



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería del Diseño

Diseño de un sistema automático de detección y  
aislamiento de fallos en instalaciones fotovoltaicas  
utilizando técnicas de Machine Learning

Trabajo Fin de Grado

Grado en Ingeniería Electrónica Industrial y Automática

AUTOR/A: Sánchez Roca, Pedro

Tutor/a: Quiles Cucarella, Eduardo

CURSO ACADÉMICO: 2022/2023

## Resumen

La innovación en sistemas fotovoltaicos, su abaratamiento y el incremento en el precio de la luz han hecho que la demanda de sistemas fotovoltaicos crezca, y con ella las investigaciones que buscan tanto aumentar su eficiencia como reducir sus pérdidas. Estas pérdidas de eficiencia están causadas por el uso cotidiano, que se ha estimado que reduce su eficiencia en un 1% anual y por fallos en el sistema. La detección de estos fallos en los sistemas fotovoltaicos se ha convertido en una tarea esencial para evitar una degradación temprana de los mismos. Con una correcta predicción de los fallos, se puede conseguir un aumento en la vida útil de los sistemas y una mayor eficiencia media.

Investigaciones similares a este trabajo se han centrado en la detección de fallos y la clasificación de unos pocos tipos de estos. Como novedad, este trabajo estudia el rendimiento de los diferentes modelos disponibles para detectar y clasificar hasta siete tipos de fallos diferentes, incluido el sombreado parcial, que es la principal causa de puntos calientes. Estos puntos calientes son un problema común en los paneles solares, capaz de provocar un incendio y, en el mejor de los casos, reducir drásticamente su eficiencia. Además, compara los algoritmos disponibles para detectar cada tipo de fallo, incluso teniendo en cuenta el modo de potencia del sistema. Para ello, se ha hecho uso de una base de datos obtenida de un estudio anterior. Esta base de datos fue creada utilizando un modelo de simulación diseñado en Simulink. Esta base de datos consta con un total de  $2.2 \times 10^6$  mediciones, incluye siete tipos de fallos: circuito abierto, caídas de tensión, sombreado parcial, inversor, sensor de retroalimentación de corriente y controlador MPPT/IPPT en fallos del convertidor elevador. Todas estas mediciones se realizaron tanto en el modo de seguimiento de máxima potencia (“Maximum Power Point Tracking”, MPPT), como en modo limitado (“Limited Power Point Tracking”, LPPT).

Para llevar a cabo este trabajo, se ha utilizado MATLAB y un complemento llamado Classification Learner. Además de hacer uso de Excel para analizar y representar todos los datos recopilados.

Una vez hecho el estudio sobre qué modelo predictivo funciona mejor para cada caso, se propone el diseño de un dispositivo capaz de desconectar el sistema, evitando que el problema cause daños al resto de la instalación e incluso de interrumpir el fallo si se trata de un problema en la red.

El objetivo principal de este trabajo es comparar entre los distintos algoritmos de machine learning disponibles para la detección y clasificación de fallos en sistemas fotovoltaicos, además de diseñar un dispositivo que evite fallos mayores e incluso interrumpa los fallos.

**Palabras clave:** Sistema fotovoltaico, panel solar, célula solar, MATLAB, mantenimiento, mantenimiento predictivo, modelos de predicción, entrenamiento supervisado, eficiencia, precisión

## Abstract

Innovation in photovoltaic systems, their cheapening, and the increase in the price of electricity have made the demand for photovoltaic systems grow, and with it, the researches that seek to increase their efficiency and reduce their losses. These efficiency losses are caused by everyday use, which has been estimated to reduce its efficiency by 1% per year, and system failures. Detecting these failures in photovoltaic systems has become essential to avoid their early degradation. With correct fault prediction, an increase in system life and higher average efficiency can be achieved.

Researches similar to this work have focused on fault detection and classification of a few types of faults. As a novelty, this work studies the performance of the different models available to detect and classify up to seven types of failures, including partial shading, which is the leading cause of hot spots. These hot spots are a common problem in solar panels, capable of causing a fire and, at best, drastically reducing their efficiency. In addition, it compares the algorithms available to detect each type of failure, even considering the system's power mode. For this, a database obtained from a previous study has been used. This database was created using a simulation model designed in Simulink. This database consists of a total of  $2.2 \times 10^6$  measurements and includes seven types of failures: open circuit, voltage drops, partial shading, inverter, current feedback sensor and MPPT / IPPT controller in failure of the elevator converter. All these measurements were performed in Maximum Power Point Tracking (MPPT) mode and Limited Power Point Tracking (LPPT).

To carry out this work, MATLAB and an add-on called Classification Learner were used. In addition to using Excel to analyze and represent all the data collected.

Once the study has been done on which predictive model works best for each case, a device capable of disconnecting the system is proposed, preventing the problem from causing damage to the rest of the installation and even interrupting the failure if it is a problem in the network.

The main objective of this work is to compare between the different machine learning algorithms available for the detection and classification of failures in photovoltaic systems, in addition to designing a device that avoids major failures and even interrupts failures.

**Keywords:** PV system, solar panel, solar cell, MATLAB, maintenance, predictive maintenance, prediction models, supervised training, efficiency, accuracy.

## Resum

La innovació en sistemes fotovoltaics, el seu abaratiment i l'increment en el preu de la llum han fet que la demanda de sistemes fotovoltaics cresca, i amb ella les investigacions que busquen tant augmentar la seua eficiència com reduir les seues pèrdues. Aquestes pèrdues d'eficiència són causades per l'ús quotidià, que s'ha estimat que redueix la seua eficiència en un 1% anual i per fallades en el sistema. La detecció d'aquestes fallades en els sistemes fotovoltaics s'ha convertit en una tasca essencial per a evitar una degradació primerenca d'aquests. Amb una correcta predicció de les fallades, es pot aconseguir un augment en la vida útil dels sistemes i una major eficiència mitjana.

Investigacions similars a aquest treball s'han centrat en la detecció de fallades i la classificació d'uns pocs tipus d'aquests. Com a novetat, aquest treball estudia el rendiment dels diferents models disponibles per a detectar i classificar fins a set tipus de fallades diferents, inclòs l'ombreig parcial, que és la principal causa de punts calents. Aquests punts calents són un problema comú en els panells solars, capaç de provocar un incendi i, en el millor dels casos, reduir dràsticament la seua eficiència. A més, compara els algorismes disponibles per a detectar cada tipus de fallada, fins i tot tenint en compte la manera de potència del sistema. Per a això, s'ha fet ús d'una base de dades obtinguda d'un estudi anterior. Aquesta base de dades va ser creada utilitzant un model de simulació dissenyat en Simulink. Aquesta base de dades consta amb un total de  $2.2 \times 10^6$  mesuraments, inclou set tipus de fallades: circuit obert, caigudes de tensió, ombreig parcial, inversor, sensor de retroalimentació de corrent i controlador MPPT/IPPT en fallades del convertidor elevador. Tots aquests mesuraments es van realitzar tant en la manera de seguiment de màxima potència ("Maximum Power Point Tracking", MPPT), com en manera limitada ("Limited Power Point Tracking", LPPT).

Per a dur a terme aquest treball, s'ha utilitzat MATLAB i un complement anomenat Classification Learner. A més de fer ús d'Excel per a analitzar i representar totes les dades recopilades.

Una vegada fet l'estudi sobre quin model predictiu funciona millor per a cada cas, es proposa el disseny d'un dispositiu capaç de desconnectar el sistema, evitant que el problema cause danys a la resta de la instal·lació i fins i tot d'interrompre la fallada si es tracta d'un problema en la xarxa.

L'objectiu principal d'aquest treball és comparar entre els diferents algorismes de machine learning disponibles per a la detecció i classificació de fallades en sistemes fotovoltaics, a més de dissenyar un dispositiu que evite fallades majors i fins i tot interrompa les fallades.

**Paraules clau:** Sistema fotovoltaic, panell solar, cèl·lula solar, MATLAB, manteniment, manteniment predictiu, models de predicció, entrenament supervisat, eficiència, precisió.

## Contenido

Resumen.....	1
Abstract.....	2
Resum .....	3
I Memoria .....	7
1. Objetivos .....	7
2. Energía solar.....	7
3. Principales fallos en sistemas fotovoltaicos .....	13
4. Mantenimiento en sistemas fotovoltaicos .....	15
4.1 Mantenimiento reactivo.....	15
4.2 Mantenimiento preventivo .....	15
4.3 Mantenimiento predictivo .....	16
5. Inteligencia artificial .....	17
5.1 Aprendizaje supervisado .....	18
5.2 Aprendizaje no supervisado .....	18
5.3 Machine Learning .....	19
5.4 Ejemplos de modelos .....	20
5.5 Métodos de validación .....	23
6. Caso de estudio.....	25
6.1 Estructura de la base de datos.....	26
7. Herramientas utilizadas .....	27
7.1 Cómo usar el complemento Classification Learner.....	28
8. Tratamiento de los datos .....	31
8.1 Cómo funciona el script .....	32
9. Resultados .....	34
9.1 Comparación entre familias y modelos .....	34
9.2 Diferencia entre modos de potencia .....	37

9.3	Diferencia entre fallos .....	40
10.	Sistema de desconexión .....	42
10.1	Procedimiento tras la detección de errores .....	42
10.1.2	Priorización de los errores detectados .....	43
10.2	Planteamiento del sistema de desconexión .....	44
10.3	Problemas al tener el sistema en circuito abierto .....	48
10.4	Elementos de desconexión .....	49
11.	Discusión.....	57
12.	Conclusión.....	58
13.	Tabla de siglas.....	60
II	Pliego de condiciones.....	62
1.	Descripción de la instalación.....	62
2.	Condiciones de los materiales .....	62
3.	Pruebas y ajustes finales o de servicio. ....	63
III	Presupuesto.....	65
1.	Mano de obra.....	65
2.	Licencias.....	65
3.	Materiales .....	66
4.	Total .....	66
	Referencias.....	67
	ANEXO 1 - RELACIÓN DEL TRABAJO CON LOS OBJETIVOS DE DESARROLLO SOSTENIBLE DE LA AGENDA 2030 .....	70
	ANEXO 2 - CÓDIGO DE TRATAMIENTO DE LA BASE DE DATOS Y CARGA DE LOS DATOS A CLASSIFICATIONLEARNER .....	71
	ANEXO 3 - CÓDIGO DE COMPROBACIÓN DE LA PRECISIÓN DEL MODELO. ....	75
	ANEXO 4 - CÓDIGO DEL GUI DEL SISTEMA DE DESCONEXIÓN .....	78

# I Memoria

## 1. Objetivos

Es bien sabido que la energía solar no produce ningún residuo ni contaminación durante su vida útil, pero cuando un módulo se rompe o tiene una reducción considerable en su eficiencia y, por lo tanto, hay que reemplazarlo, solo podemos reciclar el 6% del mismo. Esto hace que diseñar un mantenimiento adecuado sea esencial para prolongar su vida útil y reducir sus pérdidas de eficiencia. El mantenimiento predictivo puede ser útil para esta tarea, evitando que ocurran fallos en el sistema o prediciéndolos antes de que pueda causar daños mayores.

Este trabajo se centra en estudiar el desempeño de los diferentes modelos disponibles para realizar un mantenimiento predictivo. Uno de los puntos clave del trabajo es que este trabajo tiene en cuenta si el sistema está trabajando en seguimiento de máxima potencia (MPPT) o no. Este dato nos servirá para comprobar si hay alguna diferencia en la precisión de los modelos a la hora de predecir los fallos que puedan ocurrir en el sistema. Además, este trabajo analiza el rendimiento de cada modelo con cada uno de los tipos de fallo disponible en el conjunto de datos utilizado.

El objetivo es discutir las diferentes opciones para crear un sistema predictivo simple, robusto y de bajo coste para cada caso. Esto se hace utilizando los datos sin procesar proporcionados por el inversor, evitando la necesidad de utilizar cualquier sensor externo. Además, en lugar de solo detectar si el sistema está fallando, o centrarnos en un único fallo, clasificaremos los fallos que aparezcan y estudiaremos el rendimiento de cada modelo disponible a la hora de detectar cada uno de los siete tipos de fallos que tenemos. Estos fallos van desde sombreado parcial hasta interferencias de red. También se estudia si hay alguna diferencia en la capacidad de predicción estando en MPPT o en LPPT.

Por último, se estudiarán los diferentes elementos de desconexión que hay en el mercado y cuál sería conveniente utilizar en un sistema de desconexión. Adicionalmente, se creará una simulación del sistema de desconexión, con el que se podrá ver de forma gráfica cómo se comportaría todo el sistema en una situación de fallo. Y, además, demostrará la capacidad de predicción del modelo que se decida utilizar para el sistema.

## 2. Energía solar

La energía solar es un sector que ha crecido rápidamente en los últimos años y una de las fuentes de energía renovable más importantes que tenemos ahora mismo. Esta tecnología

comenzó a desarrollarse en los años 60 cuando comenzó la carrera espacial entre los Estados Unidos y la Unión Soviética. Todos los satélites lanzados necesitaban una forma de obtener electricidad para mantener sus sistemas, y la energía solar era la mejor opción disponible ya que esos satélites estaban constantemente expuestos a la radiación solar. Pero los módulos solares que utilizaban en esa época no tienen nada que ver con los que tenemos hoy en día, que cuentan con mayores eficiencias y más fiabilidad a la hora de averías.

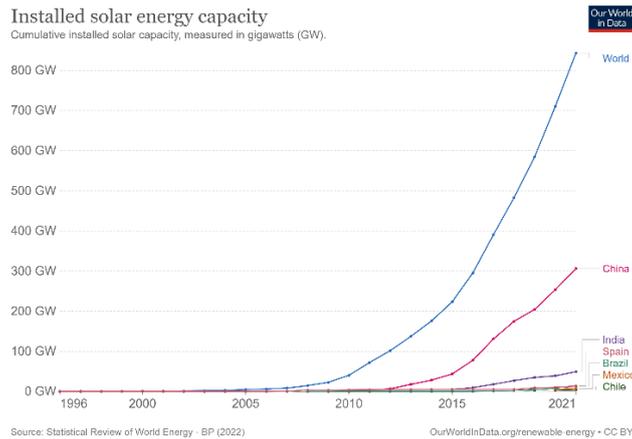


Figura 1- Capacidad instalada de energía solar. (Our World In Data, 2022)

La energía solar se ha convertido en un gran negocio, generando más de 1000TW en todo el mundo en 2021, lo que representa un aumento aproximado del 20% desde 2020. Este aumento de la producción se ha visto afectado por la escasez global de gas natural y por los avances tecnológicos, que han hecho que esta forma de obtener energía sea más eficiente y asequible, por lo que ahora no es un lujo para unas pocas personas o para grandes industrias. Ahora merece la pena tener una instalación relativamente pequeña para uso personal.

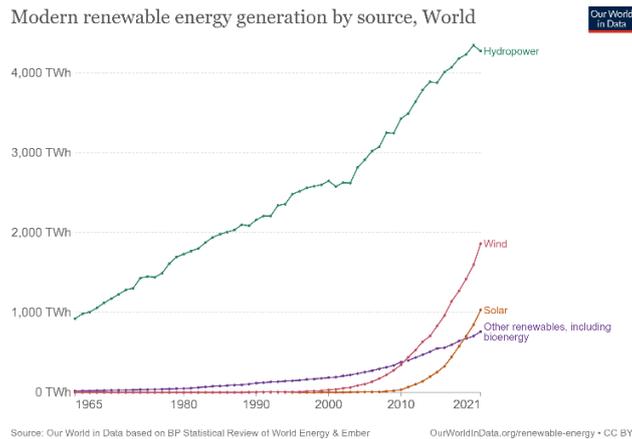


Figura 2 - Generación de energía por fuente. (Our World In Data, 2022)

Los paneles solares están formados por muchas células fotovoltaicas, responsables de convertir la luz solar en electricidad. Estas células tienen dos capas de semiconductores. Los semiconductores son materiales que pueden generar una corriente eléctrica entre conductores y aislantes. Las células solares tienen semiconductores cargados positivamente (tipo p) y negativamente (tipo n). Los electrones libres del semiconductor de tipo n saltan para llenar los huecos en el conductor de tipo p, dejando un espacio vacío. Este fenómeno ocurre porque el semiconductor de tipo n tiene electrones adicionales, y el tipo p, cargado positivamente, no tiene suficientes, por lo que los electrones son donados.

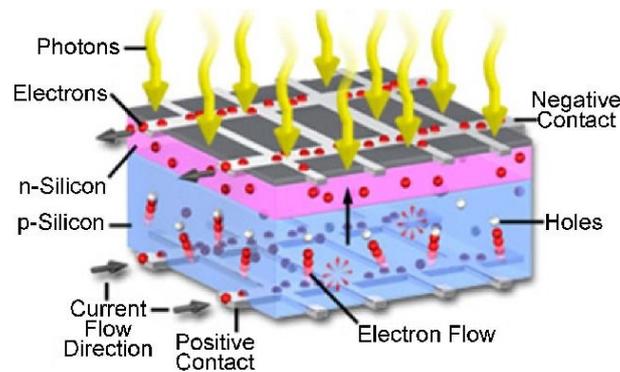


Figura 3 - Esquema de células solares. (Tabassum, 2017)

Si los semiconductores están a temperatura ambiente, es posible que no puedan conducir electricidad, pero si se calientan, lo harán. Entonces, momentáneamente, toda la célula está equilibrada, pero cuando los rayos solares alcanzan el panel los fotones pueden liberar uno de estos electrones donados, dejando un nuevo hueco para que otro electrón ocupe su lugar.

Este proceso ocurre millones de veces por segundo, y los electrones liberados se dirigen en una misma dirección. Este camino de electrones que creamos es corriente eléctrica, y esta corriente es electricidad recién generada. Este proceso es realizado en una célula, pero solo una célula no es suficiente para generar suficiente electricidad para alimentar una casa, que es la instalación solar más común. Para aumentar la producción de electricidad se han creado módulos solares conectando células en un patrón matricial. En estas conexiones, se incluyen algunos diodos, llamados diodos de bloqueo, que evitan que la corriente fluya hacia atrás. Además, también se colocan diodos de derivación, que aseguran que la corriente pueda seguir fluyendo, aunque una celda esté estropeada. Para obtener aún más potencia estos módulos se conectan en serie para aumentar el voltaje total de salida. Para aumentar la corriente disponible se conectan en paralelo. Todos estos módulos conectados crean una matriz fotovoltaica.

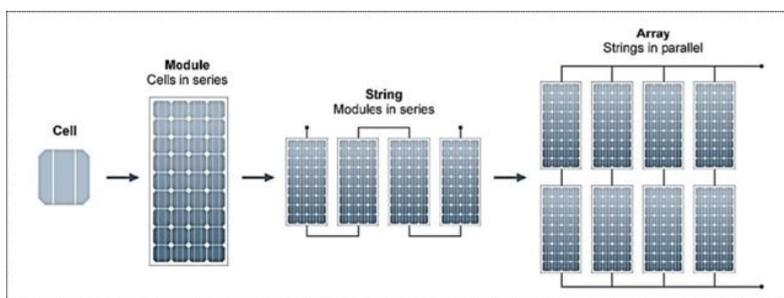


Figura 4 - Conexiones fotovoltaicas. (Abed, 2020)

Otro de los elementos necesarios en toda instalación fotovoltaica es un controlador. Este dispositivo se encarga de hacer todas las conexiones y desconexiones eléctricas. Decide si el sistema está generando suficiente electricidad para suministrar la carga, por lo que el sistema puede estar desconectado de la red de suministro o incluso si puede inyectar el excedente a la red. También decide si el sistema no es capaz de suministrar la carga y necesita obtener la energía restante de la red de suministro eléctrico. Además, si hay baterías involucradas, determina cuándo y cómo cargarlas.

La salida de los módulos es de corriente continua (CC. También llamada Direct Current, DC). Por lo general, se utiliza un inversor para transformar esta CC en corriente alterna (CA. También llamada Alternate Current, AC) con la misma frecuencia y amplitud que la red eléctrica para que pueda ser volcada en ella. En algunos casos específicos el sistema que se quiere alimentar solo necesita CC, por lo que se puede usar un convertidor DC/DC. Una desventaja de estos sistemas es que no podrá volcar el excedente a la red.

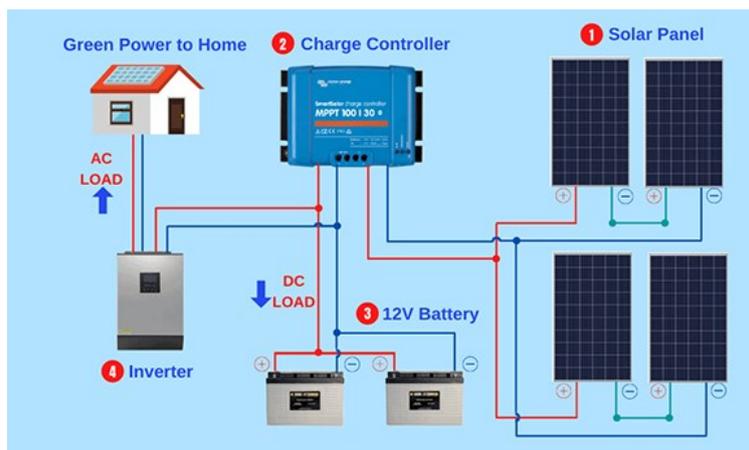


Figura 5 - Cableado PVS estándar. (Liter of light USA, 2021)

Los inversores son dispositivos que tienen como señal de entrada corriente continua (CC) y que transforman para convertirla en una señal de corriente alterna (CA) en la salida. Esta salida de CA es simétrica, con una amplitud y frecuencia diseñadas por el usuario. Normalmente la amplitud es de 230V y la frecuencia de 50Hz. Estas son las características eléctricas de la red de suministro de España por lo que para alimentar nuestros servicios públicos o volcar nuestro excedente es obligatorio que tengan esas características. Por otro lado, en un sistema aislado con otros requisitos o en un sistema en otro país, la amplitud y la frecuencia pueden variar, como en Estados Unidos, que su red de suministro es de 120V y 60Hz.

Hay tres tipos diferentes de inversor, su diferencia principal es la forma de onda de la salida, puede ser un inversor de onda cuadrada, sinusoidal modificada o sinusoidal pura. El inversor de onda cuadrada es adecuado para usar cuando la carga es puramente resistiva. El inversor de onda sinusoidal modificada puede suministrar energía a cargas resistivas y capacitivas, pero puede causar ruido en cargas inductivas. Si necesitamos alimentar cualquier tipo de carga, el inversor de onda sinusoidal pura es capaz de hacerlo.

