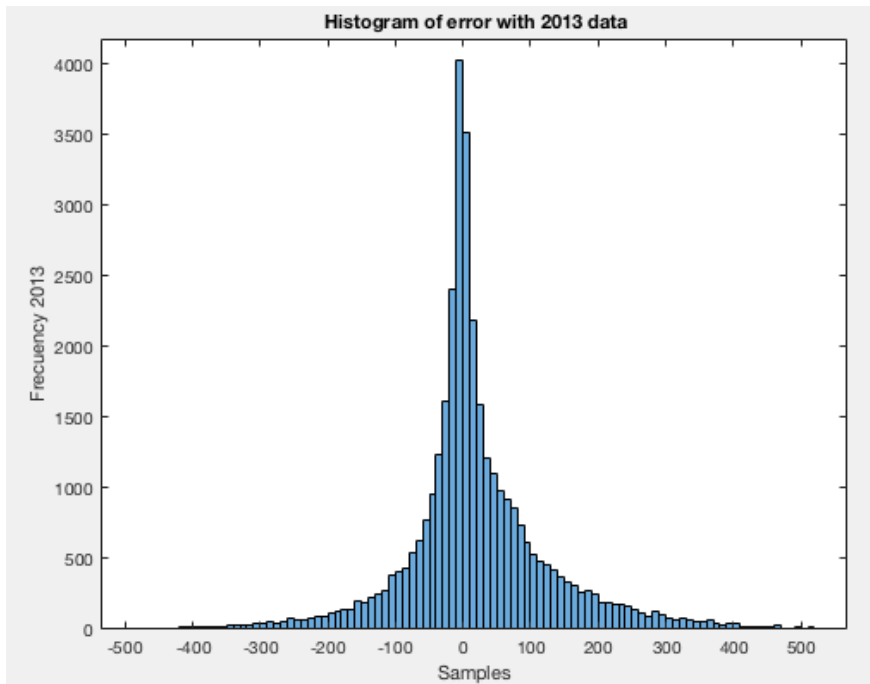


*Fig 19. Histograma error.*

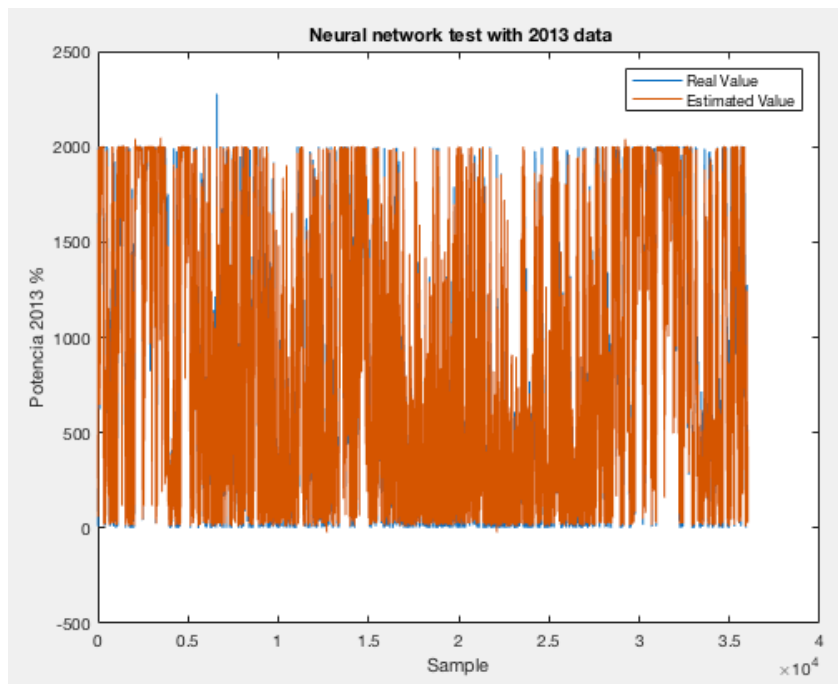


*Fig 20. Comparación datos reales/estimados.*

**Datos 2013:**

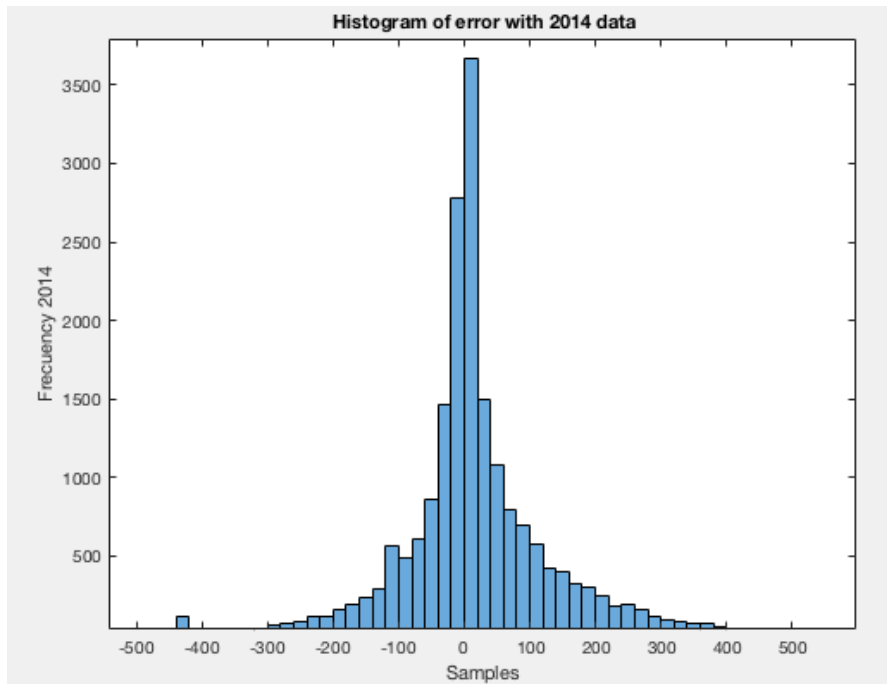


*Fig 21. Histograma error.*

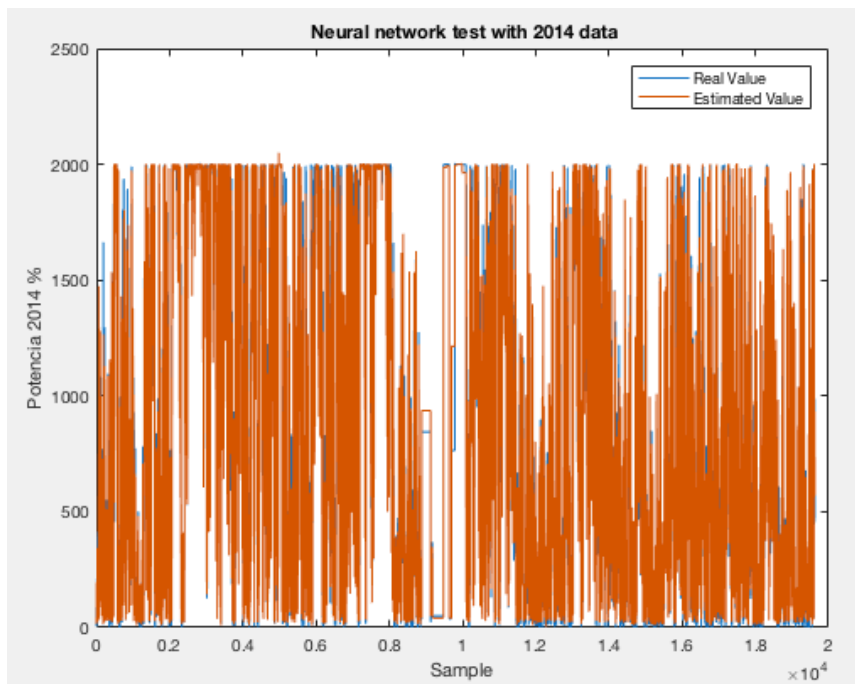


*Fig 21. Comparación datos reales/estimados.*

## Datos 2014:



*Fig 22. Histograma error.*



*Fig 23. Comparación datos reales/estimados.*

### **Comentario los histogramas y gráficas de comparación:**

Podemos observar como todos los histogramas de errores de los diferentes años están centrados en torno al cero. Los errores disminuyen al alejarse del error 0, con apenas valores en los extremos del histograma. Todos tienen una forma parecida, lo que quiere

decir que las distribuciones en los años no tienen grandes variaciones y que la red neuronal se puede usar para simular los valores de la potencia en los diferentes años. En cuanto a la distribución de los datos vemos como no existen potencias negativas y la máquina no supera los 2MW que es su potencia nominal.

Distribuciones de los errores por año:

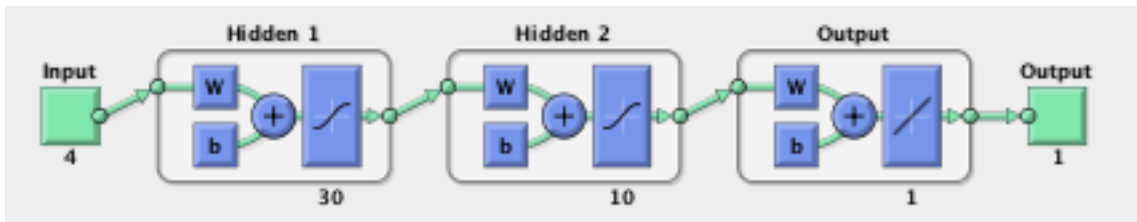
<b>Año</b>	<b>Media</b>	<b>Desviación</b>
<b>2009</b>	1,789	101,253
<b>2010</b>	0,0236	109,037
<b>2011</b>	6,697	131,472
<b>2012</b>	13,694	131,318
<b>2013</b>	19,360	123,887
<b>2014</b>	16,248	131,792

Observamos como las medias y también las desviaciones (aunque en menor medida) de los primeros años (2009, 2010, 2011) son claramente más bajas que los años posteriores. Este cambio es debido a que los comportamientos de los diferentes componentes del aerogenerador se modifican a medida que la máquina va envejeciendo.

### 3.2 Modelo 2. Red Neuronal Rodamiento NDE

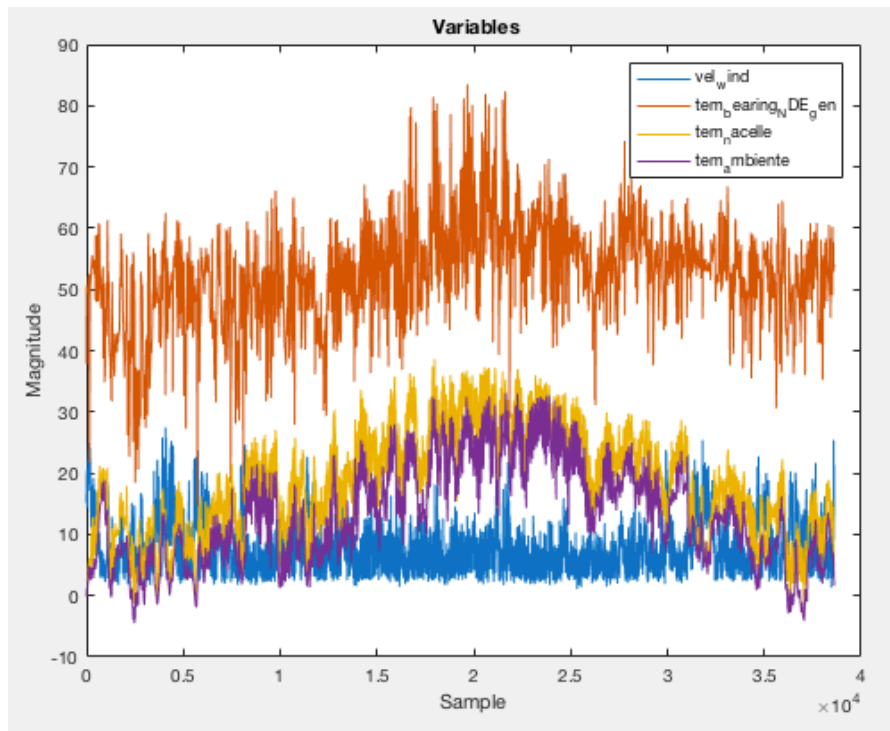
Como hemos mencionado anteriormente, se trata de simular la temperatura del rodamiento DE del aerogenerador en función de la potencia extraída, la velocidad del viento, la temperatura de la góndola y la temperatura ambiente.

Una vez filtrados y seleccionados los datos como inputs a la red neuronal, se llegó a la conclusión de que las variables que mejor servían para ajustar la red neuronal serían las anteriormente mencionadas. Es verdad que el tener cuatro inputs puede hacer más lenta la simulación, pero se gana en precisión. Se trata de una red neuronal de doble capa. El esquema es el siguiente:



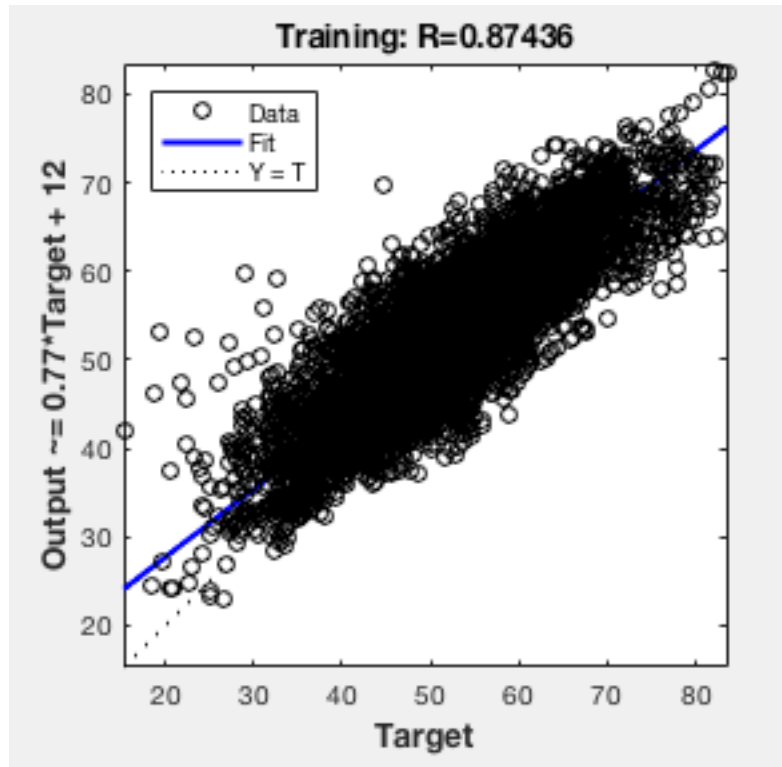
*Fig 24. Bloque red neuronal de la temperatura del rodamiento NDE en el entorno de Matlab.*

Previo a la simulación representamos los valores de los inputs a la red neuronal en función del tiempo:



*Fig 25. Inputs reales de la red neuronal.*

Podemos observar como tanto la temperatura ambiente como la de la góndola aumentan claramente durante los meses de verano. Y también una pequeña elevación de la temperatura del rodamiento durante ese mismo periodo de tiempo.



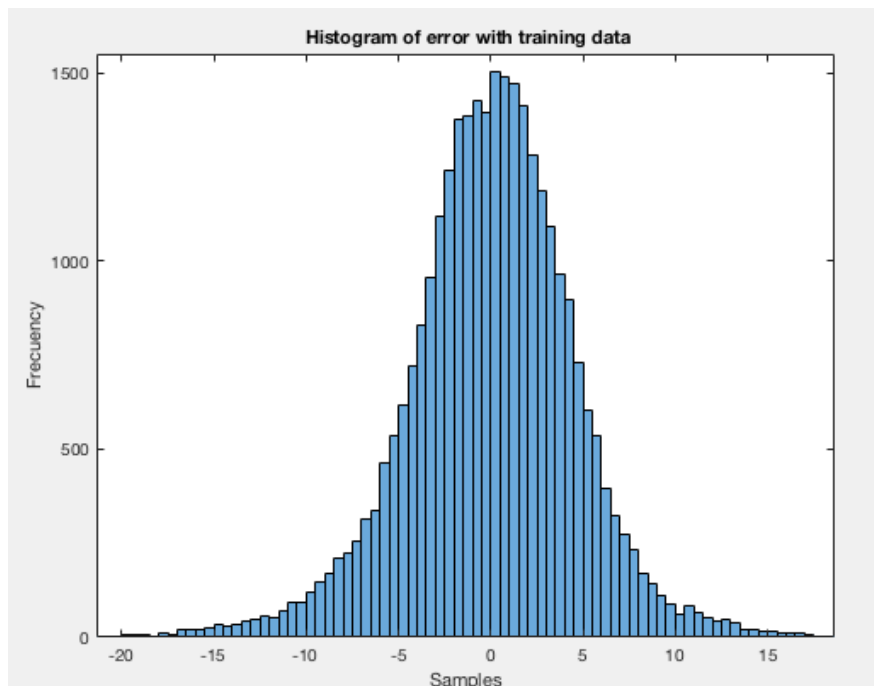
*Fig 26. Recta regresión RN Temperatura Rodamiento NDE.*

La recta de regresión con una  $R \approx 0,87$  nos muestra que la temperatura del rodamiento NDE tiene una buena relación con los inputs utilizados, a pesar de que existen puntos que se salen de la recta general.

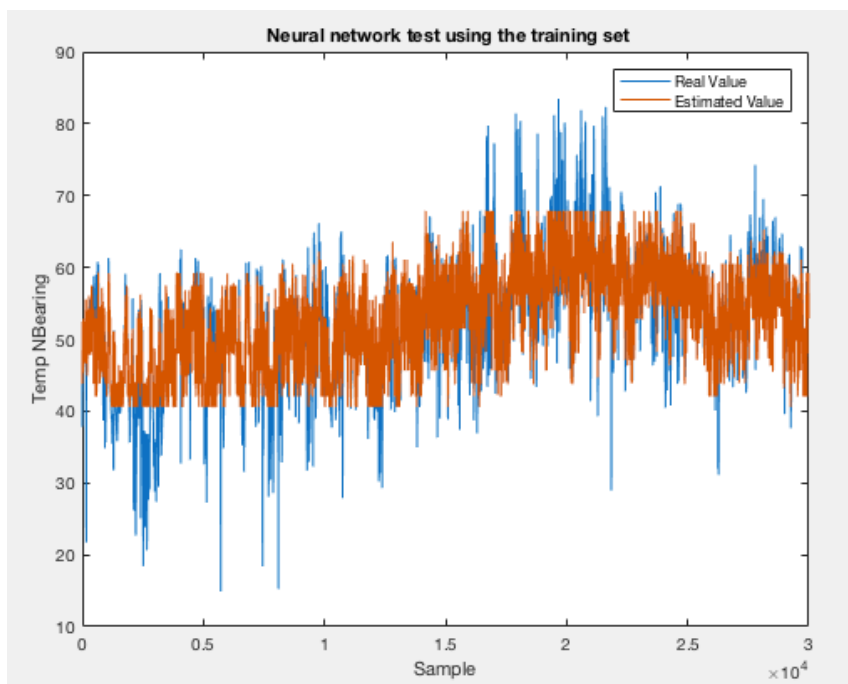
### **Estudio del error**

Como en todo modelo es importante conocer el error al que nos atenemos. Observamos los errores con los datos de entrenamiento (año 2009) y con los datos de los años siguientes. Para continuar con el estudio de la red neuronal, también representamos los valores reales frente a los simulados para observar si existen comportamientos anormales.

