

Trabajo Fin de Máster  
Máster Universitario en Sistemas de Energía  
Eléctrica

Técnicas de Inteligencia Artificial y *Big Data*  
en la Gestión Óptima de Parques Fotovoltaicos:  
Estado del Arte

Autor: Rafael Morán Corbacho

Tutores: Esther Romero Ramos

José María Maza Ortega

Dpto. Ingeniería Eléctrica  
Escuela Técnica Superior de Ingeniería  
Universidad de Sevilla

Sevilla, 2021





Trabajo Fin de Máster  
Máster Universitario en Sistemas de Energía  
Eléctrica

# **Técnicas de Inteligencia Artificial y *Big Data* en la Gestión Óptima de Parques Fotovoltaicos: Estado del Arte**

Autor:

Rafael Morán Corbacho

Tutores:

Esther Romero Ramos

Catedrática de Universidad

José María Maza Ortega

Catedrático de Universidad

Dpto. Ingeniería Eléctrica  
Escuela Técnica Superior de Ingeniería  
Universidad de Sevilla

Sevilla, 2021



Trabajo Fin de Máster: Técnicas de Inteligencia Artificial y *Big Data* en la Gestión Óptima de Parques Fotovoltaicos: Estado del Arte

Autor: Rafael Morán Corbacho

Tutores: Esther Romero Ramos José María Maza Ortega

El tribunal nombrado para juzgar el trabajo arriba indicado, compuesto por los siguientes profesores:

Presidente:

Vocal/es:

Secretario:

acuerdan otorgarle la calificación de:

El Secretario del Tribunal

Fecha:



# Agradecimientos

---

Me gustaría dedicar unas palabras de agradecimiento a todas aquellas personas que me han acompañado en mi formación hasta llegar a este punto.

A mis tutores, Esther y José María, no solo por guiarme en mi camino en el campo de la ingeniería eléctrica, sino también por darme la oportunidad de introducirme en el de la inteligencia artificial, permitiéndome descubrir y explorar un nuevo gran interés por este mundo que no termina con este proyecto.

También al resto de profesores del Departamento de Ingeniería Eléctrica de la Escuela, por contagiarme su pasión por este campo.

A mis compañeros de Ingelectus, que me han acompañado durante la realización de este trabajo y han aportado al mismo, sirviéndome de apoyo con sus amplios conocimientos en los distintos campos abordados.

No olvidarme de mis amigos y compañeros que han compartido conmigo muchos años en la Escuela, los cuales han sido pieza fundamental en mi formación y a los que debo buena parte del mérito de haber llegado a este punto.

Me gustaría terminar agradeciendo a mi familia, y en especial a mis padres, por toda la ayuda y el apoyo mostrados a lo largo no solo de mi estancia en la Escuela, sino durante todo mi desarrollo como persona más allá de lo académico.

*Rafael Morán Corbacho*  
*Sevilla, 2021*





# Resumen

---

**E**n este trabajo se lleva a cabo una revisión del estado del arte de la aplicación de técnicas de inteligencia artificial y *Big Data* a una serie de tareas relacionadas con la gestión óptima de parques fotovoltaicos.

El trabajo comienza con una introducción donde se indican los objetivos del mismo, la motivación de esta revisión del estado del arte y la estructura que se seguirá en los capítulos posteriores.

Se dedicará un capítulo a describir los principales conceptos de inteligencia artificial y *Big Data*, así como las técnicas y algoritmos de estos campos que serán de aplicación en las tareas revisadas más adelante.

Tras lo anterior, en los siguientes capítulos se revisarán las aplicaciones de estas técnicas a las tareas de modelado, control, detección y diagnóstico de falta y mantenimiento predictivo de parques fotovoltaicos conectados a red. A lo largo de esta revisión se tendrán presentes las mejoras que pueda suponer el empleo de estas técnicas en comparación a los métodos convencionales.