La parte principal de un inversor es el transformador, encargado de aumentar la tensión disponible. Si por la bobina primaria circula una corriente alterna ésta generará un flujo magnético que cambiará de sentido continuamente, lo que causará una AC más alta en la bobina secundaria. La amplitud de la señal en la bobina secundaria depende principalmente de la estructura de los bobinados, del número de vueltas que el conductor de al núcleo. Dado que nuestra fuente es CC, si conectamos esta señal al devanado primario, no creará un flujo magnético cambiante, por lo que no habrá nada en la salida. Como solución a este problema necesitamos cambiar rápidamente la dirección de la corriente, creando así un inversor de onda cuadrada simple.

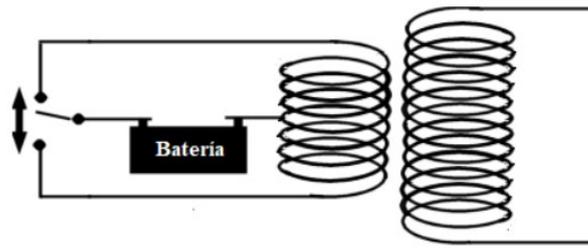


Figura 6 - Inversor de forma cuadrada simple. (MPPT Solar, s.f.)

Para obtener una salida sinusoidal a la salida del transformador necesitamos introducir en la entrada una señal sinusoidal. Esta forma de onda sinusoidal se crea utilizando un oscilador. El problema con los osciladores electrónicos es que la amplitud de la salida aumenta con el tiempo, hasta que alcanza la ganancia máxima o incluso hasta que la señal quede recortada. Este problema provoca una distorsión en la señal causada por los armónicos en altas frecuencias. Como solución a este problema se creó el puente de Wien. Este puente utiliza un amplificador con retroalimentación para crear la forma de onda deseada. Esta retroalimentación está diseñada para aumentar su resistencia a medida que aumenta la temperatura del puente debido a la frecuencia del oscilador. Una vez que la ganancia es estable, los amplificadores funcionan como si fuera un amplificador de clase A, lo que significa que es casi perfecto, reduce la distorsión drásticamente.

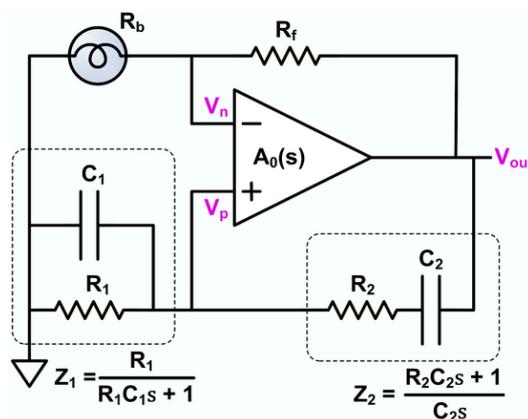


Figura 7 - Diagrama del oscilador Wien Bridge. (Wikipedia, 2022)

### 3. Principales fallos en sistemas fotovoltaicos

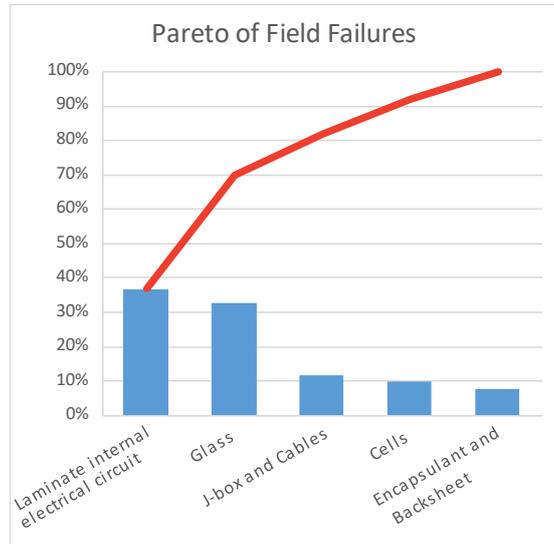


Figura 8 - Diagrama de Pareto de fallos archivadas. (Intechopen, 2018)

La figura 8 muestra la frecuencia media con la que suele ocurrir cada tipo de fallo. Alrededor del 80% de ellos podrían haberse evitado con un mantenimiento adecuado, como los relacionados con el cristal que cubre los módulos, que suele deteriorarse con la suciedad y una mal cuidado. Al igual que los problemas en cables y cajas de conexiones, al tratarse de elementos plásticos expuestos al clima se deterioran con rapidez, por lo que deben revisarse periódicamente para evitar que causen problemas mayores.

Uno de los problemas más graves y comunes en los sistemas fotovoltaicos es la aparición de puntos calientes. Estos puntos calientes son el resultado de un incremento de la temperatura en una o varias de las celdas de las que se compone un módulo. Este incremento de la temperatura ocurre cuando unas celdas en vez de generar energía pasan a actuar como una carga y la consumen. El hecho de que una célula pase a actuar como una carga lo causa un sombreado parcial. Hay dos tipos de sombreado parcial:

- Sombreado objetivo, provocado por factores que se escapan del control humano y que provocan una reducción en la irradiancia que recibe el módulo. Las nubes, la niebla o una contaminación intensa son ejemplos de este tipo de sombreado.
- Sombreado subjetivo, que a su vez se divide en sombreado dinámico, provocado por objetos que crean una sombra sobre el módulo dependiendo del ángulo del sol, y sombreado estático, causado por elementos que obstruyen los rayos solares de forma

constante. Este tipo de sombreado se puede evitar si las instalaciones se diseñan correctamente, evitando la incidencia de sombras de antenas, chimeneas, árboles u otros objetos y con una limpieza de forma periódica, eliminando cualquier residuo que haya podido aparecer en la superficie del panel solar, como restos de aves, hojas o polvo.

El sombreado dinámico es muy común en zonas residenciales, provocando una disminución de la eficiencia, entre un 2% y un 6% según el artículo (Laukamp, 2002), aunque varía dependiendo de la posición relativa entre la sombra y los diodos de protección (A. Drews, 2007)

Este consumo irregular de energía a priori solo se vería reflejado en una disminución de la eficiencia, pero si éste fenómeno anómalo se extiende en el tiempo debemos de considerar la ley de Joule. Esta ley estipula que si por un conductor (en este caso la célula que actúa como carga) circula corriente, parte de la energía cinética de los electrones será disipada en forma de calor. Teniendo en cuenta que los módulos fotovoltaicos están diseñados para generar electricidad y no para consumirla, ergo, tampoco para disipar calor, esto provocará un aumento descontrolado en la temperatura de la célula. La temperatura puede incluso superar los 100°C, pudiendo incluso llegar a causar un incendio. En caso de no causar un incendio esta elevación de temperatura causará daños irreversibles en algunas celdas del panel, provocando fallos de desajuste. Estudios realizados sobre la influencia de este deterioro han mostrado que, si no se corrige la causa de estos problemas, repercutirá en una disminución de la eficiencia del panel en un 1% por año.

Además de estos puntos calientes, el hecho de un sistema fotovoltaico esté expuesto de forma continuada a las condiciones climáticas también genera otros problemas, como es la corrosión. Los módulos fotovoltaicos están montados sobre marcos metálicos, que, aunque cuenten con un tratamiento galvanizado que previene la corrosión, si son dañados pueden comenzar a oxidarse. Por otro lado, la propia superficie de los módulos puede deteriorarse si los rayos ultravioletas traspasan la primera capa protectora y llegan hasta el material que recubre los semiconductores. Este material es conocido como EVA (Etileno-Vinil-Acetato). Cuando los rayos solares alcanzan este componente lo oxidan, provocando que aparezca en el panel una capa amarillenta que reduce ligeramente su eficiencia.

Otro tipo de problema muy común está relacionado con los cables y cajas de conexiones. Una mala soldadura, un conector que no haga buen contacto o una caja de conexiones en la que entre agua pueden provocar que el sistema falle por completo y en algunos casos incluso provocar un incendio. Este tipo de fallos se solucionan fácilmente revisando concienzudamente todas las conexiones y cajas, pero siempre está el factor del error humano.

## 4. Mantenimiento en sistemas fotovoltaicos

El mantenimiento es una tarea esencial en los sistemas fotovoltaicos. La calidad del mantenimiento puede determinar la eficiencia e incluso la vida útil de la instalación, lo que resultará en un menor gasto en reparaciones o, si se trata de un negocio, en una mayor rentabilidad.

Hay varios tipos de mantenimiento. Cada instalación fotovoltaica puede requerir un mantenimiento diferente. Este mantenimiento debe ser adecuado a necesidades tanto del propietario como de la propia instalación.

### 4.1 Mantenimiento reactivo

Este mantenimiento es el más común en los sistemas industriales ya que a veces es inevitable. Cuando se descubre que ha habido un problema se arreglan o se reemplazan los elementos defectuosos. Por lo general, este tipo de mantenimiento es adecuado para piezas pequeñas, fáciles de reparar y que no vale la pena cambiar antes, como podría ser un fusible. Además, estos componentes no deben ser críticos en el sistema, para que no causen un problema mayor que sea más complejo o incluso que no se pueda solucionar.

La principal desventaja de este sistema es que el fallo de una pieza puede desencadenar otros problemas, un deterioro de otros componentes o incluso de todo el sistema.

Por otro lado, es conveniente debido a su simplicidad.

### 4.2 Mantenimiento preventivo

Es el mantenimiento rutinario del equipo, basado en la búsqueda y reparación de problemas menores para evitar otros más significativos. Requiere planificarlo y registrar todas las inspecciones realizadas. Es útil para prevenir el deterioro y los fallos relacionados con el uso. Los fabricantes a menudo proporcionan información sobre el mantenimiento adecuado para los equipos. Las piezas y componentes se revisan en función del tiempo que han estado en funcionamiento o del uso que han tenido.

Algunas ventajas de este tipo de mantenimiento son que prolonga la vida útil de los equipos y evita averías más extensas. Además, dependiendo del tipo de equipo, puede mejorar la eficiencia.

La principal desventaja es que, en caso de tratarse de una industria, la producción puede llegar a detenerse, lo que resulta en unas pérdidas económicas.

### 4.3 Mantenimiento predictivo

Es el método más avanzado de mantenimiento. Los componentes o sistema que mantener son monitorizados constantemente, buscando algún mal funcionamiento o parámetro para pronosticar su avería. Si se predice adecuadamente, se evita una gran parada en el proceso.

La principal desventaja de esta técnica es que tiene un coste inicial sustancial ya que se necesitan sensores, software que pueda hacer esta tarea y empleados bien formados y especializados que puedan supervisar este mantenimiento. Pero todo esto puede valer la pena, ya que puede predecir un fallo completo del sistema, evitando una pérdida económica considerable y la necesidad de detener todo el sistema durante mucho tiempo.

Para comprender mejor todos los métodos de mantenimiento, podemos imaginar un ejemplo de cada uno. En el mantenimiento de un sistema fotovoltaico real podemos ver todos los métodos de mantenimiento. Por ejemplo, un fusible generalmente se cambia una vez que se funde, al igual que un cable, que se cambia una vez ha sido dañado o está muy deteriorado. Estos son ejemplos de mantenimiento reactivo.

Si, por el contrario, tenemos una planificación para la limpieza de la superficie de los módulos solares para evitar la acumulación de suciedad este es un mantenimiento preventivo. Este mantenimiento preventivo ayuda a evitar problemas de sombreado parcial, que son la principal causa de la aparición de puntos calientes, capaces de reducir drásticamente la eficiencia del módulo e incluso de causar un incendio.

Con mantenimiento preventivo podemos evitar hasta el 32% de las causas comunes de degradación de los módulos solares, como se muestra en la Figura 7. Además, mientras se limpian los paneles también se pueden verificar si hay grietas u otros daños físicos, lo que también constituye una de las principales causas de degradación del panel.

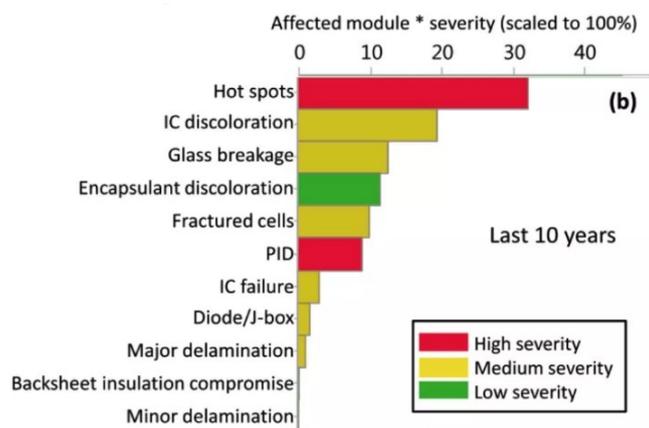


Figura 9 - Diagrama de Pareto de las causas de degradación en módulos fotovoltaicos. (Intechopen, 2018)

Y, por último, un ejemplo de mantenimiento predictivo, que es el caso de estudio de este trabajo. Un sistema que monitoree los voltajes y la corriente del sistema, creando predicciones sobre el estado del sistema. Estas predicciones nos indican si habrá un problema en el sistema fotovoltaico, lo que si completamos con un sistema de desconexión de los módulos nos puede ayudar a evitar daños mayores.

## 5. Inteligencia artificial

La inteligencia artificial (IA) es un sistema creado para imitar la inteligencia humana, capaz de realizar una tarea compleja y además seguir aprendiendo de sus acciones y de los resultados que obtiene. Esta rama del conocimiento se está volviendo esencial en otras aplicaciones científicas, como la ciencia de datos o la medicina. Actualmente en medicina se utiliza IA para el análisis de imágenes de pruebas, como rayos-X, resonancias y pruebas similares. Además, son muy útiles para utilizar la ciencia de datos en estudios médicos, tanto para buscar patrones de la causa de una enfermedad como para desarrollar tratamientos especializados para cada paciente. Nosotros, como usuarios, estamos usando IA sin siquiera darnos cuenta, como cuando recibimos recomendaciones personalizadas en algunas aplicaciones y páginas web. Estas recomendaciones se basan en nuestra huella digital y en las acciones que realizamos dentro de la página web o plataforma que utilizamos para proporcionarnos recomendaciones que cree que se ajustan a nuestros gustos. Además, estos sistemas de recomendaciones evitan que nos aparezcan recomendaciones que cree que no nos van a gustar en absoluto.

Podemos diferenciar entre dos formas en que una IA puede aprender: de forma supervisada y no supervisada.

## 5.1 Aprendizaje supervisado

Este método es una rama del llamado Machine Learning, se basa en la condición de que hayamos etiquetado previamente los datos que se utilizarán para buscar un patrón dentro de ellos sin tener que indicarle dónde buscarlo. Además, el modelo sigue mejorando su rendimiento utilizando los datos obtenidos después de su validación.

La salida de un sistema con aprendizaje supervisado puede ser un valor numérico (regresión) o una etiqueta (clasificación). La regresión se utiliza para realizar pronósticos en base a los datos introducido. Estas regresiones pueden ser lineal, muy utilizada en economía; regresión logística, para el cálculo de probabilidades; y regresión polinomial, similar a la lineal pero muy útil cuando los datos introducidos no están totalmente relacionados de forma lineal, lo que nos podría causar desviaciones en la predicción. Cuando realizamos una clasificación el sistema nos devuelve una etiqueta, dependiendo de la cantidad de etiquetas que podamos obtener estaremos hablando de clasificación binaria (dos etiquetas) o de clasificación multiclase (más de dos etiquetas). Algunos ejemplos de predicciones binarias son la detección de spam (spam/no spam) o el estado de una máquina (defectuosa/buen funcionamiento), en ambos casos solo hay dos etiquetas disponibles. Si por el contrario tenemos más de dos etiquetas estaremos hablando de clasificación multiclase, la cual se usa para clasificación de rostros y correos electrónicos, entre otros.

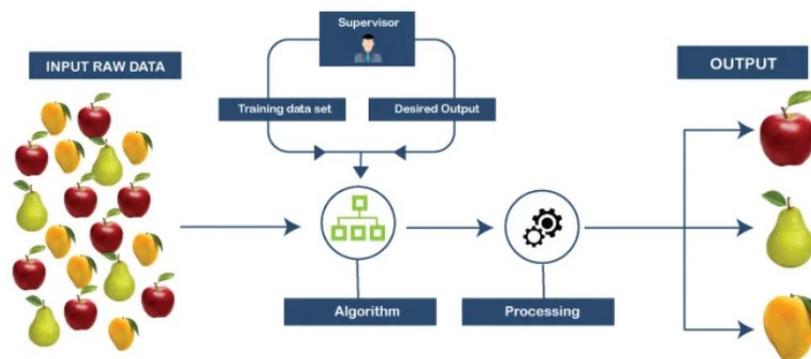


Figura 10- Diagrama aprendizaje supervisado. (Omdena, 2022)

## 5.2 Aprendizaje no supervisado

Cuando no tenemos datos etiquetados, pero queremos encontrar patrones dentro de los datos proporcionados utilizamos el aprendizaje no supervisado, otra rama del Machine Learning. Este método, sin haber sido entrenado previamente o programado de forma específica, busca los patrones ocultos en los datos sin necesidad de proporcionarle datos para ser entrenado ni de ser

supervisado. Este tipo de aprendizaje se asemeja al funcionamiento del cerebro humano para aprender nuevos conceptos.

Este aprendizaje puede utilizarse para algoritmos de agrupamiento, en el que agrupa los datos de entrada que tienen muchas similitudes con el resto; y algoritmos de asociación, que intentan relacionar las variables de los datos de entrada.

El hecho de que las predicciones se basen únicamente en los datos, en información, hace que estas predicciones carezcan de cualquier tipo de sesgo o compromiso por parte del desarrollador. Además, evita posibles errores en el etiquetado de los datos.

Como desventaja, este tipo de aprendizaje proporciona escasa información sobre cómo ha clasificado los datos, además de que no conocemos las etiquetas que utiliza para proporcionarnos los datos de salida.

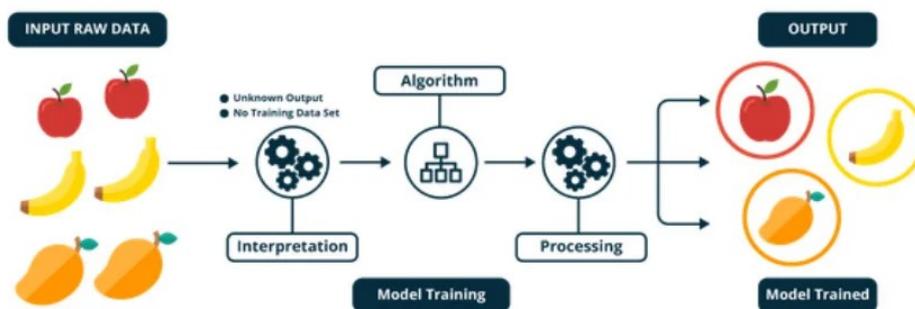


Figura 11 - Esquema aprendizaje no supervisado. (Omdena, 2022)

El aprendizaje supervisado y no supervisado están igualmente extendidos. Cada uno tiene sus ventajas e inconvenientes basadas en su forma de encontrar patrones en los datos. La principal diferencia entre ambos métodos son los "datos etiquetados", esto hace que en el aprendizaje supervisado haya un entrenamiento, un conocimiento previo, mientras que en el no supervisado se proporcionan los datos sin ser tratados. Los algoritmos de aprendizaje supervisado se entrenan en función de datos previos, mientras que los métodos no supervisados intentan encontrar un patrón en los datos no etiquetados.

### 5.3 Machine Learning

Machine learning es una rama de la inteligencia artificial que imita la forma de aprender que tenemos los humanos. Este machine learning y la inteligencia artificial a menudo son confundidos

o intercambiados. Es esencial entender que todo aprendizaje automático es inteligencia artificial, pero no al revés.

Esta rama de la IA es muy importante en la ciencia de datos, haciendo uso de métodos estadísticos ayuda a los científicos de datos a tomar decisiones sobre cuando trabajan con big data.

El funcionamiento de este sistema es un algoritmo que genera una predicción o clasificación basándose en los datos introducidos. Estos datos introducidos pueden estar etiquetados o no, por lo que se trataría de aprendizaje supervisado o no supervisado. Utilizando estos datos crea un modelo que es capaz de buscar patrones en los datos.

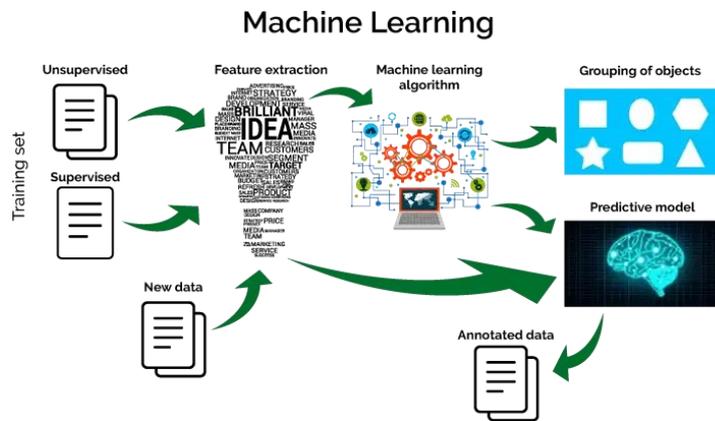


Figura 12 - Diagrama de aprendizaje automático. (Pant, 2019)

Gracias a su capacidad de hacer pronósticos, estos algoritmos pueden ser utilizados para diversos fines, como el mantenimiento predictivo. Le permite a la empresa saber cuándo se requiere mantenimiento, trabajando de manera segura y eficiente durante más tiempo.

Hoy en día, donde todo está conectado, recopilando y compartiendo datos, el aprendizaje automático es una herramienta poderosa para impulsar un negocio, mejorar la productividad y reducir los costos en diversas áreas.

## 5.4 Ejemplos de modelos

### 5.4.1 Redes neuronales

Las redes neuronales son modelos que realizan las operaciones simulando el funcionamiento del sistema nervioso humano. Tanto es su parecido con el sistema humano que estos modelos

están formados por nódulos llamados neuronas. Estas neuronas están conectadas entre sí, divididas en tres partes normalmente: capa de entrada, donde se introducen los datos a procesar; capas ocultas, donde ocurren todas las operaciones necesarias; y la capa de salida, donde encontraremos el resultado producido por la red neuronal. Estas capas pueden estar formadas una o más neuronas, no tienen un número fijo. Cada conexión entre una neurona de una capa y otra de otra capa multiplica el valor proporcionado por un valor de peso, estas multiplicaciones entre las distintas capas de neuronas hacen que el resultado final sea el que deseamos. Al principio todos estos valores de peso son aleatorios, se van ajustando de forma autónoma mientras aprende de los datos que le proporcionamos.