**Datos 2009 (entrenamiento):**

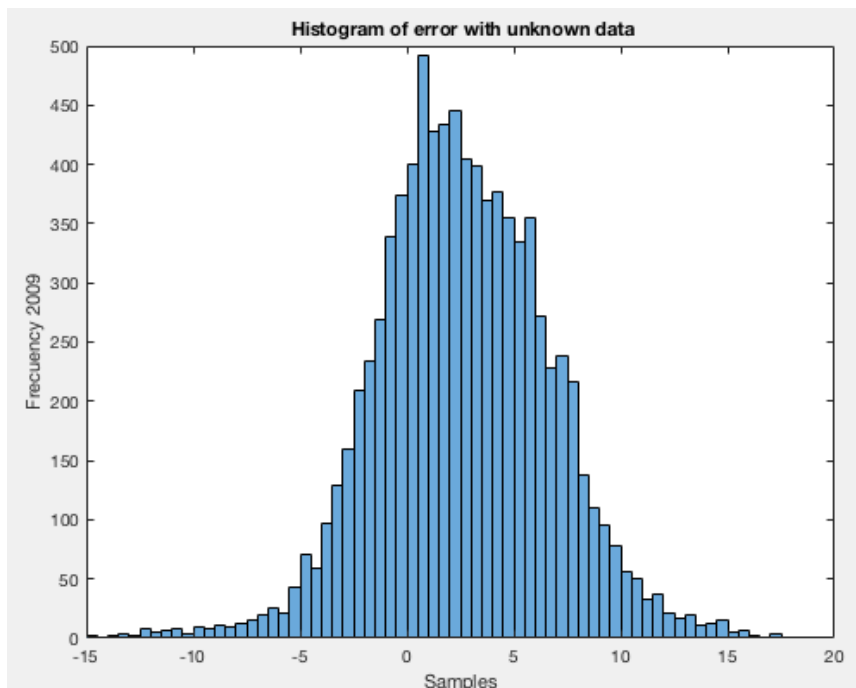


*Fig 27. Histograma error.*

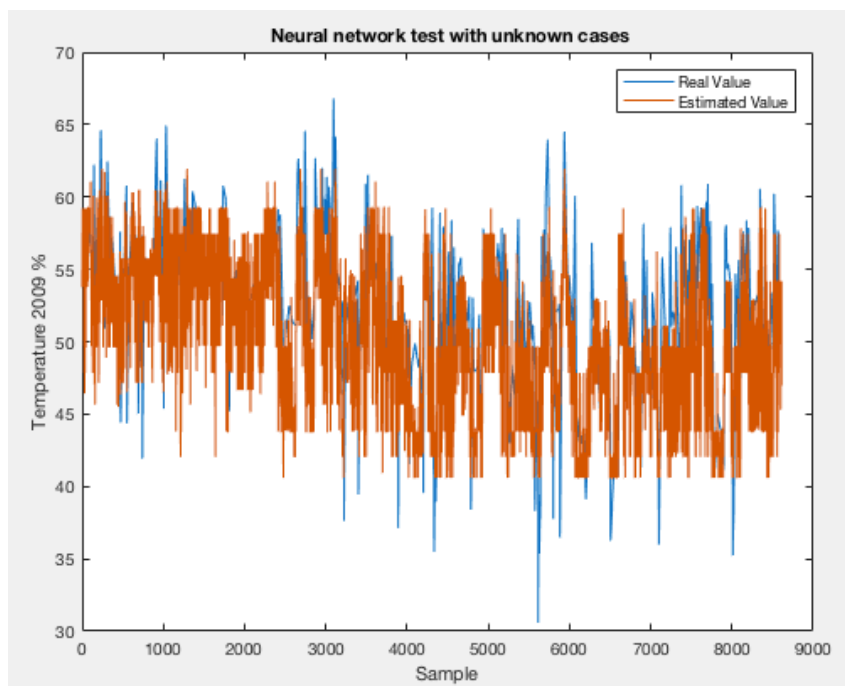


*Fig 28. Comparación datos reales/estimados.*

**Datos 2009 (simulación):**

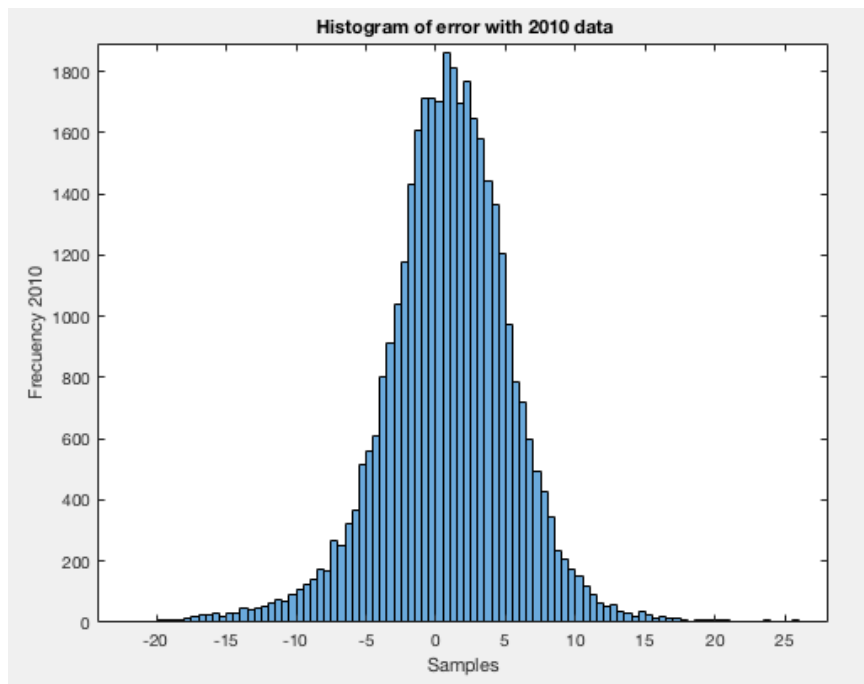


*Fig 29. Histograma error.*

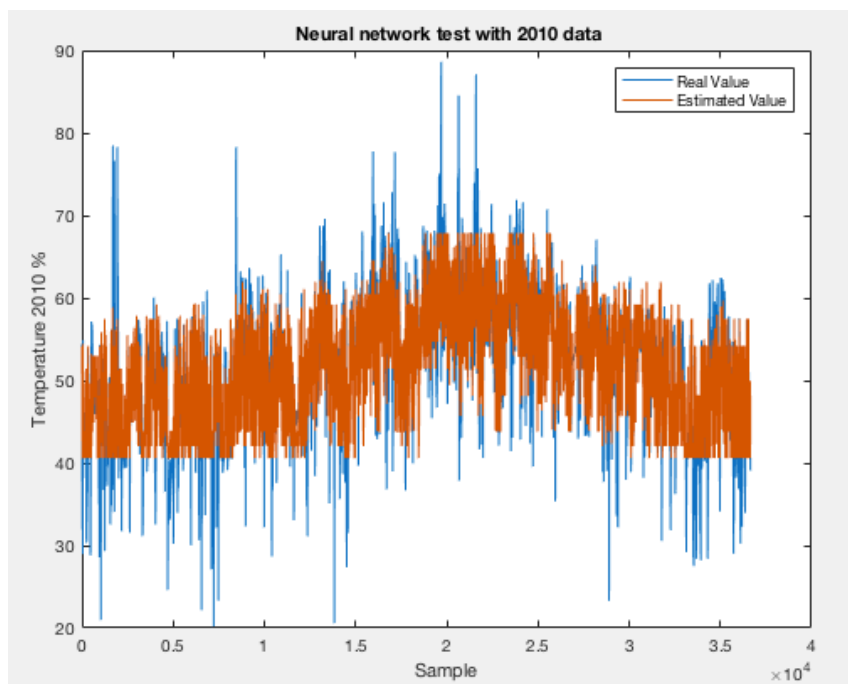


*Fig 30. Comparación datos reales/estimados.*

**Datos 2010:**

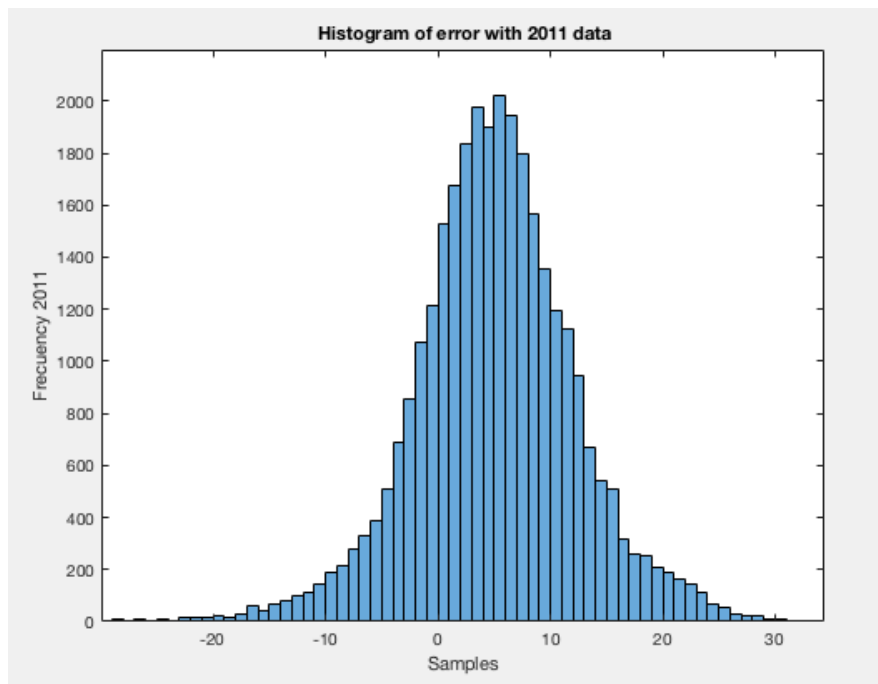


*Fig 31. Histograma error.*

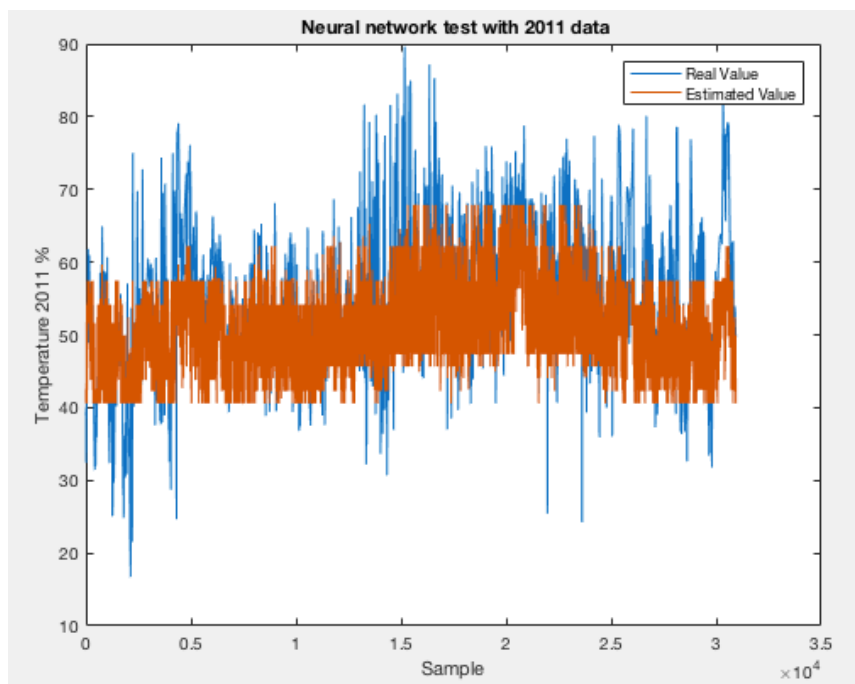


*Fig 32. Comparación datos reales/estimados.*

**Datos 2011:**

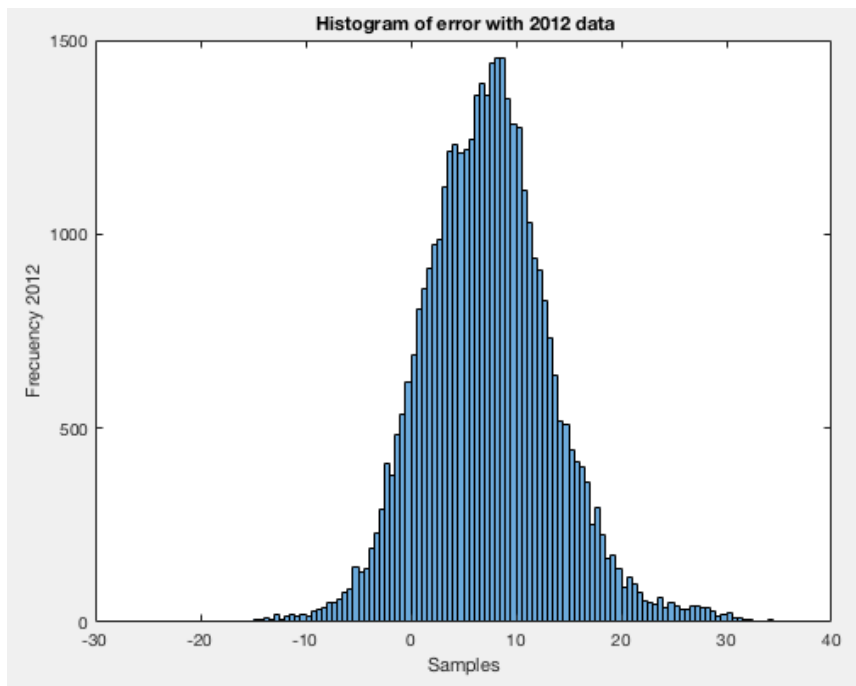


*Fig 33. Histograma error.*

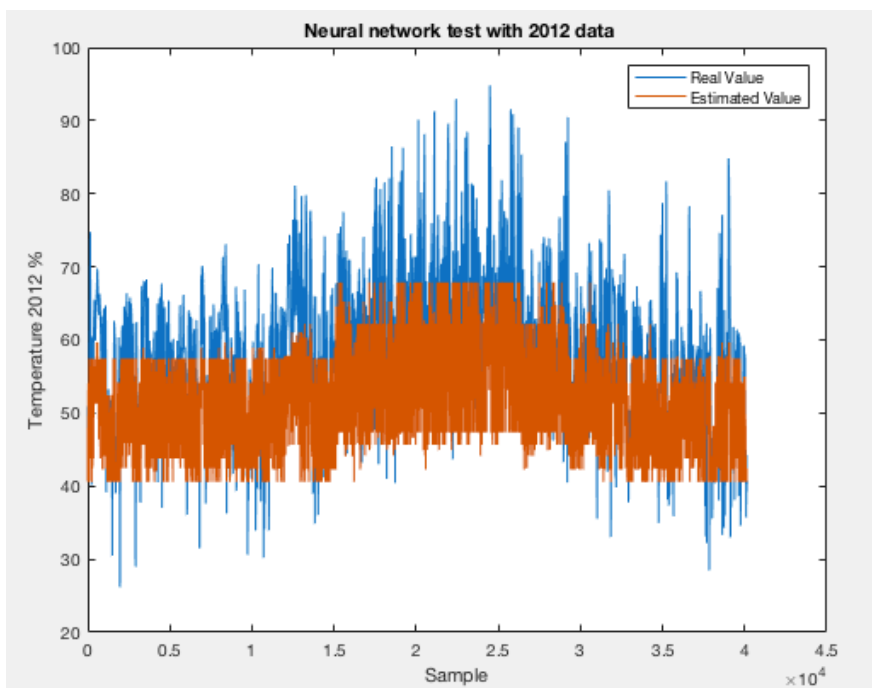


*Fig 34. Comparación datos reales/estimados.*

**Datos 2012:**

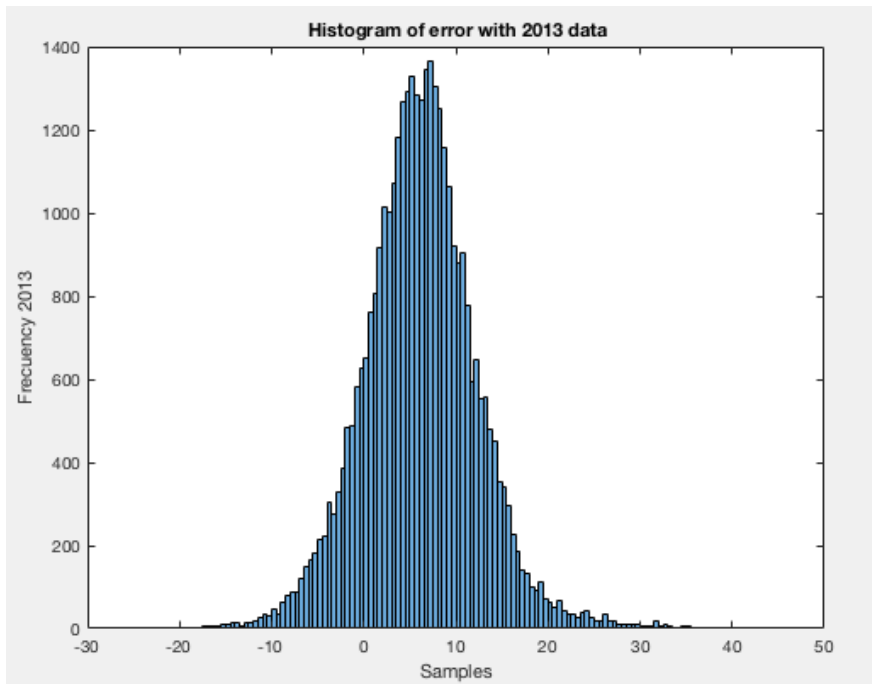


*Fig 35. Histograma error.*

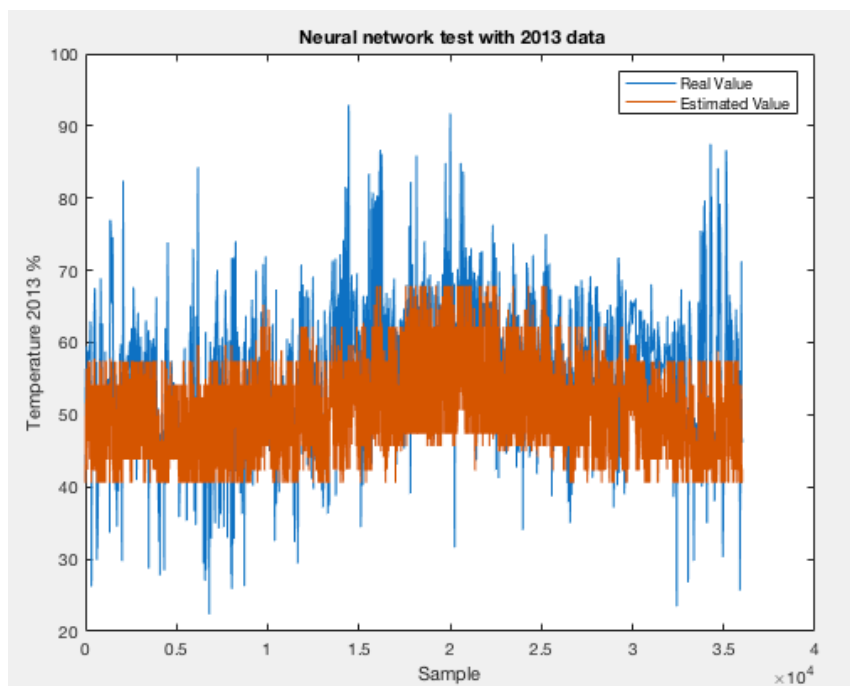


*Fig 36. Comparación datos reales/estimados.*

**Datos 2013:**

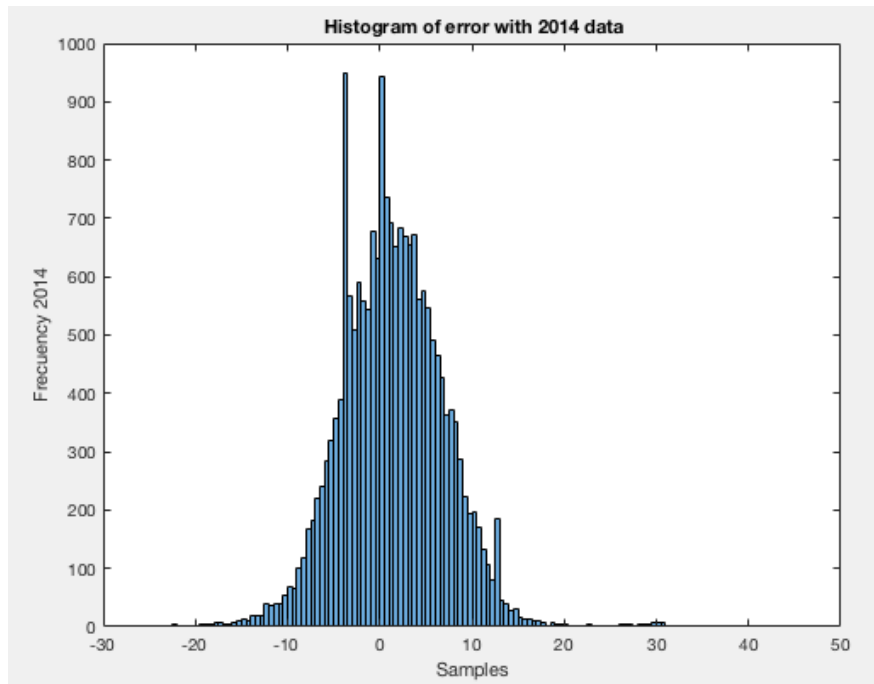


*Fig 37. Histograma error.*

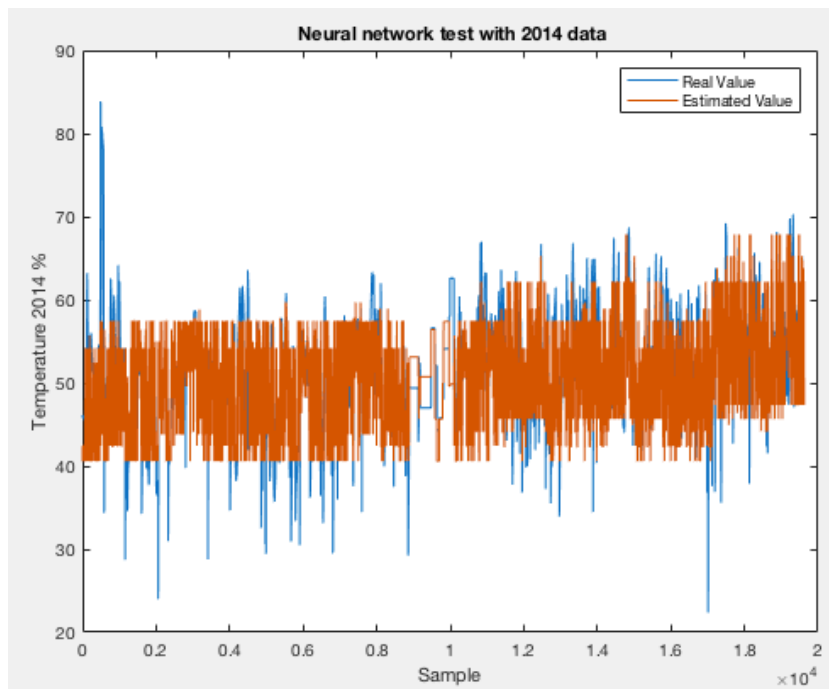


*Fig 38. Comparación datos reales/estimados.*

Datos 2014:



*Fig 39. Histograma error.*



*Fig 40. Comparación datos reales/estimados.*

### **Comentario los histogramas y gráficas de comparación:**

Podemos ver como en el caso del año 2014 los datos no son tan completos y tuvieron más errores que en los otros años, por lo que de los meses del medio del año tenemos menos información. Esto es importante pues son como hemos visto en la distribución de los datos aquellos en los que la temperatura exterior es más alta, causando que la temperatura de los rodamientos sea más alta independientemente de su funcionamiento.

Los histogramas de error están ligeramente desplazados a la derecha del cero. Esto quiere decir que nuestra red neuronal predice, de media, temperaturas por encima de las temperaturas reales. Este modelo es especialmente cauto ya que por norma general predecirá temperaturas superiores a las reales.

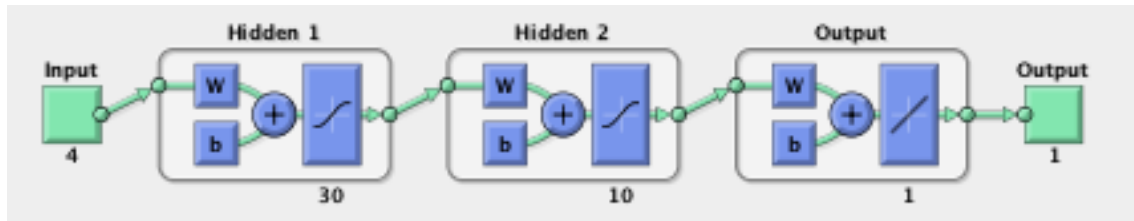
Distribuciones de los errores por año:

<b>Año</b>	<b>Media</b>	<b>Desviación</b>
<b>2009</b>	-0,0125	3,8459
<b>2010</b>	-0,0526	6,4410
<b>2011</b>	-0,05198	10,0154
<b>2012</b>	0,5182	10,2941
<b>2013</b>	-0,8599	10,4636
<b>2014</b>	-5,0195	10,2817

Observamos como las medias y las desviaciones aumentan ligeramente cada año, siendo claramente menores en los primeros años. Este cambio es debido a que los comportamientos de los diferentes componentes del aerogenerador se modifican a medida que la máquina va envejeciendo. Este comportamiento es similar al de la red neuronal anterior (Potencia).

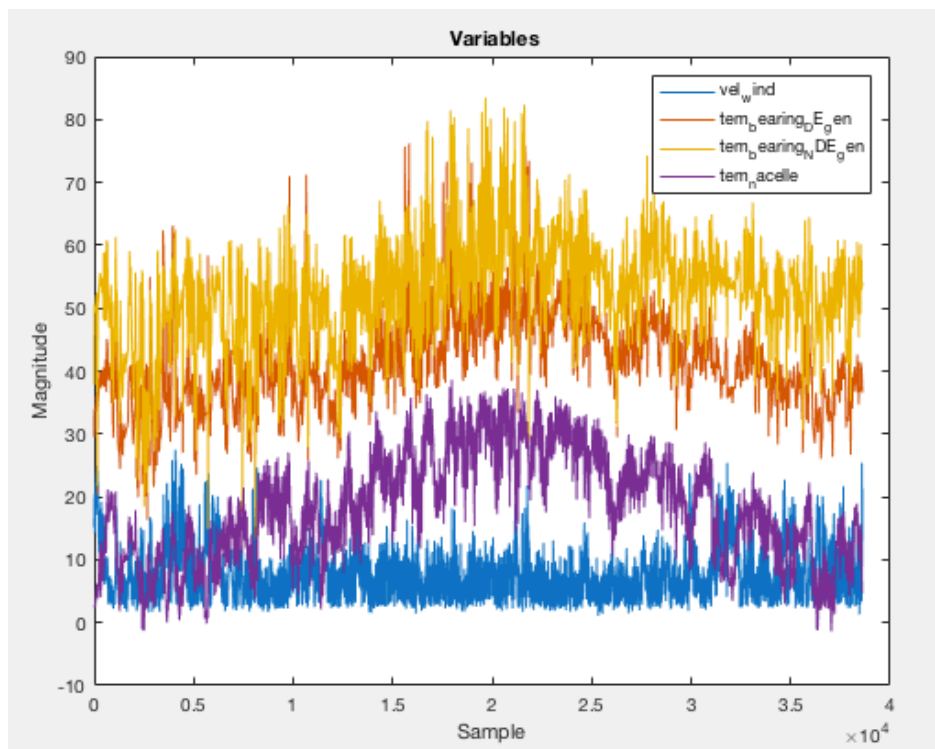
### 3.3 Modelo 3. Red Neuronal Rodamiento DE.

Esta red neuronal tiene los mismos outputs que la anterior y la misma funcionalidad. Simula la temperatura del otro rodamiento que sujeta el generador.



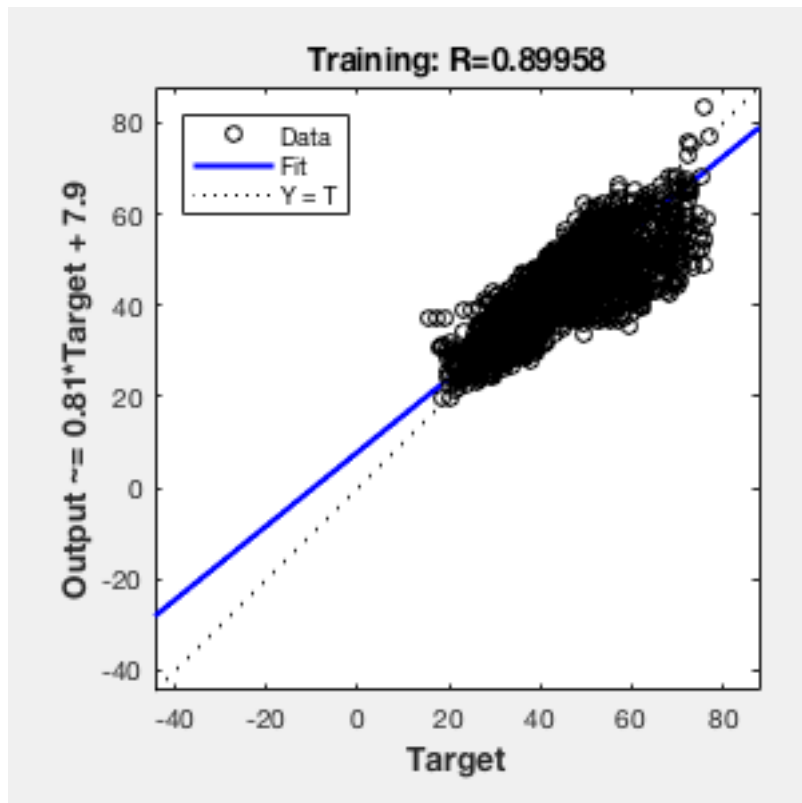
*Fig 41. Bloque red neuronal de la temperatura del rodamiento DE en el entorno de Matlab.*

Previo a la simulación representamos los valores de los inputs en función del tiempo:



*Fig 42. Inputs reales de la red neuronal.*

La gráfica tiene la misma subida de temperaturas en los meses de verano.



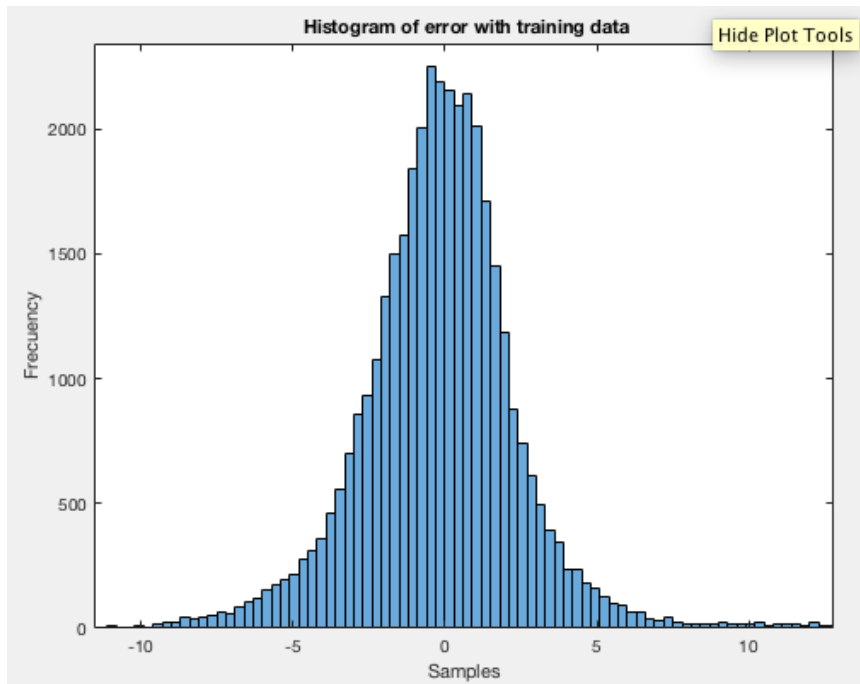
*Fig 43. Recta regresión RN Temperatura Rodamiento DE.*

La recta de regresión con una  $R \approx 0,90$  nos muestra que la temperatura tiene una buena relación con la velocidad del viento.

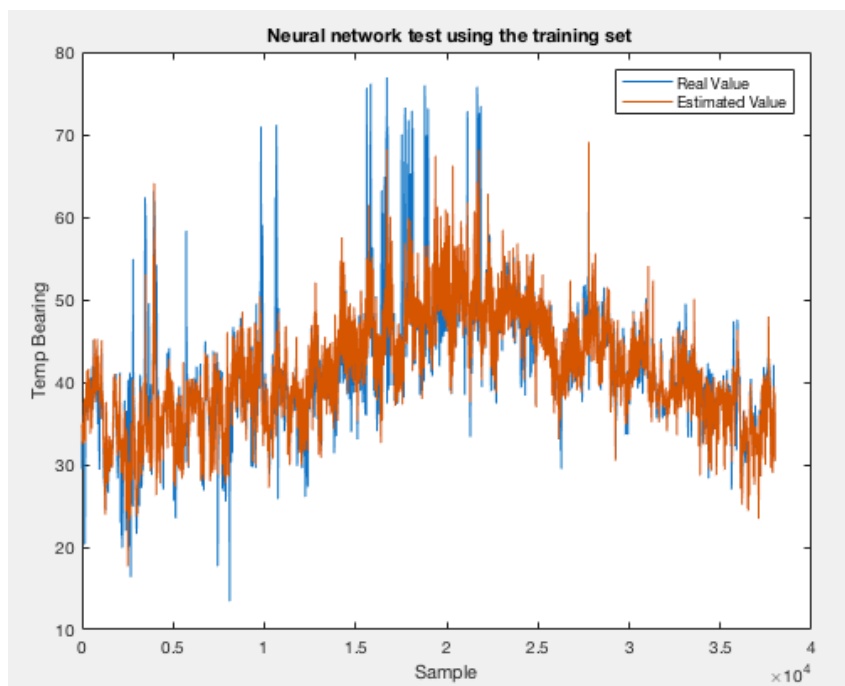
### **Estudio del error**

Como en todo modelo es importante conocer el error al que nos atenemos. Observamos los errores con los datos de entrenamiento (año 2009) y con los datos de los años siguientes. Para continuar con el estudio de la red neuronal también representamos los valores reales frente a los simulados para observar si existen comportamientos anormales.

Datos 2009 (entrenamiento):

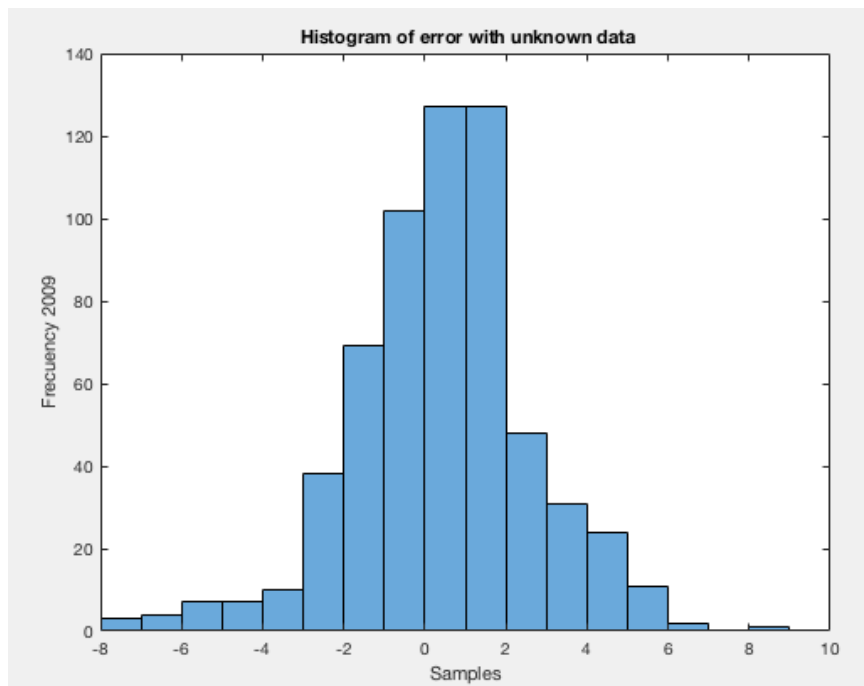


*Fig 44. Histograma error.*

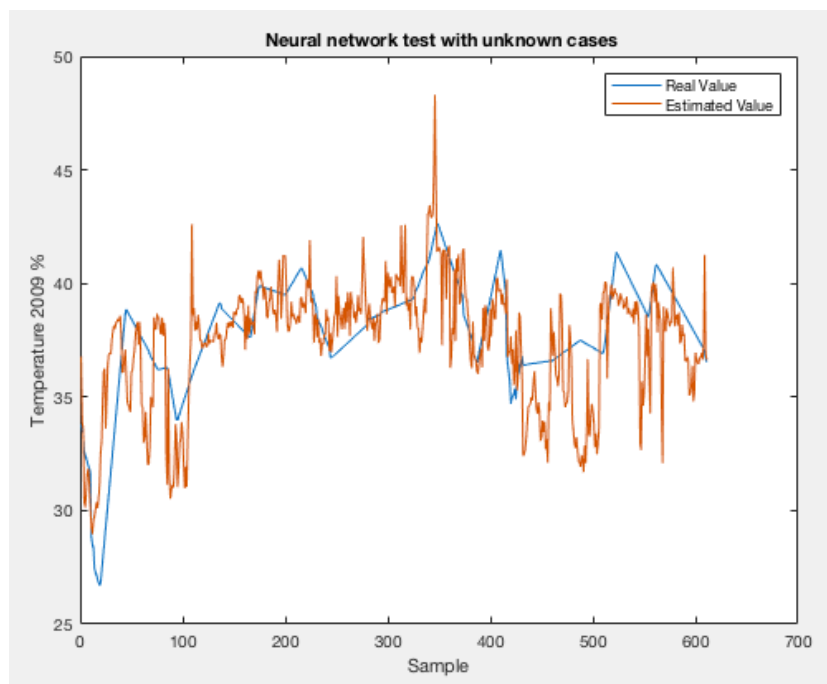


*Fig 45. Comparación datos reales/estimados.*

### Datos 2009 (simulación):



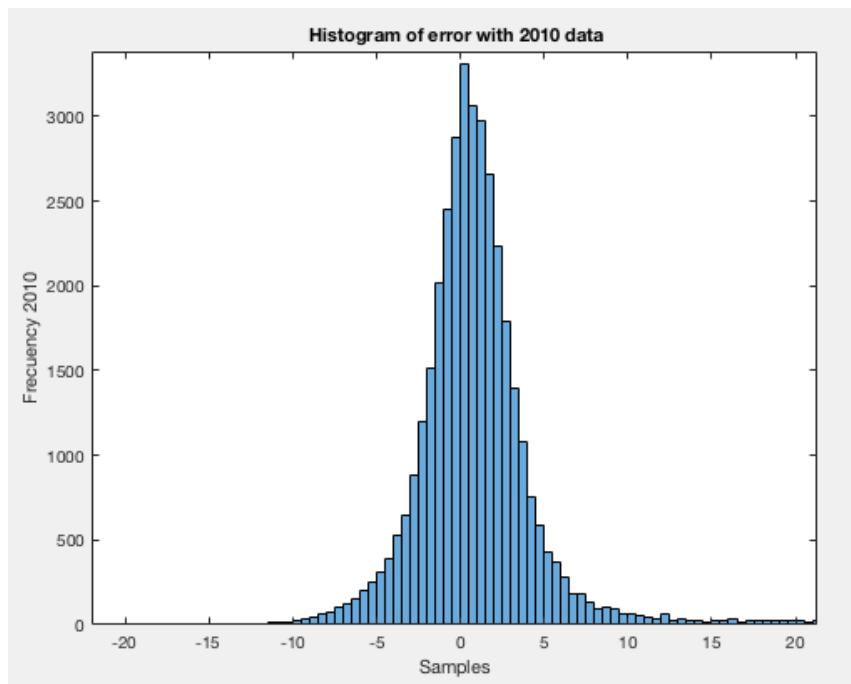
*Fig 46. Histograma error.*



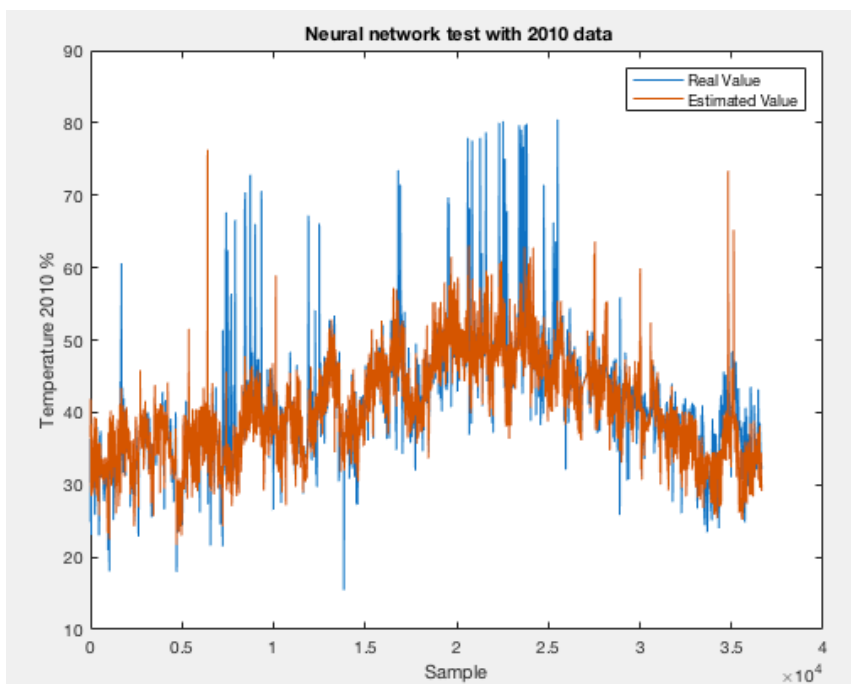
*Fig 47. Comparación datos reales/estimados.*

En este caso se eligió una diferente estrategia de entrenamiento distribuyendo los datos en entrenamiento, comprobación y simulación (90%/5%/5%). Se probó a entrenar la red neuronal con más datos del año de entrenamiento (2009). Y da mejores resultados que con la configuración habitual (60%20%20%).