Por último se esbozarán una serie de conclusiones extraídas tras la realización de esta revisión del estado del arte.



# Abstract

---

This project presents the state of the art of the application of artificial intelligence and *Big Data* techniques in the optimal management of grid-connected photovoltaic plants.

The project starts with an introduction where the purpose of this work is presented, along with its motivation and the structure of the rest of the document.

Next, artificial intelligence and *Big Data* main concepts are explained along with several techniques and algorithms that find application in the tasks studied in following sections.

In the following chapters, applications of this techniques to modelling, control and fault detection and diagnosis along with predictive maintenance of grid-connected photovoltaic farms are revised, giving special attention to the improvements that comes with the application of this techniques in comparison with conventional ones.

Finally, the conclusions extracted from this state of the art will be provided.



# Índice Abreviado

---

<i>Resumen</i>	III
<i>Abstract</i>	V
<i>Índice Abreviado</i>	VII
<b>1 Introducción</b>	<b>1</b>
<b>2 Técnicas</b>	<b>3</b>
<b>Inteligencia Artificial</b>	<b>4</b>
2.1 Machine Learning	5
2.2 Deep Learning	10
2.3 Algoritmos Bioinspirados	14
2.4 Lógica difusa	17
2.5 Procesamiento del Lenguaje Natural	20
<b>Big Data</b>	<b>21</b>
<b>3 Modelado</b>	<b>23</b>
3.1 Modelado a nivel de módulo FV	23
3.2 Modelado a nivel de inversor	25
3.3 Modelado a nivel de planta	25
3.4 Resumen	29
<b>4 Control</b>	<b>31</b>
4.1 Limitación de Potencia Activa	31
4.2 Control de frecuencia	31
4.3 Control de tensiones	34
4.4 Low-voltage Ride Through	35
4.5 Resumen	36
<b>5 Detección y diagnóstico de falta. Mantenimiento predictivo</b>	<b>37</b>
5.1 Detección y diagnóstico de falta	37
5.2 Mantenimiento predictivo	44
5.3 Resumen	45
<b>6 Conclusiones</b>	<b>47</b>

<i>Índice de Figuras</i>	51
<i>Índice de Tablas</i>	53
<i>Bibliografía</i>	55
<i>Glosario</i>	69

# Índice

---

<i>Resumen</i>	III
<i>Abstract</i>	V
<i>Índice Abreviado</i>	VII
<b>1 Introducción</b>	<b>1</b>
<b>2 Técnicas</b>	<b>3</b>
<b>Inteligencia Artificial</b>	<b>4</b>
2.1 Machine Learning	5
2.1.1 Aprendizaje Supervisado	6
Support Vector Machine	6
K-nearest neighbors	7
2.1.2 Aprendizaje No Supervisado	7
K-means	8
Correlation Clustering	8
Principal Component Analysis	8
Factorial Analysis	9
2.1.3 Aprendizaje por Refuerzo	9
2.1.4 Aprendizaje Semisupervisado	10
2.2 Deep Learning	10
2.2.1 Redes Neuronales Convolucionales	12
2.2.2 Redes Neuronales Recurrentes	13
2.3 Algoritmos Bioinspirados	14
2.3.1 Algoritmos Evolutivos	14
Algoritmo Genético	15
Evolución Diferencial	15
2.3.2 Optimización por Enjambre de Partículas	15
2.3.3 Algoritmo Colonia de Abejas Artificiales	16
2.3.4 Cuckoo Search	16
2.3.5 Algoritmo de Recocido Simulado	17
2.4 Lógica difusa	17
2.4.1 Control Difuso	18
2.4.2 Clasificación Difusa	19
2.5 Procesamiento del Lenguaje Natural	20
<b>Big Data</b>	<b>21</b>