Este tipo de modelos suele ser utilizado para las tareas más complejas como visión artificial, identificación de compuestos o reconocimiento de voz.

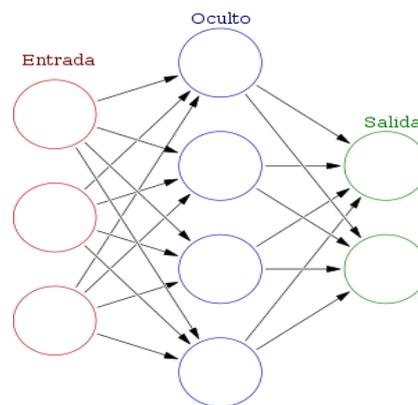


Figura 13 - Diagrama red neuronal artificial. (Red neuronal artificial, 2023)

Un ejemplo de utilización de este modelo es el artículo (F. Aziz, 2020), donde desarrollaron un sistema capaz de detectar solo cinco tipos de fallos. Para ello utilizaron una red neuronal convolucional (del inglés Convolutional Neural Network, a partir de ahora CNN) reentrenada, logrando una precisión del 73.53% en condiciones de seguimiento de máxima potencia (a partir de ahora MPPT, del inglés Maximum Power Point Tracking) y una precisión del 45.35% sin seguimiento de máxima potencia (en inglés Limited Power Point Tracking, a partir de ahora LPPT).

Otro artículo que utiliza esta misma técnica es el artículo (Tati, 2022). En este artículo se centraron en detectar los diferentes niveles de sombreado parcial en condiciones de MPPT utilizando los datos de tensión y corriente (I-V). Para realizar esta tarea, utilizaron una Red

Neuronal Artificial, logrando precisiones que van del 97% hasta el 98%. Cabe destacar, que solo entrenaron la red neuronal para detectar un tipo de fallo.

## 5.4.2 Árboles de decisión

Se trata de un modelo usado tanto en regresión como en clasificación. Su nombre viene dado por la forma en que se representa, formado por un nodo inicial, del que parten ramas que representan las posibles acciones o resultados del nodo anterior y que llegan a otro nodo, así sucesivamente hasta llegar a un nodo terminal o nodo hoja. Estos nodos hoja son todos los resultados posibles que podemos obtener del modelo.

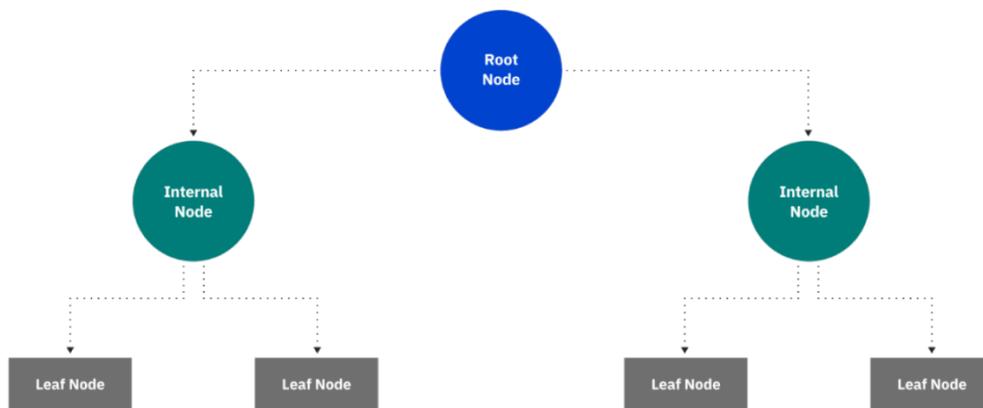


Figura 14 - Diagrama de un árbol de decisión. (IBM, s.f.)

Estos modelos están muy extendidos debido a que son fáciles de entender, pero cuando la tarea es muy compleja el árbol puede ser muy extenso. Cuando esto ocurre se utilizan otras técnicas que combinan múltiples diagramas más sencillos, pero que sumados crean un modelo muy preciso:

- Bagging: crea múltiples árboles de decisión escaneando repetidas veces los datos introducidos y le asigna un valor, un peso, a cada árbol para posteriormente combinarlos.
- Clasificador de selva aleatoria: múltiples árboles de decisión utilizados para aumentar la tasa de clasificación.
- Random forest: los árboles son creados aplicando primeramente un análisis de componentes principales a una porción aleatoria los datos de entrada.

### 5.4.3 Máquinas de soporte de vectores

Las máquinas de soporte de vectores son algoritmos que buscan el hiperplano que tenga los elementos con distintas etiquetas separados el máximo posible, es decir, con el mayor margen que se pueda obtener. Este margen es la distancia máxima entre dos vectores de soporte. Estos vectores de soporte son vectores paralelos al hiperplano, uno que pasa por el punto más cercano al hiperplano con una etiqueta y otro que pasa por el punto más cercano de la otra etiqueta. Esta búsqueda de la mayor distancia entre los vectores de soporte hace que el entramiento de estos algoritmos se asemeje mucho a in problema de optimización cuadrática.

Estos algoritmos utilizan un kernel para transformar los datos a un espacio de con un número mayor de dimensiones intentando que la separación entre etiquetas o clases sea más sencilla y con una mayor distancia.

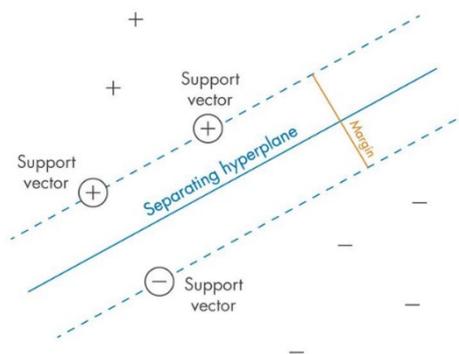


Figura 15 - Diagrama máquina de vectores de soporte. (IArtificial.net, s.f.)

El artículo (Etemadi, 2017) utilizó una máquina de soporte de vectores (Support Vector Machine, SVM) de dos etapas. A esta SVM la entrenaron utilizando únicamente datos de I-V, extrayendo las características principales de las medidas. Gracias a este modelo y a la extracción de datos, pudieron obtener una precisión de entre 91% y 95% al clasificar fallos de descompensación.

## 5.5 Métodos de validación

### 5.5.1 Validación cruzada k-fold

Esta técnica es útil cuando se trata de pocas mediciones y permite entrenar el modelo con un conjunto de datos más extenso al tiempo que evita el sobreajuste. Divide los datos en "pliegues" más pequeños y con ellos crea iteraciones con diferentes de datos de entrenamiento y de prueba.

Una vez ha completado todas las iteraciones hace una media aritmética con los resultados de cada iteración para así obtener un único resultado.

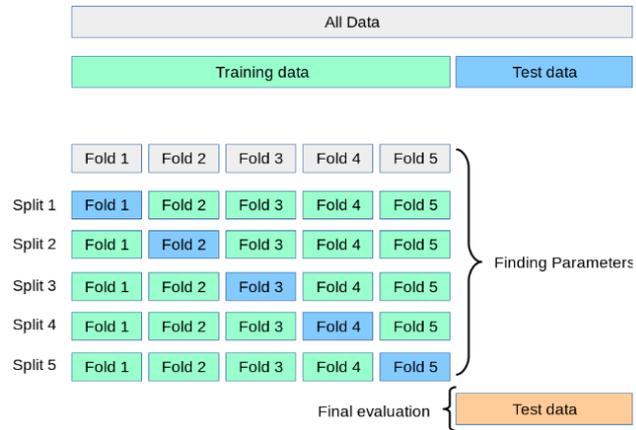


Figura 16 - Diagrama de validación cruzada. (Scikit Learn, s.f.)

### 5.5.2 Método hold-out

Es el método más sencillo, consiste en dividir los datos en dos conjuntos, uno de entrenamiento y otro de prueba. Este método tiene como ventaja su sencillez, ya que no hay que realizar operaciones adicionales. Además, su simplicidad hace que su entrenamiento sea el más rápido.

Por otro lado, este método solo es funcional cuando nuestra base de datos a entrenar es muy grande. Si aplicamos este método a una base de datos de tamaño reducido el modelo estará sobreentrenado con los datos de entrenamiento y no será capaz de dar buenos resultados para los datos de prueba o futuros datos que le añadamos.

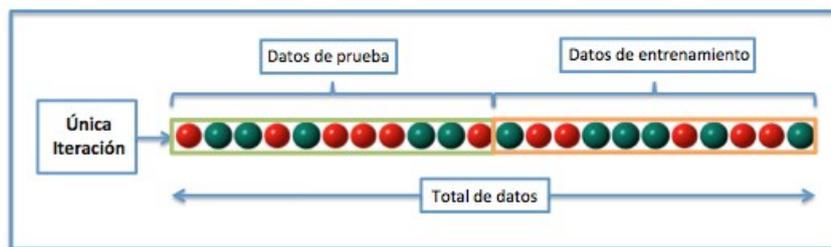


Figura 17 - Diagrama del método hold-out. (Wikipedia, 2020)

## 6. Caso de estudio

En este trabajo, se ha utilizado la base de datos proporcionada por el artículo (Bakdi, 2020) para entrenar todas las pruebas realizadas a los modelos. La base de datos consta de  $2.2 \cdot 10^6$  mediciones, incluidas las de los módulos que funcionan correctamente y las de los módulos que presentan algún tipo de fallo. La base de datos cuenta con mediciones de 7 tipos de fallos diferentes: circuito abierto, caídas de voltaje, sombreado parcial, inversor, sensor de retroalimentación de corriente y controlador MPPT / LPPT en fallos del convertidor elevador. Además, las mediciones se llevaron a cabo tanto en modo LPPT como en modo MPPT. Para cada caso, tenemos los siguientes datos:

- Tiempo: Tiempo de medición real en segundos. El muestreo promedio es  $T_s = 9.9989 \mu s$ .
- Ipv: Medición de corriente de modulo fotovoltaico.
- Vpv: Medición de voltaje de modulo fotovoltaico.
- Vdc: Medición de voltaje DC.
- IA: Fase\_A medición de corriente.
- IB: Fase\_B medición de corriente.
- IC: Fase\_C medición de corriente.
- VA: Fase\_A medición de voltaje.
- VB: Fase \_B medición de voltaje.
- VC: Fase \_C medición de voltaje.
- Iabc: Magnitud actual estimada de secuencia positiva.
- Si: Frecuencia de corriente estimada de secuencia positiva.
- Vabc: Magnitud de voltaje estimada de secuencia positiva.
- Vf: Frecuencia de voltaje estimada de secuencia positiva.

Esta base de datos fue creada registrando las medidas en un laboratorio gracias a un simulador de placas solares y otro simulador de la red eléctrica. El hecho de utilizar simuladores en lugar de elementos reales es para poder simular los fallos de forma controlada sin deteriorar ningún componente real o alterar la tensión de la red eléctrica, El simulador de placas solares lo que permite es generar la tensión que generaría un array completo de módulos fotovoltaicos utilizando un elemento del tamaño de una fuente de alimentación. A pesar de no utilizar módulos fotovoltaicos sí que utilizaron un inversor y conversor elevador real, junto a una carga de alta potencia para simular la instalación a la que iría conectado todo el sistema.

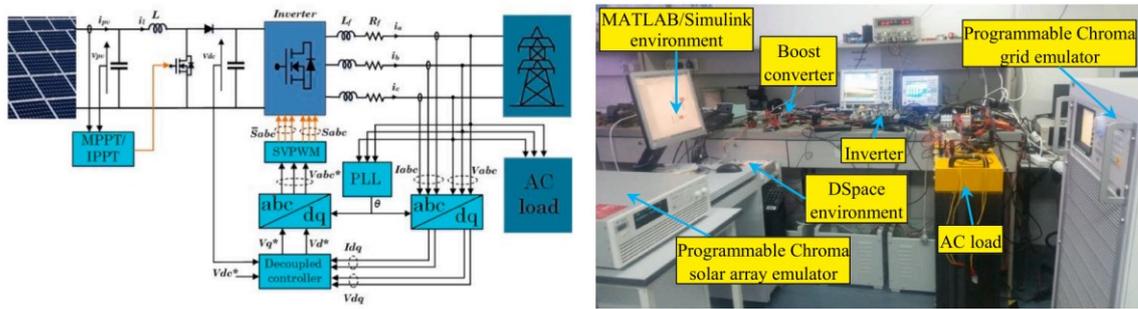


Figura 18 - Sistema utilizado para generar la base de datos.

## 6.1 Estructura de la base de datos.

Los ficheros de datos GPVS-Faults están organizados y etiquetados como "Fxy", donde:

- $x \in \{0, 1, \dots, 7\}$  representa el escenario de error, donde:
  - '0' significa experimento sin fallos. Esto se puede utilizar para el entrenamiento.
  - '1', ..., '7' son los siete tipos de fallos realizados en los experimentos.
- $y \in \{L, M\}$  representa el modo de operación durante el cual se realizó el experimento:
  - 'L' significa modo de potencia limitada (LPPT)
  - 'M' significa modo de potencia máxima (MPPT)
- Por ejemplo, "F4M" es un archivo de datos para un experimento que incluye un fallo F4 durante el modo MPPT, y "F1L" incluye el fallo F1 durante el modo LPPT.

Código de error	Medidas	Total por código de fallo
F0L	143715	284729
F0M	141014	
F1L	129013	268027
F1M	139014	
F2L	142128	286143
F2M	144015	
F3L	103497	173464
F3M	69967	
F4L	144014	288028
F4M	144014	
F5L	143015	287029
F5M	144014	
F6L	144015	288029
F6M	144014	
F7L	144015	288030
F7M	144015	
<b>Total</b>		<b>2163479</b>

Tabla 1 - Cantidad de mediciones.

Fallo	Tipo	Descripción
F1	Fallo del inversor	Error completo en uno de los seis IGBT <sup>1</sup>
F2	Error del sensor de retroalimentación	Fallo del sensor monofásico 20%
F3	Anomalía en la red eléctrica	Caídas intermitentes de voltaje
F4	Desajuste de matriz fotovoltaica	10 a 20% de sombreado parcial no homogéneo
F5	Desajuste de matriz fotovoltaica	15% de circuito abierto en matriz fotovoltaica
F6	Error del controlador MPPT/LPPT	-20% de parámetro de ganancia del controlador PI <sup>2</sup> en el controlador MPPT/IPPT del convertidor elevador
F7	Error del controlador del convertidor Boost	+20% en la constante de tiempo del controlador PI en el controlador MPPT / LPPT del convertidor elevador

*Tabla 2 - Códigos de fallo y su descripción.*

## 7. Herramientas utilizadas

Para tratar y modificar los datos se utilizó MATLAB. Se creó un script en el que los datos se cargaron y trataron para ajustarse a los requisitos para un análisis posterior. Para crear, entrenar y probar los diferentes modelos, MATLAB nos permite instalar un complemento llamado "Classification Learner" esta herramienta simplifica el proceso, permitiéndonos elegir cómo se dividen los datos en train y test. Además, esta herramienta nos ofrece 31 modelos de entrenamiento para predecir nuestros datos. La apertura y configuración de esta herramienta también están automatizadas en el script anterior.

Una vez entrenados todos los modelos, se utilizó Excel para recopilar toda la información, crear gráficos con esa información y compararlos en un análisis posterior.

<sup>1</sup> Del inglés Insulated Gate Bipolar Transistor.

<sup>2</sup> Proporcional Integral, un tipo de controlador.

## 7.1 Cómo usar el complemento Classification Learner

“Classification Learner” es una herramienta perteneciente al add-on llamado “Statistics and Machine Learning Toolbox”. Esta herramienta se utiliza para entrenar modelos de clasificación basados en datos etiquetados, lo que llamamos un aprendizaje supervisado. Es muy útil ya que de una forma sencilla podemos especificar qué modelo queremos entrenar o incluso podemos entrenar todos los modelos disponibles para que nos encuentre el que mejor funciona. Otra gran ventaja de esta herramienta es que una vez entrenaos y validados los modelos nos genera las matrices de confusión de los mismos para poder analizar su desempeño.

Para abrir esta herramienta debemos utilizar el comando “classificationLearner”, esto nos abrirá la herramienta y en caso de tenerla abierta nos preguntará si queremos cerrar la sesión ya iniciada.

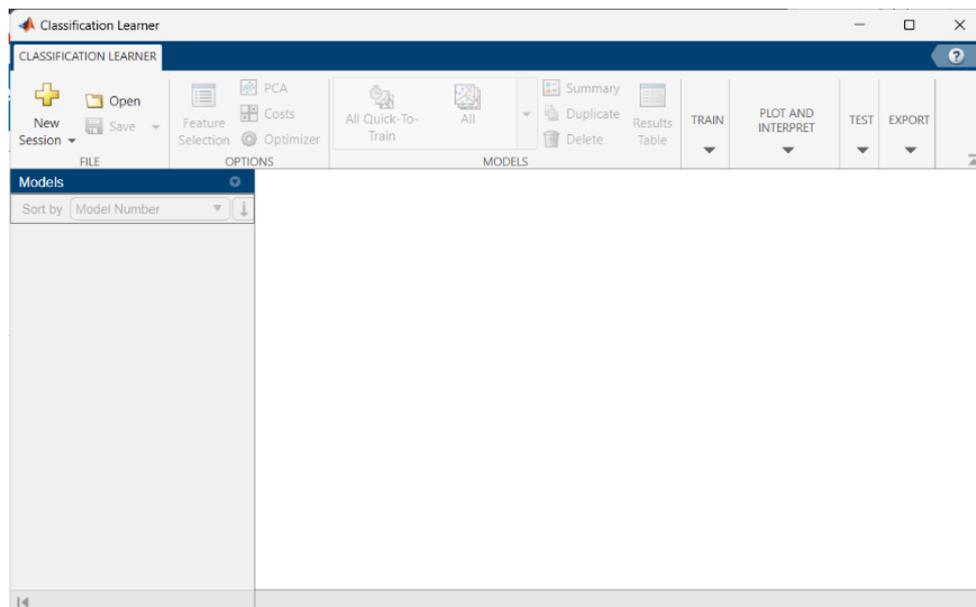


Figura 19 - Classification Learner al abrirlo por primera vez.

Al iniciarla por primera vez lo único que podremos hacer será crear una nueva sesión, es decir, añadir datos y entrenar modelos, o abrir alguna sesión que hayamos guardado anteriormente. Si creamos una nueva sesión nos abrirá una nueva ventana de diálogo que nos permitirá seleccionar los datos que vamos a usar, los cuales deben estar definidos en el Workspace y nos permite elegir el tipo de validación que queremos utilizar.

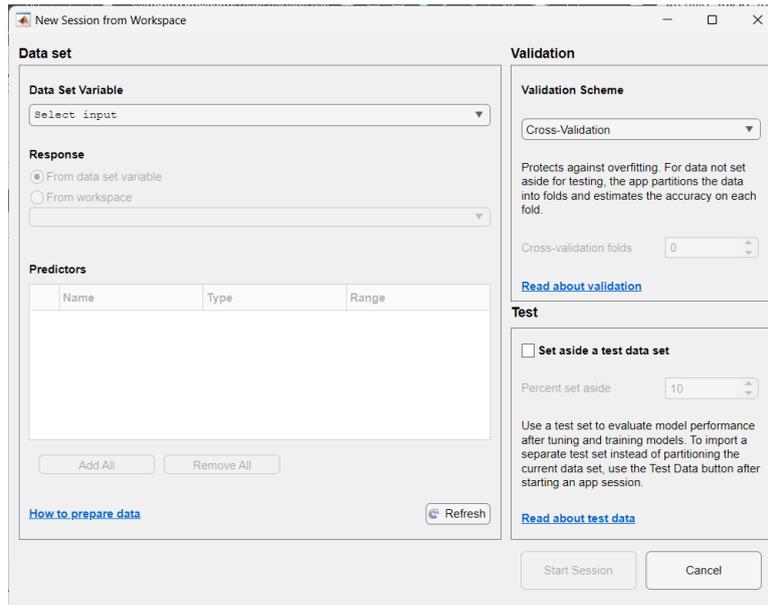


Figura 20 - Selección de datos de entreno en Classification Learner.

Una vez hemos seleccionado los datos para entrenar los modelos nos los cargará y podremos seleccionar la variable de respuesta y los predictores que utilizará para calcular esas respuestas. Para terminar de configurar la herramienta y poder empezar a entrenar los modelos debemos hacer click en “Start Session”. Con los datos ya cargados nos hará una representación de los mismos.

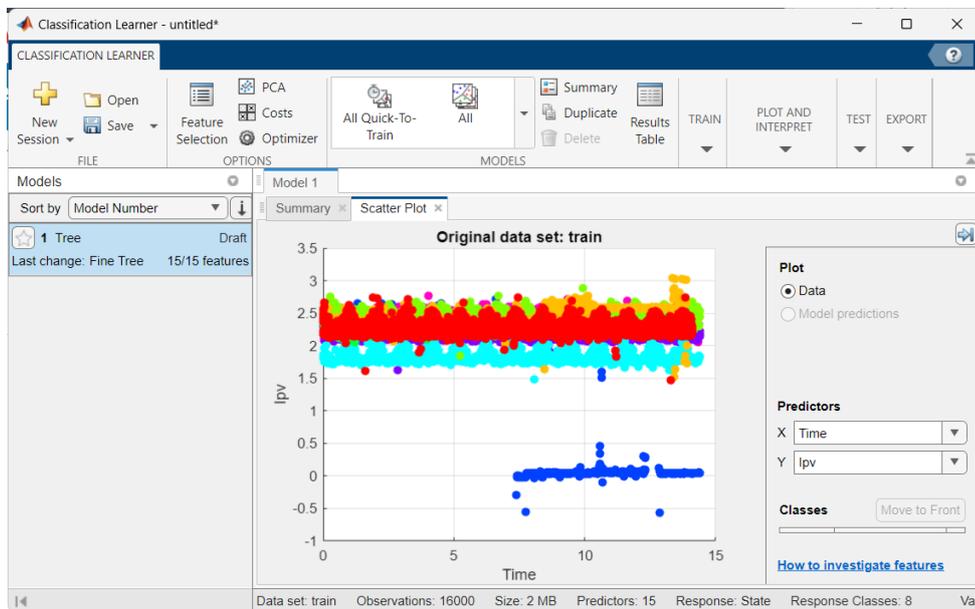


Figura 21 - Representación de los datos en Classification Learner.