**Datos 2010:**

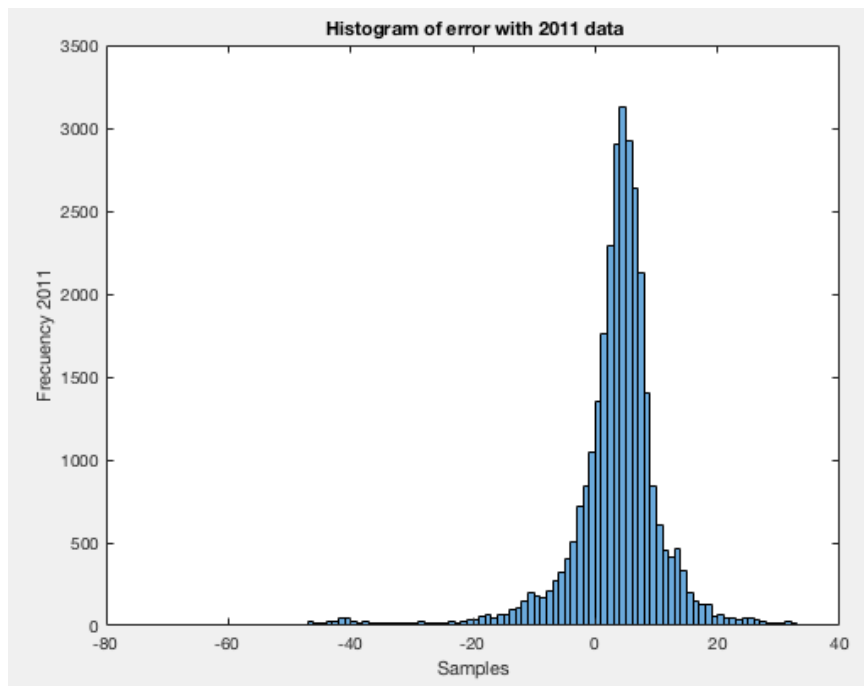


*Fig 48. Histograma error.*

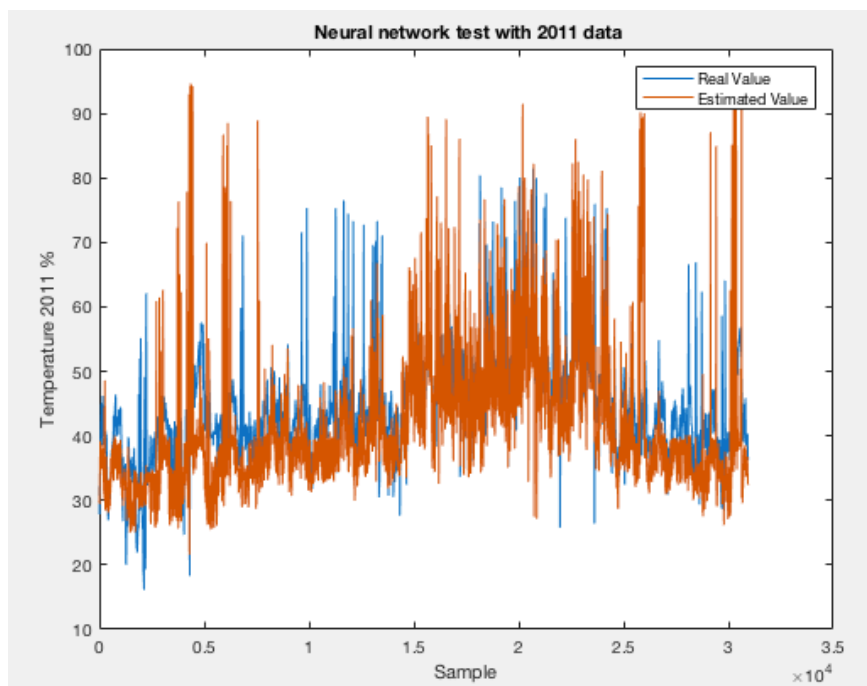


*Fig 49. Comparación datos reales/estimados*

**Datos 2011:**

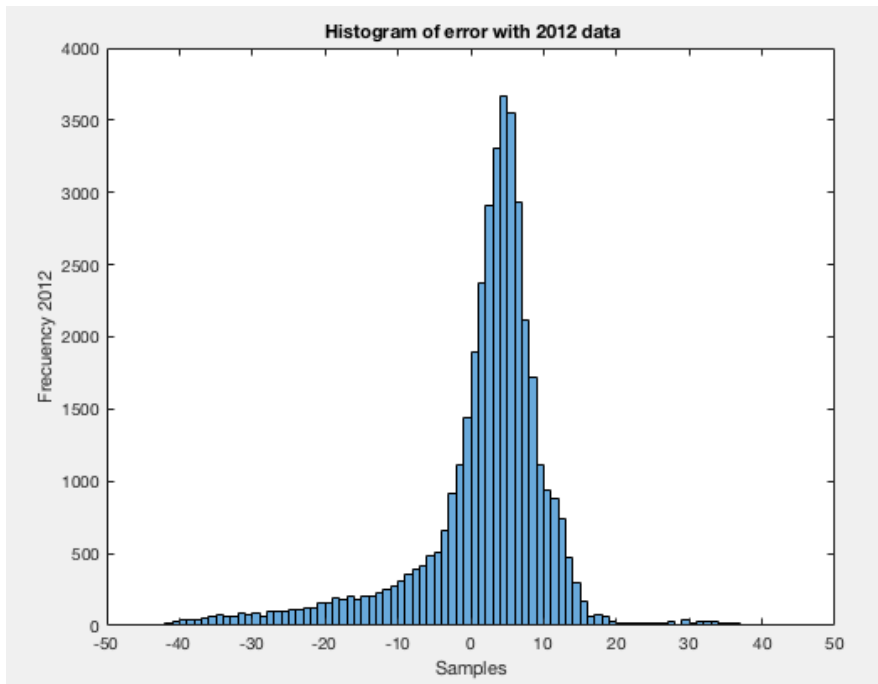


*Fig 50. Histograma error.*

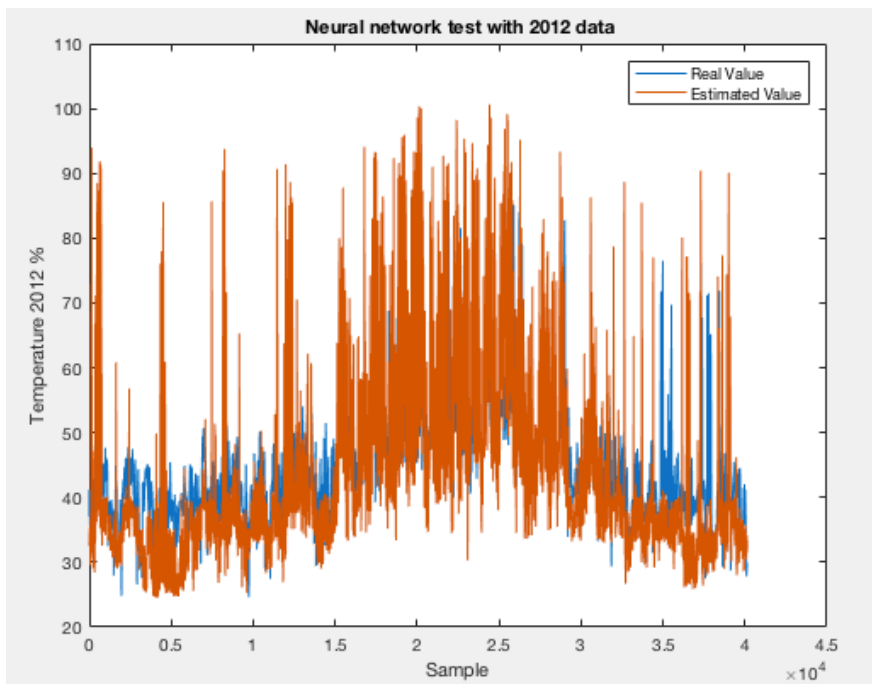


*Fig 51. Comparación datos reales/estimados.*

**Datos 2012:**

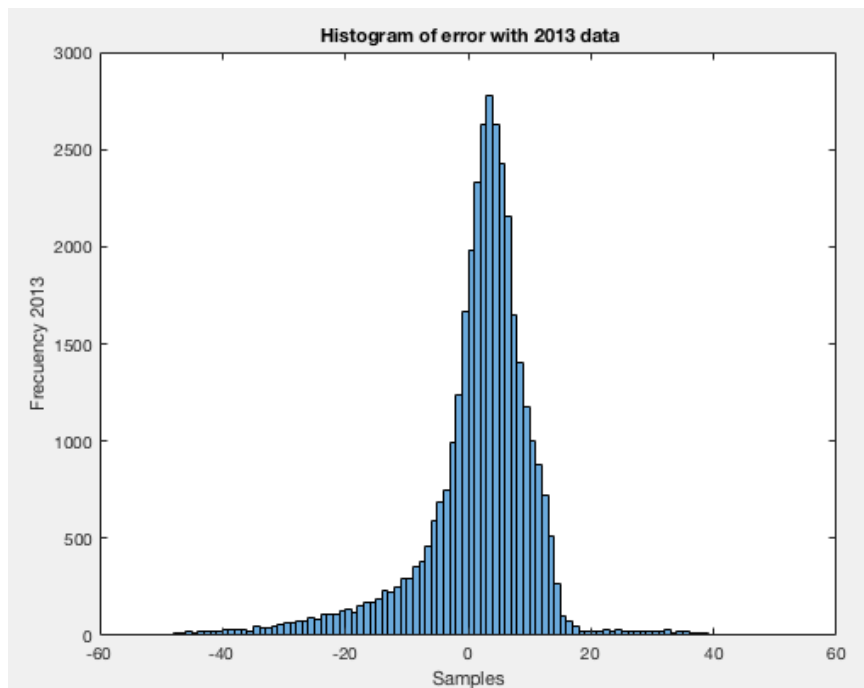


*Fig 52. Histograma error.*

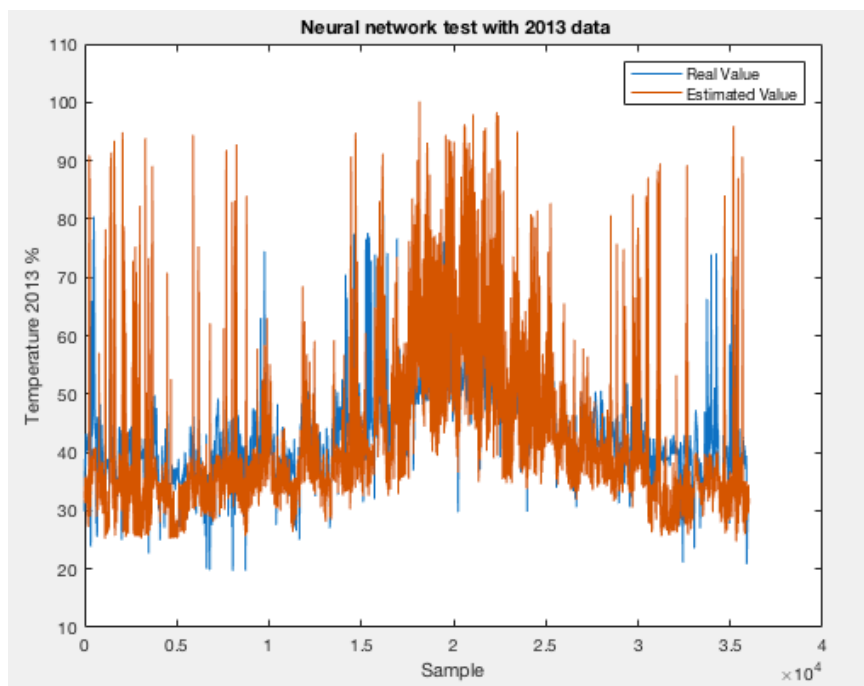


*Fig 53. Comparación datos reales/estimados.*

**Datos 2013:**

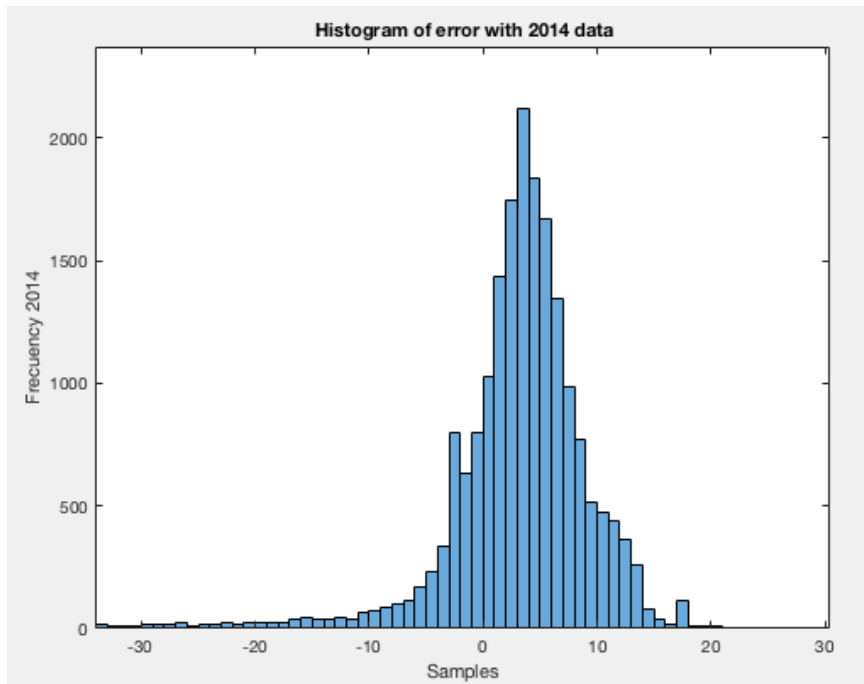


*Fig 54. Histograma error.*

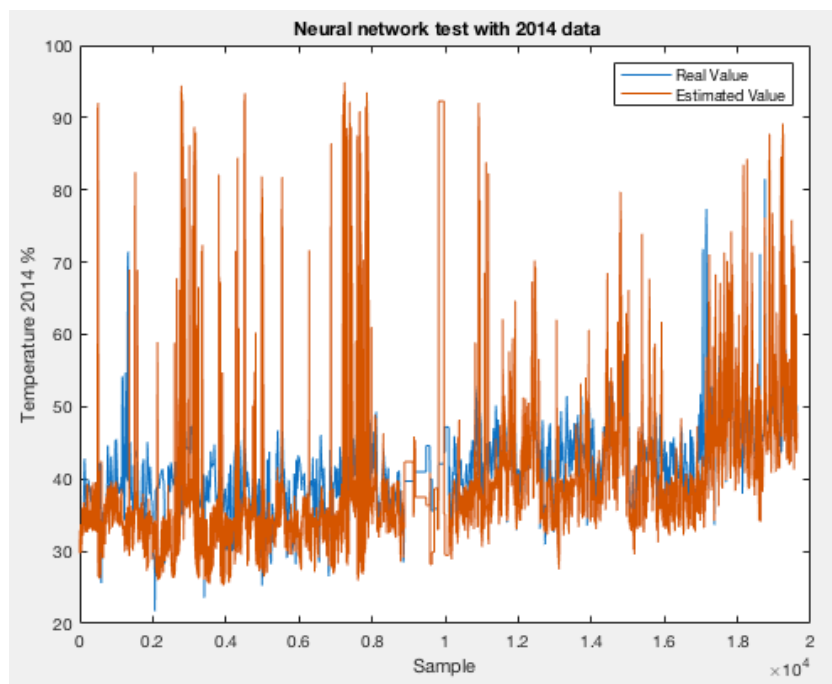


*Fig 55. Comparación datos reales/estimados.*

Datos 2014:



*Fig 56. Histograma error.*



*Fig 57. Comparación datos reales/estimados.*

### **Comentario los histogramas y gráficas de comparación:**

Los datos del 2014 tienen la misma falta de densidad en los meses de verano por las mismas razones mencionadas en la red neuronal anterior.

Los histogramas de error están ligeramente desplazados a la derecha del cero. Esto quiere decir que nuestra red neuronal predice, de media, temperaturas por encima de las temperaturas reales. Este modelo es especialmente cauto ya que por norma general predecirá temperaturas superiores a las reales.

Distribuciones de los errores por año:

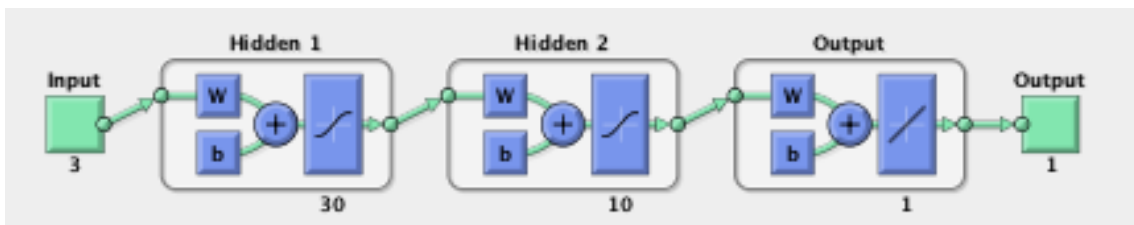
<b>Año</b>	<b>Media</b>	<b>Desviación</b>
<b>2009</b>	-0,0031	3,1501
<b>2010</b>	0,0755	4,0920
<b>2011</b>	4,0417	6,0703
<b>2012</b>	3,4643	9,2083
<b>2013</b>	2,5990	7,1024
<b>2014</b>	1,8267	5,2368

Observamos como las medias y las desviaciones se mantienen aproximadamente constantes. En esta red neuronal podemos observar como el cambio de comportamiento en el error del rodamiento DE no es tan alto como el de las redes neuronales previas, lo que mantiene el error a lo largo del tiempo.

### 3.4 Modelo 4. Red Neuronal Rodamiento de Alta de la Multiplicadora.

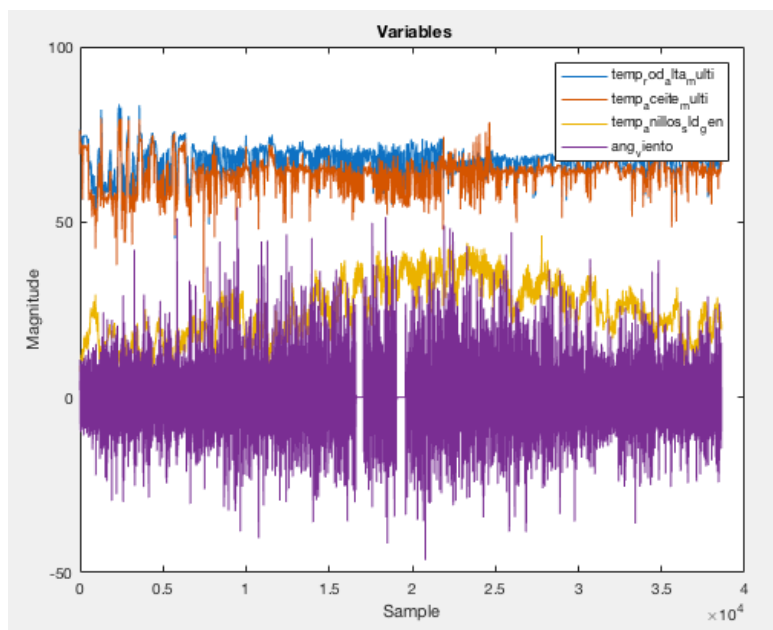
Este modelo trata de predecir con precisión la temperatura del rodamiento de alta de la multiplicadora. Los engranajes de la multiplicadora sufren constantes esfuerzos debido a que están sometidos a alta fricción. Medir la temperatura es una buena manera de predecir sus faltas, para adelantarnos y sustituir los componentes dañados a tiempo. Los datos que mejor ayudan a simular la temperatura de este rodamiento son: temperatura del aceite de la multiplicadora, ángulo del viento y la potencia generada.

Una vez filtrados y seleccionados los datos como inputs a la red neuronal, se llegó a la conclusión de que las variables que mejor servían para ajustar la red neuronal serían las anteriormente mencionadas. Es verdad que el tener tres inputs puede hacer más lenta la simulación, pero se gana mucho más en la precisión. Se trata de una red neuronal de doble capa. El esquema es el siguiente:



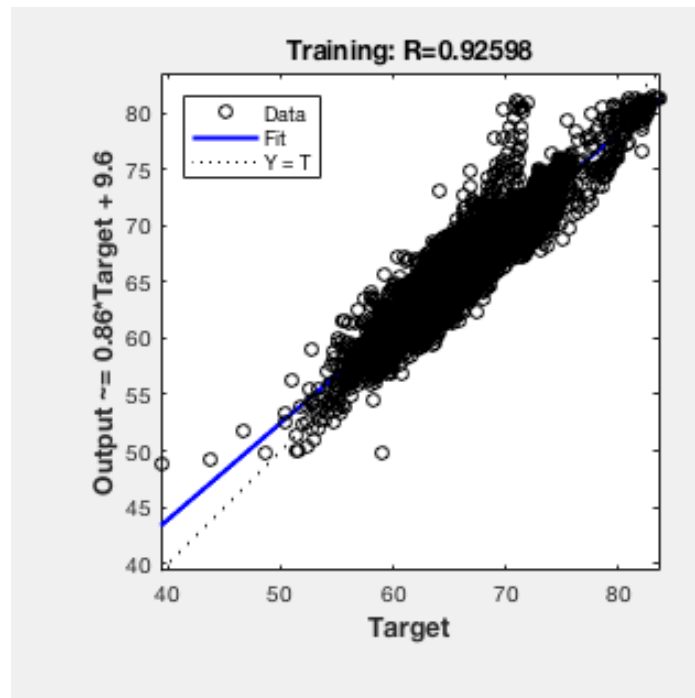
*Fig 58. Bloque red neuronal de la temperatura del rodamiento alta multiplicadora en el entorno de Matlab.*

Previo a la simulación representamos los valores de los inputs a la red neuronal en función del tiempo para ver su evolución:



*Fig 59. Inputs reales de la red neuronal.*

En la gráfica se ve como la temperatura del aceite de la multiplicadora no varía, como era de esperar. También, podemos observar una pequeña elevación de la temperatura de los anillos que unen el rotor y el estator de la máquina durante el periodo estival, así como pequeñas faltas de datos del ángulo del viento durante los meses de verano.



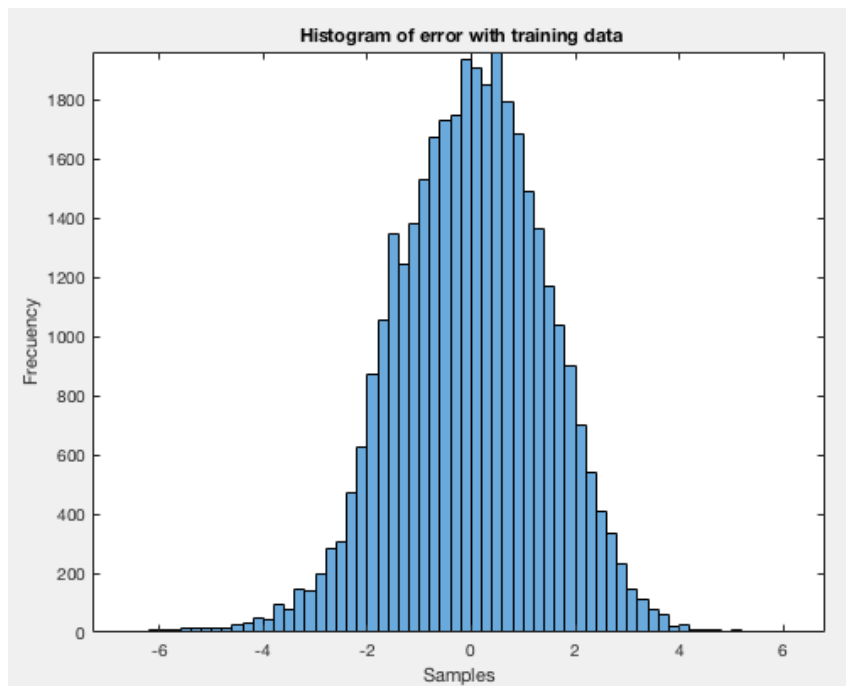
*Fig 60. Recta regresión RN Temperatura Rodamiento de alta de la multiplicadora.*

La recta de regresión con una  $R \approx 0,93$  nos muestra que la temperatura del rodamiento de alta de la multiplicadora tiene una intensa relación con los inputs utilizados, a pesar de existen puntos que se salen de la recta general. En este caso podemos ver otra pequeña nube de puntos que parece seguir otra recta diferente a la mayoría. Esto puede traducirse en que existirán errores fuera de lo normal con esta nube de puntos.

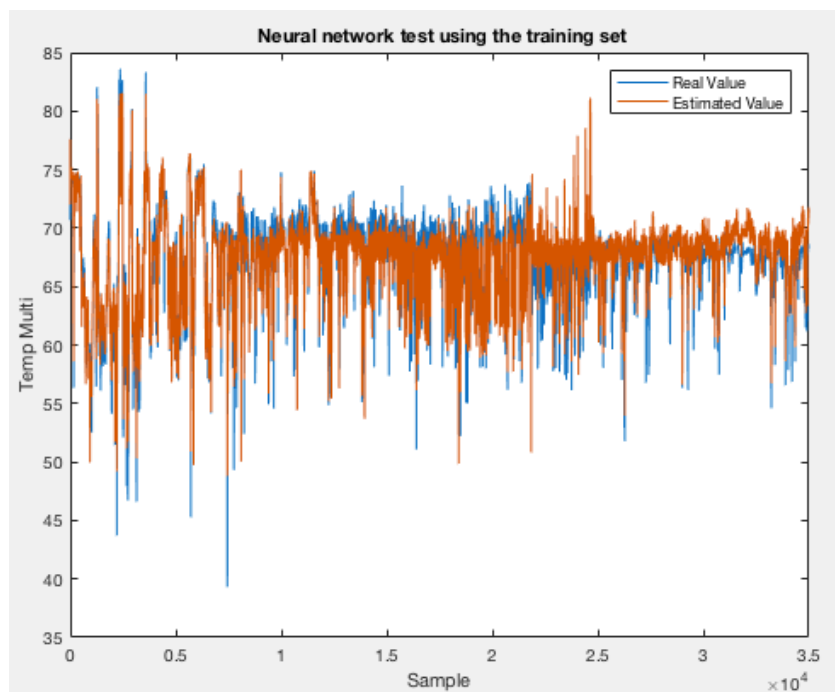
### **Estudio del error**

Como en todo modelo es importante conocer el error al que nos atenemos. Observamos los errores con los datos de entrenamiento (año 2009) y con los datos de los años siguientes. Para continuar con el estudio de la red neuronal, también representamos los valores reales frente a los simulados para observar si existen comportamientos anormales.