---

<b>3</b>	<b>Modelado</b>	<b>23</b>
3.1	Modelado a nivel de módulo FV	23
3.2	Modelado a nivel de inversor	25
3.3	Modelado a nivel de planta	25
3.4	Resumen	29
<b>4</b>	<b>Control</b>	<b>31</b>
4.1	Limitación de Potencia Activa	31
4.2	Control de frecuencia	31
4.3	Control de tensiones	34
4.4	Low-voltage Ride Through	35
4.5	Resumen	36
<b>5</b>	<b>Detección y diagnóstico de falta. Mantenimiento predictivo</b>	<b>37</b>
5.1	Detección y diagnóstico de falta	37
5.1.1	Métodos eléctricos	38
5.1.2	Métodos visuales y térmicos	42
5.2	Mantenimiento predictivo	44
5.3	Resumen	45
<b>6</b>	<b>Conclusiones</b>	<b>47</b>
	<i>Índice de Figuras</i>	51
	<i>Índice de Tablas</i>	53
	<i>Bibliografía</i>	55
	<i>Glosario</i>	69



# 1 Introducción

---

La lucha contra el cambio climático es uno de los grandes retos del siglo XXI. La reducción de la emisión de gases de efecto invernadero, en buena parte provenientes de las grandes centrales eléctricas que emplean combustibles fósiles, es uno de los principales frentes a abordar en esta lucha. En este contexto, el empleo de fuentes de energía renovables para generar electricidad en sustitución de las centrales tradicionales contaminantes se presenta como un elemento clave en la mitigación del avance del cambio climático.

Las plantas de generación fotovoltaica extraen energía de la radiación solar y proporcionan energía eléctrica que puede ser aprovechada a través de distintas soluciones. Las plantas fotovoltaicas pueden suministrar energía a pequeñas redes aisladas o bien verter su generación a las redes de distribución y transporte tradicionales, paso previo por distintos elementos como inversores o transformadores de potencia.

La extensa y creciente proliferación de este tipo de plantas de generación requiere estudiar métodos y aplicaciones para mejorar sus prestaciones, así como su integración a red.

Otro de los grandes retos actualmente es la digitalización de las redes eléctricas en todos sus niveles. Nos encontramos en la era digital, en la que continuamente se generan grandes cantidades de datos y, cada vez más, podemos disponer de los mismos de manera instantánea. Esto afecta también a los sistemas eléctricos, en los que constantemente se generan una gran cantidad y variedad de datos provenientes de muchos agentes de la red.

En este contexto, las técnicas de inteligencia artificial (IA) y *Big Data* se presentan como herramientas clave para afrontar estos retos. El principal objetivo de estas técnicas es el de tratar los datos para obtener valor a partir de ellos.

Este trabajo tiene por objetivo llevar a cabo una revisión del estado del arte de las aplicaciones de técnicas de inteligencia artificial y *Big Data* a distintas tareas en relación a las plantas fotovoltaicas conectadas a red, analizando las ventajas y limitaciones de las mismas e identificando los retos aún por resolver en este ámbito. En concreto, estas tareas son: modelado, control, detección y diagnóstico de faltas y mantenimiento predictivo.

La estructura de este trabajo es la siguiente: se comenzará por definir los conceptos de inteligencia artificial y *Big Data* y las distintas técnicas de estos campos que serán de aplicación en los posteriores capítulos; en el Capítulo 3 se hará un repaso de las aplicaciones anteriores al modelado de plantas fotovoltaicas conectadas a red a distintos niveles de agregación; en el Capítulo 4 se estudiarán las aplicaciones de estas técnicas a distintos paradigmas de control de plantas fotovoltaicas y en el Capítulo 5 se hará lo propio para tareas de detección y diagnóstico de falta y mantenimiento predictivo. Por último, en el Capítulo 6 se presentan las conclusiones extraídas tras realizar esta revisión del estado del arte.



## 2 Técnicas

---

**E**n este capítulo se describen los principales conceptos de inteligencia artificial y *Big Data*, así como las técnicas estudiadas para la aplicación en los distintos problemas objeto de este trabajo.