Además, se han habilitados los apartados de la parte superior. Estos apartados nos permiten modificar los predictores a usar, añadir funciones de coste y optimizar los modelos que vamos a usar, además de dejarnos elegir los modelos a entrenar.

Para elegir un modelo, o varios, abrimos es desplegable del apartado “Models” y elegimos un modelo y pulsamos en el apartado “Train” y en “Train selected”, esto comenzará a entrenar los modelos y a validarlos. Cuando termine de entrenar el modelo o modelos seleccionado a la izquierda nos parecerán todos, indicándonos la precisión obtenida y si los vamos seleccionando nos mostrará las matrices de confusión obtenidas.

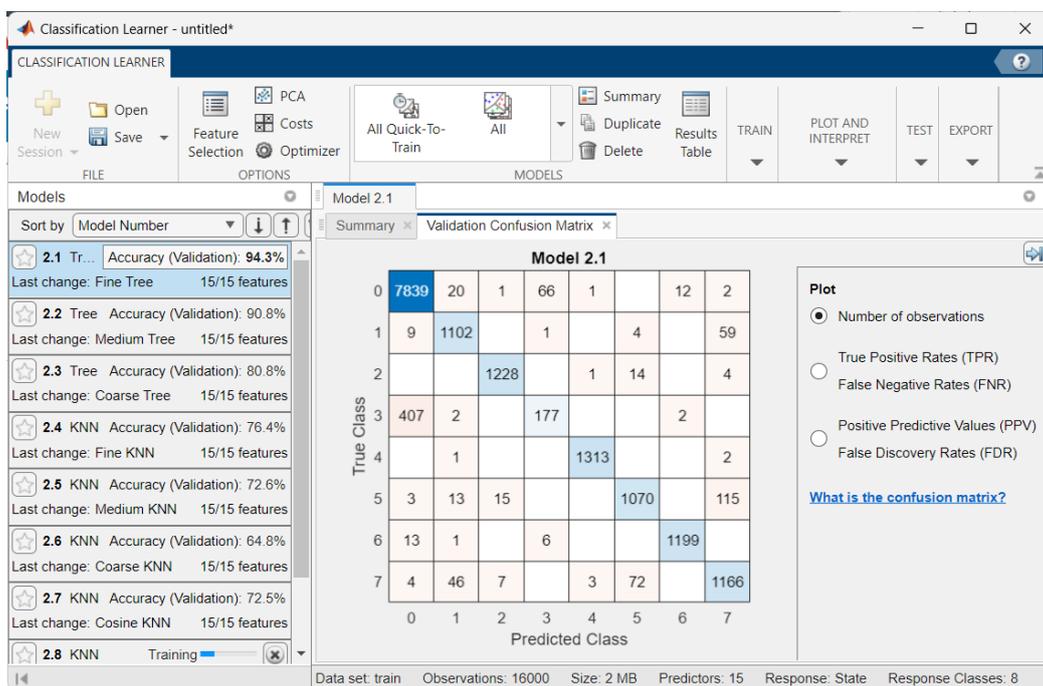


Figura 22 - Ejemplo de un modelo entrenado en Classification Learner.

Si entramos en el apartado superior “Plot and interpret” podremos crear otros diagramas y figuras que nos pueden ayudar a analizar el desempeño del modelo, como una curva ROC o un diagrama de dispersión.

Si queremos exportar el modelo podemos hacerlo en el apartado “Export” y en “Generate function”, esto nos creará una función específica para el modelo seleccionado, lo que nos permitirá usar el modelo directamente en el Workspace sin tener que abrir Classification Learner para clasificar nuevos datos.

## 8. Tratamiento de los datos

Dado que la base de datos es demasiado grande, llevará mucho tiempo entrenar y probar los 31 modelos disponibles. Para hacer que este proceso sea más rápido, permitiéndonos entrenar todos los modelos para una correcta comparación entre ellos, los datos se han reducido aleatoriamente a un conjunto de datos menor, con un total de 16000 mediciones. A pesar de que los datos se redujeron aleatoriamente, nos aseguramos de que la proporción de datos "Sin fallos" y "Con fallos" fuera similar para evitar cualquier tipo de sobreajuste, ya que la base de datos consta de casi 10 veces más datos "Con fallos" que "Sin fallos". Esta relación no se consideró con respecto a la cantidad de mediciones entre los diversos fallos, ya que la base de datos es consistente en el número de mediciones entre los diferentes fallos.

Para garantizar una reproducibilidad adecuada del entrenamiento y la prueba para una evaluación consistente de los modelos, la semilla utilizada para reducir los datos de forma aleatoria se estableció en "Default".

Dado que vamos a realizar un entrenamiento supervisado debemos de etiquetar las mediciones antes de entrenar los modelos, por lo que se agregaron dos nuevas columnas. La primera columna indica si esa medición se realizó en condiciones de MPPT o LPPT. Esta columna se ha utilizado para evaluar si ha habido alguna diferencia entre los diferentes modos cuando se trata de un modelo entrenado con datos mixtos. La segunda columna es la respuesta esperada, la etiqueta que indica si esa medición es de un módulo que funciona correctamente o tiene algún tipo de fallo. Esta etiqueta será la que los modelos necesitarán predecir utilizando los otros 15 predictores (14 mediciones y la columna que agregamos). Para poder clasificar los tipos de fallos esta etiqueta no solo indica si el sistema funciona correctamente o no, además, indica el tipo de fallo (1-7).

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16		
Time	Ipv	Vpv	Vdc	ia	ib	ic	va	vb	vc	labc	If	Vabc	Vf	PM	Fault		
0.0021	1.4919	101.0498	143.5547	-0.3701	0.4230	-0.1267	124.4019	-142.7367	19.0139		1	50.0000		1	50.0000	0	0

*Figura 23 - Predictores y respuesta.*

Todo esto se realizó utilizando un script de Matlab para hacer que el flujo de trabajo fuera más rápido y, en caso de ser necesario, rehacer los experimentos lo más rápido posible.

## 8.1 Cómo funciona el script

Un script es un archivo disponible en MATLAB que nos permite escribir y guardar líneas de código para poder ejecutarlas cuando nos sea preciso.

El Script creado en MATLAB para realizar los test de forma más automatizada está dividido en diversas secciones. Cada sección se encarga de realizar una tarea, como la de cargar los datos o la de reducir la muestra.

La primera tarea, y sección del script, es la de cargar todas las mediciones que tenemos disponibles en el programa. Esto se realiza con el comando `load();`, indicando en los paréntesis el nombre del archivo que queremos cargar. Por ejemplo `load('F5L.mat');`, este comando nos cargaría el archivo F5L con la extensión `.mat` en una tabla con el mismo nombre que el archivo, pero sin la extensión (F5L). Es importante que estos archivos estén en la misma carpeta que el script, de no ser así habría que indicar la ruta del archivo.

Una vez tenemos las mediciones cargadas en el programa debemos etiquetarlas, indicando el modo de potencia y si se trata de una medición con funcionamiento correcto o si se trata de algún error. Esto lo haremos añadiendo nuevas columnas a la tabla con el comando `F7L.Var15(:)=7;`, este comando lo que hace es asignarle a la tabla F7L, en la columna 15 el valor 7, que es el código de error de las mediciones de esa tabla. Como la tabla no contaba con 15 columnas lo que hace el comando es crearla directamente y añadir este valor a todos los elementos de la tabla. Para una mejor comprensión de los elementos de la tabla es conveniente darle un nombre apropiado a la columna, esto se hace utilizando `F0L.Properties.VariableNames{15} = 'State';`, lo que le asigna a la variable (columna) 15 el nombre de "State".

Repetiremos lo mismo en una nueva columna para indicar el modo de potencia de las mediciones. Un ejemplo de esto: `F4M.Var16(:) = 1;` y `F4M.Properties.VariableNames{16} = 'PM';`. En este caso un 1 indica MPPT y un 0 LPPT.

El siguiente paso es reducir la muestra de mediciones hasta 16000, para posibles futuras variaciones en los test se ha creado una variable que determina la cantidad de datos `data = 16000`. Para fijar la semilla para poder reproducir los datos usamos `rng('default');`. Como vamos a realizar diversos test debemos de crear diferentes tablas para cada caso. Para el primer test, en el que no hacemos distinción de los tipos de fallo, primero creamos dos tablas que contengan los datos de un funcionamiento correcto y los datos que contengan los fallos y luego

los reducimos aleatoriamente hasta que contengan el mismo número de mediciones entre ellas, para así evitar que haya algún tipo de sobreentrenamiento debido al número de mediciones de cada caso:

```
Good=[ F0L;F0M];
Bad=[ F1L;F1M;F2L;F2M;F3L;F3M;F4L;F4M;F5L;F5M;F6L;F6M;F7L;F7M;F7L];

rng('default');
tmp1 = randsample(1:height(Bad), height(Good));
tmp2 = Bad(tmp1,:);
Mix=[Good;tmp2];
```

En este caso Mix contiene el mismo número de mediciones correcta como de fallos. Se han creado tablas similares para los diferentes tests, por ejemplo, para estudiar las diferencias entre modos de potencia se usa “tmp3=[F1L;F2L;F3L;F4L;F5L;F6L;F7L];” y para estudiar cada tipo de error “Fail1=[Good;F1L;F1M];”.

El siguiente paso es reducir estas nuevas tablas hasta conseguir el número de mediciones que deseamos, para facilitar el cambio de tablas a reducir en cada test se ha utilizado el comando “switch(selector)”, que nos permite ejecutar diferentes partes de código dependiendo del valor de la variable “selector”. Cuando “selector” tiene un valor de 1 a 7 utiliza los datos de un solo tipo de fallo, cuando vale 8 utiliza datos MPPT, cuando vale 9 utiliza datos LPPT y cuando vale 10 utiliza todos los datos. Los comandos para reducir la muestra son similares a los utilizados anteriormente:

```
switch(selector)
    case 10
        index = randsample(1:height(MMix),data);
        train = MMix(index,:);
end
```

“train” será la tabla que introduciremos para entrenar y validar los modelos en los test.

Como último paso queda abrir el complemento “Classification Learner”, para automatizar este proceso se utiliza el comando “classificationLearner(train, 'State', 'CrossVal','on');” que además de abrir la herramienta la preconfigura por nosotros cargando la tabla “train”, indicando que la columna “State” es la que tiene que predecir y asignando “CrossVal” (validación cruzada) como método de partición de datos. El atributo “on” especifica que la validación cruzada sea de 5 pliegues, si utilizamos “off” sería una validación por sustitución.

El Script completo se encuentra en el ANEXO 2, en él se explica todo el código línea a línea.

## 9. Resultados

### 9.1 Comparación entre familias y modelos

Una vez preparados todos los datos, el siguiente paso fue entrenar el modelo. Como se mencionó, el complemento "Classification Learner" es la herramienta para realizar la tarea. Los 31 modelos disponibles se entrenaron utilizando todas las 16000 mediciones aleatorias exactas, dejando "Predeterminado" todos los ajustes del modelo y dividiendo los datos utilizando una técnica de "validación cruzada" con cinco pliegues.

Una vez entrenados los modelos, se recogió toda la información y resultados de los mismos, como se ve en la Tabla 3:

Nombre del modelo	Precisión (%)	Coste total	Velocidad de predicción (obj/s)	Tiempo(s) de entrenamiento
Fine Tree	85,9%	2253	140000	28,544
Medium Tree	77,8%	3546	410000	4,8581
Coarse Tree	64,1%	5739	390000	1,4732
Linear Discriminant	60,4%	6337	150000	3,5111
Quadratic Discriminant	77,3%	3625	210000	1,8517
Gaussian Naive Bayes	53,5%	7445	170000	3,8191
Kernel Naive Bayes	77,0%	3684	200	341,37
Linear SVM	82,0%	2875	52000	224,24
Quadratic SVM	90,7%	1495	26000	1635,9
Cubic SVM	87,2%	2053	19000	4306,7
Fine Gaussian SVM	79,3%	3304	1500	179,23
Medium Gaussian SVM	79,1%	3347	3300	78,047
Coarse Gaussian SVM	69,8%	4834	2100	87,958
Fine KNN <sup>3</sup>	68,3%	5069	11000	14,797
Medium KNN	68,7%	5004	11000	7,2603
Coarse KNN	60,3%	6345	8200	9,1016
Cosine KNN	68,5%	5038	11000	7,5188
Cubic KNN	67,9%	5135	290	237,88
Weighted KNN	70,7%	4686	11000	7,1996
Boosted Trees	82,5%	2804	29000	31,108
Bagged Trees	92,2%	1252	27000	17,311

<sup>3</sup> K vecinos más cercanos, en inglés K Nearest Neighbors.

Subspace Discriminant	55,1%	7187	17000	8,8583
Subspace KNN	71,6%	4551	6900	16,326
RUS Boosted Trees	72,7%	4367	36000	12,284
Narrow Neural Network	88,8%	En	270000	75,468
Medium Neural Network	91,8%	En	330000	91,132
Wide Neural Network	92,0%	En	280000	189,37
Bilayered Neural Network	89,4%	En	350000	78,093
Trilayered Neural Network	89,2%	En	360000	88,667
SVM Kernel	57,2%	6842	62000	119,68
Logistic Regression Kernel	53,6%	7422	5500	42,408

Tabla 3 - Resultados de los modelos.

Existen diferentes familias de modelos para clasificación, como las “Neural Networks” (NN). Eligiendo el modelo más preciso de cada familia, podemos ver que la familia con una precisión más alta es "Ensemble classifier", que utiliza "Bagged trees" para lograr una precisión del 92,2%. La siguiente mejor familia es "Neural Networks" utilizando una "Wide Neural Network", obteniendo una precisión del 92%.

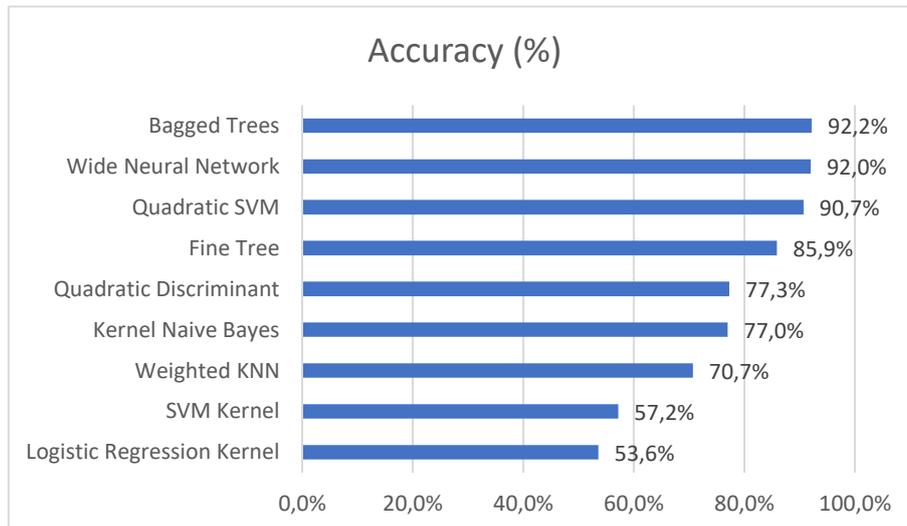


Figura 24 - Precisión del mejor modelo de cada familia.

El resultado es bastante diferente si observamos los tiempos de entrenamiento de los modelos más precisos. Podemos ver la gran diferencia de tiempo de entrenamiento que hay entre el modelo más rápido y el más lento. El más rápido es de la familia de los "Discriminant classifiers" con el modelo "Quadratic Discriminant", que solo requirió 1.85s para entrenar. Por

otro lado, la familia "Support Vector" fue la más lenta, con el modelo "Quadratic SVM" tardando más de 27 minutos en entrenar, a pesar de haber reducido el número de mediciones a 16.000.

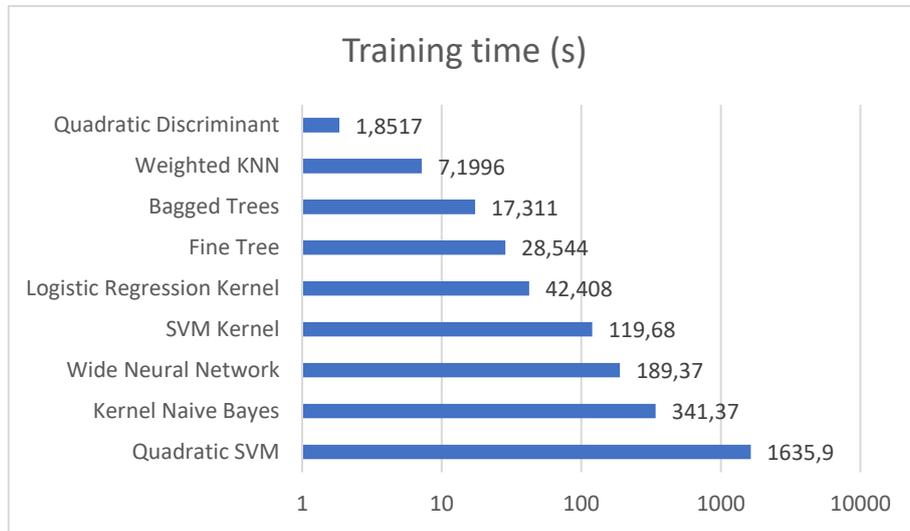


Figura 25 - Tiempo de entrenamiento del modelo más preciso de cada familia. Eje en formato logarítmico.

Ahora centrémonos en los diez modelos más precisos, independientemente de su familia. Podemos ver que entre los modelos más precisos hay algunos pertenecientes a la misma familia, como por ejemplo las Redes Neuronales.

Model Name	Precisión (%)	Coste total	Velocidad de predicción (obj/s)	Tiempo (s) de entrenamiento
Bagged Trees	92,2%	1252	27000	17,311
Wide Neural Network	92,0%	En	280000	189,37
Medium Neural Network	91,8%	En	330000	91,132
Quadratic SVM	90,7%	1495	26000	1635,9
Bilayered Neural Network	89,4%	En	350000	78,093
Trilayered Neural Network	89,2%	En	360000	88,667
Narrow Neural Network	88,8%	En	270000	75,468
Cubic SVM	87,2%	2053	19000	4306,7
Fine Tree	85,9%	2253	140000	28,544
Boosted Trees	82,5%	2804	29000	31,108

Tabla 4 - Resultados de los diez modelos con más precisión.

El más preciso es "Bagged Trees", con una precisión del 92,2% y un tiempo de entrenamiento de solo 17,31s. Las NN no están lejos de esa precisión, pero requieren más tiempo de entrenamiento.

## 9.2 Diferencia entre modos de potencia

Dado que la base de datos que hemos utilizado está compuesta por mediciones bajo MPPT y LPPT, hemos estudiado si la precisión es diferente en una condición u otra. Como antes, hemos entrenado todos los modelos disponibles en el complemento "Classification Learner" de MATLAB. Los datos se dividieron en entrenamiento y prueba con una validación cruzada de cinco pliegues y dejando como "Predeterminado" todas las propiedades del modelo.

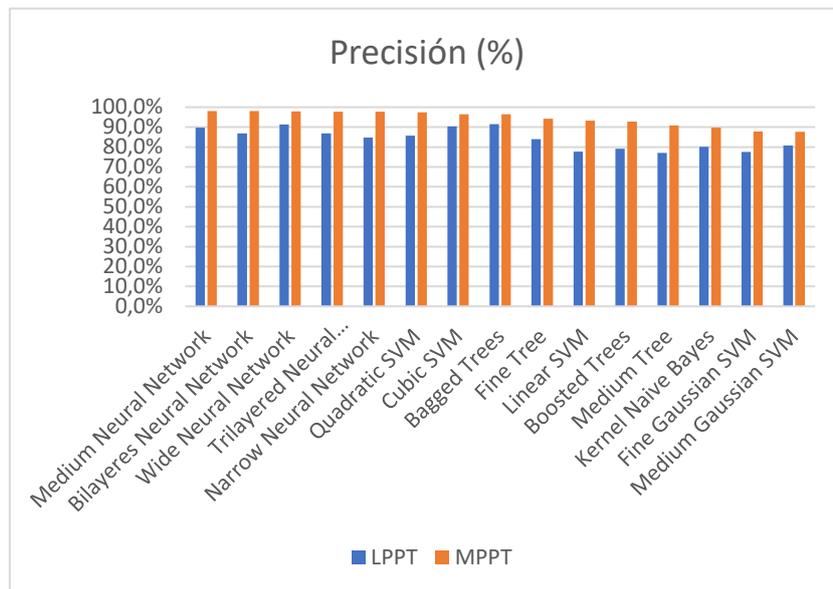


Figura 26 - Comparación de la precisión de los modelos bajo MPPT y LPPT.

Hay una ligera diferencia entre la precisión de los modelos entrenados utilizando datos MPPT y LPPT. Cuando se utilizan datos MPPT, los modelos son más precisos.

Además, el modelo más preciso para MPPT no es el más preciso para LPPT. Cuando nuestro sistema está bajo MPPT, puede valer la pena usar una "Midium Neural Network" para pronosticarlo, ya que tiene una precisión del 98.1%.

Model name	Precisión (%)
Medium Neural Network	98,1%
Bilayered Neural Network	98,0%
Wide Neural Network	97,9%
Trilayered Neural Network	97,8%
Narrow Neural Network	97,7%
Quadratic SVM	97,5%
Cubic SVM	96,5%
Bagged Trees	96,4%
Fine Tree	94,3%
Linear SVM	93,3%

Tabla 5 - Precisión de los modelos más precisos bajo MPPT.

Pero si nuestro sistema está bajo LPPT, si usamos el modelo "Bagged Trees" obtendremos unos resultados más precisos, llegando al 91.5%. Podríamos utilizar un modelo u otro, dependiendo de cómo funcione nuestro sistema fotovoltaico. Otra opción es desarrollar un algoritmo que cambie ambos dependiendo del modo de energía.

Model Name	Precisión (%)
Bagged Trees	91,5%
Wide Neural Network	91,3%
Cubic SVM	90,3%
Medium Neural Network	89,8%
Bilayered Neural Network	86,9%
Trilayered Neural Network	86,9%
Quadratic SVM	85,8%
Narrow Neural Network	84,8%
Fine Tree	83,9%
Medium Gaussian SVM	80,8%

Tabla 6 - Precisión de los modelos más precisos bajo LPPT.

Dado que "Bagged Trees" fue el modelo más preciso cuando se entrenó con ambos tipos de datos, estudiamos cómo se desempeñó en ambas condiciones por separado.