**Datos 2009 (entrenamiento):**

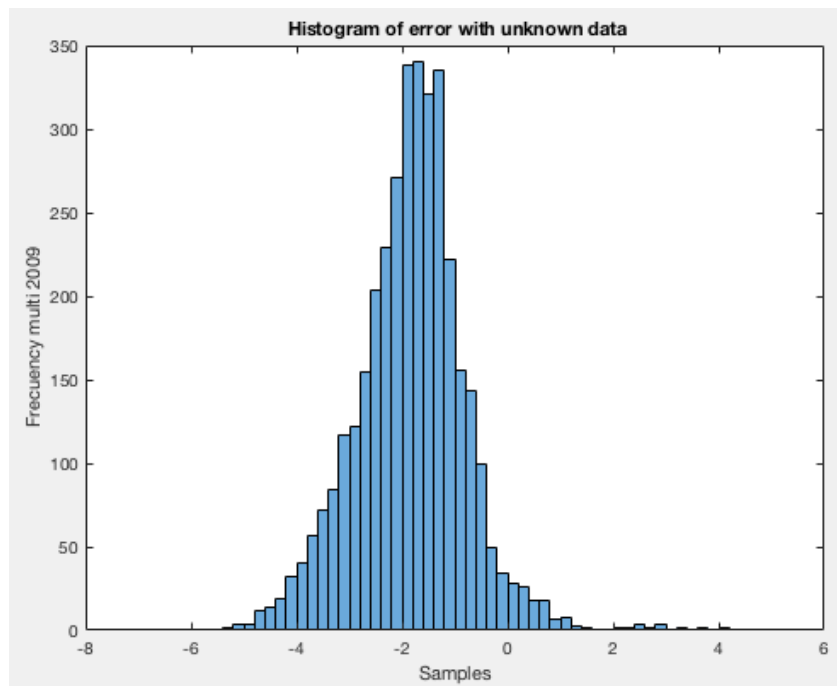


*Fig 61. Histograma error.*

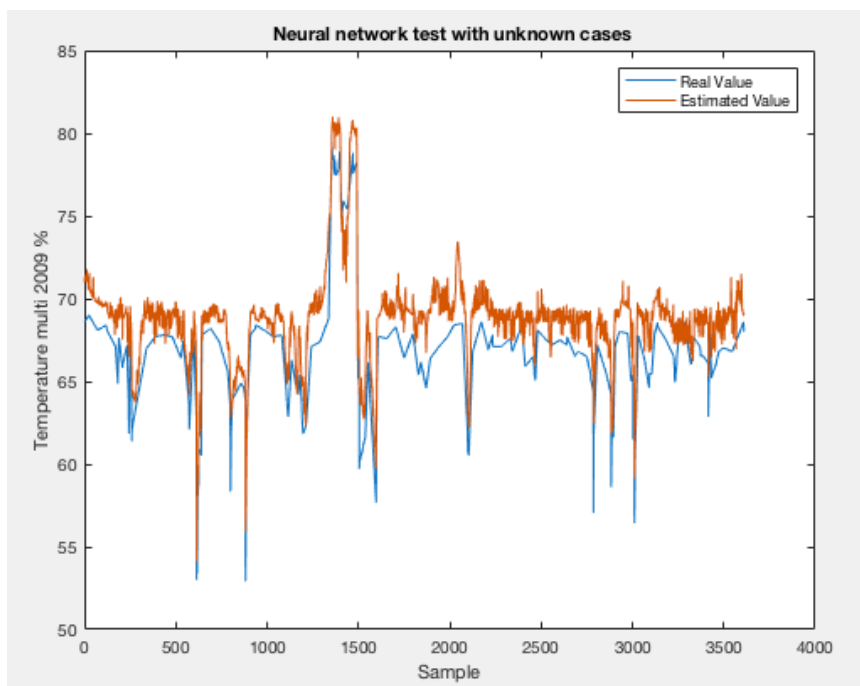


*Fig 62. Comparación datos reales/estimados.*

**Datos 2009 (simulación):**

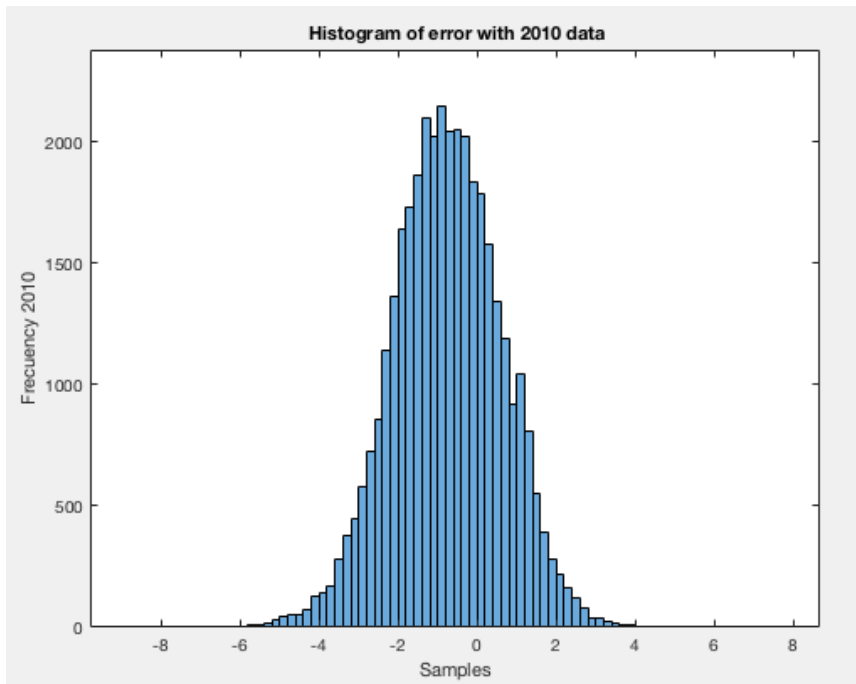


*Fig 63. Histograma error.*

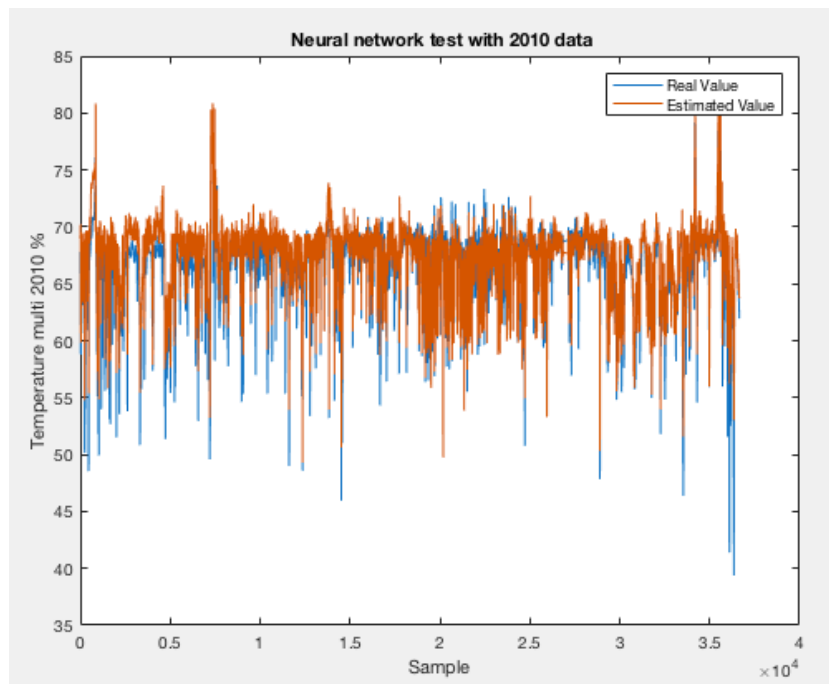


*Fig 64. Comparación datos reales/estimados.*

**Datos 2010:**

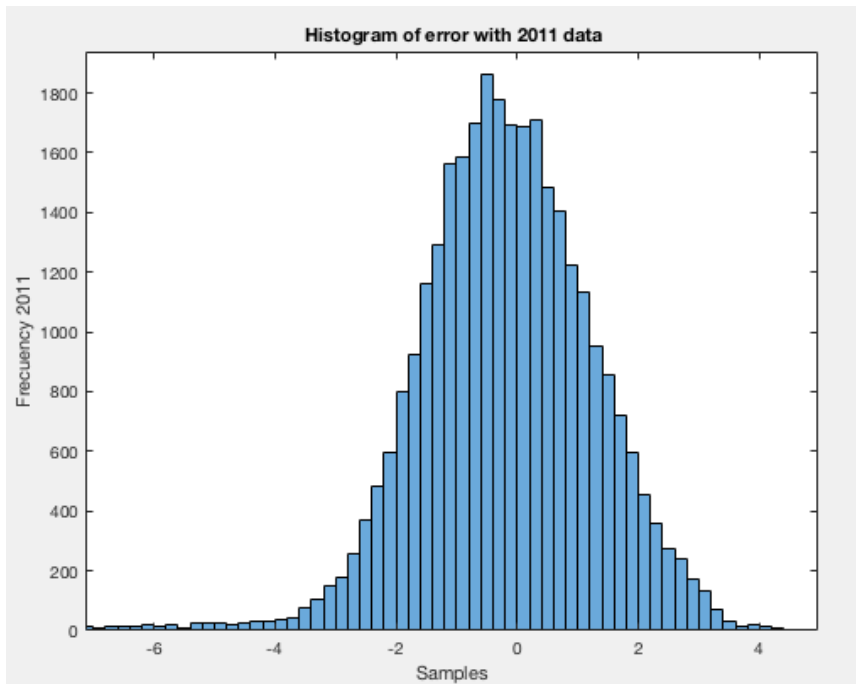


*Fig 65. Histograma error.*

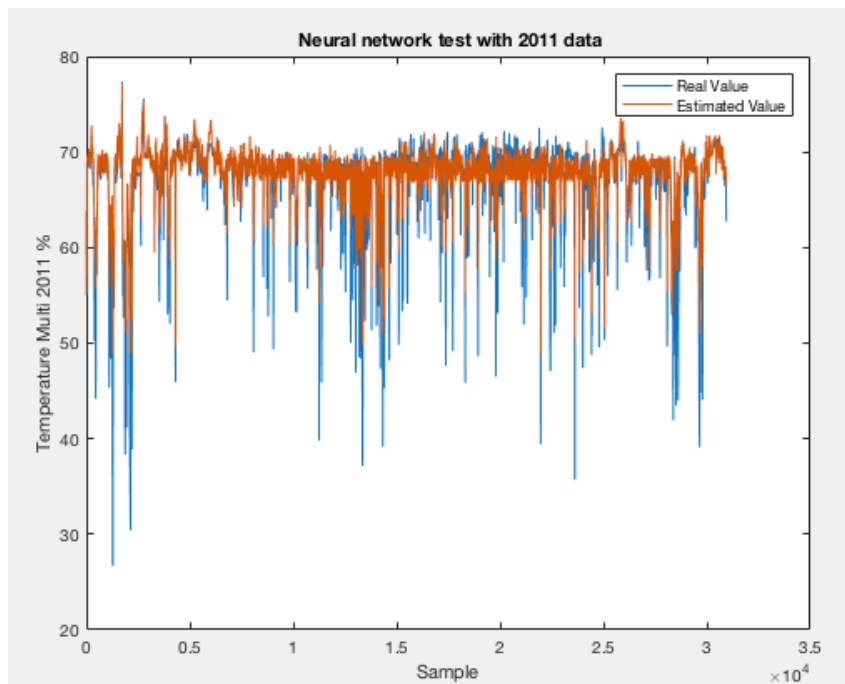


*Fig 66. Comparación datos reales/estimados.*

**Datos 2011:**

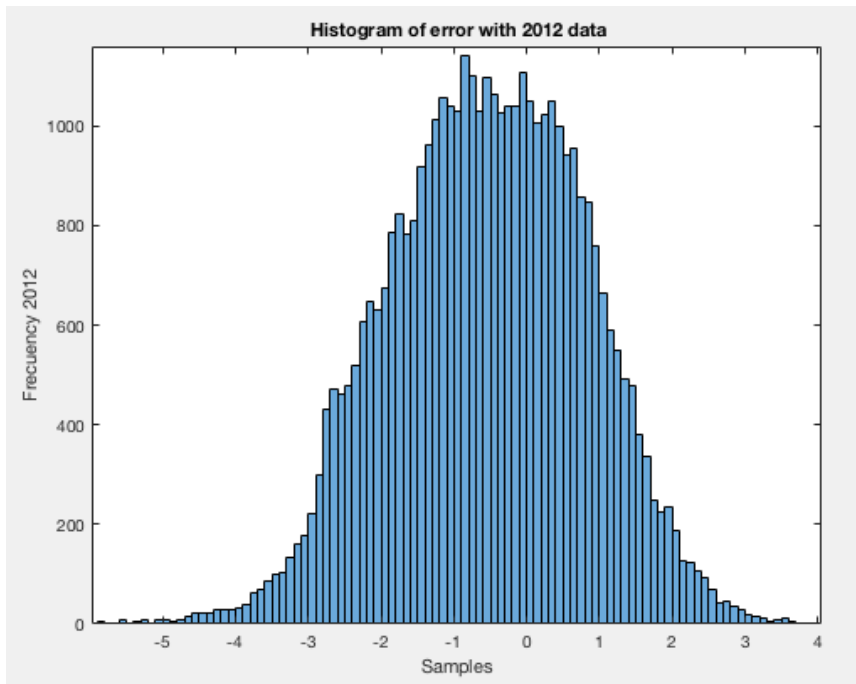


*Fig 67. Histograma error.*

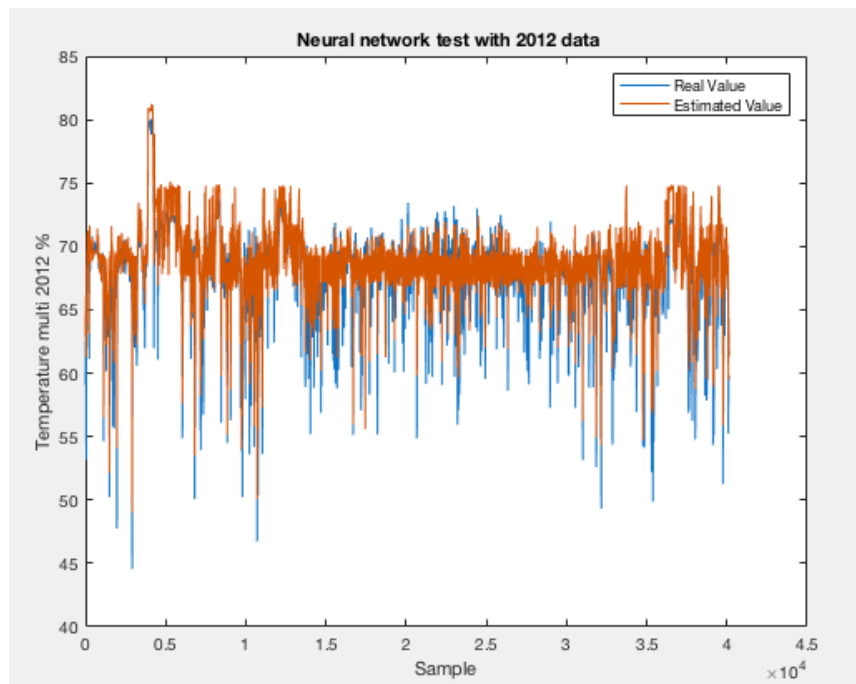


*Fig 68. Comparación datos reales/estimados.*

**Datos 2012:**

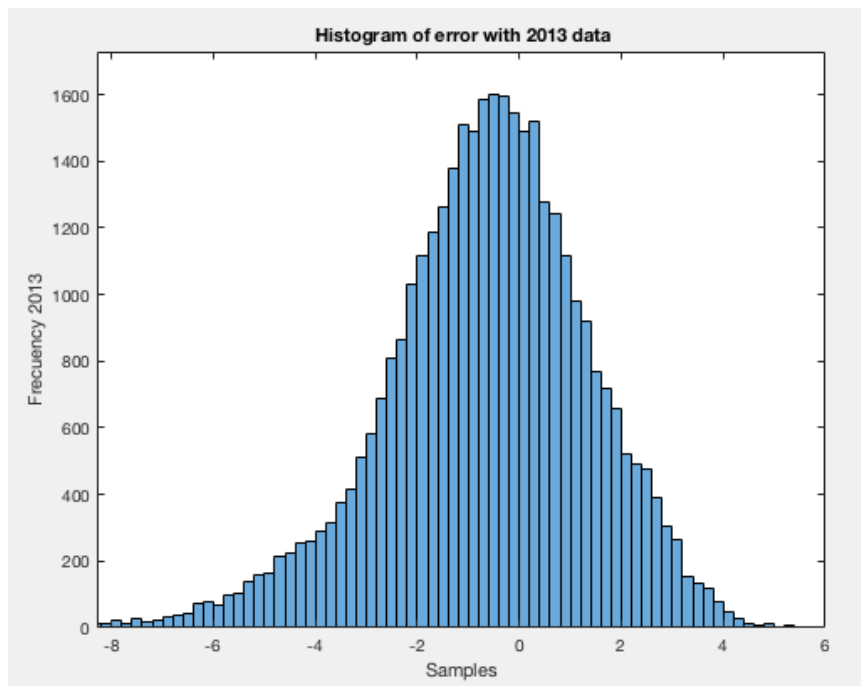


*Fig 69. Histograma error.*

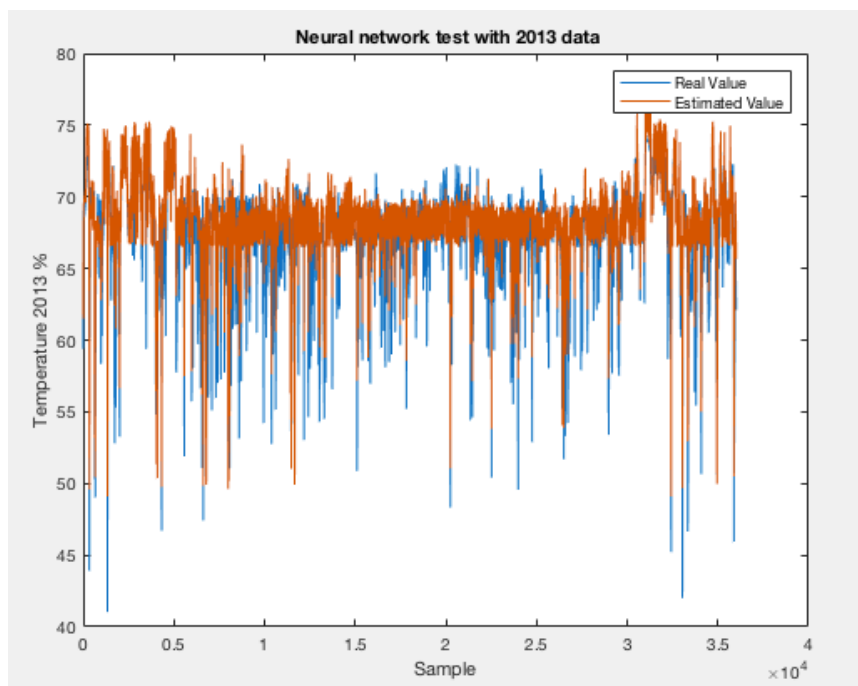


*Fig 70. Comparación datos reales/estimados.*

**Datos 2013:**

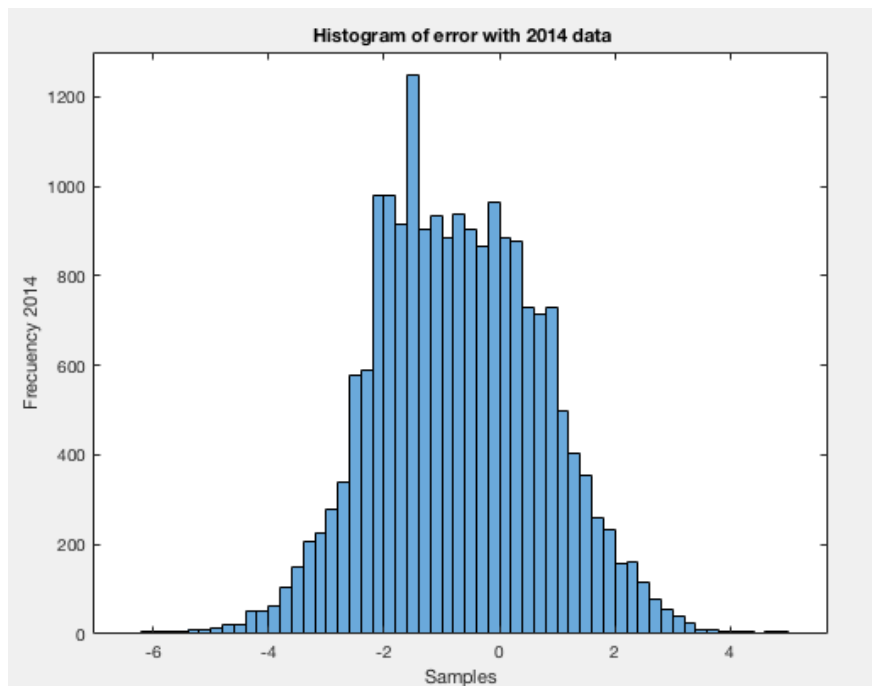


*Fig 71. Histograma error.*

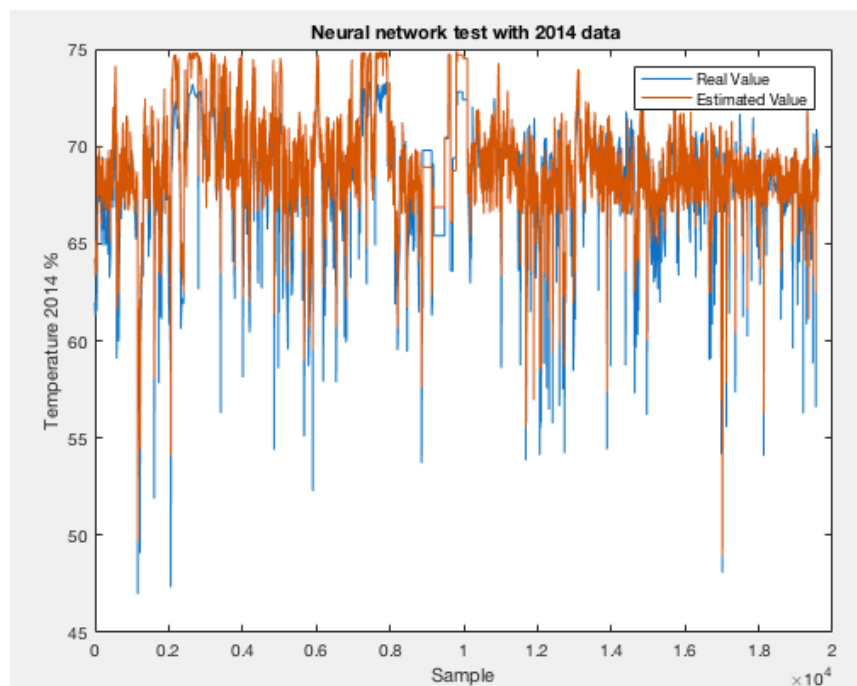


*Fig 72. Comparación datos reales/estimados.*

Datos 2014:



*Fig 73. Histograma error.*



*Fig 74. Comparación datos reales/estimados.*

### **Comentario los histogramas y gráficas de comparación:**

Podemos ver como en el caso del año 2014 los datos no son tan completos y tuvieron más errores que en los otros años, por lo que de los meses del medio del año tenemos menos información. En este caso son importantes pues son como hemos visto en la distribución de los datos aquellos en los que la temperatura exterior es más alta, causando que la temperatura de los rodamientos sea más alta independientemente de su funcionamiento.

Los histogramas de error están muy ligeramente desplazados a la izquierda del cero. Esto quiere decir que nuestra red neuronal predice, de media, temperaturas por debajo de las temperaturas reales. Si los errores fueran mayores y más habituales esto podría llegar a ser un problema ya que predecir por debajo de las temperaturas reales podría llevarnos a creer que el funcionamiento es normal cuando no lo es, imposibilitando la buena previsión de faltas.

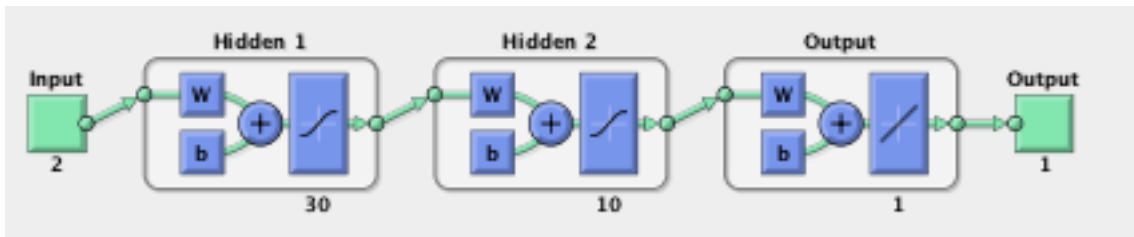
Distribuciones de los errores por año:

<b>Año</b>	<b>Media</b>	<b>Desviación</b>
<b>2009</b>	0,0168	1,4520
<b>2010</b>	-0,7618	1,4323
<b>2011</b>	-0,3289	1,9255
<b>2012</b>	-0,5251	1,3678
<b>2013</b>	-0,5622	1,9907
<b>2014</b>	-0,6702	1,5054

Observamos como las medias y las desviaciones se mantienen aproximadamente constantes. En esta red neuronal podemos observar como el cambio de comportamiento en el error del rodamiento de la multiplicadora no es tan alto como el de las dos primeras redes neuronales estudiadas, lo que mantiene el error a lo largo del tiempo.

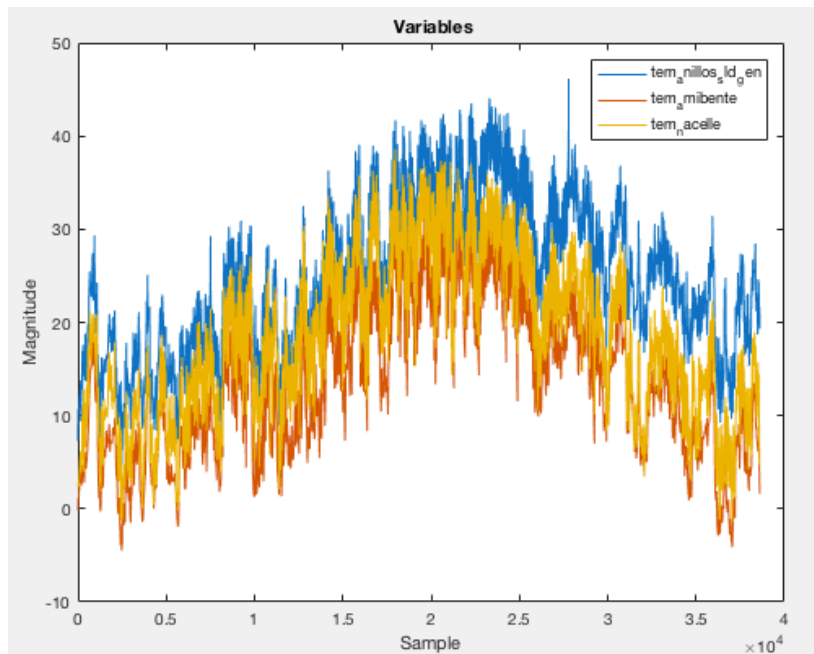
### 3.5 Modelo 5. Red Neuronal Anillos SLD del generador

La intención de este modelo es simular la temperatura de los anillos SLD del generador. [5] Son las escobillas del generador a través de las que se establece la conexión eléctrica entre las bobinas del estator y las bobinas del rotor. Los datos que mejor ayudan a simular la temperatura de los anillos son: la temperatura ambiente y la temperatura de la góndola. Una vez filtrados y seleccionados los datos como inputs a la red neuronal, se llegó a la conclusión de que las variables que mejor servían para ajustar la red neuronal serían las anteriormente mencionadas. Se trata de una red neuronal de doble capa. El esquema es el siguiente:



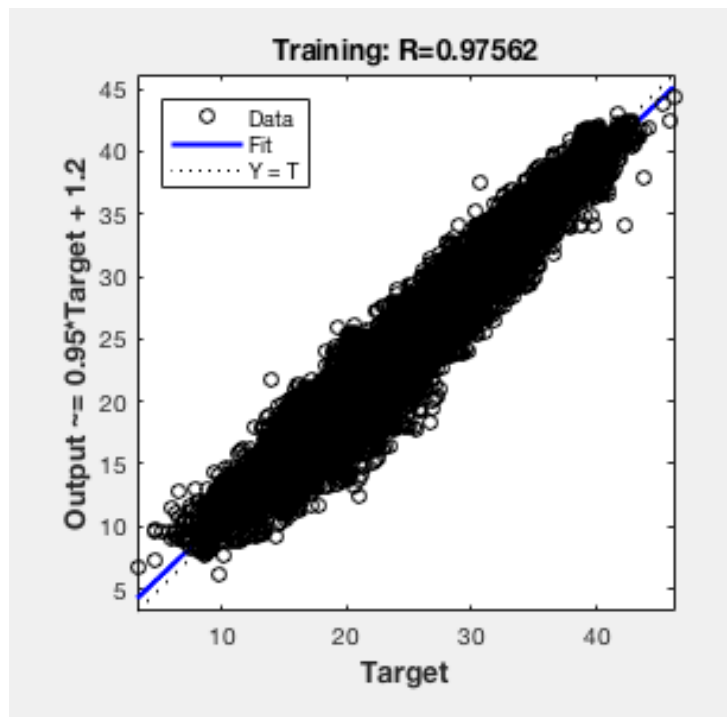
*Fig 75. Bloque red neuronal de la temperatura del rodamiento alta multiplicadora en el entorno de Matlab.*

Previo a la simulación representamos los valores de los inputs a la red neuronal en función del tiempo para ver su evolución.



*Fig 76. Inputs reales de la red neuronal.*

Podemos observar una elevación de la temperatura de los anillos que unen el rotor y el estator de la máquina durante el periodo estival, así como evidentemente la temperatura ambiente y de la góndola.



*Fig 77. Recta regresión RN Temperatura anillos SLD del aerogenerador.*

La recta de regresión con una  $R \approx 0,98$  nos muestra que la temperatura de los anillos SLD del aerogenerador tiene una intensa relación con los inputs utilizados.

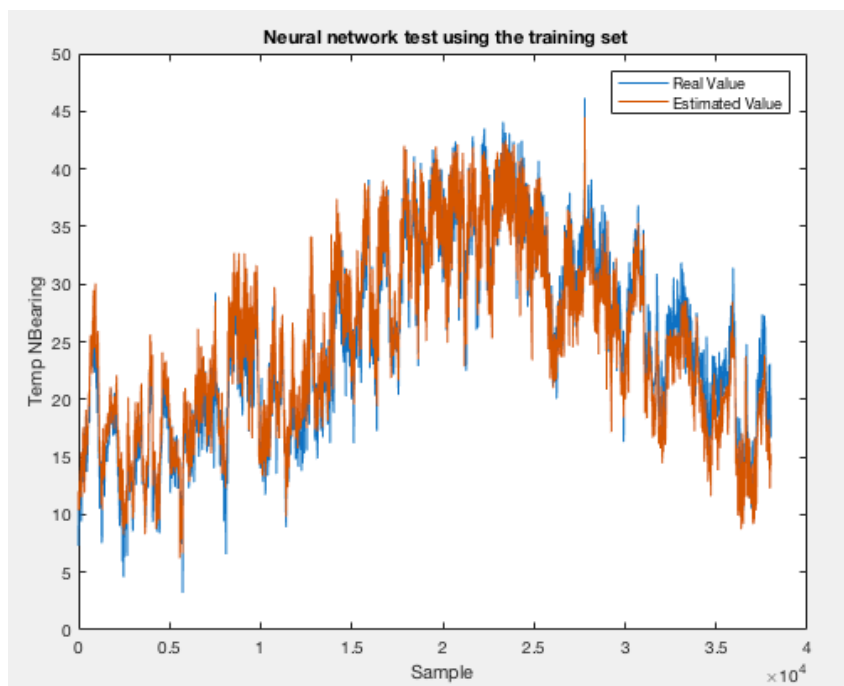
### **Estudio del error**

Como en todo modelo es importante conocer el error al que nos atenemos. Observamos los errores con los datos de entrenamiento (año 2009) y con los datos de los años siguientes. Para continuar con el estudio de la red neuronal, también representamos los valores reales frente a los simulados para observar si existen comportamientos anormales.

**Datos 2009 (entrenamiento):**

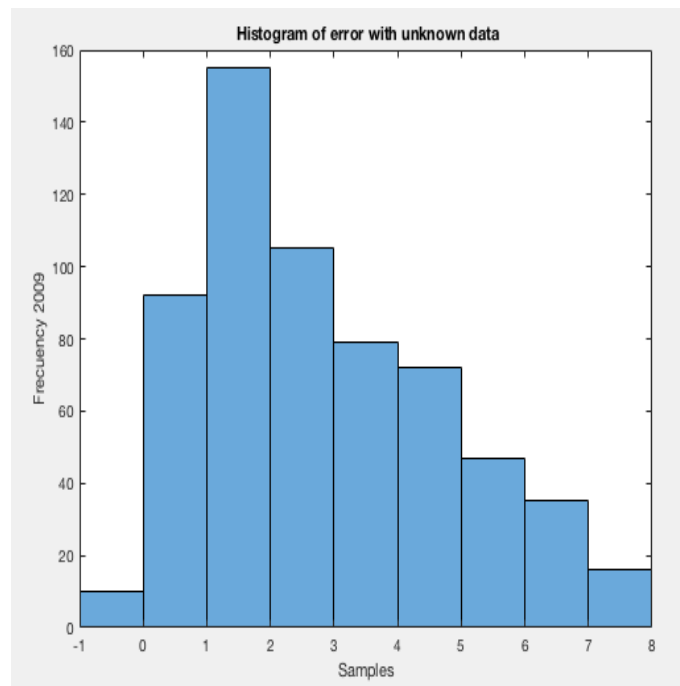


*Fig 78. Histograma error.*

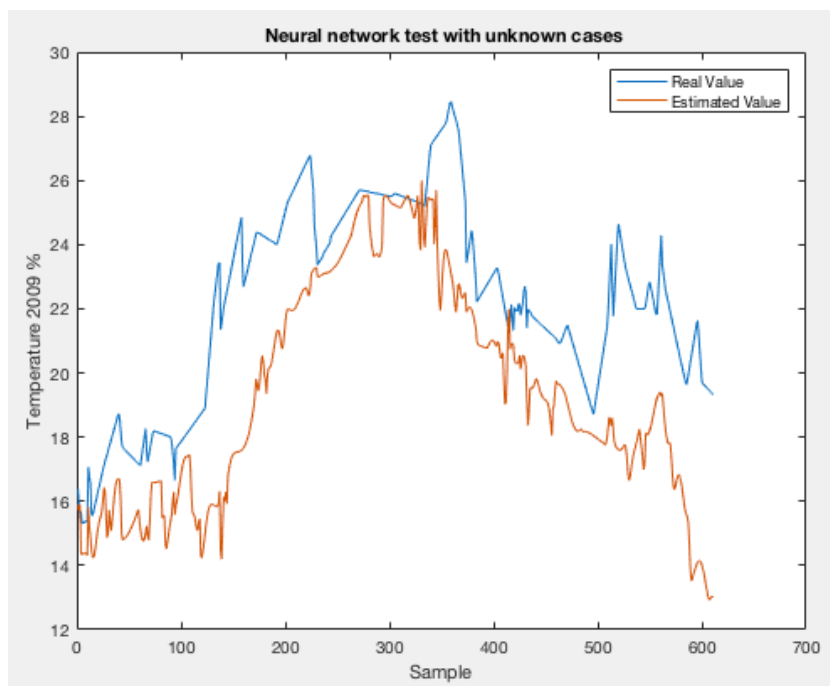


*Fig 79. Comparación datos reales/estimados.*

**Datos 2009 (simulación):**

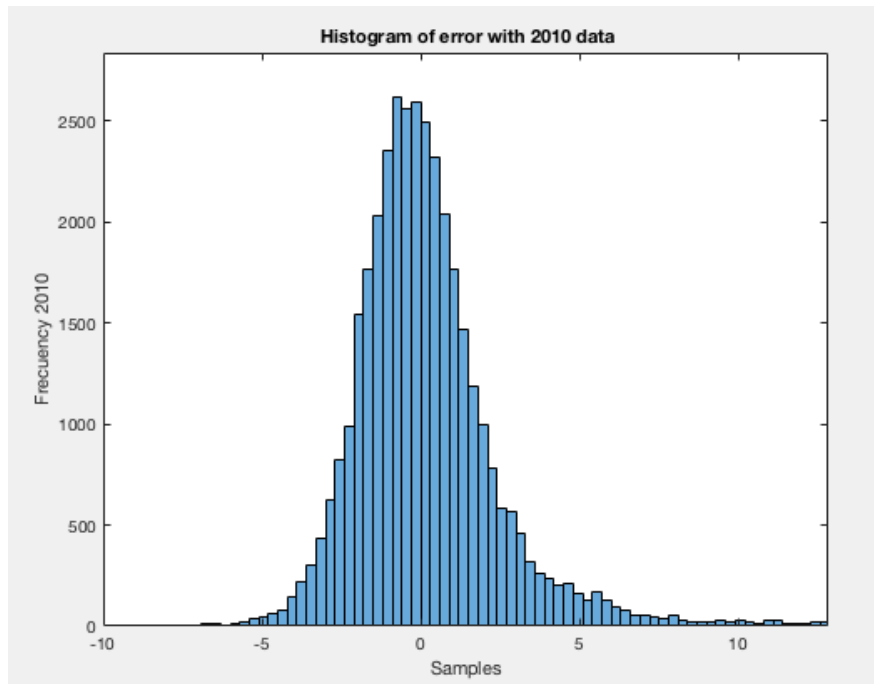


*Fig 80. Histograma error.*

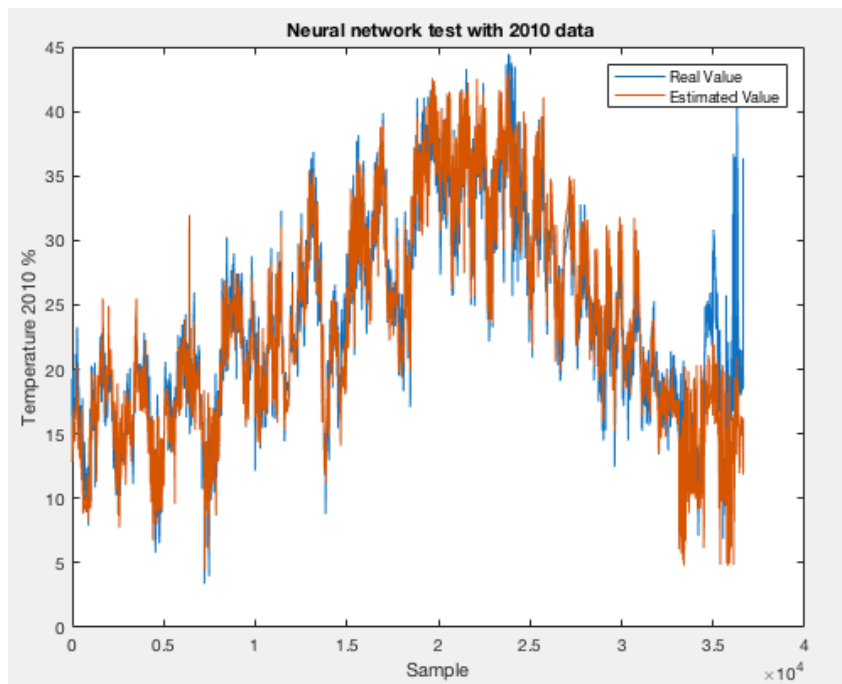


*Fig 81. Comparación datos reales/estimados.*

**Datos 2010:**

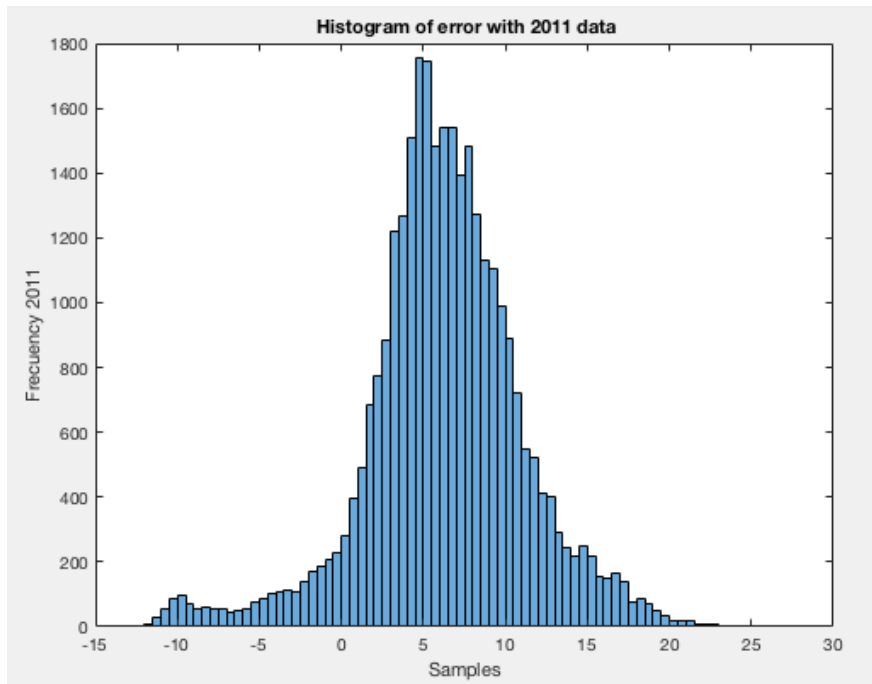


*Fig 82. Histograma error.*

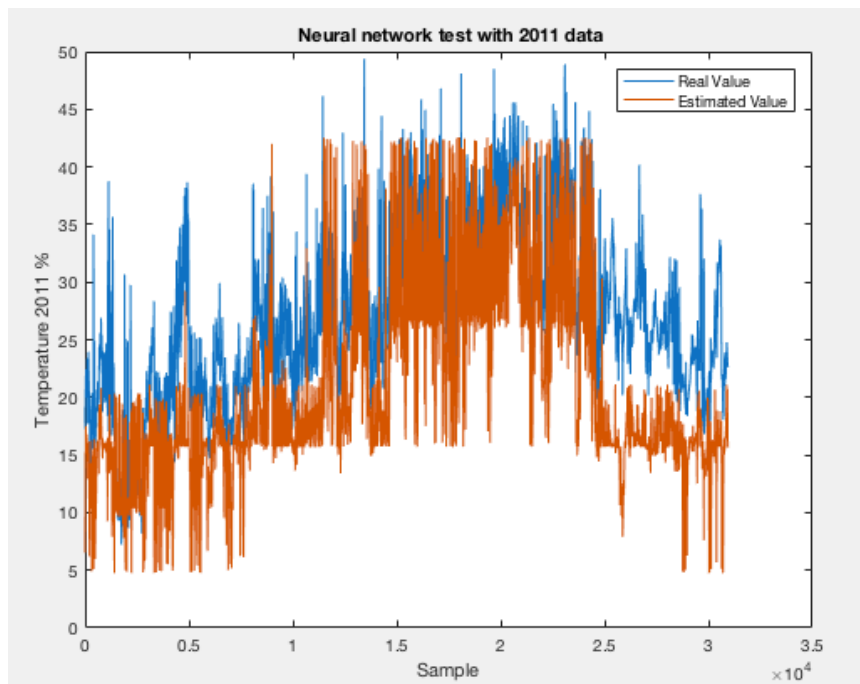


*Fig 83. Comparación datos reales/estimados.*

**Datos 2011:**

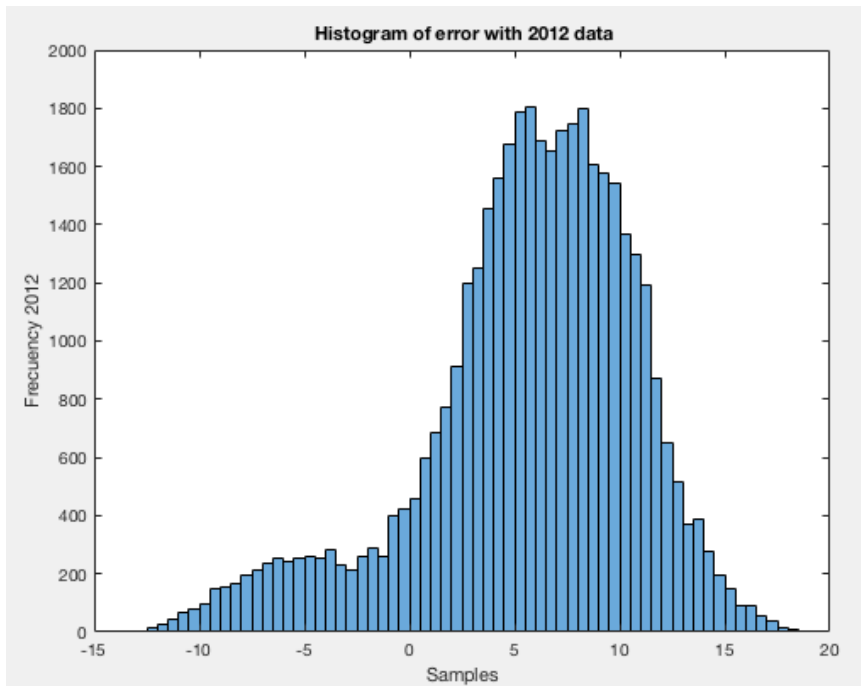


*Fig 84. Histograma error.*

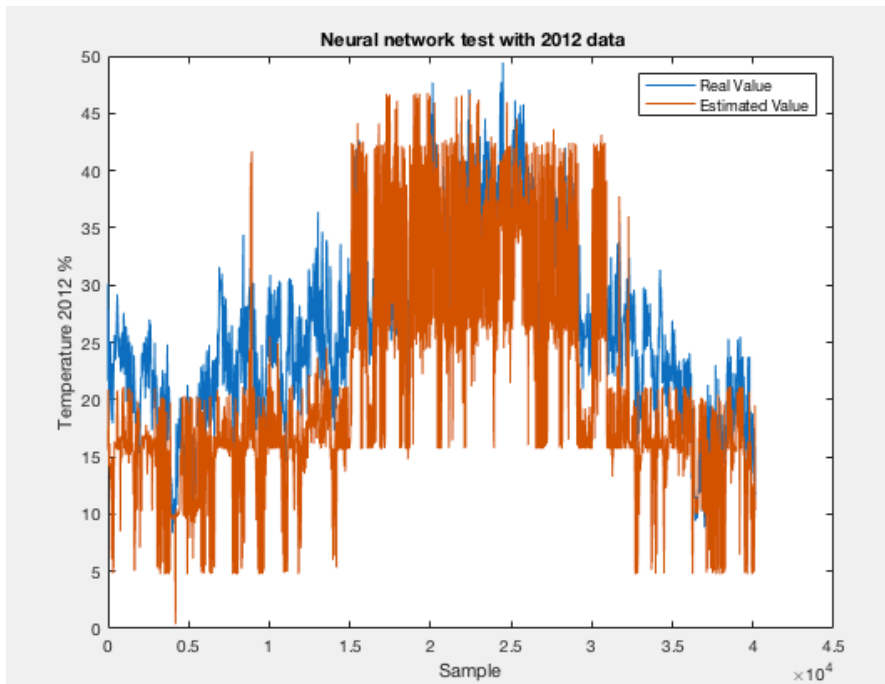


*Fig 85. Comparación datos reales/estimados.*

**Datos 2012:**

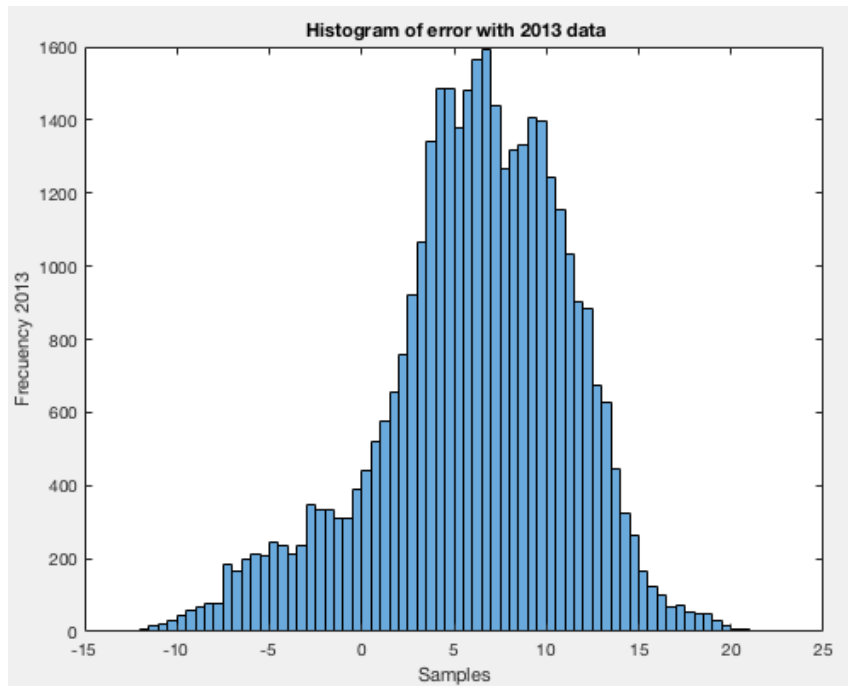


*Fig 86. Histograma error.*

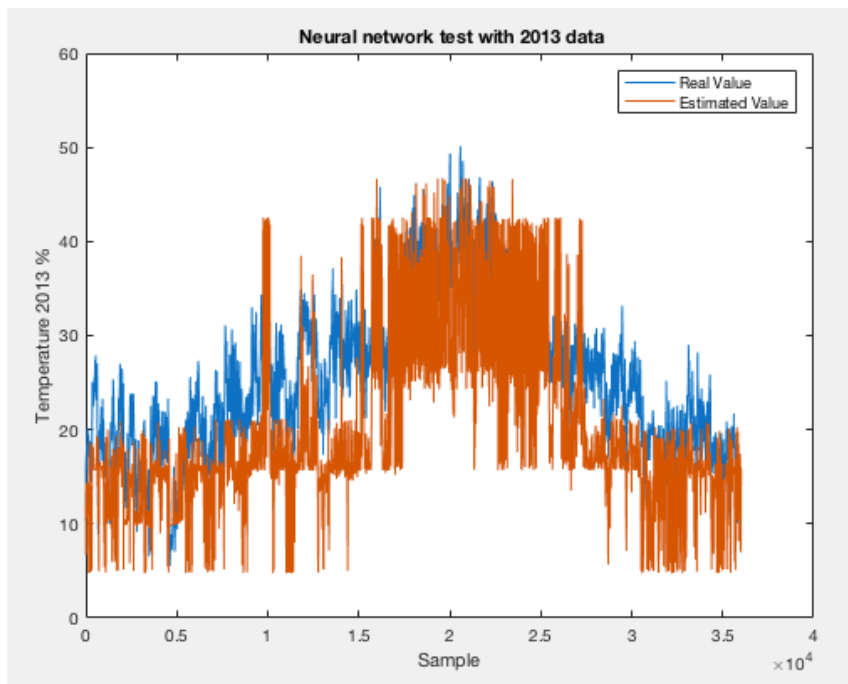


*Fig 87. Comparación datos reales/estimados.*

Datos 2013:

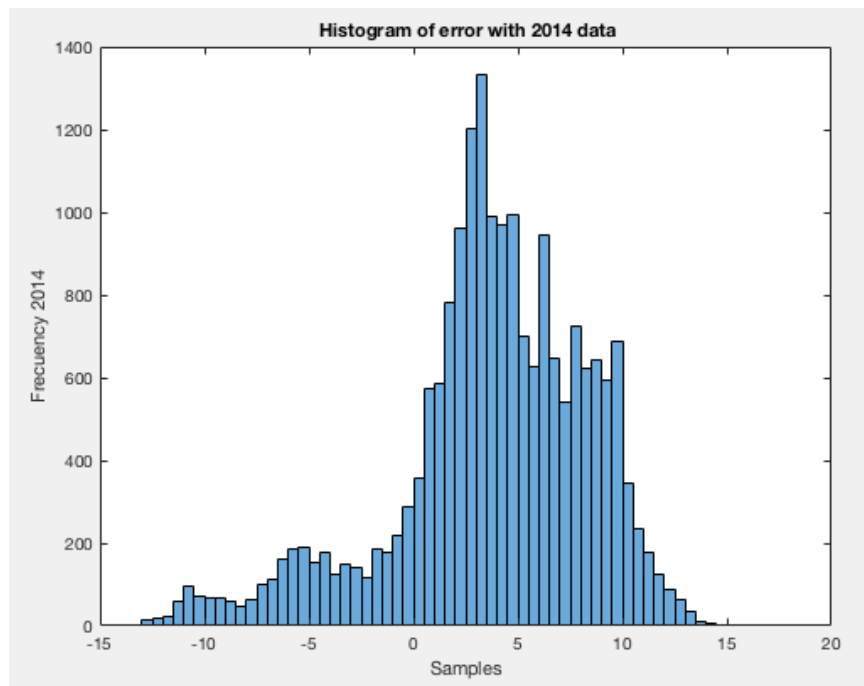


*Fig 88. Histograma error.*

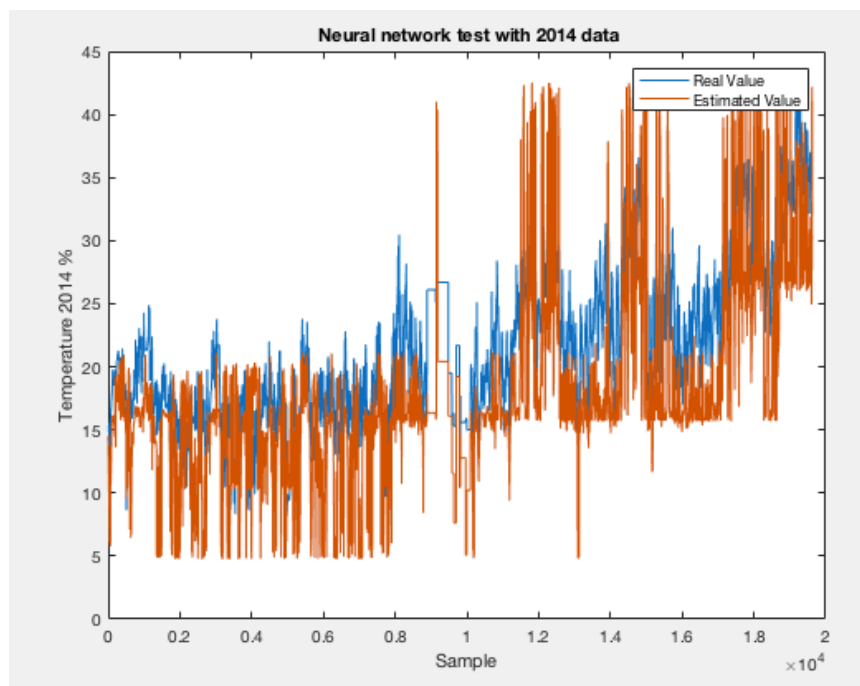


*Fig 89. Comparación datos reales/estimados.*

Datos 2014:



*Fig 90. Histograma error.*



*Fig 91. Comparación datos reales/estimados.*

### **Comentario los histogramas y gráficas de comparación:**

Podemos ver como en el caso del año 2014 los datos no son tan completos y tuvieron más errores que en los otros años, por lo que de los meses del medio del año tenemos menos información. Esto es importante, ya que como hemos visto en la distribución de los datos de verano, en los que la temperatura exterior es más alta, la temperatura de los rodamientos también lo será.