En el campo a tratar no existe una taxonomía única en cuanto a la clasificación y anidamiento de los distintos paradigmas y técnicas, sino que esta difiere según el autor consultado y el contexto en el que se desarrolla dicha clasificación y definición. Es por esto que lo que a continuación se presenta es una propuesta de taxonomía, elaborada a partir de múltiples fuentes y conforme al desarrollo posterior que se realiza de las aplicaciones de las técnicas definidas.

# Inteligencia Artificial

---

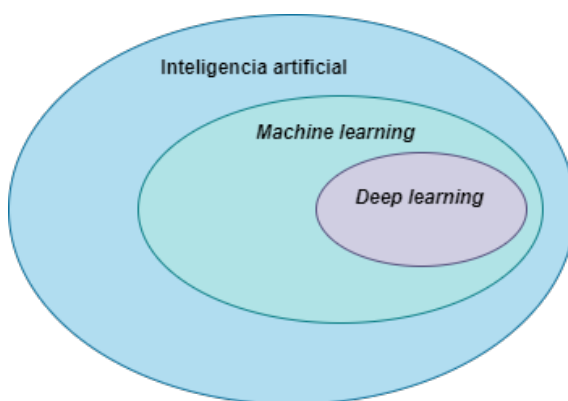
Existen múltiples definiciones del concepto de Inteligencia Artificial, si bien se pueden enmarcar en cuatro grupos atendiendo a dos conceptos fundamentales [1]. El primero de ellos está relacionado con el proceso a realizar mientras que el segundo trata de la referencia a imitar [2]. Así, se puede definir la IA como la capacidad de una máquina de realizar lo siguiente:

- Pensar como un ser humano.
- Actuar como un ser humano.
- Pensar de manera racional.
- Actuar de manera racional.

En lo referente a la actuación, entendida como el comportamiento de una máquina, surge el concepto de agente inteligente. Un agente es algo que actúa dentro de un ambiente. Así, un agente inteligente se define como un sistema que actúa de manera inteligente:

- Lo que hace es apropiado según sus circunstancias y para lograr su meta.
- Es flexible al cambio de ambiente y de metas.
- Aprende de la experiencia.
- Toma decisiones adecuadas teniendo en cuenta las limitaciones de percepción y cálculo finito.

La IA puede por tanto definirse como el estudio del diseño de agentes inteligentes [3].



**Figura 2.1** Relación entre inteligencia artificial, *machine learning* y *deep learning*.

La IA está conformada por distintas ramas, y son múltiples las clasificaciones que pueden llevarse a cabo. Este trabajo se centrará en las siguientes ramas de la IA: *machine learning*, *deep learning*, algoritmos bioinspirados, lógica difusa y procesamiento del lenguaje natural. Entre otras consideraciones, en esta clasificación se entiende el *deep learning* como un subcampo dentro del

*machine learning* (Figura 2.1) y el procesamiento del lenguaje natural como una rama independiente al *deep learning*.