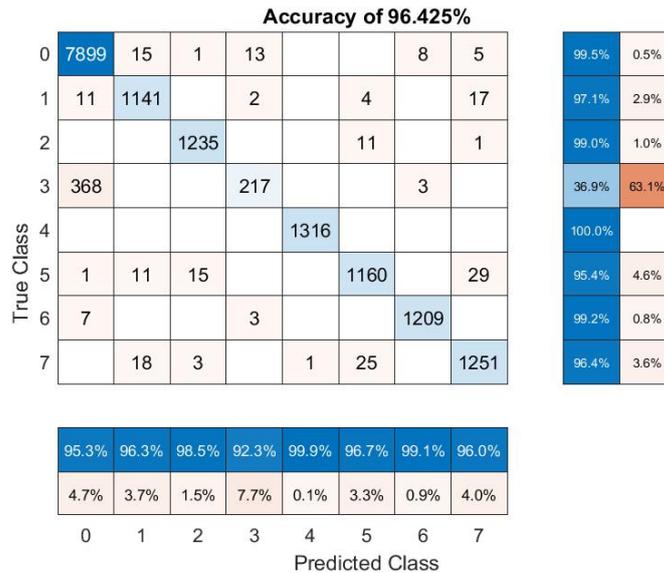


Figura 27 - Matriz de confusión del modelo "Bagged Trees" usando datos MPPT.

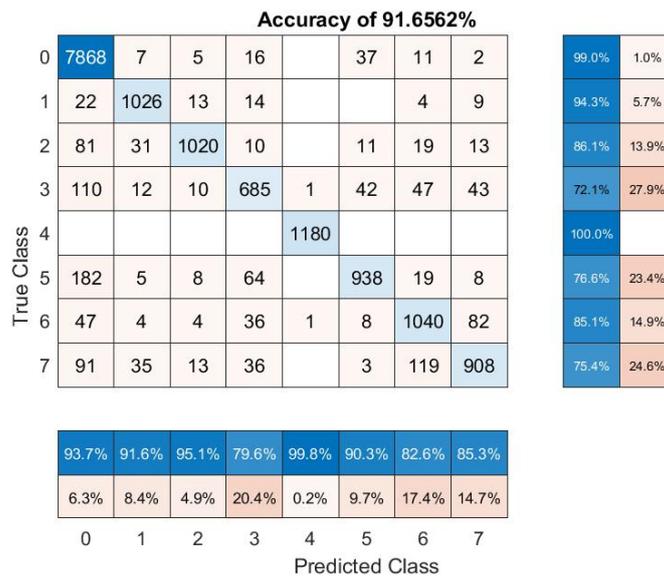
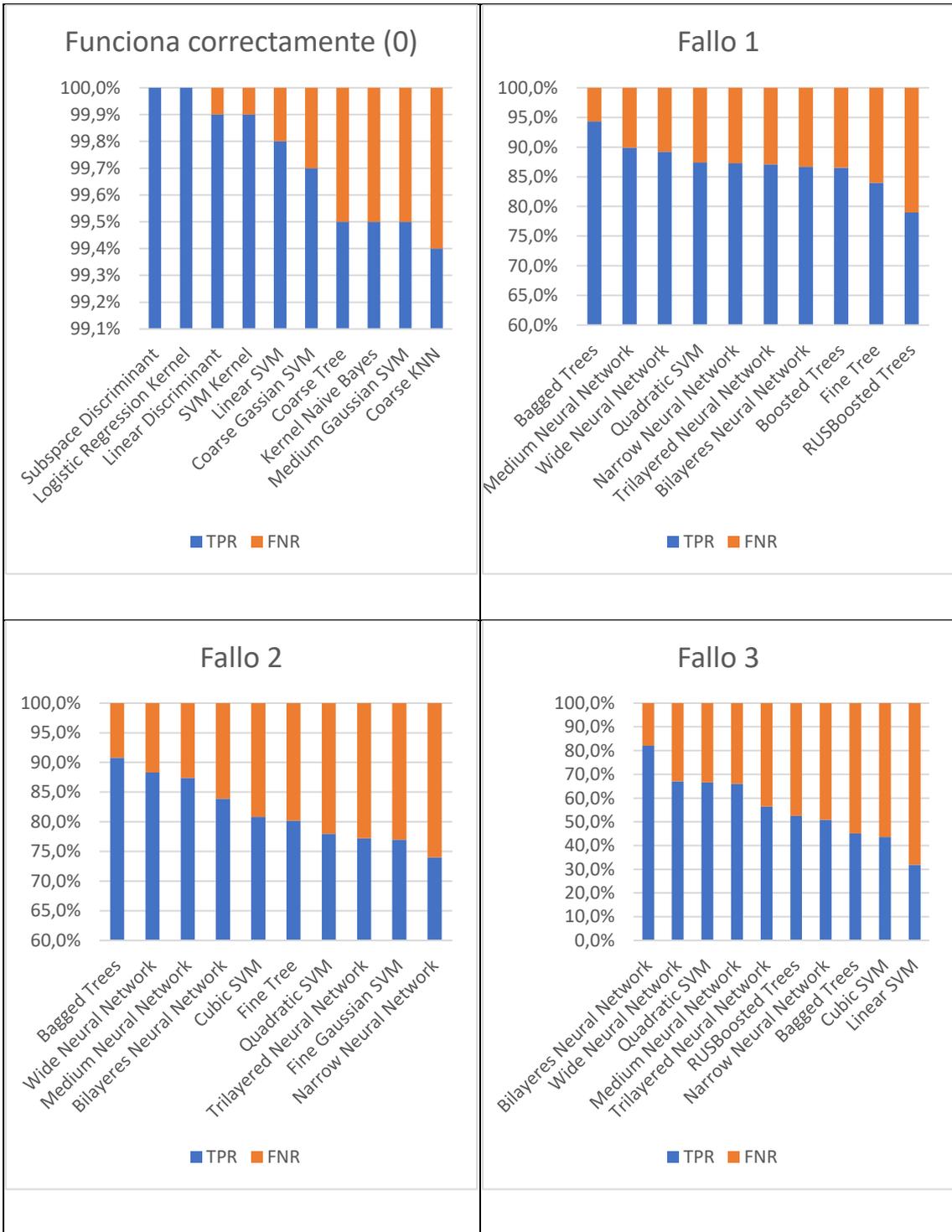


Figura 28 - Matriz de confusión del modelo "Bagged Trees" utilizando datos LPPT.

Hay algunas diferencias en los resultados. Lo primero que podemos notar es la precisión media en cada caso, hay una diferencia del 4,77%. Otro hecho a considerar es que cuando está bajo MPPT, nuestro modelo no puede predecir correctamente el "Fallo 3", que son "Anomalía de la red". Este problema en la capacidad de predicción del modelo puede deberse a los valores de potencia que se obtienen en cada caso.

### 9.3 Diferencia entre fallos



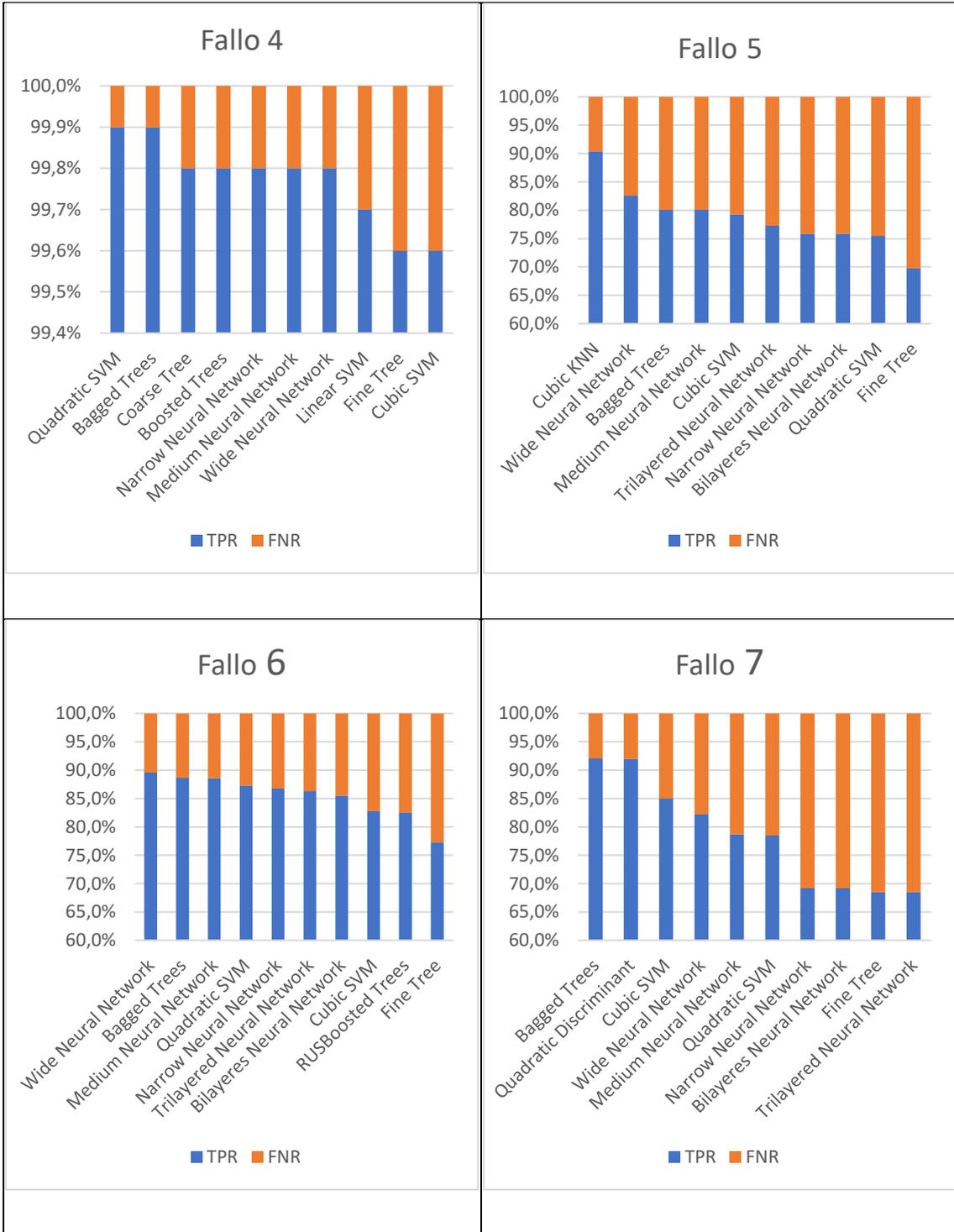


Tabla 7 - TPR y FNR de los modelos con cada tipo de fallo.

Una vez que habíamos entrenado todos los modelos, observamos sus matrices de confusión, las tasas de verdaderos positivos (TPR) y las tasas de falsos negativos (FNR), que indican la precisión de cada modelo en cada uno de los fallos.

Casi todos los modelos tenían más del 99% de precisión para pronosticar un funcionamiento correcto en nuestro sistema (Código 0) y "Fallo 4" (que se ocupa del sombreado parcial). En promedio, "Bagged Trees" y varias redes neuronales fueron los modelos que mejor funcionaron en todos los casos.

Todos los fallos están en el mismo rango de precisión, excepto el "Fallo 3", que tiene la precisión más baja. Esta diferencia con los otros tipos de fallos podría deberse a que los datos de entrenamiento no tenían suficientes mediciones de este tipo o debido a la naturaleza del fallo. En este caso, se trata de una anomalía en la red eléctrica, que es más compleja de pronosticar solo utilizando las mediciones de los módulos.

## 10. Sistema de desconexión

Observando los resultados de los tests realizados, se ha visto que para una instalación fotovoltaica en la que queremos aplicar el modelo predictivo a todos los fallos posibles es conveniente usar el modelo "Bagged Trees". Este modelo ha sido el que mejores resultados ha obtenido al ser entrenado bajo todo tipo de fallos y modos de seguimiento del punto de máxima potencia. La versatilidad de éste hace que sea idóneo para la mayoría de los sistemas fotovoltaicos ya que no se quiere sacrificar precisión global para aumentar la de un solo tipo de fallo.

Una vez se ha dedicado utilizar "Bagged Trees" en el sistema de desconexión, hay que estudiar cómo afrontar los errores que se puedan detectar. Una vez ha ocurrido un error es conveniente que un operario revise todo el sistema en busca de las causas del fallo e intente solucionarlas, pero esto puede requerir tiempo, durante el cual el sistema fotovoltaico puede seguir dañándose. Es necesario un sistema que actúe de forma inmediata cuando se detecta un fallo y avise a los trabajadores para que el sistema sea revisado.

### 10.1 Procedimiento tras la detección de errores

#### 10.1.1 Análisis de los resultados de la predicción

La interpretación de los resultados otorgados por el modelo es muy sencilla, ya que el modelo nos devuelve directamente el código del fallo detectado. Para interpretar este resultado basta con revisar la tabla de codificación de los tipos de errores.

Fallo	Tipo	Descripción
F1	Fallo del inversor	Error completo en uno de los seis IGBT
F2	Error del sensor de retroalimentación	Fallo del sensor monofásico 20%
F3	Anomalía en la red eléctrica	Caídas intermitentes de voltaje
F4	Desajuste de matriz fotovoltaica	10 a 20% de sombreado parcial no homogéneo
F5	Desajuste de matriz fotovoltaica	15% de circuito abierto en matriz fotovoltaica
F6	Error del controlador MPPT/IPPT	-20% de parámetro de ganancia del controlador PI en el controlador MPPT/IPPT del convertidor elevador
F7	Error del controlador del convertidor Boost	+20% en la constante de tiempo del controlador PI en el controlador MPPT / IPPT del convertidor elevador

*Tabla 8 - Códigos de fallo y su descripción.*

Una vez creado el modelo, la herramienta ClassificationLearner nos proporciona una medida de precisión, la cual ha sido utilizada para realizar la comparación entre los modelos. Para asegurar que esta precisión es correcta se ha realizado un test en el modelo BaggedTrees, que fue el que obtuvo una mejor puntuación al entrenarlo con todo tipo de fallos y modos de funcionamiento.

Este test consiste en realizar mil iteraciones, en cada una de ellas se elige una medida al azar entre toda la base de datos y se compara el estado real de la medida con el predicho por el modelo. Tras realizar la prueba esta nos arroja una precisión del 92%, 0.2% menos que lo que nos mostraba ClassificationLearner. Este resultado no es sorprendente ya que en el modelo nos asegurábamos de que hubiera la misma cantidad de medidas de todos los tipos de fallos y ahora, sin embargo, la toma de medidas es completamente aleatoria. Esta posible disparidad en el número de medidas de cada tipo de fallo puede provocar una diferencia en la precisión ya que, como se ha podido comprobar en el estudio del comportamiento de los modelos según el tipo de fallo, unos tipos de fallo son más complejos de predecir que otros, obteniendo una precisión menor.

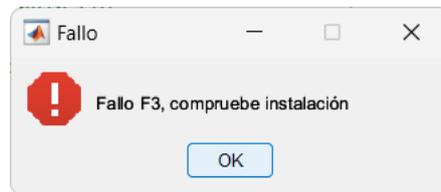
### 10.1.2 Priorización de los errores detectados

Si bien es cierto que todos los fallos aquí presentados son importantes, unos requieren especial atención debido a su naturaleza. Si en nuestro sistema fotovoltaico detectamos un fallo tipo 4, causado por sombreado parcial, no habría problema si no se realiza ninguna acción de

forma inmediata ya que no causaría daños de forma instantánea. Por otro lado, si detectamos un fallo en los IGBT (fallo tipo 1) o caídas intermitentes en el voltaje (fallo tipo 3), es imperativo que se solucione a la mayor brevedad posible, ya que de lo contrario el sistema fotovoltaico podría llegar a quedar inutilizado por completo.

### 10.1.3 Notificación y registro de los errores

Independientemente de si se toman acciones preventivas de forma inmediata o no, es importante que se notifique a los responsables del sistema fotovoltaico de que se ha detectado un fallo en el sistema. Este aviso puede ser crucial ya que, aunque el fallo que haya predicho el modelo no requiera atención inmediata, la clasificación del fallo puede ser errónea y en realidad tratarse de un fallo que sí que requiera una acción de contingencia.



*Figura 29 - Ejemplo de notificación de error.*

Además, sería recomendable que el sistema guardara un registro de los fallos detectados para poder realizar un análisis exhaustivo en caso de que un tipo de fallo ocurra de forma recurrente. Esta recurrencia puede indicar que, aunque el fallo se repare cuando se notifica, no se está solucionando la causa principal del mismo.

## 10.2 Planteamiento del sistema de desconexión

### 10.2.1 Descripción

Como se ha mencionado anteriormente, cada fallo puede requerir de una acción diferente a otro tipo. Para representar la propuesta de plan de acción que se ha diseñado de una forma visual, se ha creado una simulación simplificada que ilustra cómo funcionaría un dispositivo de desconexión basado en este mantenimiento predictivo.

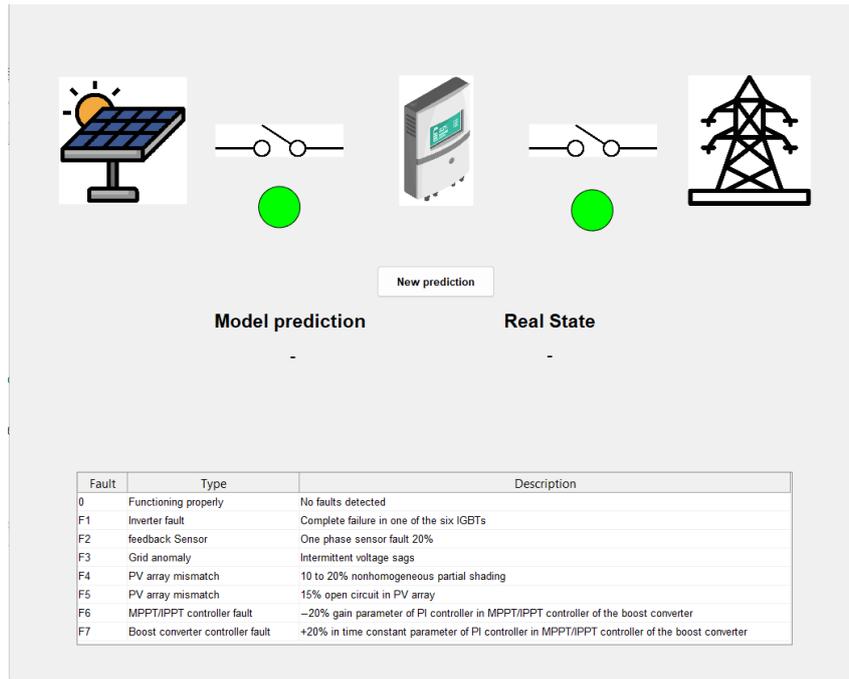


Figura 30 – Graphics User Interface (GUI) del sistema de desconexión.

Esta simulación es un GUI en la que podemos ver un pequeño diagrama de lo que sería una instalación en la que aplicar este sistema de detección de fallos: empezando por los paneles solares, que están conectados al inversor a través de un de switch (relé, contactor, etc, dependiendo de la potencia de la instalación) y por último el inversor conectado a la red eléctrica también con un dispositivo de corte de tensión alterna. La interfaz tiene en el centro un botón que nos permite generar una nueva predicción, la cual una vez procesada se mostrarán los valores predichos y los reales debajo de los textos correspondientes. Para que sea más claro qué fallos ha ocurrido se ha añadido una leyenda en la parte inferior, en la que se indica el código del fallo, el tipo y una descripción de este.

El modo en el que esta simulación funciona es que, al darle al botón de nueva predicción, se carga la base de datos y se elige de forma aleatoria una medición. Esta medición es procesada por el modelo de clasificación “Bagged Trees”, que es el que mejor desempeño tuvo realizados en otros apartados. Una vez ha sido procesada se muestra a través de texto cual ha sido el código de fallo que ha habido y cuál ha sido el predicho por el modelo. Si éstos coinciden o si difieren, se muestra a través de un texto debajo de los códigos de fallo.

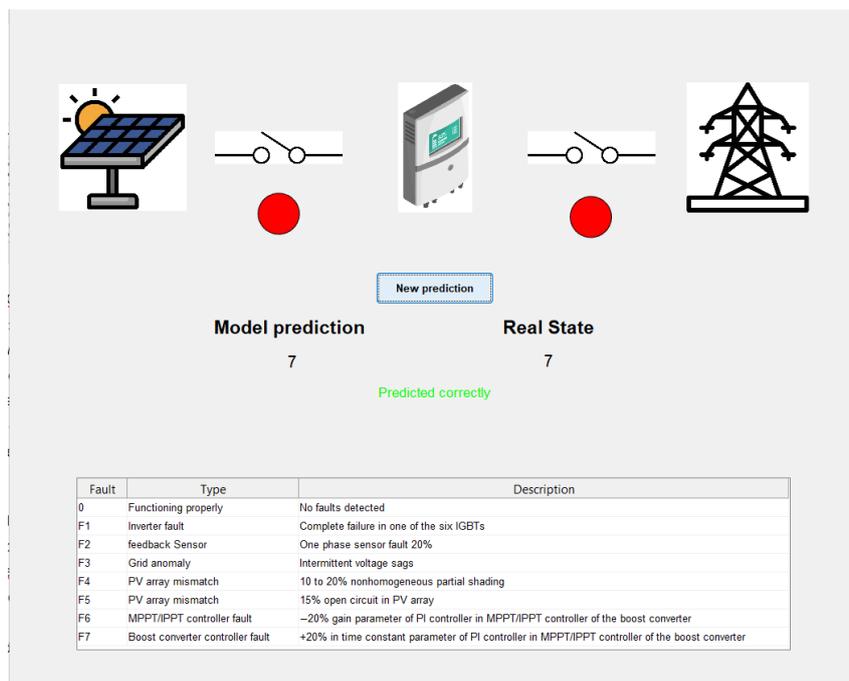


Figura 31 - Ejemplo del GUI del sistema mostrando su funcionamiento al notificar el error con código 7.

Adicionalmente, se han añadido unos “LEDs” (Light Emitting Diodes) debajo del dibujo de los switches que indicarían la acción de cada switch según el fallo predicho por el modelo. Si está en verde indica cortocircuito y si está en rojo el switch se accionaría y sería un circuito abierto, dejando aislada la instalación.

## 10.2.2 Protocolo ante fallos

Cada uno de los tipos de fallos tiene una naturaleza distinta, pueden estar causados por factores totalmente distintos y por ende ser solucionados de forma distinta. Es por esto por lo que cada uno debe de contar con su propio protocolo de actuación una vez han sido detectados. Aquí se detalla cómo se debe actuar cuando se detecta cada uno de los fallos y si los switches añadidos deben de estar en cortocircuito o en contacto abierto.