Los histogramas de error están ligeramente desplazados a la derecha del cero. Esto quiere decir que nuestra red neuronal predice, de media, temperaturas por encima de las temperaturas reales. Este modelo es especialmente cauto ya que por norma general predecirá temperaturas superiores a las reales. De las redes neuronales que hemos simulado esta es la que tiene una peor distribución del error. En los últimos años (2012-2013-2014) la distribución del error está prácticamente centrada en el 5.

Distribuciones de los errores por año:

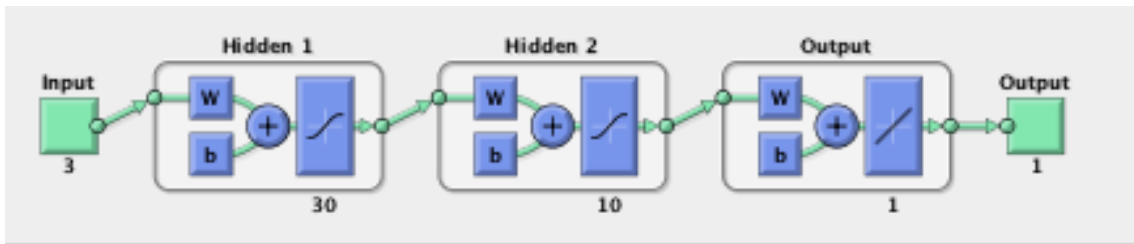
<b>Año</b>	<b>Media</b>	<b>Desviación</b>
<b>2009</b>	0,0019	1,7974
<b>2010</b>	0,5625	6,6780
<b>2011</b>	9,7167	12,9503
<b>2012</b>	13,2104	20,2164
<b>2013</b>	11,5625	17,9920
<b>2014</b>	7,0002	16,0305

Observamos como las medias y también las desviaciones (aunque en menor medida) de los primeros años (2009,2010,) son claramente más bajas que los años posteriores. Este cambio es debido a que los comportamientos de los diferentes componentes del aerogenerador se modifican a medida que la máquina va envejeciendo.

### 3.6 Modelo 6. Red Neuronal PitchAngle.

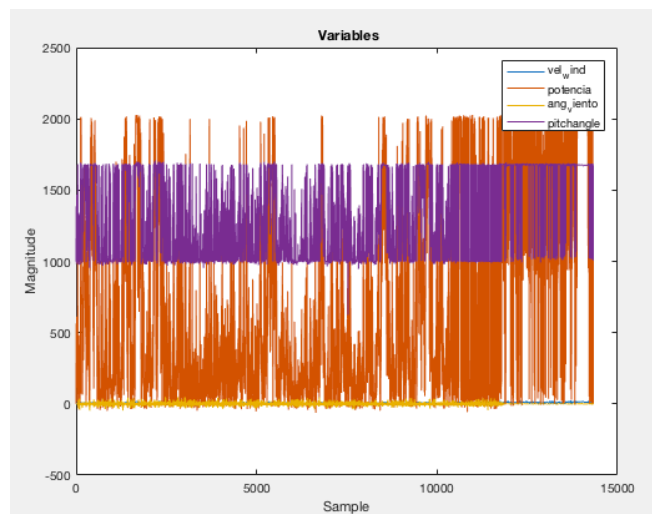
La intención de este modelo es simular el ángulo de paso de las palas. Dependiendo de la dirección del viento los aerogeneradores pueden ajustar la posición de las palas para que este incida lo más perpendicular posible, y así poder extraer la mayor cantidad de energía posible. En el caso de esta red neuronal los datos eran más reducidos y por lo tanto se agruparon todos sin tener en cuenta el año. Por esto, la red neuronal entrenada tiene mucha menor fiabilidad que las anteriores y, además, tampoco había suficientes datos para comprobar su eficacia. Los inputs seleccionados para esta red neuronal han sido: la velocidad del viento, el ángulo del viento y la potencia.

Una vez filtrados y seleccionados los datos como inputs a la red neuronal, se llegó a la conclusión de que las variables que mejor servían para ajustar la red neuronal serían las anteriormente mencionadas. Se trata de una red neuronal de doble capa. El esquema es el siguiente:

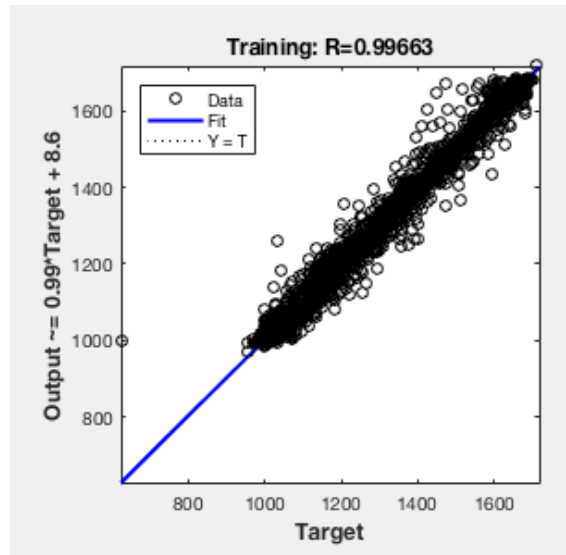


*Fig 92. Bloque red neuronal del ángulo de paso de las palas en el entorno de Matlab.*

Previamente a la simulación representamos los valores de los inputs a la red neuronal en función del tiempo para ver su evolución.



*Fig 93. Inputs reales de la red neuronal.*



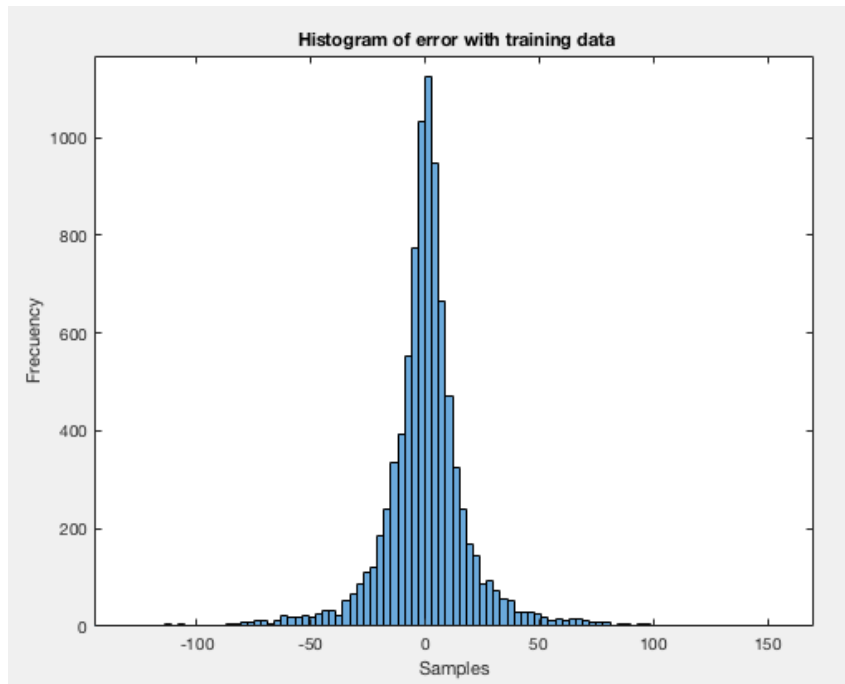
*Fig 94. Recta regresión RN Temperatura anillos SLD del aerogenerador.*

La recta de regresión con una  $R \approx 0,99$  nos muestra que la temperatura de los anillos SLD del aerogenerador tiene una intensa relación con los inputs utilizados.

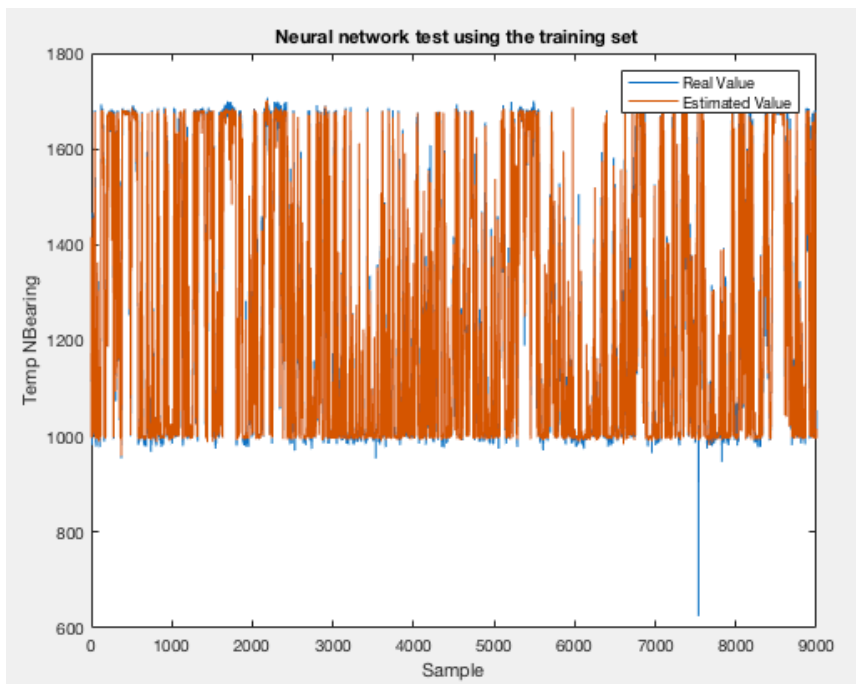
### **Estudio del error**

Como en todo modelo es importante conocer el error al que nos atenemos. Observamos los errores con los datos de entrenamiento (año 2009) y con los datos de los años siguientes. Para continuar con el estudio de la red neuronal, también representamos los valores reales frente a los simulados para observar si existen comportamientos anormales.

**Datos entrenamiento:**

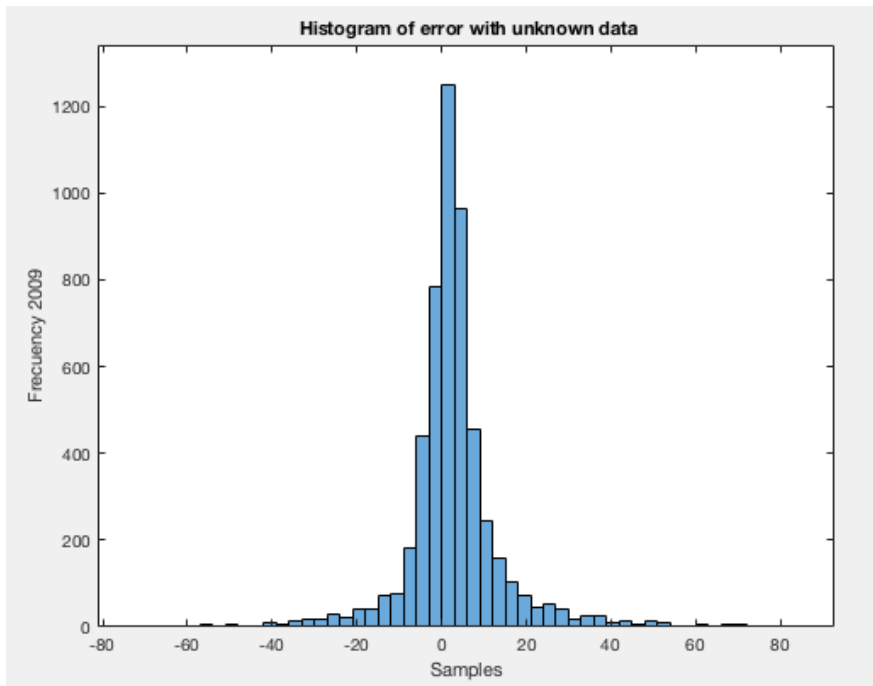


*Fig 95. Histograma error.*

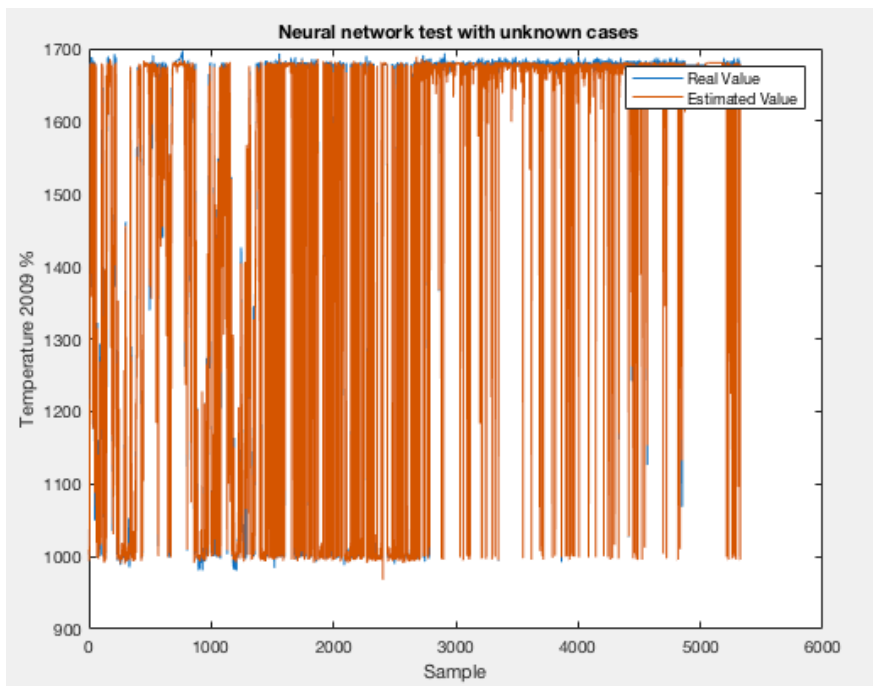


*Fig 96. Comparación datos reales/estimados.*

**Datos simulación:**



*Fig 97. Histograma error.*



*Fig 98. Comparación datos reales/estimados.*

### **Comentario los histogramas y gráficas de comparación:**

Podemos ver como los histogramas de error están en torno al cero, lo que es un buen indicativo de que la red neuronal funciona correctamente. Sin embargo, como ya hemos mencionado anteriormente, es necesaria una muestra de mayor tamaño para finalizar el entrenamiento correctamente y comprobar la fiabilidad de la red neuronal.

Distribuciones de los errores por año:

<b>Año</b>	<b>Media</b>	<b>Desviación</b>
<b>Entrenamiento</b>	0,4119	22,0288

La media del error es relativamente pequeña pero su desviación es bastante grande. En el caso de esta red neuronal no podemos sacar más conclusiones, porque no tenemos datos suficientes para poder estudiar el comportamiento con los datos de otros años que no fueran de entrenamiento.

### **Estudio del error**

Analizando la evolución de los errores en las diferentes redes neuronales podemos observar dos grupos:

1. No hay gran cambio del error: rodamiento DE y rodamiento de Alta
2. El error aumenta a medida que envejece el aerogenerador: potencia, rodamiento NDE y anillos SLD.

Una vez estudiado los errores individualmente de cada año para cada red neuronal continuamos buscando unos modelos que nos permitan entender cómo se distribuye.



#### **4. Creación de modelos de error para la detección de anomalías.**

Analizando la evolución de los errores en las diferentes redes neuronales podemos observar dos grupos:

1. No hay gran cambio del error: rodamiento DE y rodamiento de Alta
2. El error aumenta a medida que envejece el aerogenerador: potencia, rodamiento NDE y anillos SLD.

Una vez estudiado los errores individualmente de cada año para cada red neuronal continuamos buscando unos modelos que nos permitan entender cómo se distribuye.

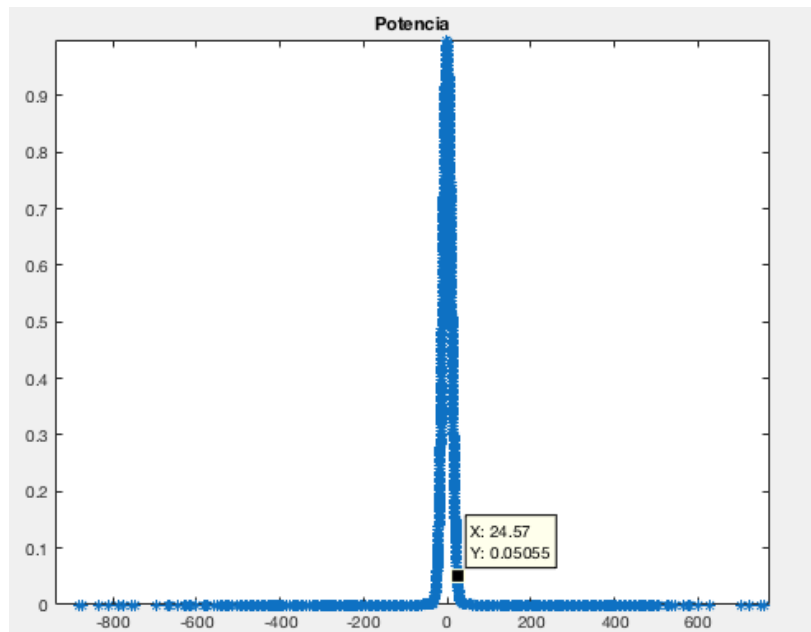
##### **4.1 Definición y determinación de anomalías.**

El objetivo final de este proyecto no es solo simular las variables elegidas con redes neuronales, si no servirse de esta simulación para poder detectar anomalías en los componentes del aerogenerador. Para ello tenemos que definir lo que consideramos error y cuantos errores repetidos implican un problema en el componente del aerogenerador.

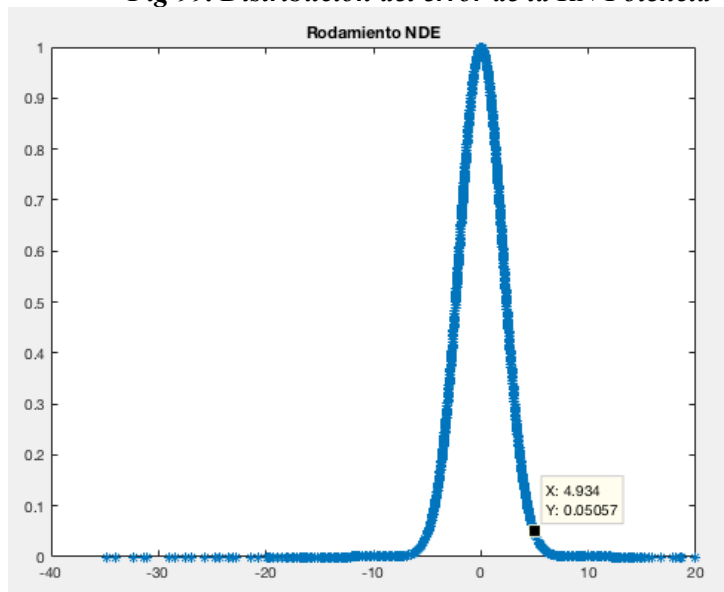
Una vez estudiados los resultados y los histogramas de error se ha decidido crear unas distribuciones normales de error de cada red neuronal. Con estas curvas podemos determinar un intervalo de confianza para cada red. Así, los datos que estén fuera del intervalo del 95% de confianza se considerarán error. Para las temperaturas el límite siempre será superior pues si las temperaturas están por debajo de la media no es preocupante para el funcionamiento salvo que sean casos muy extremos. Para realizarlos hemos utilizado los errores de las redes neuronales con los datos de entrenamiento, usando la función “mvnpdf” de Matlab.

Una vez determinados los errores se considera anomalía en el componente si se repite el error en tres muestras consecutivas, es decir, durante treinta minutos o más. Todo esto se va realizar mediante bloques función en el entorno Matlab-Simulink.

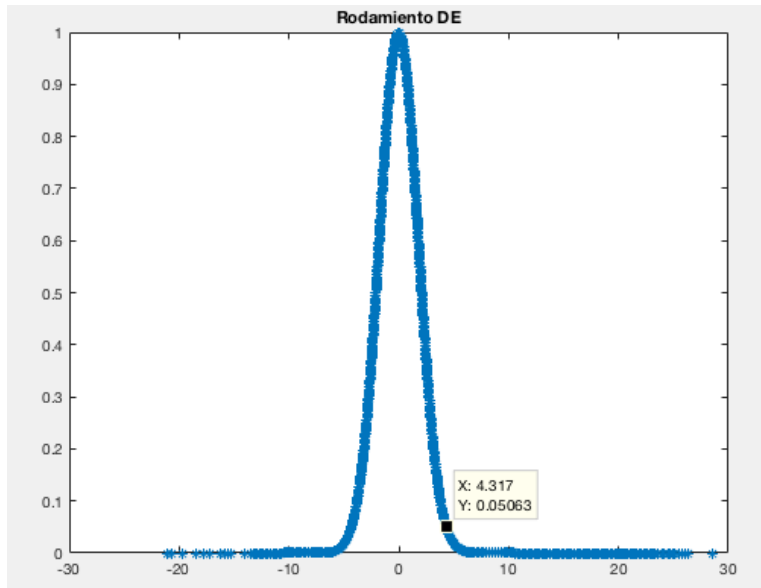
## 4.2 Gráficas normales de error:



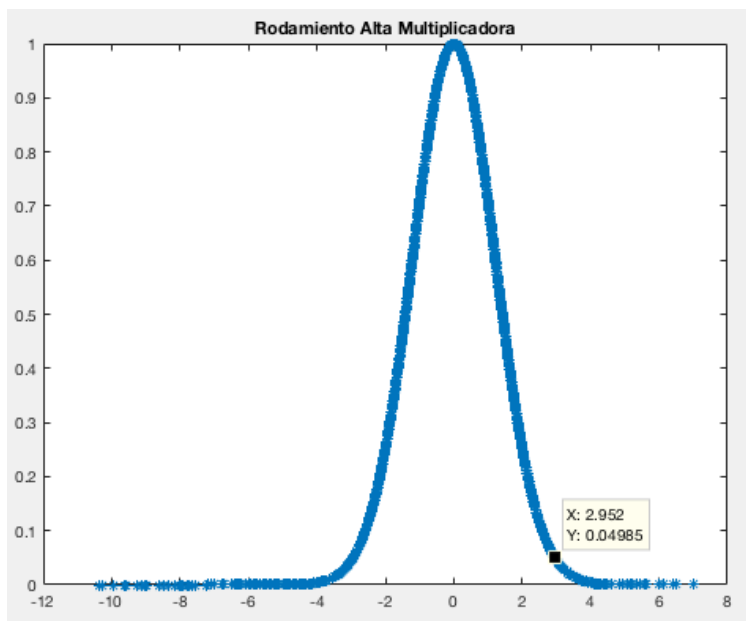
*Fig 99. Distribución del error de la RN Potencia*



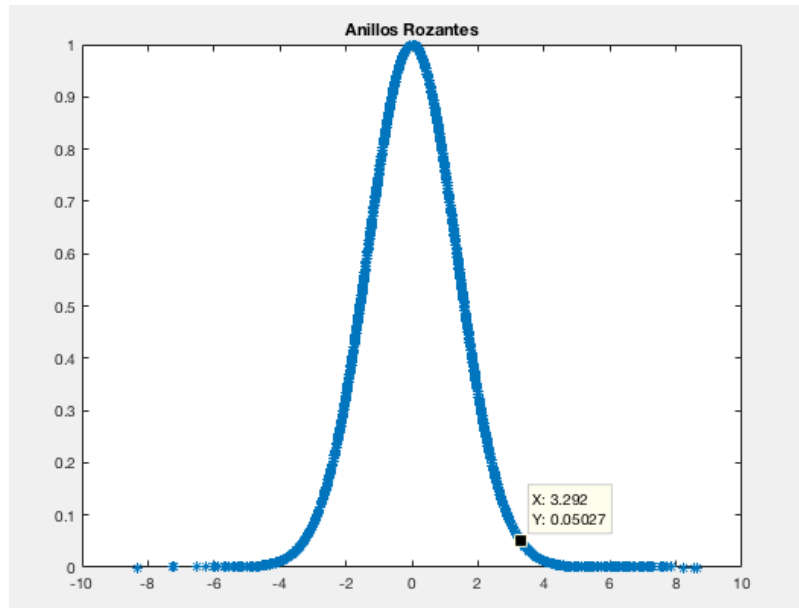
*Fig 100. Distribución del error de la RN Rodamiento NDE.*



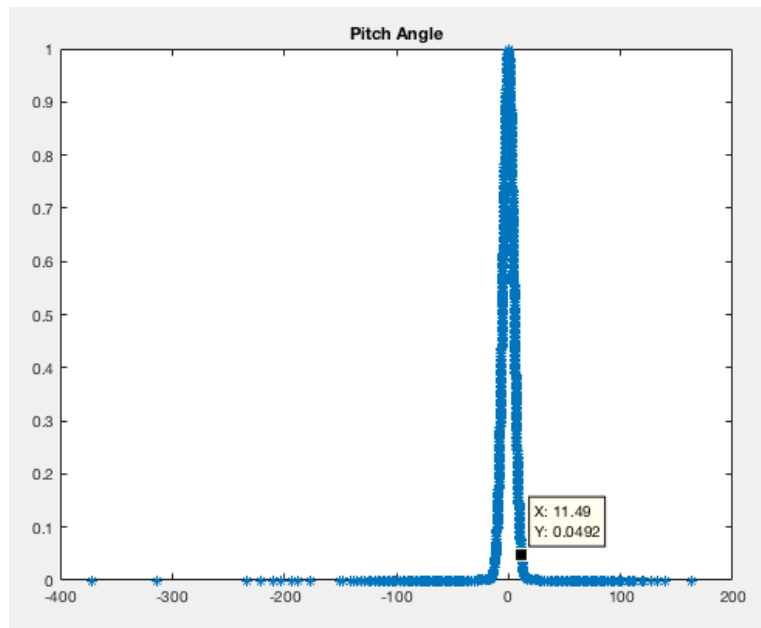
*Fig 101. Distribución del error de la RN Rodamiento DE.*



*Fig 102. Distribución del error de la RN Rodamiento Alta Multiplicadora*



*Fig 103. Distribución del error de la RN Anillos SLD.*



*Fig 104. Distribución del error de la RN PitchAngle.*

En las gráficas podemos observar la distribución del error en las redes neuronales, así como los límites usados. En la figura 106 (RN Ángulo de Paso) y la 108 (RN Potencia), que son las únicas en la que interesa el error tanto positivo como negativo ya que no es una temperatura, al ser simétricas respecto al 0 es el mismo límite por ambos lados, pero con el símbolo cambiado.

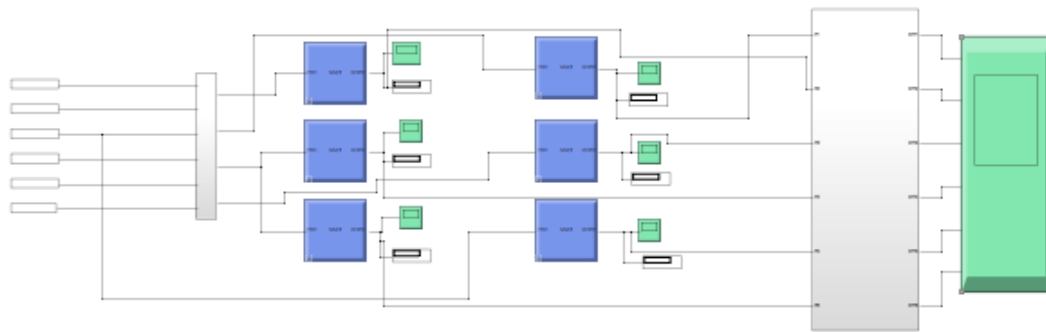
En la tabla siguiente se muestran los parámetros que llevan al programa a detectar error, y, en el caso de tres errores consecutivos, la anomalía:

Límite error NN potencia.	24,57 KW
Límite error NN rodamiento NDE.	4,934°C
Límite error NN rodamiento DE.	4,317°C
Límite error NN rodamiento alta multiplicadora.	2,952°C
Límite error NN anillos SLD.	3,292°C
Límite error NN Pitchangle.	11,49°

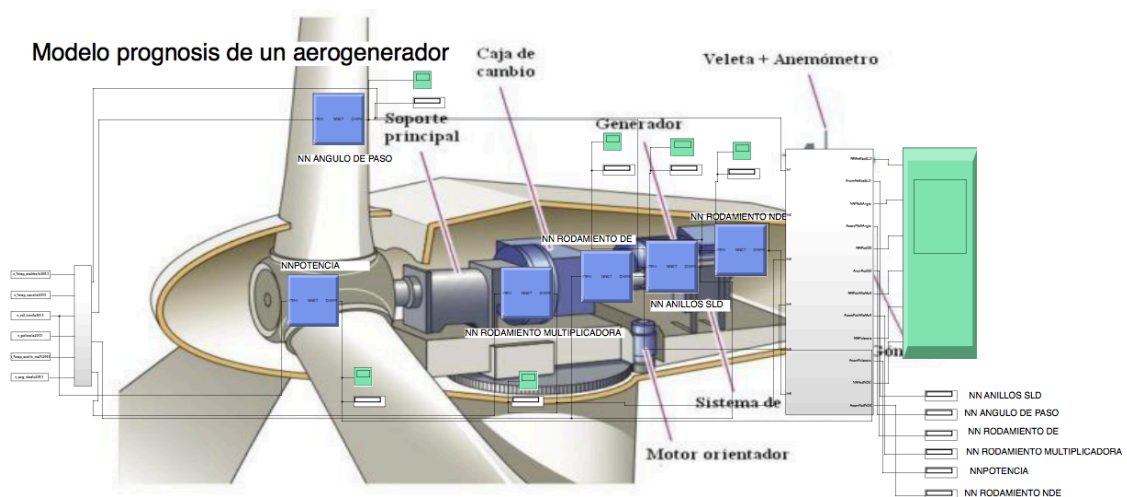


## 5. Modelo conjunto.

### 5.1 Análisis de la información que proporciona el modelo.



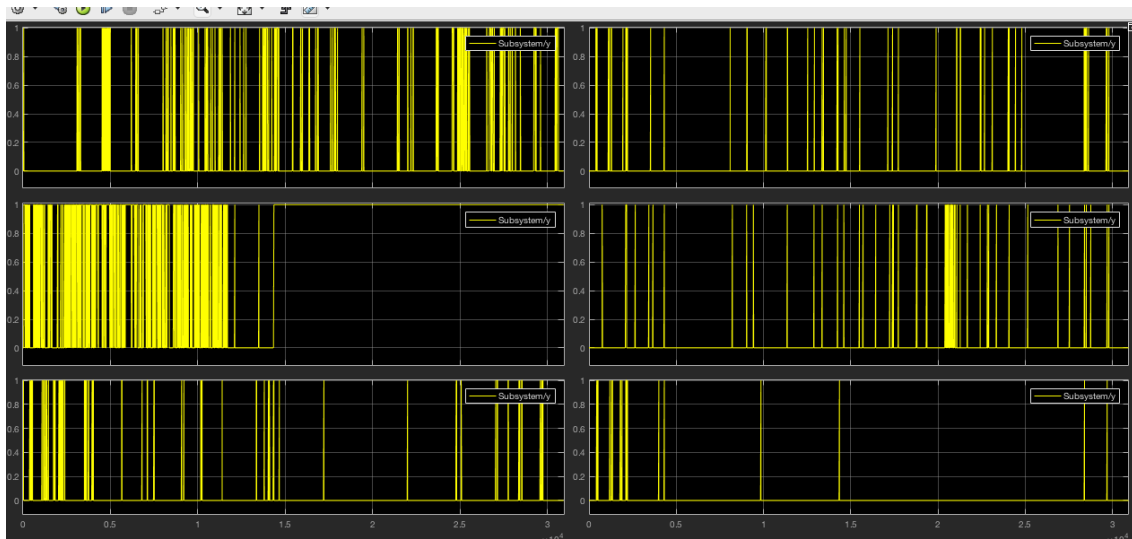
*Fig 105. Modelo esquemático*



*Fig 106. Modelo final.*

En las figuras 109 y 110 podemos ver el resultado con todas las redes neuronales en Simulink. Cada bloque red neuronal está ubicado cerca del elemento que simula. A la izquierda se encuentran los bloques de entrada y a la derecha el scope de salida. El bloque grande gris a la izquierda realiza el cálculo de los errores según el intervalo de confianza, establecido previamente, y computa las anomalías.