## 2.1 Machine Learning

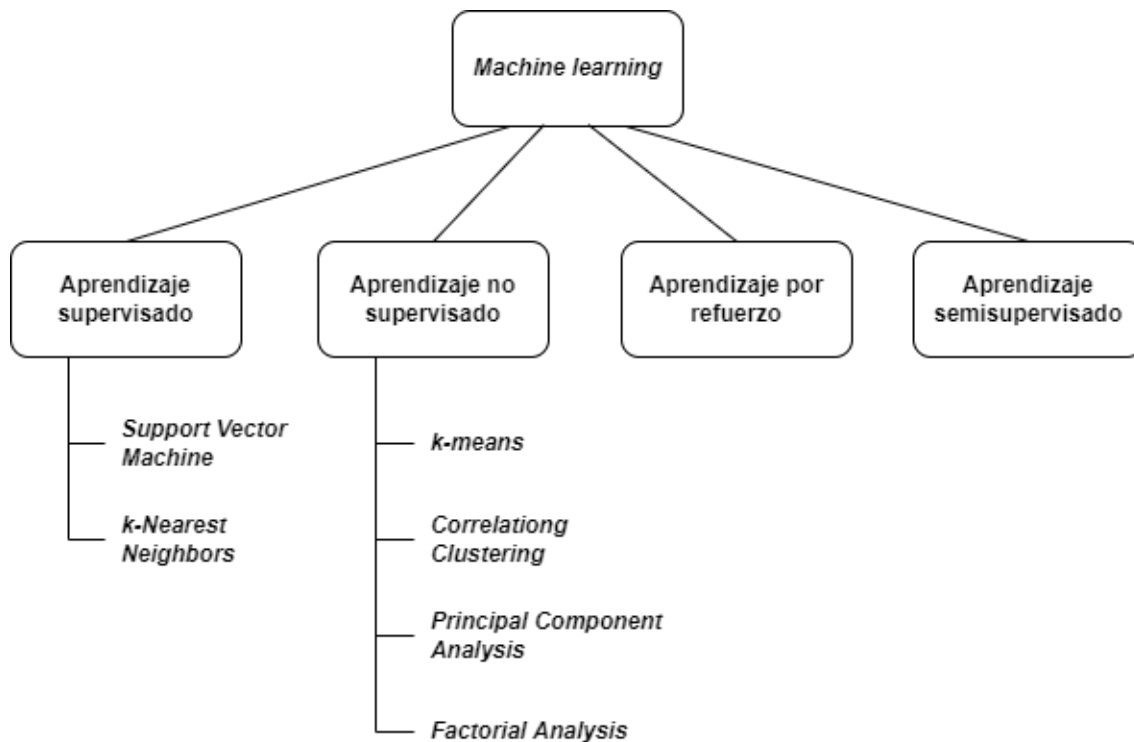
*Machine learning* (cuya traducción al español equivaldría a "aprendizaje automático" o "aprendizaje de máquina") es un subcampo de la inteligencia artificial que, según Arthur Samuel en 1959, provee a los ordenadores la capacidad de aprender sin ser explícitamente programados [4–6]. Se trata de un campo muy interdisciplinar, que toma prestados y se construye en base a conceptos de estadística, ciencias de la computación, ingeniería, ciencia cognitiva, teoría de optimización y otras muchas disciplinas [7]. Así, un algoritmo de *machine learning* es un proceso computacional que usa datos de entrada para completar una tarea sin ser explícitamente programado para llegar a una solución particular [8].

Existen múltiples algoritmos de *machine learning* como regresión lineal, regresión logística, árboles de decisión, *support vector machine* y *k-nearest neighbors* entre otros [9].

Desde el punto de vista del aprendizaje, los algoritmos de *machine learning* pueden clasificarse en cuatro categorías [10]:

- Aprendizaje supervisado (*supervised learning*).
- Aprendizaje no supervisado (*unsupervised learning*).
- Aprendizaje por refuerzo (*reinforcement learning*).
- Aprendizaje semisupervisado (*semi-supervised learning*).

La Figura 2.2 muestra la clasificación anterior, incluyendo las técnicas concretas que se definirán posteriormente en este capítulo:



**Figura 2.2** Tipos de aprendizaje automático.

De entre ellas, las más adoptadas son el aprendizaje supervisado y no supervisado. La mayoría de algoritmos de *machine learning* (en torno al 70%) son de aprendizaje supervisado. Los algoritmos de aprendizaje no supervisado supondrían entre el 10 y el 20%, mientras que los métodos semisupervisados y de aprendizaje por refuerzo serían los menos empleados [4].