- **F0**: este código de funcionamiento indica un correcto funcionamiento, por lo que no sería necesaria ninguna acción. Ambos switches estarían en circuito cerrado y los “LEDs” en verde.
- **F1**: cuando se detecta un fallo F1 indica un fallo en alguno de los IGBT del inversor, por lo que para evitar que dañe los módulos fotovoltaicos o que el problema empeore al recibir

corriente de la red eléctrica ambos switches deben estar en circuito abierto, en el GUI se indicarán en color rojo.

- F2: se trata de un fallo en el sensor de realimentación, puede estar causado por diversos factores, desde un defecto en el propio sensor hasta una mala calibración. Este sensor es crucial en un sistema fotovoltaico, ya que se encarga de asegurar un funcionamiento seguro del sistema. Si no funciona correctamente puede resultar en un mal funcionamiento del sistema fotovoltaico o una incapacidad para controlar y regular adecuadamente la corriente o voltaje. Debido a su importancia, una vez detectado se debe aislar completamente el inversor, al igual que en el código de fallo anterior, dejando en circuito abierto ambos switches.
- F3: cuando se notifica este código es debido a una anomalía en la red eléctrica. Estas anomalías suelen ser transitorias, causadas generalmente por sobretensiones, variaciones en la frecuencia o micro cortes. Para prevenir problemas en el sistema es conveniente aislar de la red, por lo que el switch que la une al inversor debe estar en circuito abierto, el inversor que conecta el inversor a los módulos fotovoltaicos puede seguir en cortocircuito (representado en color verde en el GUI), proporcionando corriente a la instalación a la que esté conectado (una casa, fábrica, etc). Al tratarse de un fallo transitorio se podría programar que el switch volviera a cerrarse (cortocircuito) pasado un tiempo, reduciendo así las pérdidas de energía que se vierte a la red y evitando que un operario lo accione manualmente.
- F4: este fallo nos indica que hay una descompensación causada por sombreado parcial. A pesar de ser una de las principales causas de puntos calientes, lo que deteriora de forma considerable la eficiencia del sistema, no es un problema que requiera una desconexión inmediata. Un protocolo adecuado sería avisar al operario para que revise y realice el mantenimiento oportuno al módulo solar, pero sin aislar nada. Sin aislar el sistema conseguimos mantener la producción de energía, pero es importante que este fallo no perdure excesivamente en el tiempo.
- F5: un fallo por circuito abierto puede ser fatal, ya no solo para la eficiencia y vida útil del sistema fotovoltaico, sino también para la seguridad de los operarios que revisan y mantienen todo el sistema. Para mantener la seguridad e integridad de toda la instalación es importante aislar los elementos relativos a la generación de energía solar todo lo posible, por lo que desconectaríamos el switch que aísla los módulos del inversor.
- F6: a pesar de que un fallo en el controlador del inversor puede ser momentáneo y no causar ningún problema no sabemos con certeza cómo esto puede repercutir tanto en los módulos como en la inyección de energía en la red. Para evitar fallos más graves lo más seguro es aislar el controlador del resto de la instalación, dejando en circuito abierto ambos switches.

- F7: se trata de un fallo en otro componente del inversor, el convertidor elevador, ocurre cuando le llega una señal incorrecta el controlador. Esta señal incorrecta puede ser causada por una ganancia descompensada, un tiempo de retardo excesivo o por simples perturbaciones en la señal. Al igual que con el código de fallo “F6”, por motivos de seguridad, es conveniente aislar el inversor, ya que una señal errónea puede provocar fallos adicionales y dañar el resto de los equipos. Ambos switches estarán en circuito abierto.

Cuando por seguridad se aísla alguno de los elementos del sistema, éstos no serán rearmados hasta que un operario o persona autorizada y con conocimientos suficientes haya comprobado toda la instalación y se haya cerciorado de que su funcionamiento es correcto.

Es posible hacer que el rearme se ejecute de forma automática transcurrido un tiempo o, a criterio de otros factores que el usuario considere convenientes (nivel de irradiancia, rearme de forma remota, etc). Aunque, no se podrá asegurar un correcto funcionamiento del sistema hasta que sea revisado de forma concienzuda.

### 10.3 Problemas al tener el sistema en circuito abierto

Diversos protocolos a seguir cuando se detectan fallos en el sistema consisten en aislar los módulos solares, dejándolos en circuito abierto. Esta configuración acarrea un problema, y es que, aunque el sistema esté sin carga los módulos siguen estando en condiciones de irradiancia, los circuitos quedan abiertos y la energía no tiene adónde ir.

Al igual que una batería que no está conectada a nada, un panel solar que está desenchufado funcionará normalmente. Sin embargo, dado que está en circuito abierto, no fluirá corriente. El resultado puede ser un sistema sobrecargado y paneles dañados. Esta energía que no se consume en ninguna carga se transforma en calor, incrementando la temperatura del panel solar, pudiendo provocar deterioro en el mismo, lo que resultaría en una reducción de la eficiencia. Es por este deterioro por lo que los módulos fotovoltaicos no deberían estar en circuito abierto demasiado tiempo.

A pesar de no ser una solución final es una acción importante por realizar cuando se detecta un fallo grave, ya que, a pesar de tener la posibilidad de degradar los paneles, evitamos daños mayores, como posibles incendios o cortocircuitos, dependiendo del fallo detectado.

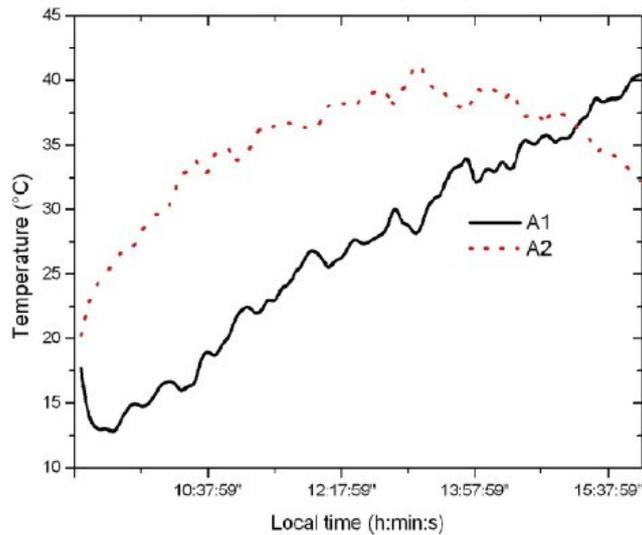


Figura 32 - Temperatura de los paneles solares en el experimento. A1 es el módulo bajo carga y A2 está en circuito abierto.

Según el artículo (Boussaid, 2016) la degradación, en potencia, de un módulo fotovoltaico en circuito abierto dejado en condiciones de irradiancia durante 100 días es del 11.8% y de 22.8% tras 300 días, mientras que el mismo módulo con carga tuvo una degradación de 7.8% en 100 días y de 18.9% en 300 días. La diferencia en la degradación supone una diferencia de prácticamente un 4% por año.

Esta degradación se descompone en una degradación de la corriente de cortocircuito del 17.8% en el panel con carga y del 20.9% en el panel en circuito abierto (tras 300 días). Una degradación del voltaje en circuito abierto del 6% y del 6.5% respectivamente, tras 300 días.

Por otro lado, el factor de forma se vio incrementado en un 4.9% al estar en cortocircuito y un 4.4% en circuito abierto tras 300 días. Este incremento en el factor de llenado quiere decir que la eficiencia en la conversión de energía ha incrementado tras este tiempo, compensando ligeramente la degradación general.

## 10.4 Elementos de desconexión

Existen varios tipos de switches que se pueden utilizar para aislar los diferentes elementos en un sistema fotovoltaico. Estos switches son componentes importantes ya que nos permiten realizar una desconexión segura y controlada entre los paneles solares y el inversor o el inversor y la red eléctrica. Algunos de los switches más comunes en los sistemas fotovoltaicos:

- Interruptor diferencial: también conocidos como disyuntores, son dispositivos de protección que se utilizan para abrir o cerrar un circuito eléctrico en respuesta a una sobrecorriente o un cortocircuito. A pesar de estar diseñados para accionarse por condiciones anómalas, nos permiten aislar el sistema de forma manual utilizando una pequeña palanca.



*Figura 33 - Interruptor diferencial.*

- Portafusibles: dispositivos encargados de proteger un fusible. Los fusibles están diseñados para proteger el circuito cuando la potencia se eleva por encima de la que éstos soportan. Adicionalmente, existen modelos de portafusibles que incorporan la función de seccionador, lo que nos permite asegurar que el paso de corriente por el sistema se ha cortado.



*Figura 34 – Portafusible.*

- Interruptores de desconexión de CC: Estos interruptores se utilizan para desconectar específicamente la corriente continua que fluye entre los paneles solares y el inversor. Estos interruptores son importantes porque permiten una desconexión segura de la corriente generada por los paneles solares antes de que llegue al inversor. Esto es especialmente relevante durante las tareas de mantenimiento, reparación o en caso de emergencia. Adicionalmente, existe un tipo de interruptor de CC que se pueden controlar

mediante una señal de corriente. Estos interruptores suelen utilizarse en sistemas fotovoltaicos para desconectar la corriente que fluye entre los paneles solares y el inversor. Al recibir una señal de control, el interruptor se activa o desactiva, permitiendo o interrumpiendo el flujo de corriente entre los paneles solares y el inversor. Esto ofrece la posibilidad de controlar de forma remota la desconexión de los paneles solares en función de las condiciones del sistema.



*Figura 35 - Interruptor de desconexión de CC.*

- Relés: son dispositivos electromagnéticos que permiten abrir o cerrar un circuito eléctrico mediante una señal de control. En un sistema fotovoltaico se pueden utilizar relés para interrumpir la conexión eléctrica entre los paneles solares y el inversor. Existen relés diseñados específicamente para aplicaciones de mayor potencia, como un sistema fotovoltaico, conocidos como relés de potencia. Un ejemplo de estos relés puede ser el G7L-X de Omron.



*Figura 36 – Relé.*

Estos relés tienen una corriente y tensión de corte más elevadas que los relés comunes. Al recibir una señal de control, el relé se activa y abre el circuito, desconectando así los módulos solares del inversor.

La utilización de relés para desconectar los módulos solares del inversor ofrece algunas ventajas., como que debido a su amplio uso y su simplicidad el stock es mayor. Además, los relés son fiables, duraderos.

- Contactores: un contactor es un tipo de interruptor electromagnético que se utiliza para controlar y abrir o cerrar circuitos eléctricos de alta potencia. Son capaces de manejar corrientes y tensiones elevadas lo que, en este caso, los hace idóneos para la desconexión de los elementos del sistema fotovoltaico.



*Figura 37 – Contactor.*

Los contactores tienen como ventaja su robustez, fiabilidad y seguridad. Además, los contactores suelen ser accionados por una señal de control, lo que permite su accionamiento de forma remota o automatizada.

- Interruptores magnetotérmicos (CA): son elementos de corte de corriente alterna que se accionan cuando ocurre sobrecarga en el sistema. Estos interruptores son importantes ya que permiten aislar el inversor de la red eléctrica. Esto es útil en situaciones de mantenimiento o en caso de fallos en el sistema, evitando que la energía generada por los paneles solares se inyecte en la red eléctrica cuando no se desea. Algunos interruptores de CA también pueden ser controlados mediante una señal de control. Esto permite un control remoto o automatizado de la desconexión de la energía generada por el sistema fotovoltaico y su inyección a la red eléctrica.



*Figura 38 - Interruptor magnetotérmico.*

Los elementos de desconexión que son capaces de accionarse mediante una señal externa nos permiten accionarlos de forma remota o incluso automatizar el proceso utilizando el modelo de predicción elegido. Esta automatización nos asegura que se tomarán medidas de forma instantánea cuando se detecte un fallo.

Para incrementar la seguridad de nuestro sistema, sobre todo cuando se efectúan labores de mantenimiento, se ha diseñado un procedimiento llamado LOTO (Lock-Out-tag-Out) que permite bloquear estos dispositivos para evitar accidentes.

#### 10.4.1 LOTO

LOTO es un sistema de bloqueo y etiquetado que se utiliza para evitar el arranque o la activación accidental de equipos durante trabajos de mantenimiento o reparación. Este procedimiento requiere el uso de dispositivos de desconexión y bloqueo adecuados, como interruptores, relés o sistemas de desconexión seguros, que permitan a los trabajadores aislar de manera efectiva los equipos y evitar su puesta en marcha mientras se realizan las tareas requeridas.



*Figura 39 - Candado LOTO siendo utilizado en u interruptor industrial.*

En España, la normativa específica que establece las disposiciones mínimas de seguridad y salud para la utilización por los trabajadores de los equipos de trabajo es el Real Decreto 1215/1997, de 18 de julio. Este Real Decreto establece las medidas necesarias durante la

utilización de los equipos de trabajo, incluyendo los dispositivos de desconexión y bloqueo. Si bien este Real Decreto no especifica directamente los requisitos técnicos para los dispositivos de desconexión y bloqueo en relación con LOTO, establece la obligación general de proporcionar los medios necesarios para realizar el procedimiento LOTO de forma efectiva y segura. Esto implica que los switches utilizados en sistemas fotovoltaicos deben ser seleccionados y configurados de acuerdo con las mejores prácticas de seguridad y para garantizar la desconexión segura de los equipos y evitar el reinicio accidental del sistema.

Es importante destacar que además del Real Decreto 1215/1997, existen otras normativas y regulaciones relacionadas con la seguridad y la prevención de riesgos laborales que deben tenerse en cuenta al seleccionar los dispositivos y establecer los procedimientos de seguridad en el contexto de la utilización de switches en sistemas fotovoltaicos.

### 10.4.2 La importancia de los elementos de desconexión

A pesar de tratarse de dispositivos aparentemente simple resulta de vital importancia en cualquier tipo de instalación eléctrica. En este caso, esta importancia se debe a que proporcionan una forma segura de desconectar los paneles solares del inversor, evitando el riesgo de descargas eléctricas y garantizando un entorno de trabajo seguro durante el mantenimiento, reparación o en situaciones de emergencia.

Por otro lado, nos permiten desconectar los paneles solares en caso de sobrecarga, cortocircuito u otras condiciones anormales que podrían dañar los paneles solares o el inversor. Esto ayuda a mantener la eficiencia y prolongar la vida útil de los componentes del sistema fotovoltaico, evitando posibles averías.

Al utilizar switches para desconectar los paneles solares, se simplifica y agiliza el proceso de mantenimiento. En lugar de tener que acceder a los cables individualmente los switches nos permiten una desconexión rápida y segura sin necesidad de manipular las conexiones eléctricas principales. Esto ahorra tiempo y esfuerzo en las tareas de inspección, limpieza o reemplazo de componentes.

### 10.4.3 Criterio para elegir elementos de desconexión

Es importante seleccionar switches de calidad y adecuados para garantizar un funcionamiento óptimo y seguro del sistema fotovoltaico. Deben ser elegidos en consonancia con las características eléctricas del resto del sistema.

Corriente soportada: deben tener una capacidad de corriente adecuada para soportar la corriente máxima que fluye entre los paneles solares y el inversor. Es esencial verificar las especificaciones de corriente nominal y corriente de corte del switch para asegurarse de que son capaces de soportar tanto la corriente que circulará normalmente por ellos, como la corriente que deben cortar al ser accionados.

Tensión de operación: deben ser capaces de manejar la tensión nominal del sistema fotovoltaico.

Tipo de interruptor: otra característica que debemos observar al seleccionar los switches adecuados es el tipo de corriente para la que está diseñado. No funcionan de la misma manera los interruptores para continua como los de alterna.

Certificaciones y normas de seguridad: al igual que el resto de los elementos, los switches están obligados a cumplir con las normas y regulaciones de seguridad eléctrica aplicables a los sistemas fotovoltaicos.

Fiabilidad y durabilidad: los switches deben ser fiables y capaces de soportar las condiciones ambientales en las que se instalarán. En muchas ocasiones estos elementos estarán a la intemperie, por lo que es importante que cumplan algún estándar de impermeabilidad, como la clasificación IP (Ingress Protection).

Es buena praxis sobredimensionar las características eléctricas de los elementos de corte de nuestro sistema. Esto quiere decir que en una instalación con una tensión nominal de 100V buscaremos elementos de corte de entre 150V e incluso 200V.

Una vez vistos estos criterios los elementos más recomendados son los contactores. Bien es cierto que si trabajamos con un valor de tensión bajo podríamos utilizar un relé y sería capaz de realizar la función que buscamos, pero esto nos limitaría mucho en caso de querer ampliar el número de módulos solares en serie ya que incrementaría la tensión y habría que cambiar los relés. Otra ventaja que nos brindan los contactores con vista a futuro es que normalmente cuentan con más de un contacto por lo que si decidiéramos añadir baterías a nuestro sistema podríamos aprovechar un contacto con más de un contacto para cortar el paso de corriente a las baterías.

En definitiva, un contactor nos brinda seguridad gracias a su robustez. Nos asegura una vida útil de larga duración, ya que está diseñado para realizar un alto número de conmutaciones. Y en caso de ampliar nuestro sistema nos seguiría sirviendo este elemento, siempre y cuando no

superemos sus valores nominales, pero por lo general están sobredimensionados. A estas ventajas habría que añadir su precio, ya que son más económicos que los interruptores de desconexión de CC.

#### 10.4.4 Regulación vigente sobre los elementos de desconexión

En España, los switches utilizados en sistemas fotovoltaicos deben cumplir con la normativa y las regulaciones existentes para garantizar la integridad del sistema y la seguridad de los trabajadores mientras lo manipulan. Algunas de las normativas aplicables hoy en día en España:

- Reglamento Electrotécnico de Baja Tensión (REBT): establece las condiciones técnicas y de seguridad que deben cumplir las instalaciones eléctricas de baja tensión en España. Aunque no sea específico para las instalaciones fotovoltaicas, todo el sistema, incluidos los switches, deben cumplir con los criterios de seguridad y calidad establecidos por el REBT.
- Norma UNE<sup>4</sup>-EN<sup>5</sup> 62477-1: esta norma establece los Requisitos de seguridad para sistemas y equipos de conversión de potencia de semiconductores. Esta norma afecta sobre todo a los inversores en las instalaciones fotovoltaicas. La norma incluye los requisitos mínimos que tienen que tener los equipos. Algunos de estos requisitos indican la protección contra descargas eléctricas que deben tener, el aislamiento adecuado para este tipo de elementos y las protecciones con las que deben contar incendios, sobretensiones y sobrecalentamiento.
- Norma UNE-EN 60947-3: esta norma es específica para la “aparata de baja tensión”, como interruptores, seccionadores y fusibles, siempre y cuando no se supere 1kV CA o 1,5kV CC. En el anexo D habla específicamente de los elementos de corte utilizados en sistemas fotovoltaicos. Estipula los criterios que deben de cumplir en cuando a capacidad de corriente, tensión de corte y demás características eléctricas.

Estas son solo algunas de las normativas y estándares aplicables en España para los switches utilizados en sistemas fotovoltaicos. Es importante consultar la legislación y normativa actualizada, ya que pueden actualizar la regulación, anulando las versiones previas como pasa con la norma UNE-EN 50548, que regulaba las cajas de conexiones de los módulos fotovoltaicos. No

---

<sup>4</sup> Una Norma Española.

<sup>5</sup> Norma Europea

solo es importante revisar la normativa nacional, además hay que conocer la normativa de cada comunidad autónoma, ya que puede variar entre una comunidad u otra.

## 11. Discusión

El primer experimento, en el que comparábamos la precisión de todos los modelos habiendo sido entrenados sin hacer ninguna distinción entre el modo de potencia o el tipo de fallo de los datos ha servido para comprobar cómo se comportarían en un escenario realista, en el que puede ocurrir cualquier fallo. Este experimento mostró que algunas familias de modelos tienen una precisión media muy alta, como las redes neuronales. Esta superioridad en la precisión puede ser debida a la forma en que estos modelos trabajan. Están formados por nodos, también llamados neuronas, como un cerebro humano. Por otro lado, "Bagged Trees" logró la precisión más alta, seguido de "Wide Neural Network". Ambos modelos serían la mejor opción para un sistema que quiere predecir y clasificar todos los tipos de fallos, independientemente de su modo de potencia.

Además, hemos descubierto que hay una diferencia en la precisión si pronosticamos el estado de un sistema en MPPT o si está en LPPT. El documento descubrió que su sistema de predicción perdía precisión cuando la irradiancia era menor. Esta disminución en la irradiancia causó que la corriente cayera, lo que también podría ser la razón de esta diferencia de precisión entre los métodos bajo diferentes tipos de datos. En nuestro caso, la diferencia de corriente podría ser causada por cómo funciona el MPPT, ajustando la impedancia de salida para que coincida con la carga y obtenga el mejor Modo de Potencia, dando la corriente máxima. Además, el modelo más preciso en cada caso no es el mismo. Si queremos pronosticar fallos en condiciones MPPT, el mejor modelo es usar una "Red neuronal media", mientras que en condiciones LPPT, la mejor opción es el modelo "Árboles embolsados". Esta mayor precisión en "Árboles embolsados" y el aumento en la precisión de "SVM cúbico" nos permiten pensar que cuando se trata de LPPT, los datos de un panel de trabajo adecuado y uno defectuoso tienen una distinción más aparente que en MPPT. Esta condición abre una ventana de posibilidades. Sin pruebas, la mejor manera de proceder en un sistema fotovoltaico que utiliza ambos modos de potencia puede ser desarrollar un algoritmo capaz de cambiar los modelos antes de pronosticar, dependiendo de su modo de potencia. Este algoritmo nos daría el mejor rendimiento en cada caso, aumentando la precisión media. (Spataru, Diagnostic method for photovoltaic systems based on light, 2015)

En cuanto al análisis del rendimiento de cada modelo cuando se entrena utilizando los mismos datos de tipo de fallo, hemos descubierto que los modelos no funcionan de la misma manera en diferentes errores. En promedio, como vimos en el primer experimento, las "redes

neuronales" y los "árboles embolsados" son el modelo dominante en cada fallo, pero en algunos de ellos "SVM" y "KNN" lograron el mayor rendimiento. Esta diferencia significa que, en estos fallos, el parámetro eléctrico de un módulo defectuoso y uno que funciona bien están más separados en grupos ("Bueno" y "Fallo"). Esta información es útil en caso de un fallo predominante en nuestro sistema fotovoltaico, como tener siempre sombreado parcial o problemas con la red. En estos casos, en lugar de utilizar un modelo de propósito general, podemos seleccionar el que mejor funcione para nuestro caso, conociendo el fallo común en nuestro sistema fotovoltaico. Además, esta diferencia en la capacidad de predicción de los modelos en función del tipo de fallo nos permite pensar que un sistema de predicción sólido podría basarse en varios modelos de predicción concurrentes y proporcionar una respuesta ponderada a los mismos, en función de su rendimiento en cada tipo de fallo. Esto requerirá más estudio.