Los resultados en el scope se visualizan de la siguiente forma:



*Fig 107. Anomalías detectadas*

De arriba abajo y de izquierda a derecha:

- Red Neuronal Anillos SLD.
- Red Neuronal Ángulo de paso de palas.
- Red Neuronal Rodamiento DE.
- Red Neuronal Rodamiento Alta Multiplicadora.
- Red Neuronal Potencia.
- Red Neuronal Rodamiento NDE.

## 5.2 Análisis de resultados.

Como hemos podido observar durante el entrenamiento y la simulación de cada red neuronal los comportamientos y la precisión de cada una son diferentes. Por eso, algunas dan mejores resultados que otras. Como ya hemos mencionado antes los datos de 2014 eran escasos, por lo tanto, los datos del error que se obtenido no se han presentado. A continuación, se presentan las anomalías de cada red neuronal.

<b>NN Potencia</b>	<b>Anomalías por hora</b>	<b>Anomalías por año</b>
<b>2009</b>	0,0113	72
<b>2010</b>	0,0231	141
<b>2011</b>	0,0142	73
<b>2012</b>	0,0320	214
<b>2013</b>	0,0095	57
<b>TOTAL</b>	0,0161	90

Observamos un comportamiento parecido en los años 2009, 2011 y 2013 con menos anomalías por año. En el 2010 pero sobretodo en el 2012 las anomalías aumentan. Podemos ver que realmente el número de anomalías no aumenta por año, sino que se mantiene más o menos constante con fluctuaciones. A priori, no indica la existencia de un fallo.

<b>NN Rodamiento NDE</b>	<b>Anomalías por hora</b>	<b>Anomalías por año</b>
<b>2009</b>	0,0029	19
<b>2010</b>	0,0109	66
<b>2011</b>	0,0045	23
<b>2012</b>	0,0100	65
<b>2013</b>	0,0056	34
<b>Total</b>	0,0063	35

Observamos un comportamiento muy parecido entre todos los años con ligeramente más anomalías al año en los años 2010 y 2013. Podemos ver que este componente no varía su comportamiento durante los años simulados. Concluimos que el comportamiento no varía y, por lo tanto, no indica la existencia de un fallo.

<b>NN Rodamiento DE</b>	<b>Anomalías por hora</b>	<b>Anomalías por año</b>
<b>2009</b>	0,0219	113
<b>2010</b>	0,0615	376
<b>2011</b>	0,0609	392
<b>2012</b>	0,0283	189
<b>2013</b>	0,0423	254
<b>Total</b>	0,0389	218

Observamos un comportamiento muy parecido entre todos los años con ligeramente más anomalías en los años 2010 y 2011. Podemos ver que este componente no varía su comportamiento durante los años simulados. Concluimos que el comportamiento realmente no varía, pero las anomalías por año son muchas lo que puede indicar un fallo constante o una simulación poco precisa.

<b>NN Rod. Alta Multiplicadora</b>	<b>Anomalías por hora</b>	<b>Anomalías por año</b>
<b>2009</b>	0,0041	26
<b>2010</b>	0,0041	25
<b>2011</b>	0,0113	58
<b>2012</b>	0,0045	30
<b>2013</b>	0,0072	43
<b>Total</b>	0,0059	33

Observamos un comportamiento muy parecido entre todos los años con ligeramente más anomalías al año en los años 2011 y 2013. Podemos ver que este componente no varía su comportamiento durante los años simulados. Podemos concluir que el comportamiento no varía y, por lo tanto, no indica la existencia de un fallo.

<b>NN Anillos SLD</b>	<b>Anomalías por hora</b>	<b>Anomalías por año</b>
<b>2009</b>	0,0010	6
<b>2010</b>	0,0037	23
<b>2011</b>	0,0498	257
<b>2012</b>	0,0603	403
<b>2013</b>	0,0557	334
<b>Total</b>	0,0358	201

Observamos un comportamiento en el que aumentan las anomalías a medida que avanzan los años. Vemos una tendencia clara hacia el aumento de la anomalía lo que puede significar que con el paso del tiempo el componente envejezca.

<b>NN PitchAngle</b>	<b>Anomalías por hora</b>	<b>Anomalías por año</b>
-	0,0768	-

En el caso de la red neuronal del PitchAngle como ya habíamos mencionado ahora, los datos han sido insuficientes para el entrenamiento correcto de la red. Por eso, es la red con mayor tasa de anomalías con mucha diferencia.

Las dos Redes neuronales de los rodamientos NDE y del de Alta de la Multiplicadora son las que menos anomalías muestran. La red neuronal de los anillos SLD, la de la potencia y la del rodamiento de DE presentan más anomalías, pero siempre dentro de un funcionamiento normal. Es la red neuronal del ángulo de paso la que presenta un comportamiento claramente anormal con anomalía casi permanente. Esto, como ya hemos mencionado anteriormente, se debe a que por la falta de datos de PitchAngle, esta red neuronal se ha entrenado menos de lo que se debería, dejando en evidencia lo importante que es el entrenamiento en una red neuronal. Cuantos más datos, más precisa y veraz es la red neuronal, en este caso no hemos tenido una muestra suficiente.

### **5.3 Conclusión y dificultades**

Este proyecto se puede dividir claramente en tres partes.

#### **5.3.1 Entrenamiento de redes neuronales.**

Esta parte es la más metódica. Usando el propio editor de redes neuronales de Matlab, simplemente había que elegir la configuración que mejores resultados daba a cada variable simulada. Se probaron diferentes configuraciones de inputs y outputs, de capas y de distribuciones de los datos de entrenamiento.

Es evidente la importancia de los datos utilizados. La muestra no debe presentar errores, para cada unidad de tiempo utilizada, en ninguna de las variables medidas. De las 6 redes neuronales creadas la que peores resultados da, es la única cuya muestra de datos era la más reducida, lo que revela la importancia de un entrenamiento completo.

Observando las distribuciones de los errores en cada red neuronal por año podemos detectar una clara tendencia del error a incrementarse según pasa el tiempo. Esto es lógico, pues los comportamientos de los componentes empeoran según se deterioran. Comparando las evoluciones de las distribuciones de error encontramos aquellos componentes que mantienen el mismo error a lo largo de los años, lo que quiere decir, que tienen menos necesidad de ser cambiados o reparados que los otros (NN del rodamiento DE, y de alta de la multiplicadora). En el caso del Rodamiento NDE el aumento de la media del error de 0,125 a 5,019 (casi del 4000%) puede indicar una avería en el 2014, ya que en los años anteriores la media del error casi no se ve afectada.

#### **5.3.2 Integración del modelo y bloques en el entorno de Simulink.**

Para poder integrar los seis modelos realizados eran necesarios numerosos bloques de llamada de variables, así como los bloques necesarios para detectar errores y anomalías. Para mayor claridad se ha intentado integrar de la mejor manera posible cada red neuronal en un diagrama de las partes de un aerogenerador.

### **5.3.3 Estudio estadístico del error.**

Para detectar la existencia de errores se han usado distribuciones normales al considerarse que los errores de las redes neuronales se adaptaban a este tipo de modelos. Un estudio más a fondo de estos errores podría determinar mejores distribuciones, que se adaptaran con más precisión a la realidad.

Además, cabe destacar la detección de anomalías ha seguido un modelo bastante sencillo, definiendo como anomalía la existencia de un error fuera del intervalo del 95% de confianza tres veces seguidas. Esta sencillez invita a diversos estudios más en profundidad sobre otras formas más eficaces de determinar estas faltas en los componentes.

#### **Conclusión.**

Este proyecto presenta una idea teórica y desarrollada informáticamente sobre un posible modelo de predicción y detección de fallos de un aerogenerador. Hemos observado como la utilización de la red neuronal, una tecnología rápida, robusta y sencilla de manejar, aplicada a una considerable muestra de datos permite crear un modelo que nos permite:

1. Simular las variables difíciles de medir: medidas de los componentes menos accesibles del interior de la góndola, como son los rodamientos del aerogenerador o de la multiplicadora.
2. Crear unos modelos de comportamiento normal que nos permitan identificar las medidas anómalas.
3. Predecir y controlar el envejecimiento de los componentes más sensibles al fallo de un aerogenerador.

En definitiva, este modelo teórico desarrollado y llevado a la práctica nos permitiría lidiar con una de las causas de los principales problemas que presentan las energías renovables: la continuidad. Es verdad que seguiremos dependiendo de que se manifiesten en la naturaleza, pero evitaremos al máximo el error de nuestras máquinas.



## 11. Bibliografía

- [NARE11] Antonio Narejos  
«Energías renovables e instalaciones industriales. Principales averías de un aerogenerador» [1].
- [MORE16] Rafael Alberto Moreno Parra  
«Redes Neuronales. Parte 1» [2].
- [FESI17] F. Fernández y L. Sigrist, «Accionamientos Eléctricos. Modelo dinámico de la máquina asíncrona. Modelo dinámico de la máquina síncrona» [3].
- [PERS11] Perso Wanadoo  
«Redes neuronales. Introducción» [4].
- [ANGE17] Ángel Gil Ramil  
«Desarrollo de indicadores para la detección del malfuncionamiento y evaluación del estado de salud en equipos monitorizados. Caso real de aplicación a aerogeneradores.» [5].
- [DROM11] Dromstorre. Asociación danesa de la industria eólica.  
«Seguridad en aerogeneradores. Sistema de freno» [6].
- [FESI17] F. Fernández y L. Sigrist, «Accionamientos Eléctricos. Esquemas de control vectorial para accionamientos» [7].

[FESI17] F. Fernández y L. Sigríst, «Accionamientos Eléctricos. Introducción al control de aerogeneradores y FACTS» [8].

[FESI17] F. Fernández y L. Sigríst, «Máquina síncrona de polos salientes. Ejes dq. Vectores Espaciales» [9].

## Anexo. Código

```
temp_aceite_multi=data_struct.temp_aceite_multi;
temp_rod_alta_multi=data_struct.temp_rod_alta_multi;
temp_rod_DE_gen=data_struct.temp_rod_DE_gen;
temp_rod_NDE_gen=data_struct.temp_rod_NDE_gen;
temp_anillos_sld_gen=data_struct.temp_anillos_sld_gen;
pres_grupo_h=data_struct.pres_grupo_h;
temp_nacelle=data_struct.temp_interior_nac;
temp_ambiente=data_struct.temp_ambiente;
w_eje_principal=data_struct.w_eje_principal;
vel_viento=data_struct.vel_viento;
ang_viento=data_struct.ang_viento;
v_red=data_struct.v_red;
pot_reactiva=data_struct.pot_reactiva;
pot_act_est=data_struct.pot_act_est;
pot_act_rot=data_struct.pot_act_rot;
potencia=data_struct.pot_total;
energia=data_struct.energia;
state=data_struct.state;

x=pres_grupo_h;
y=(w_eje_principal, vel_viento);

[ranked,weights]=relieff(x,y)

clear all
close all

load('data2009.mat');

%[nsamples m]=size(data);
data_struct=data2009.Generador;

% temp_aceite_multi=data_struct.temp_aceite_multi;
% temp_rod_alta_multi=data_struct.temp_rod_alta_multi;
temp_rod_DE_gen=data_struct.temp_rod_DE_gen;
temp_rod_NDE_gen=data_struct.temp_rod_NDE_gen;
% temp_anillos_sld_gen=data_struct.temp_anillos_sld_gen;
% pres_grupo_h=data_struct.pres_grupo_h;
temp_nacelle=data_struct.temp_interior_nac;
temp_ambiente=data_struct.temp_ambiente;
% w_eje_principal=data_struct.w_eje_principal;
vel_viento=data_struct.vel_viento;
% ang_viento=data_struct.ang_viento;
% v_red=data_struct.v_red;
% pot_reactiva=data_struct.pot_reactiva;
% pot_act_est=data_struct.pot_act_est;
% pot_act_rot=data_struct.pot_act_rot;
potencia=data_struct.pot_total;
% energia=data_struct.energia;
% state=data_struct.state;

[nsamples m]=size(potencia);
t=1:nsamples;

%plot(t, data(:,1), t, data(:,2), t, data(:,3))

figure(1)
plot(t, vel_viento, t, temp_rod_DE_gen, t, temp_rod_NDE_gen, t, temp_nacelle)
ylabel('Magnitude'); xlabel('Sample');
legend('vel_wind', 'tem_bearing_DE_gen', 'tem_bearing_NDE_gen', 'tem_nacelle');
title('Variables');

training=38000;
X=zeros(training,4); %Only first training input samples
X(:,1)=potencia(1:training); % Power
X(:,2)=vel_viento(1:training); % Wind speed
X(:,3)=temp_nacelle(1:training); %Nacelle Temperature
X(:,4)=temp_ambiente(1:training,1); %temperature in the bearing NDE

T=zeros(training,1); %Only first training target output samples
T(:,1)=temp_rod_DE_gen(1:training,1); %temperature in the bearing DE

%T(:,2)=data(1:june,9); %oil pressure guide vane position
```

```

%T=data(1:june,9); %pressure guide vane position
T=T';
X=X';

net = feedforwardnet([30 10], 'trainlm');
net = configure(net,X,T);

%Criteria for splitting the training set

net.divideFcn='dividerand';
net.divideParam.trainRatio = 60/100;%60,20,20
net.divideParam.valRatio = 20/100;
net.divideParam.testRatio = 20/100;

% Inicializaci3n de pesos
net=init(net);
% Se ajustan diversos par3metros del entrenamiento

net.trainParam.epochs = 10000;
net.trainParam.lr = 0.002;
net.trainParam.show = 50;
net.trainParam.goal = 1e-5;
net.trainParam.max_fail =40; % Failures allowed in validation

[net, traza] = train(net,X,T);
%view(net)
Yp = net(X);
perf = perform(net,Yp,X);

% Training error representation
figure(2)
plot(traza.epoch, traza.perf, traza.epoch, traza.vperf, traza.epoch, traza.tperf);
legend('training', 'Validation', 'Test');
ylabel('Cuadratic Error');
xlabel('Epoch');

% Prediction of the NN with training set

P = sim(net,X);

% Representation predicted and real values in the output

figure(3);
%plot(X,T, 'sr',X,Z,'X')
plot(1:training, T(1,:),1:training, P(1,:))
ylabel('Temp Bearing'); xlabel('Sample');
legend('Real Value', 'Estimated Value');
title(' Neural network test using the training set');

figure(4)
histogram(T(1,:)-P(1,:))
ylabel('Frecuency'); xlabel('Samples');
title(' Histogram of error with training data');

datos= [T-P];
[mediaentreno,desviacionentreno] = normfit(datos);
% Se va a obtener la respuesta que dar3a la red entrenada cuando se le
% pasan como entradas casos cuya salida se desconoce

Xt=zeros(nsamples-training,4); %Only first training samples
Xt(:,1)=potencia(training+1:nsamples); % Power
Xt(:,2)=vel_viento(training+1:nsamples); % Wind speed
Xt(:,3)=temp_nacelle(training+1:nsamples); %Nacelle Temperature
Xt(:,4)=temp_ambiente(training+1:nsamples); %temperature in the bearing NDE

Tt=zeros(nsamples-training,1); %Only first 4000 samples
Tt(:,1)=temp_rod_DE_gen(training+1:nsamples);
%Tt(:,2)=data(june+1:nsamples,9);
%Tt=data(june+1:nsamples,9);
Tt=Tt';
Xt=Xt';

Pt = sim(net,Xt);

% Se representa la salida estimada por la red para las entradas dadas y
% tambi3n lo que tendr3a que haber dado

figure(30);
%plot(X,T, 'sr',X,Z,'X')

```

```

plot(1:nsamples-training, Tt(1,:),1:nsamples-training, Pt(1,:))
ylabel('Temperature 2009 %'); xlabel('Sample');
legend('Real Value', 'Estimated Value');
title('Neural network test with unknown cases');

figure(31)
histogram(Tt(1,:)-Pt(1,:))
ylabel('Frecuency 2009'); xlabel('Samples');
title('Histogram of error with unknown data');

%%
%-----Probar2010-----
load('data2010');

data_struct2010=data2010.Generador;

% temp_aceite_multi2010=data_struct2010.temp_aceite_multi;
% temp_rod_alta_multi2010=data_struct2010.temp_rod_alta_multi;
temp_rod_DE_gen2010=data_struct2010.temp_rod_DE_gen;
temp_rod_NDE_gen2010=data_struct2010.temp_rod_NDE_gen;
% temp_anillos_sld_gen2010=data_struct2010.temp_anillos_sld_gen;
% pres_grupo_h2010=data_struct2010.pres_grupo_h;
temp_nacelle2010=data_struct2010.temp_interior_nac;
temp_ambiente2010=data_struct2010.temp_ambiente;
% w_eje_principal2010=data_struct2010.w_eje_principal;
vel_viento2010=data_struct2010.vel_viento;
% ang_viento2010=data_struct2010.ang_viento;
% v_red2010=data_struct2010.v_red;
% pot_reactiva2010=data_struct2010.pot_reactiva;
% pot_act_est2010=data_struct2010.pot_act_est;
% pot_act_rot2010=data_struct2010.pot_act_rot;
potencia2010=data_struct2010.pot_total;
% energia2010=data_struct2010.energia;
% state2010=data_struct2010.state;

[nsamples m]=size(potencia2010);
t=1:nsamples;

Xt=zeros(nsamples,3); %Only first training samples
Xt(:,1)=potencia2010(1:nsamples); % Power
Xt(:,2)=vel_viento2010(1:nsamples); % Wind speed
Xt(:,3)=temp_nacelle2010(1:nsamples); %Nacele Temperature
Xt(:,4)=temp_ambiente2010(1:nsamples); %temperature in the bearing NDE

Tt=zeros(nsamples,1); %Only first 4000 samples
Tt(:,1)=temp_rod_DE_gen2010(1:nsamples);
%Tt(:,2)=data(june+1:nsamples,9);
%Tt=data(june+1:nsamples,9);
Tt=Tt';
Xt=Xt';

Pt = sim(net,Xt);

% Se representa la salida estimada por la red para las entradas dadas y
% tambii¿n lo que tendri¿a que haber dado

figure(32);
%plot(X,T, 'sr',X,Z,'X')
plot(1:nsamples, Tt(1,:),1:nsamples, Pt(1,:))
ylabel('Temperature 2010 %'); xlabel('Sample');
legend('Real Value', 'Estimated Value');
title('Neural network test with 2010 data');

figure(33)
histogram(Tt(1,:)-Pt(1,:))
ylabel('Frecuency 2010'); xlabel('Samples');
title('Histogram of error with 2010 data');

datos= [Tt-Pt];
[media2010,desviacion2010] = normfit(datos);

%%
%-----Probar2011-----
load('data2011');

data_struct2011=data2011.Generador;

% temp_aceite_multi2011=data_struct2011.temp_aceite_multi;
% temp_rod_alta_multi2011=data_struct2011.temp_rod_alta_multi;
temp_rod_DE_gen2011=data_struct2011.temp_rod_DE_gen;
temp_rod_NDE_gen2011=data_struct2011.temp_rod_NDE_gen;
% temp_anillos_sld_gen2011=data_struct2011.temp_anillos_sld_gen;
% pres_grupo_h2011=data_struct2011.pres_grupo_h;

```

```

temp_nacelle2011=data_struct2011.temp_interior_nac;
temp_ambiente2011=data_struct2011.temp_ambiente;
% w_eje_principal2011=data_struct2011.w_eje_principal;
vel_viento2011=data_struct2011.vel_viento;
% ang_viento2011=data_struct2011.ang_viento;
% v_red2011=data_struct2011.v_red;
% pot_reactiva=2011data_struct2011.pot_reactiva;
% pot_act_est2011=data_struct2011.pot_act_est;
% pot_act_rot2011=data_struct2011.pot_act_rot;
potencia2011=data_struct2011.pot_total;
% energia2011=data_struct2011.energia;
% state2011=data_struct2011.state;

[nsamples m]=size(potencia2011);
t=1:nsamples;

Xt=zeros(nsamples,4); %Only first training samples
Xt(:,1)=potencia2011(1:nsamples); % Power
Xt(:,2)=vel_viento2011(1:nsamples); % Wind speed
Xt(:,3)=temp_nacelle2011(1:nsamples); %Nacelle Temperature
Xt(:,4)=temp_ambiente2011(1:nsamples); %temperature in the bearing NDE

Tt=zeros(nsamples,1); %Only first 4000 samples
Tt(:,1)=temp_rod_DE_gen2011(1:nsamples);
%Tt(:,2)=data(june+1:nsamples,9);
%Tt=data(june+1:nsamples,9);
Tt=Tt';
Xt=Xt';

Pt = sim(net,Xt);

% Se representa la salida estimada por la red para las entradas dadas y
% tambii&#n lo que tendri&#a que haber dado

figure(34);
%plot(X,T, 'sr',X,Z,'X')
plot(1:nsamples, Tt(1,:),1:nsamples, Pt(1,:))
ylabel('Temperature 2011 %'); xlabel('Sample');
legend('Real Value', 'Estimated Value');
title(' Neural network test with 2011 data');

figure(35)
histogram(Tt(1,:)-Pt(1,:))
ylabel('Frecuency 2011'); xlabel('Samples');
title(' Histogram of error with 2011 data');

datos= [Tt-Pt];
[media2011,desviacion2011] = normfit(datos);

%%
%-----Probar2012-----
load('data2012');

data_struct2012=data2012.Generador;

% temp_aceite_multi2012=data_struct2012.temp_aceite_multi;
% temp_rod_alta_multi2012=data_struct2012.temp_rod_alta_multi;
temp_rod_DE_gen2012=data_struct2012.temp_rod_DE_gen;
temp_rod_NDE_gen2012=data_struct2012.temp_rod_NDE_gen;
% temp_anillos_sld_gen2012=data_struct2012.temp_anillos_sld_gen;
% pres_grupo_h2012=data_struct2012.pres_grupo_h;
temp_nacelle2012=data_struct2012.temp_interior_nac;
temp_ambiente2012=data_struct2012.temp_ambiente;
% w_eje_principal2012=data_struct2012.w_eje_principal;
vel_viento2012=data_struct2012.vel_viento;
% ang_viento2012=data_struct2012.ang_viento;
% v_red2012=data_struct2012.v_red;
% pot_reactiva2012=data_struct2012.pot_reactiva;
% pot_act_est2012=data_struct2012.pot_act_est;
% pot_act_rot2012=data_struct2012.pot_act_rot;
potencia2012=data_struct2012.pot_total;
% energia2012=data_struct2012.energia;
% state2012=data_struct2012.state;

[nsamples m]=size(potencia2012);
t=1:nsamples;

Xt=zeros(nsamples,4); %Only first training samples
Xt(:,1)=potencia2012(1:nsamples); % Power
Xt(:,2)=vel_viento2012(1:nsamples); % Wind speed
Xt(:,3)=temp_nacelle2012(1:nsamples); %Nacelle Temperature
Xt(:,4)=temp_ambiente2012(1:nsamples); %temperature in the bearing NDE

Tt=zeros(nsamples,1); %Only first 4000 samples

```

```

Tt(:,1)=temp_rod_DE_gen2012(1:nsamples);
%Tt(:,2)=data(june+1:nsamples,9);
%Tt=data(june+1:nsamples,9);
Tt=Tt';
Xt=Xt';

Pt = sim(net,Xt);

% Se representa la salida estimada por la red para las entradas dadas y
% tambii¿n lo que tendrí¿a que haber dado

figure(36);
%plot(X,T, 'sr',X,Z,'X')
plot(1:nsamples, Tt(1,:),1:nsamples, Pt(1,:))
ylabel('Temperature 2012 %'); xlabel('Sample');
legend('Real Value', 'Estimated Value');
title(' Neural network test with 2012 data');

figure(37)
histogram(Tt(1:)-Pt(1:))
ylabel('Frecuency 2012'); xlabel('Samples');
title(' Histogram of error with 2012 data');

datos= [Tt-Pt];
[media2012,desviacion2012] = normfit(datos);

%%
%-----Probar2013-----
load('data2013');

data_struct2013=data2013.Generador;

% temp_aceite_multi2013=data_struct2013.temp_aceite_multi;
% temp_rod_alta_multi2013=data_struct2013.temp_rod_alta_multi;
temp_rod_DE_gen2013=data_struct2013.temp_rod_DE_gen;
temp_rod_NDE_gen2013=data_struct2013.temp_rod_NDE_gen;
% temp_anillos_sld_gen2013=data_struct2013.temp_anillos_sld_gen;
% pres_grupo_h2013=data_struct2013.pres_grupo_h;
temp_nacelle2013=data_struct2013.temp_interior_nac;
temp_ambiente2013=data_struct2013.temp_ambiente;
% w_eje_principal2013=data_struct2013.w_eje_principal;
vel_viento2013=data_struct2013.vel_viento;
% ang_viento2013=data_struct2013.ang_viento;
% v_red2013=data_struct2013.v_red;
% pot_reactiva2013=data_struct2013.pot_reactiva;
% pot_act_est2013=data_struct2013.pot_act_est;
% pot_act_rot2013=data_struct2013.pot_act_rot;
potencia2013=data_struct2013.pot_total;
% energia2013=data_struct2013.energia;
% state2013=data_struct2013.state;

[nsamples m]=size(potencia2013);
t=1:nsamples;

Xt=zeros(nsamples,4); %Only first training samples
Xt(:,1)=potencia2013(1:nsamples); % Power
Xt(:,2)=vel_viento2013(1:nsamples); % Wind speed
Xt(:,3)=temp_nacelle2013(1:nsamples); %Nacele Temperature
Xt(:,4)=temp_ambiente2013(1:nsamples); %temperature in the bearing NDE

Tt=zeros(nsamples,1); %Only first 4000 samples
Tt(:,1)=temp_rod_DE_gen2013(1:nsamples);
%Tt(:,2)=data(june+1:nsamples,9);
%Tt=data(june+1:nsamples,9);
Tt=Tt';
Xt=Xt';

Pt = sim(net,Xt);

% Se representa la salida estimada por la red para las entradas dadas y
% tambii¿n lo que tendrí¿a que haber dado

figure(38);
%plot(X,T, 'sr',X,Z,'X')
plot(1:nsamples, Tt(1,:),1:nsamples, Pt(1,:))
ylabel('Temperature 2013 %'); xlabel('Sample');
legend('Real Value', 'Estimated Value');
title(' Neural network test with 2013 data');

figure(39)
histogram(Tt(1:)-Pt(1:))
ylabel('Frecuency 2013'); xlabel('Samples');
title(' Histogram of error with 2013 data');

```

```

datos= [Tt-Pt];
[media2013,desviacion2013] = normfit(datos);

%%
%-----Probar2014-----
load('data2014');

data_struct2014=data2014.Generador;

% temp_aceite_multi2014=data_struct2014.temp_aceite_multi;
% temp_rod_alta_multi2014=data_struct2014.temp_rod_alta_multi;
temp_rod_DE_gen2014=data_struct2014.temp_rod_DE_gen;
temp_rod_NDE_gen2014=data_struct2014.temp_rod_NDE_gen;
% temp_anillos_sld_gen2014=data_struct2014.temp_anillos_sld_gen;
% pres_grupo_h2014=data_struct2014.pres_grupo_h;
temp_nacelle2014=data_struct2014.temp_interior_nac;
temp_ambiente2014=data_struct2014.temp_ambiente;
% w_eje_principal2014=data_struct2014.w_eje_principal;
vel_viento2014=data_struct2014.vel_viento;
% ang_viento2014=data_struct2014.ang_viento;
% v_red2014=data_struct2014.v_red;
% pot_reactiva2014=data_struct2014.pot_reactiva;
% pot_act_est2014=data_struct2014.pot_act_est;
% pot_act_rot2014=data_struct2014.pot_act_rot;
potencia2014=data_struct2014.pot_total;
% energia2014=data_struct2014.energia;
% state2014=data_struct2014.state;

[nsamples m]=size(potencia2014);
t=1:nsamples;

Xt=zeros(nsamples,4); %Only first training samples
Xt(:,1)=potencia2014(1:nsamples); % Power
Xt(:,2)=vel_viento2014(1:nsamples); % Wind speed
Xt(:,3)=temp_nacelle2014(1:nsamples); %Nacele Temperature
Xt(:,4)=temp_ambiente2014(1:nsamples); %temperature in the bearing NDE

Tt=zeros(nsamples,1); %Only first 4000 samples
Tt(:,1)=temp_rod_DE_gen2014(1:nsamples);
%Tt(:,2)=data(june+1:nsamples,9);
%Tt=data(june+1:nsamples,9);
Tt=Tt';
Xt=Xt';

Pt = sim(net,Xt);

% Se representa la salida estimada por la red para las entradas dadas y
% tambii¿n lo que tendrí¿a que haber dado

figure(40);
%plot(X,T, 'sr',X,Z,'X')
plot(1:nsamples, Tt(1,:),1:nsamples, Pt(1,:))
ylabel('Temperature 2014 %'); xlabel('Sample');
legend('Real Value', 'Estimated Value');
title(' Neural network test with 2014 data');

figure(41)
histogram(Tt(1:)-Pt(1,:))
ylabel('Frecuency 2014'); xlabel('Samples');
title(' Histogram of error with 2014 data');

datos= [Tt-Pt];
[media2014,desviacion2014] = normfit(datos);

clear all
close all

load('data2009.mat');

% [nsamples m]=size(data);
data_struct=data2009.Generador;

%temp_aceite_multi=data_struct.temp_aceite_multi;
% temp_rod_alta_multi=data_struct.temp_rod_alta_multi;
% temp_rod_DE_gen=data_struct.temp_rod_DE_gen;
temp_rod_NDE_gen=data_struct.temp_rod_NDE_gen;
% temp_anillos_sld_gen=data_struct.temp_anillos_sld_gen;
% pres_grupo_h=data_struct.pres_grupo_h;
temp_nacelle=data_struct.temp_interior_nac;
temp_ambiente=data_struct.temp_ambiente;
% w_eje_principal=data_struct.w_eje_principal;

```

```

vel_viento=data_struct.vel_viento;
% ang_viento=data_struct.ang_viento;
% v_red=data_struct.v_red;
% pot_reactiva=data_struct.pot_reactiva;
% pot_act_est=data_struct.pot_act_est;
% pot_act_rot=data_struct.pot_act_rot;
potencia=data_struct.pot_total;
% energia=data_struct.energia;
% state=data_struct.state;

[nsamples m]=size(potencia);
t=1:nsamples;

%plot(t, data(:,1), t, data(:,2), t, data(:,3))

figure(1)
plot(t, vel_viento, t, temp_rod_NDE_gen, t, temp_nacelle, t, temp_ambiente)
ylabel('Magnitude'); xlabel('Sample');
legend('vel_wind', 'tem_bearing_NDE_gen', 'tem_nacelle', 'tem_ambiente');
title(' Variables');

training=30000;
X=zeros(training,4); %Only first training input samples
X(:,1)=potencia(1:training); % Power
X(:,2)=vel_viento(1:training); % Wind speed
X(:,3)=temp_nacelle(1:training); %Nacelle Temperature
X(:,4)=temp_ambiente(1:training,1); %temperature in the bearing DE

T=zeros(training,1); %Only first training target output samples
T(:,1)=temp_rod_NDE_gen(1:training,1); %temperature in the bearing DE

% T(:,2)=data(1:june,9); %oil pressure guide vane position
% T=data(1:june,9); %pressure guide vane position
T=T';
X=X';

net = feedforwardnet([30 10], 'trainlm');
net = configure(net,X,T);

%Criteria for splitting the training set

net.divideFcn='dividerand';
net.divideParam.trainRatio = 60/100;%60,20,20
net.divideParam.valRatio = 20/100;
net.divideParam.testRatio = 20/100;

%Inicializaci3n de pesos
net=init(net);
%Se ajustan diversos par3metros del entrenamiento

net.trainParam.epochs = 10000;
net.trainParam.lr = 0.002;
net.trainParam.show = 50;
net.trainParam.goal = 1e-5;
net.trainParam.max_fail =40; % Failures allowed in validation

[net, traza] = train(net,X,T);
view(net)
Yp = net(X);
perf = perform(net,Yp,X);

%Training error representation
figure(2)
plot(traza.epoch, traza.perf, traza.epoch, traza.vperf, traza.epoch, traza.tperf);
legend('training', 'Validation', 'Test');
ylabel('Quadratic Error');
xlabel('Epoch');

%Prediction of the NN with training set

P = sim(net,X);

%Representation predicted and real values in the output

figure(3);
%plot(X,T, 'sr',X,Z,'X')
plot(1:training, T(1,:),1:training, P(1,:))

```

```

ylabel('Temp NBearing'); xlabel('Sample');
legend('Real Value', 'Estimated Value');
title(' Neural network test using the training set');

figure(4)
histogram(T(1,:)-P(1,:))
ylabel('Frecuency'); xlabel('Samples');
title(' Histogram of error with training data');

datos= [T-P];
[mediaentrenamiento,desviacionentrenamiento] = normfit(datos);

%Se va a obtener la respuesta que daría la red entrenada cuando se le
%pasan como entradas casos cuya salida se desconoce

Xt=zeros(nsamples-training,4); %Only first training samples
Xt(:,1)=potencia(training+1:nsamples); % Power
Xt(:,2)=vel_viento(training+1:nsamples); % Wind speed
Xt(:,3)=temp_nacelle(training+1:nsamples); %Nacelle Temperature
Xt(:,4)=temp_ambiente(training+1:nsamples); %temperature in the bearing DE

Tt=zeros(nsamples-training,1); %Only first 4000 samples
Tt(:,1)=temp_rod_NDE_gen(training+1:nsamples);
% Tt(:,2)=data(june+1:nsamples,9);
% Tt(:,3)=data(june+1:nsamples,9);
Tt=Tt';
Xt=Xt';

Pt = sim(net,Xt);

%Se representa la salida estimada por la red para las entradas dadas y
%también lo que tendría que haber dado

figure(30);
%plot(X,T, 'sr',X,Z,'X')
plot(1:nsamples-training, Tt(1,:),1:nsamples-training, Pt(1,:))
ylabel('Temperature 2009 %'); xlabel('Sample');
legend('Real Value', 'Estimated Value');
title(' Neural network test with unknown cases');

figure(31)
histogram(Tt(1,:)-Pt(1,:))
ylabel(' Frecuency 2009 '); xlabel('Samples');
title(' Histogram of error with unknown data');

%%
%-----Probar2010-----
load('data2010');

data_struct2010=data2010.Generador;

% temp_aceite_multi2010=data_struct2010.temp_aceite_multi;
% temp_rod_alta_multi2010=data_struct2010.temp_rod_alta_multi;
%temp_rod_DE_gen2010=data_struct2010.temp_rod_DE_gen;
temp_rod_NDE_gen2010=data_struct2010.temp_rod_NDE_gen;
% temp_anillos_sld_gen2010=data_struct2010.temp_anillos_sld_gen;
% pres_grupo_h2010=data_struct2010.pres_grupo_h;
temp_nacelle2010=data_struct2010.temp_interior_nac;
temp_ambiente2010=data_struct2010.temp_ambiente;
% w_eje_principal2010=data_struct2010.w_eje_principal;
vel_viento2010=data_struct2010.vel_viento;
% ang_viento2010=data_struct2010.ang_viento;
% v_red2010=data_struct2010.v_red;
% pot_reactiva2010=data_struct2010.pot_reactiva;
% pot_act_est2010=data_struct2010.pot_act_est;
% pot_act_rot2010=data_struct2010.pot_act_rot;
potencia2010=data_struct2010.pot_total;
% energia2010=data_struct2010.energia;
% state2010=data_struct2010.state;

[nsamples m]=size(potencia2010);
t=1:nsamples;

Xt=zeros(nsamples,4); %Only first training samples
Xt(:,1)=potencia2010(1:nsamples); % Power
Xt(:,2)=vel_viento2010(1:nsamples); % Wind speed
Xt(:,3)=temp_nacelle2010(1:nsamples); %Nacelle Temperature
Xt(:,4)=temp_ambiente2010(1:nsamples); %temperature in the bearing DE

Tt=zeros(nsamples,1); %Only first 4000 samples
Tt(:,1)=temp_rod_NDE_gen2010(1:nsamples);
%Tt(:,2)=data(june+1:nsamples,9);
%Tt(:,3)=data(june+1:nsamples,9);