### 2.1.1 Aprendizaje Supervisado

El aprendizaje supervisado es un paradigma dentro del *machine learning* en el que el objetivo es aprender una función a partir de unos datos de entrada. Los datos de entrada consisten en pares de objetos de entrada y salidas deseadas. La salida de la función puede ser un valor continuo (en los problemas de regresión) o una etiqueta de clase del objeto de entrada (en los problemas de clasificación). La tarea del "aprendiz" supervisado es la de predecir el valor de la función para cualquiera de los objetos de entrada tras haber visto un conjunto de ejemplos de entrenamiento. Para lograr esto, el aprendiz debe ser capaz de generalizar a partir de los datos de entrada para aplicar lo aprendido a situaciones no vistas durante el entrenamiento [11].

Los algoritmos de aprendizaje supervisado son entrenados usando ejemplos etiquetados, de manera que, dados unos datos de entrada, la salida deseada es conocida. El algoritmo recibe un conjunto de entradas junto con sus correspondientes salidas correctas y aprende comparando sus predicciones con estas últimas para posteriormente modificar el modelo predictivo de acuerdo a los errores cometidos [4].

Un ejemplo de problema que puede ser resuelto a partir de métodos de aprendizaje supervisado puede ser distinguir entre imágenes de naranjas y manzanas. A pesar de que cada naranja y cada manzana son únicas, los humanos somos capaces de distinguir entre unas y otras. En lugar de programar en detalle una máquina con muchos parámetros representativos de naranjas y manzanas; la idea es programar para que la máquina pueda aprender a distinguir las a través de la experiencia. Para esto se entrena a la máquina con muchas imágenes que llevan emparejadas una etiqueta indicando si pertenecen a la clase "naranja" o "manzana". Esto permite que el "aprendiz" lidie con las similitudes y diferencias cuando los objetos a clasificar tienen muchas variables dentro de una misma clase pero aún así tienen cualidades fundamentales que los identifican. Lo que es más importante, una vez la máquina haya completado el entrenamiento será capaz de reconocer una naranja o una manzana que no haya visto nunca antes [8].

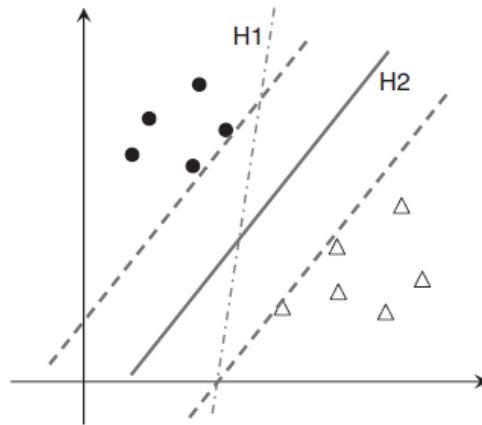
A continuación, se definen algunos algoritmos de aprendizaje supervisado, que serán de aplicación en capítulos posteriores.

#### **Support Vector Machine**

*Support Vector Machine* (SVM o máquina de vectores de soporte) es un sistema de clasificación supervisada que encuentra el hiperplano óptimo que separa los puntos de un conjunto de datos de manera que generalice bien para nuevos datos [12–17]. Este hiperplano recibe el nombre de hiperplano de máximo margen y maximiza la distancia entre los puntos más cercanos de cada clase, llamados a su vez vectores soporte [18].

El objetivo no consiste únicamente en clasificar de manera correcta los datos de entrenamiento, sino también maximizar el margen para una mejor generalización. En la Figura 2.3 puede verse cómo, a pesar de que ambos hiperplanos separan correctamente los ejemplos de ambas clases, H2 tiene un mayor margen que H1.

Los principios de SVM sirven de base para otros métodos, como *Support Vector Regression* (SVR) [20], que emplea los conceptos de vectores soporte para su aplicación en problemas de regresión; o *Relevance Vector Machine* (RVM) [21], que emplea inferencia bayesiana para obtener resultados probabilísticos en lugar de deterministas, reduciendo así el coste computacional del algoritmo.