En lo relativo al sistema de desconexión, se han elegido los componentes idóneos para el sistema, teniendo en cuenta las características técnicas de los elementos de desconexión, siempre buscando la fiabilidad y la posibilidad de accionarlos con una señal de disparo. Para controlar estos elementos se ha simulado un GUI con el que se representa gráficamente el funcionamiento del mismo y se ha comprobado que el modelo es capaz de predecir correctamente los fallos a partir de las medidas proporcionadas.

## 12. Conclusión

En conclusión, el mantenimiento predictivo es una técnica útil capaz de detectar prematuramente un fallo en un sistema fotovoltaico. Esta detección temprana evita la degradación del sistema, prolongando su vida útil y aumentando su eficiencia. A pesar de que puede requerir una alta inversión para desarrollarse, sus ventajas se traducirán en un mantenimiento más barato.

En este trabajo, hemos podido entrenar y probar con éxito varios modelos predictivos utilizando una base de datos previamente existente. Comparándolos, "Bagged Trees" fue el más preciso, pero no muy lejos de la mayoría de los modelos de red neuronal. Debido a esta estrecha brecha, no podemos decir que uno es mejor que el otro, puede depender del modo de utilización o de las características del sistema fotovoltaico.

Con el desarrollo de nuevas tecnologías y técnicas computacionales complejas, una red neuronal será la tendencia a seguir cuando se quieran predecir todo tipo de fallos. Este tipo de modelos son capaces de lograr precisiones muy altas independientemente del tipo de fallo. Otros modelos funcionan mejor pero únicamente al predecir fallos concretos.

El aprendizaje automático, y las IAs en general, es un área que está creciendo exponencialmente. Este crecimiento puede cambiar los modelos y técnicas usados, permitiéndonos crear otros modelos o incluso lograr un mejor rendimiento utilizando los actuales.

Por otro lado, se ha estudiado los distintos tipos de elementos de desconexión. Se ha comprobado que hay una amplia variedad de componentes que nos pueden resultar más o menos útiles en función de las necesidades del sistema.

A pesar de no haber implementado el sistema de desconexión de forma tangible, con el GIU se ha podido observar la capacidad de predicción del sistema y cómo se comportaría este en los diferentes escenarios que se le pueden plantear.

Es importante destacar que este sistema está pensado para generación de energía solar con baja tensión. En grandes plantas solares se utilizan otras técnicas y elementos para la desconexión. Una de las técnicas utilizadas es la conocida carga suave, que incrementa la tensión de los contactos que no están energizados hasta niveles similares a la salida de los módulos fotovoltaicos. Esta equiparación de tensiones evita que se creen arcos que generaría picos de corriente y tensión capaces de dañar nuestro sistema y probablemente dar falsas alarma de fallo.

Una mejora que se podría aplicar a este proyecto sería implementar un sensor de irradiancia. Añadiendo un sensor de irradiancia podríamos modificar el dispositivo para que no realice predicciones cuando la irradiancia no llegue al umbral establecido, evitando así resultados erróneos. Esto se basa en que el artículo (Sergiu Spataru, 2015) descubrió que con una irradiancia menor de  $500\text{W}/\text{m}^2$  el modelo de predicción era menos preciso. Los fallos causados por sombreado parcial y por un incremento en la resistencia serie son mucho más claros en los parámetros I-V con una alta irradiancia.

Otra de las mejoras que se podría hacer es entrenar el modelo con una cantidad de datos mayor. En este proyecto no se ha realizado por no contar con la potencia necesaria en el dispositivo que genera las predicciones. En el presupuesto se ha estimado el coste de un ordenador de alta potencia con el que sí que se podría entrenar el modelo con una mayor cantidad de mediciones.

### 13. Tabla de siglas

<b>Siglas</b>	<b>Significado en español</b>	<b>Significado en inglés</b>
<b>MPPT</b>		Maximum Power Point Tracking
<b>LPPT</b>		Limited Power Point Tracking
<b>CNN</b>		Convolutional Neural Network
<b>SVM</b>		Support Vector Machine
<b>CC &amp; DC</b>	Corriente Continua	Direct Current
<b>CA &amp; AC</b>	Corriente Alterna	Alternate Current
<b>IA &amp; AI</b>	Inteligencia Artificial	Artificial Intelligente
<b>IGBT</b>		Insulated Gate Bipolar Transistor
<b>KNN</b>		K Nearest Neighbour
<b>NN</b>		Neural Networko
<b>TPR</b>		True Positive Ratio
<b>FNR</b>		Faulse Positive Ration
<b>GUI</b>		Graphics User Interface
<b>LED</b>		Light Emitthing Diode
<b>LOTO</b>		Lock Out Tag Out
<b>IP</b>		Ingress Protection
<b>REBT</b>	Reglamento Electrotécnico de Baja Tensión	
<b>UNE</b>	Una Norma Española	
<b>EN</b>	Norma Europea	
<b>PCB</b>		Printed Circuit Board
<b>ODS</b>	Objetivos Desarrollo Sostenible	
<b>PI</b>	Proporcional Integral	
<b>EVA</b>	Etileno-Vinil-Acetato	
<b>(I-V)</b>	Tensión y corriente	



## II Pliego de condiciones

Para poder realizar un presupuesto del coste de implementar el sistema de desconexión automática en una instalación fotovoltaica se va a realizar el pliego de condiciones, pudiendo evaluar así el impacto económico del sistema a pesar de no ser un caso real.

### 1. Descripción de la instalación

Se quiere implementar el sistema de desconexión automática basado en machine learning en una instalación fotovoltaica de una vivienda. Esta instalación tiene una tensión de 24V y una corriente de 50A.

Este sistema de desconexión incluirá un elemento de desconexión entre los módulos fotovoltaicos y el inversor y otro elemento entre el inversor y la red eléctrica. Estos elementos deben de ser capaces de accionarse mediante una señal de corriente. Adicionalmente, se incluirá un ordenador que será el encargado de realizar el machine learning y de generar la señal para controlar los elementos de desconexión.

Teniendo en cuenta la regulación vigente, en caso de no poder escoger componentes que nos permita accionarlo a través de una señal de corriente y a su vez esté diseñado para LOTO, debemos añadir un switch adicional que sí esté capacitado para este procedimiento. Esto nos permitirá conservar la posibilidad de aislar el circuito de forma remota a la vez que se cuentan con elementos que aseguran la seguridad de los trabajadores mientras manipulan el sistema.

### 2. Condiciones de los materiales

Los elementos de desconexión elegidos para el sistema son unos contactores, tanto para aislar los módulos solares del inversor como el inversor de la red eléctrica. El motivo de esta elección es su robustez, fiabilidad y su capacidad para accionarse con un circuito de disparo.

Este tipo de elementos conmutan con una señal de 12-24V, por lo que será necesario añadir una fuente de tensión adicional que controle esa tensión. Para evitar que los posibles fallos tanto de los módulos como de la red eléctrica interfiera en la tensión de referencia con la que se alimentará el circuito de disparo, esta tensión provendrá de unas baterías.

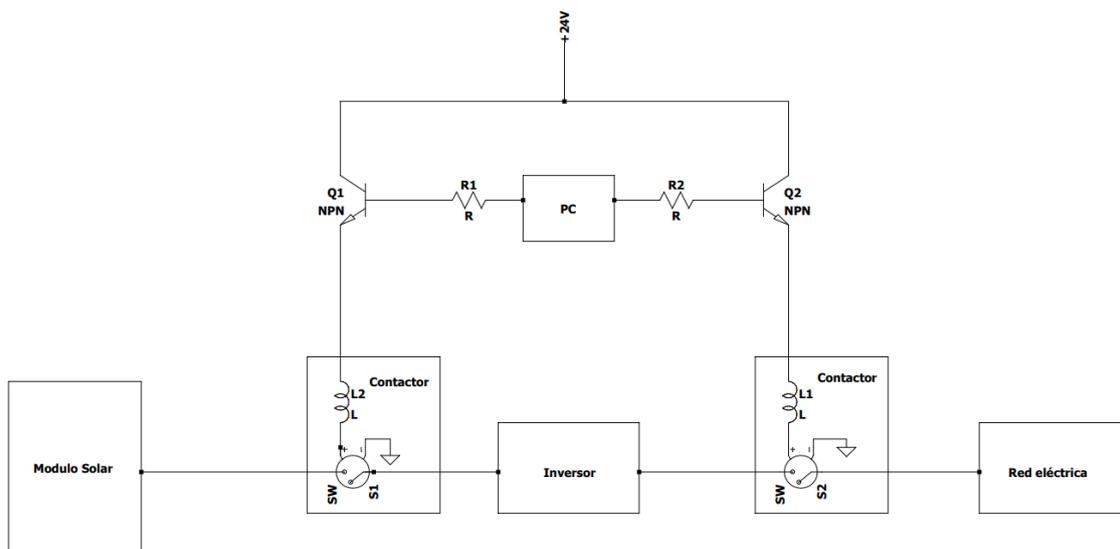


Figura 40 - Diagrama del sistema de desconexión planteado.

Tras realizar una búsqueda de componentes que cumplan las características requeridas se ha decidido comprar:

- Como contactor de DC el KILOVAC LEV200. Soporta tensiones en continua de hasta 900V y 500A, muy sobredimensionado para esta instalación. El circuito de disparo necesario para este componente es de 24V.
- Como contactor de AC hemos optado por Schneider con el Contactor Tesys D. Cuenta con dos contactos normalmente abiertos y otros dos normalmente cerrados, lo que, en caso de requerirlo el cliente, podría modificar la configuración del contactor. Es capaz de controlar 440V y 65A con un circuito de control de 24V.
- Y para las baterías se ha buscado unas de alta capacidad para que en caso de requerirlo se pudieran usar también para almacenar energía. Finalmente se han elegido la batería AGM de 12V y 150Ah de Tensite.

### 3. Pruebas y ajustes finales o de servicio.

Se comprobará la instalación simulando en el ordenador los diferentes tipos de fallos que el sistema es capaz de detectar. Estas simulaciones servirán para cerciorarse de que los elementos de desconexión se accionan de forma correcta.



### III Presupuesto

#### 1. Mano de obra

En este apartado se incluyen los costes relacionados a la mano de obra, tanto del ingeniero, como del electricista que instala el sistema de desconexión.

Se estima que el ingeniero requiere de unas 240h tanto para realizar el estudio de los modelos de predicción, con los test que ello requiere, como para implementar el software necesario en el sistema de desconexión.

Por otro lado, el electricista se estima que requerirá de 40h para implementar el sistema de desconexión en un sistema fotovoltaico ya instalado.

Mano de obra			
Descripción	Nº de horas	€/h	Total
Ingeniero	240	15,38 €	3.691,20 €
Electricista	40	10,59 €	423,60 €
<b>TOTAL</b>			<b>4.114,80 €</b>

Tabla 9 - Presupuesto de la mano de obra.

#### 2. Licencias

Este apartado incluye todo el desglose del coste que representan las licencias requeridas para todo el proyecto, desde MatLab para el análisis, como Excel para el estudio.

De MatLab se ha cogido la licencia de un año llamada "Standard", que permite ser usado en negocios pequeños que no se consideren "Start-up".

Office 365 nos ofrece sus servicios en todas sus plataformas durante un año. Principalmente se utilizará Excel para realizar gráficos y análisis de datos y Word para redactar los informes necesarios.

Licencias			
Descripción	Cantidad	Precio/ud	Total
Matlab	1	860,00 €	860,00 €
Office 365	1	69,00 €	69,00 €
<b>TOTAL</b>			<b>929,00 €</b>

Tabla 10 - Presupuesto de las licencias necesarias.

### 3. Materiales

En este apartado se incluyen todos los gastos en materiales. Teniendo en cuenta la compra de un ordenador, que será el que contenga el modelo de predicción y por tanto realice todas las operaciones necesarias. Además, se incluye el precio de la placa de circuito impreso (PCB) que contendrá los transistores y resistencias del circuito de disparo.

<b>Materiales</b>			
<b>Descripción</b>	<b>Unidades</b>	<b>Precio/ud</b>	<b>Total</b>
Contactador Tesys D	1	455,28 €	455,28€
Contactador LEV200A4ANA	1	104,68 €	104,68 €
Batería AGM 12V	2	174,22 €	348,44 €
PCB	1	50,00 €	50,00 €
Cable necesario	1	40,00 €	40,00 €
Ordenador	1	1.000,00 €	1.000,00 €
<b>TOTAL</b>			<b>1.998,40 €</b>

*Tabla 11 - Presupuesto de los materiales utilizados.*

### 4. Total

Una vez se ha presentado el desglose procedemos a calcular el gasto total. Este total ya incluye todos los impuestos ya que los precios seleccionados en el desglose ya los contienen, son precios al por mayor, sin ningún tipo de descuento.

<b>Mano de obra</b>	4.114,80 €
<b>Licencias</b>	929,00 €
<b>Materiales</b>	1.998,40 €
<b>TOTAL</b>	<b>7.042,20 €</b>

*Tabla 12 - Presupuesto total.*

## Referencias

- (2021, March 27). Retrieved from Liter of light USA: <https://www.literoflightusa.org/solar-panel-diagrams/>
- A. Drews, A. d. (2007, April). *ScienceDirect*. Retrieved from <https://www.sciencedirect.com/science/article/pii/S0038092X06002040>
- Abed, A. &. (2020, July). *Research Gate*. Retrieved from [https://www.researchgate.net/publication/342736081\\_Performance\\_Improvement\\_of\\_CIGS\\_PV\\_Solar\\_Grid\\_Tied\\_System\\_Using\\_Planer\\_Concentrators\\_Case\\_Study\\_Baghda\\_d/figures?lo=1](https://www.researchgate.net/publication/342736081_Performance_Improvement_of_CIGS_PV_Solar_Grid_Tied_System_Using_Planer_Concentrators_Case_Study_Baghda_d/figures?lo=1)
- Bakdi. (2020, Septiembre 22). Retrieved from [https://www.academia.edu/44084296/GPVS\\_Faults\\_Experimental\\_Data\\_for\\_fault\\_scenarios\\_in\\_grid\\_connected\\_PV\\_systems\\_under\\_MPPT\\_and\\_IPPT\\_modes](https://www.academia.edu/44084296/GPVS_Faults_Experimental_Data_for_fault_scenarios_in_grid_connected_PV_systems_under_MPPT_and_IPPT_modes)
- Beny. (2022, April 28). Retrieved from [https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.beny.com%2Fsolar-inverter-installation-guide%2F&psig=AOvVaw1sVaElK03yZ016nkRh1f0Y&ust=1667927622621000&source=images&cd=vfe&ved=0CA4QjhxqFwoTCPC7wt\\_InPsCFQAAAAAdAAAAABA2](https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.beny.com%2Fsolar-inverter-installation-guide%2F&psig=AOvVaw1sVaElK03yZ016nkRh1f0Y&ust=1667927622621000&source=images&cd=vfe&ved=0CA4QjhxqFwoTCPC7wt_InPsCFQAAAAAdAAAAABA2)
- Boussaid, M. &. (2016, Septiembre). *Researchgate*. Retrieved from [https://www.researchgate.net/publication/308891423\\_Solar\\_cell\\_degradation\\_under\\_open\\_circuit\\_condition\\_in\\_out-doors-in\\_desert\\_region](https://www.researchgate.net/publication/308891423_Solar_cell_degradation_under_open_circuit_condition_in_out-doors-in_desert_region)
- Etemadi, Z. Y. (2017, November). *Line-to-Line Fault Detection for Photovoltaic Arrays Based on Multiresolution Signal Decomposition and Two-Stage Support Vector Machine*. Retrieved from IEEEXplorer: <https://ieeexplore.ieee.org/document/7926372>
- F. Aziz, A. U. (2020, February 28). *A Novel Convolutional Neural Network-Based Approach for Fault Classification in Photovoltaic Arrays*. Retrieved from IEEEXplorer: <https://ieeexplore.ieee.org/document/9018018>
- Fadhel, S. D. (2019, February ). *PV shading fault detection and classification based on I-V curve using principal component analysis: Application to isolated PV system*. Retrieved

- from ScienceDirect:  
<https://www.sciencedirect.com/science/article/abs/pii/S0038092X18312234>
- IArtificial.net. (n.d.). Retrieved from <https://www.iartificial.net/maquinas-de-vectores-de-soporte-svm/>
- IBM. (n.d.). *IBM*. Retrieved from <https://www.ibm.com/es-es/topics/decision-trees>
- Intechopen*. (2018, November 20). Retrieved from <https://www.intechopen.com/chapters/68937>
- Laukamp, H. S. (2002). *IEA*. Retrieved from [https://iea-pvps.org/wp-content/uploads/2020/01/rep7\\_08.pdf](https://iea-pvps.org/wp-content/uploads/2020/01/rep7_08.pdf)
- MPPT Solar*. (n.d.). Retrieved from <https://www.mpptsolar.com/es/esquema-funcionamiento-inversor.html>
- Omdena*. (2022, March 21). Retrieved from <https://omdena.com/blog/supervised-and-unsupervised-machine-learning/>
- Our World In Data*. (2022, July 08). Retrieved from <https://ourworldindata.org/grapher/installed-solar-pv-capacity>
- Our World In Data*. (2022, July 08). Retrieved from <https://ourworldindata.org/grapher/modern-renewable-prod>
- Pant, A. (2019, January 11). *Towards Data Science*. Retrieved from <https://towardsdatascience.com/workflow-of-a-machine-learning-project-ec1dba419b94>
- Red neuronal artificial*. (2023, Febrero 15). Retrieved from Wikipedia:  
[https://es.wikipedia.org/wiki/Red\\_neuronal\\_artificial](https://es.wikipedia.org/wiki/Red_neuronal_artificial)
- ScienceDirect*. (2021, February). Retrieved from <https://www.sciencedirect.com/science/article/abs/pii/S0142061520300600>
- Scikit Learn*. (n.d.). Retrieved from [https://scikit-learn.org/stable/modules/cross\\_validation.html](https://scikit-learn.org/stable/modules/cross_validation.html)
- Sergiu Spataru, D. S. (2015, Septiembre). *ScienceDirect*. Retrieved from <https://www.sciencedirect.com/science/article/pii/S0038092X1500328X>

Spataru, S. S. (2015, September). *Diagnostic method for photovoltaic systems based on light*.

Retrieved from ScienceDirect:

<https://www.sciencedirect.com/science/article/abs/pii/S0038092X1500328X>

Spataru, S. S. (2015, July 9). *ScienceDirect* . Retrieved from Diagnostic method for photovoltaic systems based on light I–V measurements:

<https://www.sciencedirect.com/science/article/abs/pii/S0038092X1500328X>

Tabassum, M. &. (2017, May). *Research Gate*. Retrieved from

[https://www.researchgate.net/publication/316644750\\_Design\\_Development\\_of\\_Maximum\\_Power\\_Point\\_Tracker\\_for\\_Solar\\_Module\\_Using\\_Microcontroller/figures](https://www.researchgate.net/publication/316644750_Design_Development_of_Maximum_Power_Point_Tracker_for_Solar_Module_Using_Microcontroller/figures)

Tati, F. &. (2022, March 11). *Intelligent shading fault detection in a PV system with MPPT control using neural network technique*. . Retrieved from ResearchGate:

[https://www.researchgate.net/publication/359180119\\_Intelligent\\_shading\\_fault\\_detection\\_in\\_a\\_PV\\_system\\_with\\_MPPT\\_control\\_using\\_neural\\_network\\_technique](https://www.researchgate.net/publication/359180119_Intelligent_shading_fault_detection_in_a_PV_system_with_MPPT_control_using_neural_network_technique)

*Upgrad*. (n.d.). Retrieved from <https://www.upgrad.com/machine-learning-course/>

Wikipedia. (2020, Marzo 5). Retrieved from

[https://es.wikipedia.org/w/index.php?title=Validaci%C3%B3n\\_cruzada&oldid=124047241](https://es.wikipedia.org/w/index.php?title=Validaci%C3%B3n_cruzada&oldid=124047241)

Wikipedia. (2022, December 23). *Oscilador de puente de Wien*. Retrieved from Wikipedia:

[https://es.wikipedia.org/wiki/Oscilador\\_de\\_puente\\_de\\_Wien](https://es.wikipedia.org/wiki/Oscilador_de_puente_de_Wien)

**ANEXO 1 - RELACIÓN DEL TRABAJO CON LOS OBJETIVOS DE DESARROLLO SOSTENIBLE DE LA AGENDA 2030**

Grado de relación del trabajo con los Objetivos de Desarrollo Sostenible (ODS).

Objetivos de Desarrollo Sostenible	Alto	Medio	Bajo	No Procede
ODS 1. <b>Fin de la pobreza.</b>				X
ODS 2. <b>Hambre cero.</b>				X
ODS 3. <b>Salud y bienestar.</b>		X		
ODS 4. <b>Educación de calidad.</b>				X
ODS 5. <b>Igualdad de género.</b>				X
ODS 6. <b>Agua limpia y saneamiento.</b>				X
ODS 7. <b>Energía asequible y no contaminante.</b>	X			
ODS 8. <b>Trabajo decente y crecimiento económico.</b>			X	
ODS 9. <b>Industria, innovación e infraestructuras.</b>	X			
ODS 10. <b>Reducción de las desigualdades.</b>				X
ODS 11. <b>Ciudades y comunidades sostenibles.</b>	X			
ODS 12. <b>Producción y consumo responsables.</b>				X
ODS 13. <b>Acción por el clima.</b>		X		
ODS 14. <b>Vida submarina.</b>				X
ODS 15. <b>Vida de ecosistemas terrestres.</b>				X
ODS 16. <b>Paz, justicia e instituciones sólidas.</b>				X
ODS 17. <b>Alianzas para lograr objetivos.</b>				X

Un dispositivo capaz de realizar un mantenimiento predictivo en los sistemas fotovoltaicos aumentaría la vida útil de los mismos, haciéndolos más resilientes (ODS 9) y, por lo tanto, garantizando el acceso de la población a la energía solar (ODS 7). Por otro lado, también contribuiría en la participación de la energía solar en el mix energético (ODS 11). Este nuevo mix energético significaría la reducción de otras fuentes de energía, como la quema de carbón entre otros combustibles fósiles que generan grandes cantidades de CO<sub>2</sub> al ser quemados (ODS 13).