```

```

Tt=Tt';
Xt=Xt';

Pt = sim(net,Xt);

% Se representa la salida estimada por la red para las entradas dadas y
% también lo que tendríãa que haber dado

figure(32);
%plot(X,T, 'sr',X,Z,'X')
plot(1:nsamples, Tt(1,:),1:nsamples, Pt(1,:))
ylabel('Temperature 2010 %'); xlabel('Sample');
legend('Real Value', 'Estimated Value');
title(' Neural network test with 2010 data');

figure(33)
histogram(Tt(1,:)-Pt(1,:))
ylabel('Frecuency 2010'); xlabel('Samples');
title(' Histogram of error with 2010 data');

datos= [Tt-Pt];
[media2010,desviacion2010] = normfit(datos);

%%
%-----Probar2011-----
load('data2011');

data_struct2011=data2011.Generador;

% temp_aceite_multi2011=data_struct2011.temp_aceite_multi;
% temp_rod_alta_multi2011=data_struct2011.temp_rod_alta_multi;
% temp_rod_DE_gen2011=data_struct2011.temp_rod_DE_gen;
temp_rod_NDE_gen2011=data_struct2011.temp_rod_NDE_gen;
% temp_anillos_sld_gen2011=data_struct2011.temp_anillos_sld_gen;
% pres_grupo_h2011=data_struct2011.pres_grupo_h;
temp_nacelle2011=data_struct2011.temp_interior_nac;
temp_ambiente2011=data_struct2011.temp_ambiente;
% w_eje_principal2011=data_struct2011.w_eje_principal;
vel_viento2011=data_struct2011.vel_viento;
% ang_viento2011=data_struct2011.ang_viento;
% v_red2011=data_struct2011.v_red;
% pot_reactiva=2011data_struct2011.pot_reactiva;
% pot_act_est2011=data_struct2011.pot_act_est;
% pot_act_rot2011=data_struct2011.pot_act_rot;
potencia2011=data_struct2011.pot_total;
% energia2011=data_struct2011.energia;
% state2011=data_struct2011.state;

[nsamples m]=size(potencia2011);
t=1:nsamples;

Xt=zeros(nsamples,4); %Only first training samples
Xt(:,1)=potencia2011(1:nsamples); % Power
Xt(:,2)=vel_viento2011(1:nsamples); % Wind speed
Xt(:,3)=temp_nacelle2011(1:nsamples); %Nacele Temperature
Xt(:,4)=temp_ambiente2011(1:nsamples); %temperature in the bearing DE

Tt=zeros(nsamples,1); %Only first 4000 samples
Tt(:,1)=temp_rod_NDE_gen2011(1:nsamples);
%Tt(:,2)=data(june+1:nsamples,9);
%Tt=data(june+1:nsamples,9);
Tt=Tt';
Xt=Xt';

Pt = sim(net,Xt);

% Se representa la salida estimada por la red para las entradas dadas y
% también lo que tendríãa que haber dado

figure(34);
%plot(X,T, 'sr',X,Z,'X')
plot(1:nsamples, Tt(1,:),1:nsamples, Pt(1,:))
ylabel('Temperature 2011 %'); xlabel('Sample');
legend('Real Value', 'Estimated Value');
title(' Neural network test with 2011 data');

figure(35)
histogram(Tt(1,:)-Pt(1,:))
ylabel('Frecuency 2011'); xlabel('Samples');
title(' Histogram of error with 2011 data');

datos= [Tt-Pt];
[media2011,desviacion2011] = normfit(datos);

```

```

%%
%-----Probar2012-----
load('data2012');

data_struct2012=data2012.Generador;

% temp_aceite_multi2012=data_struct2012.temp_aceite_multi;
% temp_rod_alta_multi2012=data_struct2012.temp_rod_alta_multi;
% temp_rod_DE_gen2012=data_struct2012.temp_rod_DE_gen;
temp_rod_NDE_gen2012=data_struct2012.temp_rod_NDE_gen;
% temp_anillos_sld_gen2012=data_struct2012.temp_anillos_sld_gen;
% pres_grupo_h2012=data_struct2012.pres_grupo_h;
temp_nacelle2012=data_struct2012.temp_interior_nac;
temp_ambiente2012=data_struct2012.temp_ambiente;
% w_eje_principal2012=data_struct2012.w_eje_principal;
vel_viento2012=data_struct2012.vel_viento;
% ang_viento2012=data_struct2012.ang_viento;
% v_red2012=data_struct2012.v_red;
% pot_reactiva2012=data_struct2012.pot_reactiva;
% pot_act_est2012=data_struct2012.pot_act_est;
% pot_act_rot2012=data_struct2012.pot_act_rot;
potencia2012=data_struct2012.pot_total;
% energia2012=data_struct2012.energia;
% state2012=data_struct2012.state;

[nsamples m]=size(potencia2012);
t=1:nsamples;

Xt=zeros(nsamples,4); %Only first training samples
Xt(:,1)=potencia2012(1:nsamples); % Power
Xt(:,2)=vel_viento2012(1:nsamples); % Wind speed
Xt(:,3)=temp_nacelle2012(1:nsamples); %Nacele Temperature
Xt(:,4)=temp_ambiente2012(1:nsamples); %temperature in the bearing DE

Tt=zeros(nsamples,1); %Only first 4000 samples
Tt(:,1)=temp_rod_NDE_gen2012(1:nsamples);
%Tt(:,2)=data(june+1:nsamples,9);
%Tt=data(june+1:nsamples,9);
Tt=Tt';
Xt=Xt';

Pt = sim(net,Xt);

% Se representa la salida estimada por la red para las entradas dadas y
% tambii¿n lo que tendri¿a que haber dado

figure(36);
%plot(X,T, 'sr',X,Z,'X')
plot(1:nsamples, Tt(1,:),1:nsamples, Pt(1,:))
ylabel('Temperature 2012 %'); xlabel('Sample');
legend('Real Value', 'Estimated Value');
title(' Neural network test with 2012 data');

figure(37)
histogram(Tt(1,:)-Pt(1,:))
ylabel(' Frecuency 2012'); xlabel('Samples');
title(' Histogram of error with 2012 data');

datos= [Tt-Pt];
[media2012,desviacion2012] = normfit(datos);

%%
%-----Probar2013-----
load('data2013');

data_struct2013=data2013.Generador;

% temp_aceite_multi2013=data_struct2013.temp_aceite_multi;
% temp_rod_alta_multi2013=data_struct2013.temp_rod_alta_multi;
% temp_rod_DE_gen2013=data_struct2013.temp_rod_DE_gen;
temp_rod_NDE_gen2013=data_struct2013.temp_rod_NDE_gen;
% temp_anillos_sld_gen2013=data_struct2013.temp_anillos_sld_gen;
% pres_grupo_h2013=data_struct2013.pres_grupo_h;
temp_nacelle2013=data_struct2013.temp_interior_nac;
temp_ambiente2013=data_struct2013.temp_ambiente;
% w_eje_principal2013=data_struct2013.w_eje_principal;
vel_viento2013=data_struct2013.vel_viento;
% ang_viento2013=data_struct2013.ang_viento;
% v_red2013=data_struct2013.v_red;
% pot_reactiva2013=data_struct2013.pot_reactiva;
% pot_act_est2013=data_struct2013.pot_act_est;
% pot_act_rot2013=data_struct2013.pot_act_rot;
potencia2013=data_struct2013.pot_total;
% energia2013=data_struct2013.energia;

```

```

% state2013=data_struct2013.state;

[nsamples m]=size(potencia2013);
t=1:nsamples;

Xt=zeros(nsamples,4); %Only first training samples
Xt(:,1)=potencia2013(1:nsamples); % Power
Xt(:,2)=vel_viento2013(1:nsamples); % Wind speed
Xt(:,3)=temp_nacelle2013(1:nsamples); %Nacele Temperature
Xt(:,4)=temp_ambiente2013(1:nsamples); %temperature in the bearing DE

Tt=zeros(nsamples,1); %Only first 4000 samples
Tt(:,1)=temp_rod_NDE_gen2013(1:nsamples);
%Tt(:,2)=data(june+1:nsamples,9);
%Tt=data(june+1:nsamples,9);
Tt=Tt';
Xt=Xt';

Pt = sim(net,Xt);

% Se representa la salida estimada por la red para las entradas dadas y
% tambii¿n lo que tendri¿a que haber dado

figure(38);
%plot(X,T, 'sr',X,Z,'X')
plot(1:nsamples, Tt(1,:),1:nsamples, Pt(1,:))
ylabel(' Temperature 2013 %'); xlabel('Sample');
legend('Real Value', 'Estimated Value');
title(' Neural network test with 2013 data');

figure(39)
histogram(Tt(1,:)-Pt(1,:))
ylabel(' Frecuency 2013'); xlabel('Samples');
title(' Histogram of error with 2013 data');

datos= [Tt-Pt];
[media2013,desviacion2013] = normfit(datos);

%%
%-----Probar2014-----
load('data2014');

data_struct2014=data2014.Generador;

% temp_aceite_multi2014=data_struct2014.temp_aceite_multi;
% temp_rod_alta_multi2014=data_struct2014.temp_rod_alta_multi;
% temp_rod_DE_gen2014=data_struct2014.temp_rod_DE_gen;
temp_rod_NDE_gen2014=data_struct2014.temp_rod_NDE_gen;
% temp_anillos_sld_gen2014=data_struct2014.temp_anillos_sld_gen;
% pres_grupo_h2014=data_struct2014.pres_grupo_h;
temp_nacelle2014=data_struct2014.temp_interior_nac;
temp_ambiente2014=data_struct2014.temp_ambiente;
% w_eje_principal2014=data_struct2014.w_eje_principal;
vel_viento2014=data_struct2014.vel_viento;
% ang_viento2014=data_struct2014.ang_viento;
% v_red2014=data_struct2014.v_red;
% pot_reactiva2014=data_struct2014.pot_reactiva;
% pot_act_est2014=data_struct2014.pot_act_est;
% pot_act_rot2014=data_struct2014.pot_act_rot;
potencia2014=data_struct2014.pot_total;
% energia2014=data_struct2014.energia;
% state2014=data_struct2014.state;

[nsamples m]=size(potencia2014);
t=1:nsamples;

Xt=zeros(nsamples,4); %Only first training samples
Xt(:,1)=potencia2014(1:nsamples); % Power
Xt(:,2)=vel_viento2014(1:nsamples); % Wind speed
Xt(:,3)=temp_nacelle2014(1:nsamples); %Nacele Temperature
Xt(:,4)=temp_ambiente2014(1:nsamples); %temperature in the bearing DE

Tt=zeros(nsamples,1); %Only first 4000 samples
Tt(:,1)=temp_rod_NDE_gen2014(1:nsamples);
%Tt(:,2)=data(june+1:nsamples,9);
%Tt=data(june+1:nsamples,9);
Tt=Tt';
Xt=Xt';

Pt = sim(net,Xt);

% Se representa la salida estimada por la red para las entradas dadas y
% tambii¿n lo que tendri¿a que haber dado

```

```

figure(40);
%plot(X,T, 'sr',X,Z,'X')
plot(1:nsamples, Tt(1,:),1:nsamples, Pt(1,:))
ylabel('Temperature 2014 %'); xlabel('Sample');
legend('Real Value', 'Estimated Value');
title(' Neural network test with 2014 data');

figure(41)
histogram(Tt(1,:)-Pt(1,:))
ylabel('Frecuency 2014'); xlabel('Samples');
title(' Histogram of error with 2014 data');

datos= [Tt-Pt];
[media2014,desviacion2014] = normfit(datos);

clear all
close all

load('data2009.mat');

[nsamples m]=size(data);
data_struct=data2009.Generador;

temp_aceite_multi=data_struct.temp_aceite_multi;
temp_rod_alta_multi=data_struct.temp_rod_alta_multi;
% temp_rod_DE_gen=data_struct.temp_rod_DE_gen;
% temp_rod_NDE_gen=data_struct.temp_rod_NDE_gen;
temp_anillos_sld_gen=data_struct.temp_anillos_sld_gen;
% pres_grupo_h=data_struct.pres_grupo_h;
% temp_nacelle=data_struct.temp_interior_nac;
% temp_ambiente=data_struct.temp_ambiente;
% w_eje_principal=data_struct.w_eje_principal;
% vel_viento=data_struct.vel_viento;
ang_viento=data_struct.ang_viento;
% v_red=data_struct.v_red;
% pot_reactiva=data_struct.pot_reactiva;
% pot_act_est=data_struct.pot_act_est;
% pot_act_rot=data_struct.pot_act_rot;
potencia=data_struct.pot_total;
% energia=data_struct.energia;
% state=data_struct.state;

[nsamples m]=size(potencia);
t=1:nsamples;

%plot(t, data(:,1), t, data(:,2), t, data(:,3))

figure(1)
plot(t, temp_rod_alta_multi, t, temp_aceite_multi, t, temp_anillos_sld_gen, t, ang_viento, t,
potencia)
ylabel('Magnitude'); xlabel('Sample');
legend('temp_rod_alta_multi', 'temp_aceite_multi', 'temp_anillos_sld_gen', 'ang_viento',
'potencia');
title(' Variables');

figure(2)
plot(t, temp_rod_alta_multi, t, temp_aceite_multi, t, temp_anillos_sld_gen, t, ang_viento)
ylabel('Magnitude'); xlabel('Sample');
legend('temp_rod_alta_multi', 'temp_aceite_multi', 'temp_anillos_sld_gen', 'ang_viento');
title(' Variables');

figure(8)
plot(t, temp_aceite_multi)
ylabel('Magnitude'); xlabel('Sample');
title('Temp_aceite_multi');

figure(9)
plot(t, temp_anillos_sld_gen)
ylabel('Magnitude'); xlabel('Sample');
title('Temp_anillos_sld');

figure(10)
plot(t, potencia)
ylabel('Magnitude'); xlabel('Sample');
title('Potencia');

%%
training=35000;
X=zeros(training,3); %Only first training input samples
X(:,1)=temp_aceite_multi(1:training); % Power

```

```

X(:,2)=ang_viento(1:training); % Wind speed
%X(:,4)=temp_anillos_sld_gen(1:training,1); %temperatura ambiente
X(:,3)=potencia(1:training,1);

T=zeros(training,1); %Only first training target output samples
T(:,1)=temp_rod_alta_multi(1:training,1); %temperature in the bearing DE

%T(:,2)=data(1:june,9); %soil pressure guide vane position
%T=data(1:june,9); %pressure guide vane position
T=T';
X=X';

net = feedforwardnet([30 10], 'trainlm');
net = configure(net,X,T);

%Criteria for splitting the training set

net.divideFcn='dividerand';
net.divideParam.trainRatio = 70/100;%60,20,20
net.divideParam.valRatio = 20/100;
net.divideParam.testRatio = 10/100;

% Inicializaciï¿½n de pesos
net=init(net);
% Se ajustan diversos parï¿½metros del entrenamiento

net.trainParam.epochs = 10000;
net.trainParam.lr = 0.002;
net.trainParam.show = 50;
net.trainParam.goal = 1e-5;
net.trainParam.max_fail =40; % Failures allowed in validation

[net, traza] = train(net,X,T);
%view(net)
Yp = net(X);
perf = perform(net,Yp,X);

% Training error representation
figure(2)
plot(traza.epoch, traza.perf, traza.epoch, traza.vperf, traza.epoch, traza.tperf);
legend('training', 'Validation', 'Test');
ylabel('Cuadratic Error');
xlabel('Epoch');

% Prediction of the NN with training set

P = sim(net,X);

% Representation predicted and real values in the output

figure(3);
%plot(X,T, 'sr',X,Z,'X')
plot(1:training, T(1,:),1:training, P(1,:))
ylabel('Temp Multi'); xlabel('Sample');
legend('Real Value', 'Estimated Value');
title(' Neural network test using the training set');

figure(4)
histogram(T(1,:)-P(1,:))
ylabel('Frecuency'); xlabel('Samples');
title(' Histogram of error with training data');

datos= [T-P];
[mediaentreno,desviacionentreno] = normfit(datos);

% Se va a obtener la respuesta que darï¿½a la red entrenada cuando se le
% pasan como entradas casos cuya salida se desconoce

Xt=zeros(nsamples-training,3); %Only first training samples
Xt(:,1)=temp_aceite_multi(training+1:nsamples); % Power
Xt(:,2)=ang_viento(training+1:nsamples); % Wind speed
%Xt(:,4)=temp_anillos_sld_gen(training+1:nsamples); %temperatura ambiente
Xt(:,3)=potencia(training+1:nsamples); %temperatura ambiente

Tt=zeros(nsamples-training,1); %Only first 4000 samples
Tt(:,1)=temp_rod_alta_multi(training+1:nsamples);

```

```

Tt=Tt';
Xt=Xt';

Pt = sim(net,Xt);

% Se representa la salida estimada por la red para las entradas dadas y
% también lo que tendríãa que haber dado

figure(30);
%plot(X,T, 'sr',X,Z,'X')
plot(1:nsamples-training, Tt(1,:),1:nsamples-training, Pt(1,:))
ylabel('Temperature multi 2009 %'); xlabel('Sample');
legend('Real Value', 'Estimated Value');
title(' Neural network test with unknown cases');

figure(31)
histogram(Tt(1,:)-Pt(1,:))
ylabel('Frecuency multi 2009'); xlabel('Samples');
title(' Histogram of error with unknown data');

%%
%-----Probar2010-----
load('data2010');

data_struct2010=data2010.Generador;

temp_aceite_multi2010=data_struct2010.temp_aceite_multi;
temp_rod_alta_multi2010=data_struct2010.temp_rod_alta_multi;
% temp_rod_DE_gen2010=data_struct2010.temp_rod_DE_gen;
% temp_rod_NDE_gen2010=data_struct2010.temp_rod_NDE_gen;
%temp_anillos_sld_gen2010=data_struct2010.temp_anillos_sld_gen;
% pres_grupo_h2010=data_struct2010.pres_grupo_h;
% temp_nacelle2010=data_struct2010.temp_interior_nac;
% temp_ambiente2010=data_struct2010.temp_ambiente;
% w_eje_principal2010=data_struct2010.w_eje_principal;
% vel_viento2010=data_struct2010.vel_viento;
ang_viento2010=data_struct2010.ang_viento;
% v_red2010=data_struct2010.v_red;
% pot_reactiva2010=data_struct2010.pot_reactiva;
% pot_act_est2010=data_struct2010.pot_act_est;
% pot_act_rot2010=data_struct2010.pot_act_rot;
potencia2010=data_struct2010.pot_total;
% energia2010=data_struct2010.energia;
% state2010=data_struct2010.state;

[nsamples m]=size(potencia2010);
t=1:nsamples;

Xt=zeros(nsamples,3); %Only first training samples
Xt(:,1)=temp_aceite_multi2010(1:nsamples); % Power
Xt(:,2)=ang_viento2010(1:nsamples); % Wind speed
Xt(:,4)=temp_anillos_sld_gen(1:nsamples); % Wind speed
Xt(:,3)=potencia2010(1:nsamples); %temperature in the bearing NDE

Tt=zeros(nsamples,1); %Only first 4000 samples
Tt(:,1)=temp_rod_alta_multi2010(1:nsamples);

Tt=Tt';
Xt=Xt';

Pt = sim(net,Xt);

% Se representa la salida estimada por la red para las entradas dadas y
% también lo que tendríãa que haber dado

figure(32);
%plot(X,T, 'sr',X,Z,'X')
plot(1:nsamples, Tt(1,:),1:nsamples, Pt(1,:))
ylabel('Temperature multi 2010 %'); xlabel('Sample');
legend('Real Value', 'Estimated Value');
title(' Neural network test with 2010 data');

figure(33)
histogram(Tt(1,:)-Pt(1,:))
ylabel('Frecuency 2010'); xlabel('Samples');
title(' Histogram of error with 2010 data');

datos= [Tt-Pt];
[media2010,desviacion2010] = normfit(datos);

%%
%-----Probar2011-----
load('data2011');

```

```

data_struct2011=data2011.Generador;

temp_aceite_multi2011=data_struct2011.temp_aceite_multi;
temp_rod_alta_multi2011=data_struct2011.temp_rod_alta_multi;
% temp_rod_DE_gen2011=data_struct2011.temp_rod_DE_gen;
% temp_rod_NDE_gen2011=data_struct2011.temp_rod_NDE_gen;
temp_anillos_sld_gen2011=data_struct2011.temp_anillos_sld_gen;
% pres_grupo_h2011=data_struct2011.pres_grupo_h;
%temp_nacelle2011=data_struct2011.temp_interior_nac;
% temp_ambiente2011=data_struct2011.temp_ambiente;
% w_eje_principal2011=data_struct2011.w_eje_principal;
% vel_viento2011=data_struct2011.vel_viento;
ang_viento2011=data_struct2011.ang_viento;
% v_red2011=data_struct2011.v_red;
% pot_reactiva=2011data_struct2011.pot_reactiva;
% pot_act_est2011=data_struct2011.pot_act_est;
% pot_act_rot2011=data_struct2011.pot_act_rot;
potencia2011=data_struct2011.pot_total;
% energia2011=data_struct2011.energia;
% state2011=data_struct2011.state;

[nsamples m]=size(potencia2011);
t=1:nsamples;

Xt=zeros(nsamples,3); %Only first training samples
Xt(:,1)=temp_aceite_multi2011(1:nsamples); % Power
Xt(:,2)=ang_viento2011(1:nsamples); % Wind speed
%Xt(:,4)=temp_anillos_sld_gen2011(1:nsamples); % Wind speed
Xt(:,3)=potencia2011(1:nsamples); %temperature in the bearing NDE

Tt=zeros(nsamples,1); %Only first 4000 samples
Tt(:,1)=temp_rod_alta_multi2011(1:nsamples);

Tt=Tt';
Xt=Xt';

Pt = sim(net,Xt);

% Se representa la salida estimada por la red para las entradas dadas y
% también lo que tendríamos que haber dado

figure(34);
plot(1:nsamples, Tt(1,:),1:nsamples, Pt(1,:))
ylabel('Temperature Multi 2011 %'); xlabel('Sample');
legend('Real Value', 'Estimated Value');
title('Neural network test with 2011 data');

figure(35)
histogram(Tt(1,:)-Pt(1,:))
ylabel('Frecuency 2011'); xlabel('Samples');
title('Histogram of error with 2011 data');

datos= [Tt-Pt];
[media2011,desviacion2011] = normfit(datos);

%%
%-----Probar2012-----
load('data2012');

data_struct2012=data2012.Generador;

temp_aceite_multi2012=data_struct2012.temp_aceite_multi;
temp_rod_alta_multi2012=data_struct2012.temp_rod_alta_multi;
% temp_rod_DE_gen2012=data_struct2012.temp_rod_DE_gen;
% temp_rod_NDE_gen2012=data_struct2012.temp_rod_NDE_gen;
temp_anillos_sld_gen2012=data_struct2012.temp_anillos_sld_gen;
% pres_grupo_h2012=data_struct2012.pres_grupo_h;
% temp_nacelle2012=data_struct2012.temp_interior_nac;
% temp_ambiente2012=data_struct2012.temp_ambiente;
% w_eje_principal2012=data_struct2012.w_eje_principal;
% vel_viento2012=data_struct2012.vel_viento;
ang_viento2012=data_struct2012.ang_viento;
% v_red2012=data_struct2012.v_red;
% pot_reactiva2012=data_struct2012.pot_reactiva;
% pot_act_est2012=data_struct2012.pot_act_est;
% pot_act_rot2012=data_struct2012.pot_act_rot;
potencia2012=data_struct2012.pot_total;
% energia2012=data_struct2012.energia;
% state2012=data_struct2012.state;

[nsamples m]=size(potencia2012);
t=1:nsamples;

```

```

Xt=zeros(nsamples,3); %Only first training samples
Xt(:,1)=temp_aceite_multi2012(1:nsamples); % Power
Xt(:,2)=ang_viento2012(1:nsamples); % Wind speed
Xt(:,4)=temp_anillos_sld_gen2012(1:nsamples); % Wind speed
Xt(:,3)=potencia2012(1:nsamples); %temperature in the bearing NDE

Tt=zeros(nsamples,1); %Only first 4000 samples
Tt(:,1)=temp_rod_alta_multi2012(1:nsamples);

Tt=Tt';
Xt=Xt';

Pt = sim(net,Xt);

% Se representa la salida estimada por la red para las entradas dadas y
% tambii¿n lo que tendri¿a que haber dado

figure(36);
%plot(X,T, 'sr',X,Z,'X')
plot(1:nsamples, Tt(1,:),1:nsamples, Pt(1,:))
ylabel('Temperature multi 2012 %'); xlabel('Sample');
legend('Real Value', 'Estimated Value');
title(' Neural network test with 2012 data');

figure(37)
histogram(Tt(1:)-Pt(1,:))
ylabel('Frecuency 2012'); xlabel('Samples');
title(' Histogram of error with 2012 data');

datos= [Tt-Pt];
[media2012,desviacion2012] = normfit(datos);

%%
%-----Probar2013-----
load('data2013');

data_struct2013=data2013.Generador;

temp_aceite_multi2013=data_struct2013.temp_aceite_multi;
temp_rod_alta_multi2013=data_struct2013.temp_rod_alta_multi;
% temp_rod_DE_gen2013=data_struct2013.temp_rod_DE_gen;
% temp_rod_NDE_gen2013=data_struct2013.temp_rod_NDE_gen;
temp_anillos_sld_gen2013=data_struct2013.temp_anillos_sld_gen;
% pres_grupo_h2013=data_struct2013.pres_grupo_h;
% temp_nacelle2013=data_struct2013.temp_interior_nac;
% temp_ambiente2013=data_struct2013.temp_ambiente;
% w_eje_principal2013=data_struct2013.w_eje_principal;
% vel_viento2013=data_struct2013.vel_viento;
ang_viento2013=data_struct2013.ang_viento;
% v_red2013=data_struct2013.v_red;
% pot_reactiva2013=data_struct2013.pot_reactiva;
% pot_act_est2013=data_struct2013.pot_act_est;
% pot_act_rot2013=data_struct2013.pot_act_rot;
potencia2013=data_struct2013.pot_total;
% energia2013=data_struct2013.energia;
% state2013=data_struct2013.state;

[nsamples m]=size(potencia2013);
t=1:nsamples;

Xt=zeros(nsamples,3); %Only first training samples
Xt(:,1)=temp_aceite_multi2013(1:nsamples); % Power
Xt(:,2)=ang_viento2013(1:nsamples); % Wind speed
Xt(:,4)=temp_anillos_sld_gen2013(1:nsamples); % Wind speed
Xt(:,3)=potencia(1:nsamples); %temperature in the bearing NDE

Tt=zeros(nsamples,1); %Only first 4000 samples
Tt(:,1)=temp_rod_alta_multi2013(1:nsamples);

Tt=Tt';
Xt=Xt';

Pt = sim(net,Xt);

% Se representa la salida estimada por la red para las entradas dadas y
% tambii¿n lo que tendri¿a que haber dado

figure(38);
%plot(X,T, 'sr',X,Z,'X')
plot(1:nsamples, Tt(1,:),1:nsamples, Pt(1,:))
ylabel('Temperature 2013 %'); xlabel('Sample');
legend('Real Value', 'Estimated Value');
title(' Neural network test with 2013 data');

```

```

figure(39)
histogram(Tt(1,:)-Pt(1,:))
ylabel('Frecuency 2013'); xlabel('Samples');
title(' Histogram of error with 2013 data');

datos= [Tt-Pt];
[media2013,desviacion2013] = normfit(datos);

%%
%-----Probar2014-----
load('data2014');

data_struct2014=data2014.Generador;

temp_aceite_multi2014=data_struct2014.temp_aceite_multi;
temp_rod_alta_multi2014=data_struct2014.temp_rod_alta_multi;
% temp_rod_DE_gen2014=data_struct2014.temp_rod_DE_gen;
% temp_rod_NDE_gen2014=data_struct2014.temp_rod_NDE_gen;
temp_anillos_sld_gen2014=data_struct2014.temp_anillos_sld_gen;
% pres_grupo_h2014=data_struct2014.pres_grupo_h;
% temp_nacelle2014=data_struct2014.temp_interior_nac;
% temp_ambiente2014=data_struct2014.temp_ambiente;
% w_eje_principal2014=data_struct2014.w_eje_principal;
% vel_viento2014=data_struct2014.vel_viento;
ang_viento2014=data_struct2014.ang_viento;
% v_red2014=data_struct2014.v_red;
% pot_reactiva2014=data_struct2014.pot_reactiva;
% pot_act_est2014=data_struct2014.pot_act_est;
% pot_act_rot2014=data_struct2014.pot_act_rot;
potencia2014=data_struct2014.pot_total;
% energia2014=data_struct2014.energia;
% state2014=data_struct2014.state;

[nsamples m]=size(potencia2014);
t=1:nsamples;

Xt=zeros(nsamples,3); %Only first training samples
Xt(:,1)=temp_aceite_multi2014(1:nsamples); % Power
Xt(:,2)=ang_viento2014(1:nsamples); % Wind speed
%Xt(:,4)=temp_anillos_sld_gen2014(1:nsamples); % Wind speed
Xt(:,3)=potencia2014(1:nsamples); %temperature in the bearing NDE

Tt=zeros(nsamples,1); %Only first 4000 samples
Tt(:,1)=temp_rod_alta_multi2014(1:nsamples);

Tt=Tt';
Xt=Xt';

Pt = sim(net,Xt);

% Se representa la salida estimada por la red para las entradas dadas y
% tambi n lo que tendr a que haber dado

figure(40);
%plot(X,T, 'sr',X,Z,'X')
plot(1:nsamples, Tt(1,:),1:nsamples, Pt(1,:))
ylabel('Temperature 2014 %'); xlabel('Sample');
legend('Real Value', 'Estimated Value');
title(' Neural network test with 2014 data');

figure(41)
histogram(Tt(1,:)-Pt(1,:))
ylabel('Frecuency 2014'); xlabel('Samples');
title(' Histogram of error with 2014 data');

datos= [Tt-Pt];
[media2014,desviacion2014] = normfit(datos);

clear all
close all

load('data2009.mat');

%[nsamples m]=size(data);
data_struct=data2009.Generador;

temp_aceite_multi=data_struct.temp_aceite_multi;
temp_rod_alta_multi=data_struct.temp_rod_alta_multi;
temp_rod_DE_gen=data_struct.temp_rod_DE_gen;

```

```

temp_rod_NDE_gen=data_struct.temp_rod_NDE_gen;
temp_anillos_sld_gen=data_struct.temp_anillos_sld_gen;
pres_grupo_h=data_struct.pres_grupo_h;
temp_nacelle=data_struct.temp_interior_nac;
temp_ambiente=data_struct.temp_ambiente;
w_eje_principal=data_struct.w_eje_principal;
vel_viento=data_struct.vel_viento;
ang_viento=data_struct.ang_viento;
v_red=data_struct.v_red;
pot_reactiva=data_struct.pot_reactiva;
pot_act_est=data_struct.pot_act_est;
pot_act_rot=data_struct.pot_act_rot;
potencia=data_struct.pot_total;
energia=data_struct.energia;
state=data_struct.state;

[nsamples m]=size(potencia);
t=1:nsamples;

%plot(t, data(:,1), t, data(:,2), t, data(:,3))

figure(1)
plot(t, vel_viento, t, potencia)
ylabel('Magnitude'); xlabel('Sample');
legend('vel_wind');
title(' Variables');

training=30000;
X=zeros(training,1); %Only first training input samples
X(:,1)=vel_viento(1:training); % Wind speed

T=zeros(training,1); %Only first training target output samples
T(:,1)=potencia(1:training,1); %temperature in the bearing DE

%T(:,2)=data(1:june,9); %soil pressure guide vane position
%T=data(1:june,9); %pressure guide vane position
T=T';
X=X';

net = feedforwardnet([30 10], 'trainlm');
net = configure(net,X,T);

%Criteria for splitting the training set

net.divideFcn='dividerand';
net.divideParam.trainRatio = 60/100;%60,20,20
net.divideParam.valRatio = 20/100;
net.divideParam.testRatio = 20/100;

% Inicializaciï¿½n de pesos
net=init(net);
% Se ajustan diversos parï¿½metros del entrenamiento

net.trainParam.epochs = 10000;
net.trainParam.lr = 0.002;
net.trainParam.show = 50;
net.trainParam.goal = 1e-5;
net.trainParam.max_fail =40; % Failures allowed in validation

[net, traza] = train(net,X,T);
%view(net)
Yp = net(X);
perf = perform(net,Yp,X);

% Training error representation
figure(2)
plot(traza.epoch, traza.perf, traza.epoch, traza.vperf, traza.epoch, traza.tperf);
legend('training', 'Validation', 'Test');
ylabel('Quadratic Error');
xlabel(' Epoch');

% Prediction of the NN with training set

P = sim(net,X);

% Representation predicted and real values in the output