## ANEXO 2 - CÓDIGO DE TRATAMIENTO DE LA BASE DE DATOS Y CARGA DE LOS DATOS A CLASSIFICATIONLEARNER

```
% Cargamos toda la base de datos
load('F5L.mat');
load('F3M.mat');
load('F2L.mat');
load('F6M.mat');
load('F7L.mat');
load('F1L.mat');
load('F0M.mat');
load('F4L.mat');
load('F5M.mat');
load('F3L.mat');
load('F2M.mat');
load('F6L.mat');
load('F1M.mat');
load('F7M.mat');
load('F0L.mat');
load('F4M.mat');

% Añadimos una variable que indica el Código de fallo
F7L.Var15(:) = 7;
F6L.Var15(:) = 6;
F5L.Var15(:) = 5;
F4L.Var15(:) = 4;
F3L.Var15(:) = 3;
F2L.Var15(:) = 2;
F1L.Var15(:) = 1;
F0L.Var15(:) = 0;
F0M.Var15(:) = 0;
F1M.Var15(:) = 1;
F2M.Var15(:) = 2;
F3M.Var15(:) = 3;
F4M.Var15(:) = 4;
F5M.Var15(:) = 5;
F6M.Var15(:) = 6;
F7M.Var15(:) = 7;
F7M.Properties.VariableNames{15} = 'State';
F6M.Properties.VariableNames{15} = 'State';
F5M.Properties.VariableNames{15} = 'State';
F4M.Properties.VariableNames{15} = 'State';
F3M.Properties.VariableNames{15} = 'State';
F2M.Properties.VariableNames{15} = 'State';
F1M.Properties.VariableNames{15} = 'State';
F0M.Properties.VariableNames{15} = 'State';
F0L.Properties.VariableNames{15} = 'State';
F1L.Properties.VariableNames{15} = 'State';
F2L.Properties.VariableNames{15} = 'State';
F3L.Properties.VariableNames{15} = 'State';
F4L.Properties.VariableNames{15} = 'State';
F5L.Properties.VariableNames{15} = 'State';
F6L.Properties.VariableNames{15} = 'State';
F7L.Properties.VariableNames{15} = 'State';
```

```

%% Añadimos una variable que indica el modo de potencia en el que trabaja
% MPPT = 1 LPPT = 0
F7L.Var16(:) = 0;
F6L.Var16(:) = 0;
F5L.Var16(:) = 0;
F4L.Var16(:) = 0;
F3L.Var16(:) = 0;
F2L.Var16(:) = 0;
F1L.Var16(:) = 0;
F0L.Var16(:) = 0;
F0M.Var16(:) = 1;
F1M.Var16(:) = 1;
F2M.Var16(:) = 1;
F3M.Var16(:) = 1;
F4M.Var16(:) = 1;
F5M.Var16(:) = 1;
F6M.Var16(:) = 1;
F7M.Var16(:) = 1;
F7M.Properties.VariableNames{16} = 'PM';
F6M.Properties.VariableNames{16} = 'PM';
F5M.Properties.VariableNames{16} = 'PM';
F4M.Properties.VariableNames{16} = 'PM';
F3M.Properties.VariableNames{16} = 'PM';
F2M.Properties.VariableNames{16} = 'PM';
F1M.Properties.VariableNames{16} = 'PM';
F0M.Properties.VariableNames{16} = 'PM';
F0L.Properties.VariableNames{16} = 'PM';
F1L.Properties.VariableNames{16} = 'PM';
F2L.Properties.VariableNames{16} = 'PM';
F3L.Properties.VariableNames{16} = 'PM';
F4L.Properties.VariableNames{16} = 'PM';
F5L.Properties.VariableNames{16} = 'PM';
F6L.Properties.VariableNames{16} = 'PM';
F7L.Properties.VariableNames{16} = 'PM';

%% Creamos las diferentes tablas que necesitaremos para entrenar los modelos.
Estas tablas se crean mezclando las mediciones de los diferentes códigos de
fallos

data = 16000; % Número total de mediciones que usaremos para entrenar

Good=[F0L;F0M];
Bad=[F1L;F1M;F2L;F2M;F3L;F3M;F4L;F4M;F5L;F5M;F6L;F6M;F7L];

rng('default'); % Las mediciones se eligen de forma totalmente aleatoria
tmp1 = randsample(1:height(Bad), height(Good));
tmp2 = Bad(tmp1,:);
Mix=[Good;tmp2];

rng('default')
tmp3=[F1L;F2L;F3L;F4L;F5L;F6L;F7L];
tmp4 = randsample(1:height(tmp3), height(F0L));
tmp5 = tmp3(tmp4,:);
LMix=[F0L;tmp5];

rng('default')
tmp6=[F1M;F2M;F3M;F4M;F5M;F6M;F7M];

```

```

tmp7 = randsample(1:height(tmp6), height(F0M));
tmp8 = tmp6(tmp7,:);
MMix=[F0M;tmp8];

Fail1=[Good;F1L;F1M];
Fail2=[Good;F2L;F2M];
Fail3=[Good;F3L;F3M];
Fail4=[Good;F4L;F4M];
Fail5=[Good;F5L;F5M];
Fail6=[Good;F6L;F6M];
Fail7=[Good;F7L;F7M];

%% Una vez mezcladas las mediciones se reduce la muestra
selector = 8;
% Aquí se puede seleccionar qué medidas se utilizan para entrenar el modelo,
utilizado para realizar los diferentes tests de forma más sencilla

% 1-7 códigos de error
% 8 Mix
% 9 LMix
% 10 MMix

rng('default')

switch(selector)
    case 1
        index = randsample(1:height(Fail1), data);
        train = Fail1(index,:);

    case 2
        index = randsample(1:height(Fail2), data);
        train = Fail2(index,:);

    case 3
        index = randsample(1:height(Fail3), data);
        train = Fail3(index,:);

    case 4
        index = randsample(1:height(Fail4), data);
        train = Fail4(index,:);

    case 5
        index = randsample(1:height(Fail5), data);
        train = Fail5(index,:);

    case 6
        index = randsample(1:height(Fail6), data);
        train = Fail6(index,:);

    case 7
        index = randsample(1:height(Fail7), data);
        train = Fail7(index,:);

    case 8
        rng('default')
        index = randsample(1:height(Mix), data);
        train = Mix(index,:);

```

```
case 9
index = randsample(1:height(LMix),data);
train = LMix(index,:);

case 10
index = randsample(1:height(MMix),data);
train = MMix(index,:);

end
%% Test
%index = randsample(1:height(Mix), round(height(Mix)/4));
%test = Mix(index,:);

%% Abrimos ClassificationLearner con la información cargada
classificationLearner(train, 'State', 'CrossVal','on');
```

### ANEXO 3 - CÓDIGO DE COMPROBACIÓN DE LA PRECISIÓN DEL MODELO

```
%Cargamos el modelo de predicción ya entrenado
load('BaggedTrees_Model.mat');

%Cargamos los datos para testear el sistema
load('F5L.mat');
load('F3M.mat');
load('F2L.mat');
load('F6M.mat');
load('F7L.mat');
load('F1L.mat');
load('F0M.mat');
load('F4L.mat');
load('F5M.mat');
load('F3L.mat');
load('F2M.mat');
load('F6L.mat');
load('F1M.mat');
load('F7M.mat');
load('F0L.mat');
load('F4M.mat');

%% Añadimos una variable que indica el modo de potencia en el que trabaja
% MPPT = 1 LPPT = 0
F7L.Var15(:) = 0;
F6L.Var15(:) = 0;
F5L.Var15(:) = 0;
F4L.Var15(:) = 0;
F3L.Var15(:) = 0;
F2L.Var15(:) = 0;
F1L.Var15(:) = 0;
F0L.Var15(:) = 0;
F0M.Var15(:) = 1;
F1M.Var15(:) = 1;
F2M.Var15(:) = 1;
F3M.Var15(:) = 1;
F4M.Var15(:) = 1;
F5M.Var15(:) = 1;
F6M.Var15(:) = 1;
F7M.Var15(:) = 1;
F7M.Properties.VariableNames{15} = 'PM';
F6M.Properties.VariableNames{15} = 'PM';
F5M.Properties.VariableNames{15} = 'PM';
F4M.Properties.VariableNames{15} = 'PM';
F3M.Properties.VariableNames{15} = 'PM';
F2M.Properties.VariableNames{15} = 'PM';
F1M.Properties.VariableNames{15} = 'PM';
F0M.Properties.VariableNames{15} = 'PM';
F0L.Properties.VariableNames{15} = 'PM';
F1L.Properties.VariableNames{15} = 'PM';
F2L.Properties.VariableNames{15} = 'PM';
F3L.Properties.VariableNames{15} = 'PM';
F4L.Properties.VariableNames{15} = 'PM';
F5L.Properties.VariableNames{15} = 'PM';
F6L.Properties.VariableNames{15} = 'PM';
```

```

F7L.Properties.VariableNames{15} = 'PM';

%% Añadimos una variable que indica el Código de fallo
F7L.Var16(:) = 7;
F6L.Var16(:) = 6;
F5L.Var16(:) = 5;
F4L.Var16(:) = 4;
F3L.Var16(:) = 3;
F2L.Var16(:) = 2;
F1L.Var16(:) = 1;
F0L.Var16(:) = 0;
F0M.Var16(:) = 0;
F1M.Var16(:) = 1;
F2M.Var16(:) = 2;
F3M.Var16(:) = 3;
F4M.Var16(:) = 4;
F5M.Var16(:) = 5;
F6M.Var16(:) = 6;
F7M.Var16(:) = 7;
F7M.Properties.VariableNames{16} = 'Fault';
F6M.Properties.VariableNames{16} = 'Fault';
F5M.Properties.VariableNames{16} = 'Fault';
F4M.Properties.VariableNames{16} = 'Fault';
F3M.Properties.VariableNames{16} = 'Fault';
F2M.Properties.VariableNames{16} = 'Fault';
F1M.Properties.VariableNames{16} = 'Fault';
F0M.Properties.VariableNames{16} = 'Fault';
F0L.Properties.VariableNames{16} = 'Fault';
F1L.Properties.VariableNames{16} = 'Fault';
F2L.Properties.VariableNames{16} = 'Fault';
F3L.Properties.VariableNames{16} = 'Fault';
F4L.Properties.VariableNames{16} = 'Fault';
F5L.Properties.VariableNames{16} = 'Fault';
F6L.Properties.VariableNames{16} = 'Fault';
F7L.Properties.VariableNames{16} = 'Fault';

%% Se agrupa toda la base de datos en una sola tabla
Good=[F0L;F0M];
Bad=[F1L;F1M;F2L;F2M;F3L;F3M;F4L;F4M;F5M;F5L;F6M;F6L;F7M;F7L];
all=[Good;Bad];

%% Creamos las variables que contabilizarán los errores y aciertos
bien=0;
mal=0;

%% Bucle de 100 iteraciones que coge una variable a azar, la predice y
almacena si es correcta o incorrecta
for i=0:+1:1000

    tmp1 = randsample(1:height(Bad), height(Good));
    tmp2 = Bad(tmp1,:);
    to_predict = all(randsample(1:height(all),1),:);
    real_type = to_predict(1,16);
    to_predict(:, "Fault") = [];
    predicted = BaggedTrees_Model.predictFcn(to_predict);

```

```
    if predicted == real_type{1,1}
      bien=bien+1;
    else
      mal=mal+1;
    end
  end
end
bien/10
```

## ANEXO 4 - CÓDIGO DEL GUI DEL SISTEMA DE DESCONEXIÓN

```
function varargout = GUI(varargin)

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @GUI_OpeningFcn, ...
                  'gui_OutputFcn',  @GUI_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before GUI is made visible.
function GUI_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to GUI (see VARARGIN)

fault = {'0' 'F1' 'F2' 'F3' 'F4' 'F5' 'F6' 'F7'};
fault = fault';
description = {'No faults detected'
              'Complete failure in one of the six IGBTs'
              'One phase sensor fault 20%'
              'Intermittent voltage sags'
              '10 to 20% nonhomogeneous partial shading'
              '15% open circuit in PV array'
              '-20% gain parameter of PI controller in MPPT/IPPT controller
of the boost converter'
              '+20% in time constant parameter of PI controller in MPPT/IPPT
controller of the boost converter'};
type = {'Functioning properly'
       'Inverter fault'
       'feedback Sensor'
       'Grid anomaly'
       'PV array mismatch'
       'PV array mismatch'
       'MPPT/IPPT controller fault'
       'Boost converter controller fault'};

tabla = table(fault, type, description, 'VariableNames', {'Fault', 'Type',
'Description'});
```

```

tabla = table2cell(tabla);
set(handles.Faults, 'data', tabla);

axes(handles.panelSolar);
imageData = imread('D:\OneDrive -
UPV\Universidad\TFG\GUI\Images\SolarPanel.jpg');
imshow(imageData);

axes(handles.sw1);
imageData = imread('D:\OneDrive -
UPV\Universidad\TFG\GUI\Images\OpenSwitch.jpg');
imshow(imageData);

axes(handles.inverter);
imageData = imread('D:\OneDrive -
UPV\Universidad\TFG\GUI\Images\Inverter.png');
imshow(imageData);

axes(handles.sw2);
imageData = imread('D:\OneDrive -
UPV\Universidad\TFG\GUI\Images\OpenSwitch.jpg');
imshow(imageData);

axes(handles.grid);
imageData = imread('D:\OneDrive - UPV\Universidad\TFG\GUI\Images\Grid.jpg');
imshow(imageData);

axes(handles.led1);
estado1 = 'verde'; % Cambia esto según tu lógica
% Dibuja el círculo relleno según el estado
if strcmp(estado1, 'rojo')
    colorRelleno = 'r'; % Rojo
elseif strcmp(estado1, 'verde')
    colorRelleno = 'g'; % Verde
end
radio = 8; % Tamaño del radio del LED
posX = 0; % Posición X del centro del LED
posY = 0; % Posición Y del centro del LED
% Dibuja el círculo relleno
rectangle('Position', [posX-radio, posY-radio, 2*radio, 2*radio],
'Curvature', [1 1], 'FaceColor', colorRelleno);
% Ajusta el aspecto del eje
axis(handles.led1, 'equal');
axis(handles.led1, 'off');

axes(handles.led2);
% Obtén el estado del LED1 (puede ser una variable o condición)
estado2 = 'verde'; % Cambia esto según tu lógica
% Dibuja el círculo relleno según el estado
if strcmp(estado2, 'rojo')
    colorRelleno = 'r'; % Rojo
elseif strcmp(estado2, 'verde')
    colorRelleno = 'g'; % Verde
end
radio = 8; % Tamaño del radio del LED
posX = 0; % Posición X del centro del LED
posY = 0; % Posición Y del centro del LED

```

```

% Dibuja el círculo relleno
rectangle('Position', [posX-radio, posY-radio, 2*radio, 2*radio],
'Curvature', [1 1], 'FaceColor', colorRelleno);
% Ajusta el aspecto del eje
axis(handles.led2, 'equal');
axis(handles.led2, 'off');

% Choose default command line output for GUI
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes GUI wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = GUI_OutputFcn(hObject, eventdata, handles)
% varargout cell array for returning output args (see VARARGOUT);
% hObject handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject handle to pushbutton1 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

%Cargamos el modelo ya entrenado
load('BaggedTrees_Model.mat');

%Cargamos los datos para testear el sistema
load('F5L.mat');
load('F3M.mat');
load('F2L.mat');
load('F6M.mat');
load('F7L.mat');
load('F1L.mat');
load('F0M.mat');
load('F4L.mat');
load('F5M.mat');
load('F3L.mat');
load('F2M.mat');
load('F6L.mat');
load('F1M.mat');
load('F7M.mat');
load('F0L.mat');
load('F4M.mat');

% Add a "Power Mode" (PW) variable 0=limited 1=max
F7L.Var15(:) = 0;

```

```

F6L.Var15(:) = 0;
F5L.Var15(:) = 0;
F4L.Var15(:) = 0;
F3L.Var15(:) = 0;
F2L.Var15(:) = 0;
F1L.Var15(:) = 0;
F0L.Var15(:) = 0;
F0M.Var15(:) = 1;
F1M.Var15(:) = 1;
F2M.Var15(:) = 1;
F3M.Var15(:) = 1;
F4M.Var15(:) = 1;
F5M.Var15(:) = 1;
F6M.Var15(:) = 1;
F7M.Var15(:) = 1;
F7M.Properties.VariableNames{15} = 'PM';
F6M.Properties.VariableNames{15} = 'PM';
F5M.Properties.VariableNames{15} = 'PM';
F4M.Properties.VariableNames{15} = 'PM';
F3M.Properties.VariableNames{15} = 'PM';
F2M.Properties.VariableNames{15} = 'PM';
F1M.Properties.VariableNames{15} = 'PM';
F0M.Properties.VariableNames{15} = 'PM';
F0L.Properties.VariableNames{15} = 'PM';
F1L.Properties.VariableNames{15} = 'PM';
F2L.Properties.VariableNames{15} = 'PM';
F3L.Properties.VariableNames{15} = 'PM';
F4L.Properties.VariableNames{15} = 'PM';
F5L.Properties.VariableNames{15} = 'PM';
F6L.Properties.VariableNames{15} = 'PM';
F7L.Properties.VariableNames{15} = 'PM';

```

```

% Add a "State variable 0=good 1=faulty

```

```

F7L.Var16(:) = 7;
F6L.Var16(:) = 6;
F5L.Var16(:) = 5;
F4L.Var16(:) = 4;
F3L.Var16(:) = 3;
F2L.Var16(:) = 2;
F1L.Var16(:) = 1;
F0L.Var16(:) = 0;
F0M.Var16(:) = 0;
F1M.Var16(:) = 1;
F2M.Var16(:) = 2;
F3M.Var16(:) = 3;
F4M.Var16(:) = 4;
F5M.Var16(:) = 5;
F6M.Var16(:) = 6;
F7M.Var16(:) = 7;
F7M.Properties.VariableNames{16} = 'Fault';
F6M.Properties.VariableNames{16} = 'Fault';
F5M.Properties.VariableNames{16} = 'Fault';
F4M.Properties.VariableNames{16} = 'Fault';
F3M.Properties.VariableNames{16} = 'Fault';
F2M.Properties.VariableNames{16} = 'Fault';
F1M.Properties.VariableNames{16} = 'Fault';
F0M.Properties.VariableNames{16} = 'Fault';

```

```

F0L.Properties.VariableNames{16} = 'Fault';
F1L.Properties.VariableNames{16} = 'Fault';
F2L.Properties.VariableNames{16} = 'Fault';
F3L.Properties.VariableNames{16} = 'Fault';
F4L.Properties.VariableNames{16} = 'Fault';
F5L.Properties.VariableNames{16} = 'Fault';
F6L.Properties.VariableNames{16} = 'Fault';
F7L.Properties.VariableNames{16} = 'Fault';

%Creamos una tabla con el mismo número de mediciones de buen funcionamiento y
de fallo
Good=[F0L;F0M];
Bad=[F1L;F1M;F2L;F2M;F3L;F3M;F4L;F4M;F5L;F5M;F6L;F6M;F7L];
tmp1 = randsample(1:height(Bad), height(Good));
tmp2 = Bad(tmp1,:);
all=[Good;tmp2];

while 1
    to_predict = all(randsample(1:height(all),1),:);

    %Almacenamos el estado real de la medición
    real_type = to_predict(:, "Fault")
    a = table2array(real_type);
    real_type_str = num2str(a);
    if ~(real_type_str == handles.real.String)
        break;
    end
end

%Eliminamos la columna que indica el tipo de fallo
to_predict(:, "Fault") = [];

% Predicción
predicted = BaggedTrees_Model.predictFcn(to_predict)

% Convierte predicted en una cadena de caracteres
predicted_str = num2str(predicted(1,1));
set(handles.predicted, 'String', predicted_str);

a = table2array(real_type);
real_type_str = num2str(a);
set(handles.real, 'String', real_type_str);

if(predicted_str == real_type_str)
    set(handles.correct, 'String', 'Predicted correctly');
    set(handles.correct, 'ForegroundColor', 'g');
else
    set(handles.correct, 'String', 'Wrong prediction');
    set(handles.correct, 'ForegroundColor', 'r');
end
global estado1 estado2
switch predicted_str
    case '0'
        estado1 = 'verde';
        estado2 = 'verde';
    case '1'
        estado1 = 'rojo';

```

```

        estado2 = 'rojo';
    case '2'
        estado1 = 'rojo';
        estado2 = 'rojo';
    case '3'
        estado1 = 'verde';
        estado2 = 'rojo';
    case '4'
        estado1 = 'rojo';
        estado2 = 'verde';
    case '5'
        estado1 = 'rojo';
        estado2 = 'verde';
    case '6'
        estado1 = 'rojo';
        estado2 = 'rojo';
    case '7'
        estado1 = 'rojo';
        estado2 = 'rojo';
end

axes(handles.led1);
if strcmp(estado1, 'rojo')
    colorRelleno = 'r'; % Rojo
elseif strcmp(estado1, 'verde')
    colorRelleno = 'g'; % Verde
end
display(estado1)
radio = 8; % Tamaño del radio del LED
posX = 0; % Posición X del centro del LED
posY = 0; % Posición Y del centro del LED
% Dibuja el círculo relleno
rectangle('Position', [posX-radio, posY-radio, 2*radio, 2*radio],
'Curvature', [1 1], 'FaceColor', colorRelleno);

% Ajusta el aspecto del eje
axis(handles.led1, 'equal');
axis(handles.led1, 'off');

axes(handles.led2);
if strcmp(estado2, 'rojo')
    colorRelleno = 'r'; % Rojo
elseif strcmp(estado2, 'verde')
    colorRelleno = 'g'; % Verde
end
radio = 8; % Tamaño del radio del LED
posX = 0; % Posición X del centro del LED
posY = 0; % Posición Y del centro del LED
% Dibuja el círculo relleno
rectangle('Position', [posX-radio, posY-radio, 2*radio, 2*radio],
'Curvature', [1 1], 'FaceColor', colorRelleno);
% Ajusta el aspecto del eje
axis(handles.led2, 'equal');
axis(handles.led2, 'off');

% --- Executes on key press with focus on pushbutton1 and none of its
controls.

```

```
function pushbutton1_KeyPressFcn(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  structure with the following fields (see
MATLAB.UI.CONTROL.UICONTROL)
%    Key: name of the key that was pressed, in lower case
%    Character: character interpretation of the key(s) that was pressed
%    Modifier: name(s) of the modifier key(s) (i.e., control, shift)
pressed
% handles    structure with handles and user data (see GUIDATA)

% --- Executes during object deletion, before destroying properties.
function figure1_DeleteFcn(hObject, eventdata, handles)
% hObject    handle to figure1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```