```

```

figure(3);
%plot(X,T, 'sr',X,Z,'X')
plot(1:training, T(1,:),1:training, P(1,:))
ylabel('Potencia'); xlabel('Sample');
legend('Real Value', 'Estimated Value');
title(' Neural network test using the training set');

figure(4)
histogram(T(1,:)-P(1,:))
ylabel('Frecuency'); xlabel('Samples');
title(' Histogram of error with training data');

datos= [T-P];
[mediaentren,desviacionentren] = normfit(datos);

% Se va a obtener la respuesta que daría la red entrenada cuando se le
% pasan como entradas casos cuya salida se desconoce

Xt=zeros(nsamples-training,1); %Only first training samples
Xt(:,1)=vel_viento(training+1:nsamples); % Wind speed

Tt=zeros(nsamples-training,1); %Only first 4000 samples
Tt(:,1)=potencia(training+1:nsamples);

Tt=Tt';
Xt=Xt';

Pt = sim(net,Xt);

% Se representa la salida estimada por la red para las entradas dadas y
% también lo que tendría que haber dado

figure(30);
%plot(X,T, 'sr',X,Z,'X')
plot(1:nsamples-training, Tt(1,:),1:nsamples-training, Pt(1,:))
ylabel('Potencia 2009 '); xlabel('Sample');
legend('Real Value', 'Estimated Value');
title(' Neural network test with unknown cases');

figure(31)
histogram(Tt(1,:)-Pt(1,:))
ylabel('Frecuency 2009'); xlabel('Samples');
title(' Histogram of error with unknown data');

datos= [Tt-Pt];
[media2009,desviacion2009] = normfit(datos);

%%
%-----Probar2010-----
load('data2010');

data_struct2010=data2010.Generador;

% temp_aceite_multi2010=data_struct2010.temp_aceite_multi;
% temp_rod_alta_multi2010=data_struct2010.temp_rod_alta_multi;
% temp_rod_DE_gen2010=data_struct2010.temp_rod_DE_gen;
% temp_rod_NDE_gen2010=data_struct2010.temp_rod_NDE_gen;
% temp_anillos_sld_gen2010=data_struct2010.temp_anillos_sld_gen;
% pres_grupo_h2010=data_struct2010.pres_grupo_h;
% temp_nacelle2010=data_struct2010.temp_interior_nac;
% temp_ambiente2010=data_struct2010.temp_ambiente;
% w_eje_principal2010=data_struct2010.w_eje_principal;
vel_viento2010=data_struct2010.vel_viento;
% ang_viento2010=data_struct2010.ang_viento;
% v_red2010=data_struct2010.v_red;
% pot_reactiva2010=data_struct2010.pot_reactiva;
% pot_act_est2010=data_struct2010.pot_act_est;
% pot_act_rot2010=data_struct2010.pot_act_rot;
potencia2010=data_struct2010.pot_total;
% energia2010=data_struct2010.energia;
% state2010=data_struct2010.state;

[nsamples m]=size(potencia2010);
t=1:nsamples;

Xt=zeros(nsamples,1); %Only first training samples
Xt(:,1)=vel_viento2010(1:nsamples); % Wind speed

Tt=zeros(nsamples,1); %Only first 4000 samples
Tt(:,1)=potencia2010(1:nsamples);

```

```

%Tt(:,2)=data(june+1:nsamples,9);
%Tt=data(june+1:nsamples,9);
Tt=Tt';
Xt=Xt';

Pt = sim(net,Xt);

% Se representa la salida estimada por la red para las entradas dadas y
% también lo que tendríamos que haber dado

figure(32);
%plot(X,T, 'sr',X,Z,'X')
plot(1:nsamples, Tt(1,:),1:nsamples, Pt(1,:))
ylabel('Potencia 2010 %'); xlabel('Sample');
legend('Real Value', 'Estimated Value');
title(' Neural network test with 2010 data');

figure(33)
histogram(Tt(1,:)-Pt(1,:))
ylabel('Frecuency 2010'); xlabel('Samples');
title(' Histogram of error with 2010 data');

datos= [Tt-Pt];
[media2010,desviacion2010] = normfit(datos);

%%
%-----Probar2011-----
load('data2011');

data_struct2011=data2011.Generador;

% temp_aceite_multi2011=data_struct2011.temp_aceite_multi;
% temp_rod_alta_multi2011=data_struct2011.temp_rod_alta_multi;
% temp_rod_DE_gen2011=data_struct2011.temp_rod_DE_gen;
% temp_rod_NDE_gen2011=data_struct2011.temp_rod_NDE_gen;
% temp_anillos_sld_gen2011=data_struct2011.temp_anillos_sld_gen;
% pres_grupo_h2011=data_struct2011.pres_grupo_h;
% temp_nacelle2011=data_struct2011.temp_interior_nac;
% temp_ambiente2011=data_struct2011.temp_ambiente;
% w_eje_principal2011=data_struct2011.w_eje_principal;
vel_viento2011=data_struct2011.vel_viento;
% ang_viento2011=data_struct2011.ang_viento;
% v_red2011=data_struct2011.v_red;
% pot_reactiva=2011data_struct2011.pot_reactiva;
% pot_act_est2011=data_struct2011.pot_act_est;
% pot_act_rot2011=data_struct2011.pot_act_rot;
potencia2011=data_struct2011.pot_total;
% energia2011=data_struct2011.energia;
% state2011=data_struct2011.state;

[nsamples m]=size(potencia2011);
t=1:nsamples;

Xt=zeros(nsamples,1); %Only first training samples
Xt(:,1)=vel_viento2011(1:nsamples); % Wind speed

Tt=zeros(nsamples,1); %Only first 4000 samples
Tt(:,1)=potencia2011(1:nsamples);
%Tt(:,2)=data(june+1:nsamples,9);
%Tt=data(june+1:nsamples,9);
Tt=Tt';
Xt=Xt';

Pt = sim(net,Xt);

% Se representa la salida estimada por la red para las entradas dadas y
% también lo que tendríamos que haber dado

figure(34);
%plot(X,T, 'sr',X,Z,'X')
plot(1:nsamples, Tt(1,:),1:nsamples, Pt(1,:))
ylabel('Potencia 2011 %'); xlabel('Sample');
legend('Real Value', 'Estimated Value');
title(' Neural network test with 2011 data');

figure(35)
histogram(Tt(1,:)-Pt(1,:))
ylabel('Frecuency 2011'); xlabel('Samples');
title(' Histogram of error with 2011 data');

datos= [Tt-Pt];
[media2011,desviacion2011] = normfit(datos);
%%

```

```

%-----Probar2012-----
load('data2012');

data_struct2012=data2012.Generador;

% temp_aceite_multi2012=data_struct2012.temp_aceite_multi;
% temp_rod_alta_multi2012=data_struct2012.temp_rod_alta_multi;
% temp_rod_DE_gen2012=data_struct2012.temp_rod_DE_gen;
% temp_rod_NDE_gen2012=data_struct2012.temp_rod_NDE_gen;
% temp_anillos_sld_gen2012=data_struct2012.temp_anillos_sld_gen;
% pres_grupo_h2012=data_struct2012.pres_grupo_h;
% temp_nacelle2012=data_struct2012.temp_interior_nac;
% temp_ambiente2012=data_struct2012.temp_ambiente;
% w_eje_principal2012=data_struct2012.w_eje_principal;
vel_viento2012=data_struct2012.vel_viento;
% ang_viento2012=data_struct2012.ang_viento;
% v_red2012=data_struct2012.v_red;
% pot_reactiva2012=data_struct2012.pot_reactiva;
% pot_act_est2012=data_struct2012.pot_act_est;
% pot_act_rot2012=data_struct2012.pot_act_rot;
potencia2012=data_struct2012.pot_total;
% energia2012=data_struct2012.energia;
% state2012=data_struct2012.state;

[nsamples m]=size(potencia2012);
t=1:nsamples;

Xt=zeros(nsamples,1); %Only first training samples
Xt(:,1)=vel_viento2012(1:nsamples); % Wind speed

Tt=zeros(nsamples,1); %Only first 4000 samples
Tt(:,1)=potencia2012(1:nsamples);
%Tt(:,2)=data(june+1:nsamples,9);
%Tt=data(june+1:nsamples,9);
Tt=Tt';
Xt=Xt';

Pt = sim(net,Xt);

% Se representa la salida estimada por la red para las entradas dadas y
% también lo que tendríamos a que haber dado

figure(36);
%plot(X,T, 'sr',X,Z,'X')
plot(1:nsamples, Tt(1,:),1:nsamples, Pt(1,:))
ylabel('Potencia 2012 %'); xlabel('Sample');
legend('Real Value', 'Estimated Value');
title(' Neural network test with 2012 data');

figure(37)
histogram(Tt(1,:)-Pt(1,:))
ylabel('Frecuency 2012'); xlabel('Samples');
title(' Histogram of error with 2012 data');

datos= [Tt-Pt];
[media2012,desviacion2012] = normfit(datos);

%%
%-----Probar2013-----
load('data2013');

data_struct2013=data2013.Generador;

% temp_aceite_multi2013=data_struct2013.temp_aceite_multi;
% temp_rod_alta_multi2013=data_struct2013.temp_rod_alta_multi;
% temp_rod_DE_gen2013=data_struct2013.temp_rod_DE_gen;
% temp_rod_NDE_gen2013=data_struct2013.temp_rod_NDE_gen;
% temp_anillos_sld_gen2013=data_struct2013.temp_anillos_sld_gen;
% pres_grupo_h2013=data_struct2013.pres_grupo_h;
% temp_nacelle2013=data_struct2013.temp_interior_nac;
% temp_ambiente2013=data_struct2013.temp_ambiente;
% w_eje_principal2013=data_struct2013.w_eje_principal;
vel_viento2013=data_struct2013.vel_viento;
% ang_viento2013=data_struct2013.ang_viento;
% v_red2013=data_struct2013.v_red;
% pot_reactiva2013=data_struct2013.pot_reactiva;
% pot_act_est2013=data_struct2013.pot_act_est;
% pot_act_rot2013=data_struct2013.pot_act_rot;
potencia2013=data_struct2013.pot_total;
% energia2013=data_struct2013.energia;
% state2013=data_struct2013.state;

[nsamples m]=size(potencia2013);

```

```

t=1:nsamples;

Xt=zeros(nsamples,1); %Only first training samples
Xt(:,1)=vel_viento2013(1:nsamples); % Wind speed

Tt=zeros(nsamples,1); %Only first 4000 samples
Tt(:,1)=potencia2013(1:nsamples);
%Tt(:,2)=data(june+1:nsamples,9);
%Tt=data(june+1:nsamples,9);
Tt=Tt';
Xt=Xt';

Pt = sim(net,Xt);

% Se representa la salida estimada por la red para las entradas dadas y
% tambii¿n lo que tendri¿a que haber dado

figure(38);
%plot(X,T, 'sr',X,Z,'X')
plot(1:nsamples, Tt(1,:),1:nsamples, Pt(1,:))
ylabel('Potencia 2013 %'); xlabel('Sample');
legend('Real Value', 'Estimated Value');
title(' Neural network test with 2013 data');

figure(39)
histogram(Tt(1,:)-Pt(1,:))
ylabel(' Frecuency 2013'); xlabel('Samples');
title(' Histogram of error with 2013 data');

datos= [Tt-Pt];
[media2013,desviacion2013] = normfit(datos);

%%
%-----Probar2014-----
load('data2014');

data_struct2014=data2014.Generador;

% temp_aceite_multi2014=data_struct2014.temp_aceite_multi;
% temp_rod_alta_multi2014=data_struct2014.temp_rod_alta_multi;
% temp_rod_DE_gen2014=data_struct2014.temp_rod_DE_gen;
% temp_rod_NDE_gen2014=data_struct2014.temp_rod_NDE_gen;
% temp_anillos_sld_gen2014=data_struct2014.temp_anillos_sld_gen;
% pres_grupo_h2014=data_struct2014.pres_grupo_h;
% temp_nacelle2014=data_struct2014.temp_interior_nac;
% temp_ambiente2014=data_struct2014.temp_ambiente;
% w_eje_principal2014=data_struct2014.w_eje_principal;
vel_viento2014=data_struct2014.vel_viento;
% ang_viento2014=data_struct2014.ang_viento;
% v_red2014=data_struct2014.v_red;
% pot_reactiva2014=data_struct2014.pot_reactiva;
% pot_act_est2014=data_struct2014.pot_act_est;
% pot_act_rot2014=data_struct2014.pot_act_rot;
potencia2014=data_struct2014.pot_total;
% energia2014=data_struct2014.energia;
% state2014=data_struct2014.state;

[nsamples m]=size(potencia2014);
t=1:nsamples;

Xt=zeros(nsamples,1); %Only first training samples
Xt(:,1)=vel_viento2014(1:nsamples); % Wind speed

Tt=zeros(nsamples,1); %Only first 4000 samples
Tt(:,1)=potencia2014(1:nsamples);
%Tt(:,2)=data(june+1:nsamples,9);
%Tt=data(june+1:nsamples,9);
Tt=Tt';
Xt=Xt';

Pt = sim(net,Xt);

% Se representa la salida estimada por la red para las entradas dadas y
% tambii¿n lo que tendri¿a que haber dado

figure(40);
%plot(X,T, 'sr',X,Z,'X')
plot(1:nsamples, Tt(1,:),1:nsamples, Pt(1,:))
ylabel('Potencia 2014 %'); xlabel('Sample');
legend('Real Value', 'Estimated Value');
title(' Neural network test with 2014 data');

figure(41)

```

```

histogram(Tt(1,:)-Pt(1,:))
ylabel('Frecuency 2014'); xlabel('Samples');
title(' Histogram of error with 2014 data');

datos= [Tt-Pt];
[media2014,desviacion2014] = normfit(datos);

clear all
close all

load('data2009.mat');

%[nsamples m]=size(data);
data_struct=data2009.Generador;

% temp_aceite_multi=data_struct.temp_aceite_multi;
temp_rod_alta_multi=data_struct.temp_rod_alta_multi;
% temp_rod_DE_gen=data_struct.temp_rod_DE_gen;
% temp_rod_NDE_gen=data_struct.temp_rod_NDE_gen;
temp_anillos_sld_gen=data_struct.temp_anillos_sld_gen;
% pres_grupo_h=data_struct.pres_grupo_h;
temp_nacelle=data_struct.temp_interior_nac;
temp_ambiente=data_struct.temp_ambiente;
% w_eje_principal=data_struct.w_eje_principal;
% vel_viento=data_struct.vel_viento;
% ang_viento=data_struct.ang_viento;
% v_red=data_struct.v_red;
% pot_reactiva=data_struct.pot_reactiva;
% pot_act_est=data_struct.pot_act_est;
% pot_act_rot=data_struct.pot_act_rot;
potencia=data_struct.pot_total;
% energia=data_struct.energia;
% state=data_struct.state;

[nsamples m]=size(potencia);
t=1:nsamples;

% plot(t, data(:,1), t, data(:,2), t, data(:,3))

figure(1)
plot(t, temp_anillos_sld_gen, t, temp_ambiente, t, temp_nacelle)
ylabel(' Magnitude'); xlabel(' Sample ');
legend('tem_anillos_sld_gen', 'tem_amibente', 'tem_nacelle');
title(' Variables');

training=38000;
X=zeros(training,2); %Only first training input samples
X(:,1)=temp_ambiente(1:training); % Power
%X(:,2)=temp_rod_alta_multi(1:training); % Wind speed
X(:,2)=temp_nacelle(1:training); %Nacele Temperature

T=zeros(training,1); %Only first training target output samples
T(:,1)=temp_anillos_sld_gen(1:training,1); %temperature in the bearing DE

T=T';
X=X';

net = feedforwardnet([30 10], 'trainlm');
net = configure(net,X,T);

%Criteria for splitting the training set

net.divideFcn='dividerand';
net.divideParam.trainRatio = 60/100;%60,20,20
net.divideParam.valRatio = 20/100;
net.divideParam.testRatio = 20/100;

%Inicializacin de pesos
net=init(net);
%Se ajustan diversos parí¿metros del entrenamiento

net.trainParam.epochs = 10000;
net.trainParam.lr = 0.002;
net.trainParam.show = 50;
net.trainParam.goal = 1e-5;

```

```

net.trainParam.max_fail =40; % Failures allowed in validation

[net, traza] = train(net,X,T);
view(net)
Yp = net(X);
perf = perform(net,Yp,X);

%Training error representation
figure(2)
plot(traza.epoch, traza.perf, traza.epoch, traza.vperf, traza.epoch, traza.tperf);
legend('training', 'Validation', 'Test');
ylabel('Cuadratic Error');
xlabel('Epoch');

%Prediction of the NN with training set

P = sim(net,X);

%Representation predicted and real values in the output

figure(3);
plot(X,T, 'sr',X,Z,'X')
plot(1:training, T(1,:),1:training, P(1,:))
ylabel('Temp NBearing'); xlabel('Sample');
legend('Real Value', 'Estimated Value');
title('Neural network test using the training set');

figure(4)
histogram(T(1,:)-P(1,:))
ylabel('Frecuency'); xlabel('Samples');
title('Histogram of error with training data');

datos= [T-P];
[mediaentreno,desviacionentreno] = normfit(datos);

%Se va a obtener la respuesta que daría la red entrenada cuando se le
%pasan como entradas casos cuya salida se desconoce

Xt=zeros(nsamples-training,2); %Only first training samples
Xt(:,1)=temp_ambiente(training+1:nsamples); % Power
Xt(:,2)=temp_rod_alta_multi(training+1:nsamples); % Wind speed
Xt(:,2)=temp_nacelle(training+1:nsamples); %Nacelle Temperature

Tt=zeros(nsamples-training,1); %Only first 4000 samples
Tt(:,1)=temp_anillos_sld_gen(training+1:nsamples);

Tt=Tt';
Xt=Xt';

Pt = sim(net,Xt);

%Se representa la salida estimada por la red para las entradas dadas y
%también lo que tendría que haber dado

figure(30);
% plot(X,T, 'sr',X,Z,'X')
plot(1:nsamples-training, Tt(1,:),1:nsamples-training, Pt(1,:))
ylabel('Temperature 2009 '); xlabel('Sample');
legend('Real Value', 'Estimated Value');
title('Neural network test with unknown cases');

figure(31)
histogram(Tt(1,:)-Pt(1,:))
ylabel('Frecuency 2009'); xlabel('Samples');
title('Histogram of error with unknown data');

%%
%-----Probar2010-----
load('data2010');

data_struct2010=data2010.Generador;

% temp_aceite_multi2010=data_struct2010.temp_aceite_multi;
temp_rod_alta_multi2010=data_struct2010.temp_rod_alta_multi;
% temp_rod_DE_gen2010=data_struct2010.temp_rod_DE_gen;
% temp_rod_NDE_gen2010=data_struct2010.temp_rod_NDE_gen;
temp_anillos_sld_gen2010=data_struct2010.temp_anillos_sld_gen;
% pres_grupo_h2010=data_struct2010.pres_grupo_h;
temp_nacelle2010=data_struct2010.temp_interior_nac;
temp_ambiente2010=data_struct2010.temp_ambiente;
% w_eje_principal2010=data_struct2010.w_eje_principal;

```

```

% vel_viento2010=data_struct2010.vel_viento;
% ang_viento2010=data_struct2010.ang_viento;
% v_red2010=data_struct2010.v_red;
% pot_reactiva2010=data_struct2010.pot_reactiva;
% pot_act_est2010=data_struct2010.pot_act_est;
% pot_act_rot2010=data_struct2010.pot_act_rot;
potencia2010=data_struct2010.pot_total;
% energia2010=data_struct2010.energia;
% state2010=data_struct2010.state;

[nsamples m]=size(potencia2010);
t=1:nsamples;

Xt=zeros(nsamples,2); %Only first training samples
Xt(:,1)=temp_ambiente2010(1:nsamples); % Power
%Xt(:,2)=temp_rod_alta_multi2010(1:nsamples); % Wind speed
Xt(:,2)=temp_nacelle2010(1:nsamples); %Nacelle Temperature

Tt=zeros(nsamples,1); %Only first 4000 samples
Tt(:,1)=temp_anillos_sld_gen2010(1:nsamples);
%Tt(:,2)=data(june+1:nsamples,9);
%Tt=data(june+1:nsamples,9);
Tt=Tt';
Xt=Xt';

Pt = sim(net,Xt);

% Se representa la salida estimada por la red para las entradas dadas y
% tambii¿n lo que tendri¿a que haber dado

figure(32);
%plot(X,T, 'sr',X,Z,'X')
plot(1:nsamples, Tt(1,:),1:nsamples, Pt(1,:))
ylabel('Temperature 2010 %'); xlabel('Sample');
legend('Real Value', 'Estimated Value');
title(' Neural network test with 2010 data');

figure(33)
histogram(Tt(1,:)-Pt(1,:))
ylabel(' Frecuency 2010'); xlabel('Samples');
title(' Histogram of error with 2010 data');

datos= [Tt-Pt];
[media2010,desviacion2010] = normfit(datos);

%%
%-----Probar2011-----
load('data2011');

data_struct2011=data2011.Generador;

% temp_aceite_multi2011=data_struct2011.temp_aceite_multi;
temp_rod_alta_multi2011=data_struct2011.temp_rod_alta_multi;
% temp_rod_DE_gen2011=data_struct2011.temp_rod_DE_gen;
% temp_rod_NDE_gen2011=data_struct2011.temp_rod_NDE_gen;
temp_anillos_sld_gen2011=data_struct2011.temp_anillos_sld_gen;
% pres_grupo_h2011=data_struct2011.pres_grupo_h;
temp_nacelle2011=data_struct2011.temp_interior_nac;
temp_ambiente2011=data_struct2011.temp_ambiente;
% w_eje_principal2011=data_struct2011.w_eje_principal;
%vel_viento2011=data_struct2011.vel_viento;
% ang_viento2011=data_struct2011.ang_viento;
% v_red2011=data_struct2011.v_red;
% pot_reactiva=2011data_struct2011.pot_reactiva;
% pot_act_est2011=data_struct2011.pot_act_est;
% pot_act_rot2011=data_struct2011.pot_act_rot;
potencia2011=data_struct2011.pot_total;
% energia2011=data_struct2011.energia;
% state2011=data_struct2011.state;

[nsamples m]=size(potencia2011);
t=1:nsamples;

Xt=zeros(nsamples,2); %Only first training samples
Xt(:,1)=temp_ambiente2011(1:nsamples); % Power
%Xt(:,2)=temp_rod_alta_multi2011(1:nsamples); % Wind speed
Xt(:,2)=temp_nacelle2011(1:nsamples); %Nacelle Temperature

Tt=zeros(nsamples,1); %Only first 4000 samples
Tt(:,1)=temp_anillos_sld_gen2011(1:nsamples);
%Tt(:,2)=data(june+1:nsamples,9);
%Tt=data(june+1:nsamples,9);
Tt=Tt';
Xt=Xt';

```

```

Pt = sim(net,Xt);

% Se representa la salida estimada por la red para las entradas dadas y
% también lo que tendrí%a que haber dado

figure(34);
%plot(X,T, 'sr',X,Z,'X')
plot(1:nsamples, Tt(1,:),1:nsamples, Pt(1,:))
ylabel('Temperature 2011 %'); xlabel('Sample');
legend('Real Value', 'Estimated Value');
title('Neural network test with 2011 data');

figure(35)
histogram(Tt(1,:)-Pt(1,:))
ylabel('Frecuency 2011'); xlabel('Samples');
title('Histogram of error with 2011 data');

datos= [Tt-Pt];
[media2011,desviacion2011] = normfit(datos);

%%
%-----Probar2012-----
load('data2012');

data_struct2012=data2012.Generador;

% temp_aceite_multi2012=data_struct2012.temp_aceite_multi;
temp_rod_alta_multi2012=data_struct2012.temp_rod_alta_multi;
% temp_rod_DE_gen2012=data_struct2012.temp_rod_DE_gen;
% temp_rod_NDE_gen2012=data_struct2012.temp_rod_NDE_gen;
temp_anillos_sld_gen2012=data_struct2012.temp_anillos_sld_gen;
% pres_grupo_h2012=data_struct2012.pres_grupo_h;
temp_nacelle2012=data_struct2012.temp_interior_nac;
temp_ambiente2012=data_struct2012.temp_ambiente;
% w_eje_principal2012=data_struct2012.w_eje_principal;
% vel_viento2012=data_struct2012.vel_viento;
% ang_viento2012=data_struct2012.ang_viento;
% v_red2012=data_struct2012.v_red;
% pot_reactiva2012=data_struct2012.pot_reactiva;
% pot_act_est2012=data_struct2012.pot_act_est;
% pot_act_rot2012=data_struct2012.pot_act_rot;
potencia2012=data_struct2012.pot_total;
% energia2012=data_struct2012.energia;
% state2012=data_struct2012.state;

[nsamples m]=size(potencia2012);
t=1:nsamples;

Xt=zeros(nsamples,2); %Only first training samples
Xt(:,1)=temp_ambiente2012(1:nsamples); % Power
%Xt(:,3)=temp_rod_alta_multi2012(1:nsamples); % Wind speed
Xt(:,2)=temp_nacelle2012(1:nsamples); %Nacelle Temperature

Tt=zeros(nsamples,1); %Only first 4000 samples
Tt(:,1)=temp_anillos_sld_gen2012(1:nsamples);
%Tt(:,2)=data(june+1:nsamples,9);
%Tt=data(june+1:nsamples,9);
Tt=Tt';
Xt=Xt';

Pt = sim(net,Xt);

% Se representa la salida estimada por la red para las entradas dadas y
% también lo que tendrí%a que haber dado

figure(36);
%plot(X,T, 'sr',X,Z,'X')
plot(1:nsamples, Tt(1,:),1:nsamples, Pt(1,:))
ylabel('Temperature 2012 %'); xlabel('Sample');
legend('Real Value', 'Estimated Value');
title('Neural network test with 2012 data');

figure(37)
histogram(Tt(1,:)-Pt(1,:))
ylabel('Frecuency 2012'); xlabel('Samples');
title('Histogram of error with 2012 data');

datos= [Tt-Pt];
[media2012,desviacion2012] = normfit(datos);

%%
%-----Probar2013-----

```

```

load('data2013');

data_struct2013=data2013.Generador;

% temp_aceite_multi2013=data_struct2013.temp_aceite_multi;
temp_rod_alta_multi2013=data_struct2013.temp_rod_alta_multi;
% temp_rod_DE_gen2013=data_struct2013.temp_rod_DE_gen;
% temp_rod_NDE_gen2013=data_struct2013.temp_rod_NDE_gen;
temp_anillos_sld_gen2013=data_struct2013.temp_anillos_sld_gen;
% pres_grupo_h2013=data_struct2013.pres_grupo_h;
temp_nacelle2013=data_struct2013.temp_interior_nac;
temp_ambiente2013=data_struct2013.temp_ambiente;
% w_eje_principal2013=data_struct2013.w_eje_principal;
% vel_viento2013=data_struct2013.vel_viento;
% ang_viento2013=data_struct2013.ang_viento;
% v_red2013=data_struct2013.v_red;
% pot_reactiva2013=data_struct2013.pot_reactiva;
% pot_act_est2013=data_struct2013.pot_act_est;
% pot_act_rot2013=data_struct2013.pot_act_rot;
potencia2013=data_struct2013.pot_total;
% energia2013=data_struct2013.energia;
% state2013=data_struct2013.state;

[nsamples m]=size(potencia2013);
t=1:nsamples;

Xt=zeros(nsamples,2); %Only first training samples
Xt(:,1)=temp_ambiente2013(1:nsamples); % Power
%Xt(:,3)=temp_rod_alta_multi2013(1:nsamples); % Wind speed
Xt(:,2)=temp_nacelle2013(1:nsamples); %Nacelle Temperature

Tt=zeros(nsamples,1); %Only first 4000 samples
Tt(:,1)=temp_anillos_sld_gen2013(1:nsamples);
%Tt(:,2)=data(june+1:nsamples,9);
%Tt=data(june+1:nsamples,9);
Tt=Tt';
Xt=Xt';

Pt = sim(net,Xt);

% Se representa la salida estimada por la red para las entradas dadas y
% tambii¿n lo que tendrí¿a que haber dado

figure(38);
%plot(X,T, 'sr',X,Z,'X')
plot(1:nsamples, Tt(1,:),1:nsamples, Pt(1,:))
ylabel('Temperature 2013 %'); xlabel('Sample');
legend('Real Value', 'Estimated Value');
title(' Neural network test with 2013 data');

figure(39)
histogram(Tt(1,:)-Pt(1,:))
ylabel('Frecuency 2013'); xlabel('Samples');
title(' Histogram of error with 2013 data');

datos= [Tt-Pt];
[media2013,desviacion2013] = normfit(datos);

%%
%-----Probar2014-----
load('data2014');

data_struct2014=data2014.Generador;

% temp_aceite_multi2014=data_struct2014.temp_aceite_multi;
temp_rod_alta_multi2014=data_struct2014.temp_rod_alta_multi;
% temp_rod_DE_gen2014=data_struct2014.temp_rod_DE_gen;
% temp_rod_NDE_gen2014=data_struct2014.temp_rod_NDE_gen;
temp_anillos_sld_gen2014=data_struct2014.temp_anillos_sld_gen;
% pres_grupo_h2014=data_struct2014.pres_grupo_h;
temp_nacelle2014=data_struct2014.temp_interior_nac;
temp_ambiente2014=data_struct2014.temp_ambiente;
% w_eje_principal2014=data_struct2014.w_eje_principal;
% vel_viento2014=data_struct2014.vel_viento;
% ang_viento2014=data_struct2014.ang_viento;
% v_red2014=data_struct2014.v_red;
% pot_reactiva2014=data_struct2014.pot_reactiva;
% pot_act_est2014=data_struct2014.pot_act_est;
% pot_act_rot2014=data_struct2014.pot_act_rot;
potencia2014=data_struct2014.pot_total;
% energia2014=data_struct2014.energia;
% state2014=data_struct2014.state;

[nsamples m]=size(potencia2014);

```

```

t=1:nsamples;

Xt=zeros(nsamples,2); %Only first training samples
Xt(:,1)=temp_ambiente2014(1:nsamples); % Power
Xt(:,3)=temp_rod_alta_multi2014(1:nsamples); % Wind speed
Xt(:,2)=temp_nacelle2014(1:nsamples); %Nacele Temperature

Tt=zeros(nsamples,1); %Only first 4000 samples
Tt(:,1)=temp_anillos_sld_gen2014(1:nsamples);
%Tt(:,2)=data(june+1:nsamples,9);
%Tt=data(june+1:nsamples,9);
Tt=Tt';
Xt=Xt';

Pt = sim(net,Xt);

% Se representa la salida estimada por la red para las entradas dadas y
% tambiiñn lo que tendriñka que haber dado

figure(40);
%plot(X,T, 'sr',X,Z,'X')
plot(1:nsamples, Tt(1,:),1:nsamples, Pt(1,:))
ylabel('Temperature 2014 %'); xlabel('Sample');
legend('Real Value', 'Estimated Value');
title(' Neural network test with 2014 data');

figure(41)
histogram(Tt(1,:)-Pt(1,:))
ylabel('Frecuency 2014'); xlabel('Samples');
title(' Histogram of error with 2014 data');

datos= [Tt-Pt];
[media2014,desviacion2014] = normfit(datos);

clear all
close all

load('datapitchangl.mat');

%[nsamples m]=size(data);
data_struct=datapitchangl.Generador;

% temp_aceite_multi=data_struct.temp_aceite_multi;
% temp_rod_alta_multi=data_struct.temp_rod_alta_multi;
% temp_rod_DE_gen=data_struct.temp_rod_DE_gen;
% temp_rod_NDE_gen=data_struct.temp_rod_NDE_gen;
% temp_anillos_sld_gen=data_struct.temp_anillos_sld_gen;
% pres_grupo_h=data_struct.pres_grupo_h;
% temp_nacelle=data_struct.temp_interior_nac;
% temp_ambiente=data_struct.temp_ambiente;
% w_eje_principal=data_struct.w_eje_principal;
vel_viento=data_struct.vel_viento;
ang_viento=data_struct.ang_viento;
% v_red=data_struct.v_red;
% pot_reactiva=data_struct.pot_reactiva;
% pot_act_est=data_struct.pot_act_est;
% pot_act_rot=data_struct.pot_act_rot;
potencia=data_struct.pot_total;
% energia=data_struct.energia;
% state=data_struct.state;
pitchangle=data_struct.pitchangle;

[nsamples m]=size(potencia);
t=1:nsamples;

%plot(t, data(:,1), t, data(:,2), t, data(:,3))

figure(1)
plot(t, vel_viento, t, potencia, t, ang_viento, t, pitchangle)
ylabel('Magnitude'); xlabel('Sample');
legend('vel_wind', 'potencia', 'ang_viento', 'pitchangle');
title(' Variables');

training=9000;
X=zeros(training,3); %Only first training input samples
X(:,1)=potencia(1:training); % Power
X(:,2)=vel_viento(1:training); % Wind speed
X(:,3)=ang_viento(1:training); %Nacele Temperature
% X(:,4)=temp_rod_DE_gen(1:training,1); %temperature in the bearing DE

```

```

T=zeros(training,1); %Only first training target output samples
T(:,1)=pitchangle(1:training,1); %temperature in the bearing DE

%T(:,2)=data(1:june,9); %oil pressure guide vane position
%T=data(1:june,9); %pressure guide vane position
T=T';
X=X';

net = feedforwardnet([30 10], 'trainlm');
net = configure(net,X,T);

%Criteria for splitting the training set

net.divideFcn='dividerand';
net.divideParam.trainRatio = 60/100;%60,20,20
net.divideParam.valRatio = 20/100;
net.divideParam.testRatio = 20/100;

% Inicializaci3n de pesos
net=init(net);
% Se ajustan diversos par3metros del entrenamiento

net.trainParam.epochs = 10000;
net.trainParam.lr = 0.002;
net.trainParam.show = 50;
net.trainParam.goal = 1e-5;
net.trainParam.max_fail =40; % Failures allowed in validation

[net, traza] = train(net,X,T);
%view(net)
Yp = net(X);
perf = perform(net,Yp,X);

% Training error representation
figure(2)
plot(traza.epoch, traza.perf, traza.epoch, traza.vperf, traza.epoch, traza.tperf);
legend('training', 'Validation', 'Test');
ylabel('Cuadratic Error');
xlabel('Epoch');

% Prediction of the NN with training set

P = sim(net,X);

% Representation predicted and real values in the output

figure(3);
%plot(X,T, 'sr',X,Z,'X')
plot(1:training, T(1,:),1:training, P(1,:))
ylabel('Temp NBearing'); xlabel('Sample');
legend('Real Value', 'Estimated Value');
title(' Neural network test using the training set');

figure(4)
histogram(T(1,:)-P(1,:))
ylabel('Frecuency'); xlabel('Samples');
title(' Histogram of error with training data');

datos= [T-P];
[mediaentreno,desviacionentreno] = normfit(datos);

% Se va a obtener la respuesta que dar3a la red entrenada cuando se le
% pasan como entradas casos cuya salida se desconoce

Xt=zeros(nsamples-training,3); %Only first training samples
Xt(:,1)=potencia(training+1:nsamples); % Power
Xt(:,2)=vel_viento(training+1:nsamples); % Wind speed
Xt(:,3)=ang_viento(training+1:nsamples); %Nacele Temperature

Tt=zeros(nsamples-training,1); %Only first 4000 samples
Tt(:,1)=pitchangle(training+1:nsamples);
%Tt(:,2)=data(june+1:nsamples,9);
%Tt=data(june+1:nsamples,9);
Tt=Tt';
Xt=Xt';

Pt = sim(net,Xt);

```

```
% Se representa la salida estimada por la red para las entradas dadas y
% también lo que tendríamos que haber dado
```

```
figure(30);
%plot(X,T, 'sr',X,Z,'X')
plot(1:nsamples-training, Tt(1,:),1:nsamples-training, Pt(1,:))
ylabel('Temperature 2009 %'); xlabel('Sample');
legend('Real Value', 'Estimated Value');
title(' Neural network test with unknown cases');
```

```
figure(31)
histogram(Tt(1,:)-Pt(1,:))
ylabel('Frecuency 2009'); xlabel('Samples');
title(' Histogram of error with unknown data');
```

```
datos= [Tt-Pt];
[media2010,desviacion2010] = normfit(datos);
```