**Figura 2.3** SVM. Ejemplos de hiperplanos separadores [19].

### ***K*-nearest neighbors**

*K*-nearest neighbors (*k*-NN o *k*-vecinos más próximos) es un algoritmo de aprendizaje supervisado simple que puede usarse para resolver problemas tanto de clasificación como de regresión. Es sencillo y fácil de implementar, pero tiene el inconveniente de que se vuelve significativamente lento a medida que el tamaño del conjunto de datos aumenta [22].

El algoritmo almacena todos los datos de entrada y clasifica cada nuevo dato de acuerdo a la clase de la mayoría de sus *k* vecinos más próximos en el conjunto de datos. En general, para obtener los vecinos más cercanos de cada dato, este método emplea la distancia euclídea como medida de la distancia entre pares de datos [23].

La principal desventaja del algoritmo *k*-NN reside en el hecho de que clasificar nuevos datos es computacionalmente costoso, ya que requiere calcular la distancia con todos los puntos del conjunto de datos para obtener sus *k* vecinos más próximos. Se trata de un método de aprendizaje vago (*lazy learning*), dado que no generaliza a partir de los datos de entrenamiento, sino que los almacena todos. Esto hace que trabaje con grandes volúmenes de datos y por tanto incurra en altos costes de computación [24]. Además de esto, el desempeño del algoritmo depende del valor de *k* que se seleccione.

Este método resulta útil en problemas donde los objetos puedan tener muchas etiquetas [25].

### **2.1.2 Aprendizaje No Supervisado**

El aprendizaje no supervisado es un método de aprendizaje automático en el que un modelo se adapta a las observaciones. Se distingue del aprendizaje supervisado en el hecho de que no existe un resultado *a priori*. Los problemas de aprendizaje no supervisado suponen encontrar (aprender) patrones dentro de los datos de entrada cuando éstos no llevan aparejados una salida específica [11].

En los algoritmos de aprendizaje no supervisado sólo se dispone de los datos de entrada, los cuales no están etiquetados, y no existe una salida predefinida: no hay un "supervisor" [10]. El objetivo de los problemas de aprendizaje no supervisado es explorar los datos y descubrir estructuras en ellos [4].

La tarea de aprendizaje no supervisado más común es la de *clustering* (agrupamiento): detectar potenciales grupos entre los datos de entrada [2]. *Clustering* es una forma de clasificar los datos crudos de manera razonable y encontrar patrones ocultos que puedan existir en los conjuntos de datos [26]. Es un proceso de agrupar datos en grupos (*clusters*) inconexos de manera que los datos pertenecientes a un mismo cluster sean similares mientras que aquellos que pertenezcan a clusters distintos difieran entre sí [27].

A continuación se describen brevemente algunos de los algoritmos de *clustering* más extendidos, cuya aplicación al campo objeto de estudio se verá en capítulos posteriores.

### ***K-means***

El algoritmo *K-means* (o *k*-medias) es uno de los algoritmos de aprendizaje no supervisado más simples que resuelven problemas de *clustering*. Es un método numérico, no supervisado, no determinista e iterativo. Destaca por su simpleza y rapidez, lo que lo hace muy eficaz a la hora de generar buenos resultados de *clustering* en muchas aplicaciones prácticas.

Se trata de un algoritmo de *clustering* particional, que consiste en clasificar los datos de entrada en *k clusters* distintos a través de un proceso iterativo, convergiendo a un mínimo local y dando como resultado clusters compactos e independientes.

El algoritmo consta de dos fases: la primera consiste en seleccionar *k* centros de manera aleatoria, donde el valor de *k* es fijo y definido previamente. Una vez hecho esto, se asocia cada dato del conjunto de datos al centro más cercano (normalmente se emplea la distancia euclídea). La primera fase queda completada cuando ya no queden datos sin asociar a un centro, y de esta forma queda conformado un agrupamiento inicial. El siguiente paso consiste en recalcular los *k* nuevos centroides como el baricentro de los clusters generados en la primera fase. Este proceso se repite hasta que al iterar no se producen cambios en las posiciones de los centros o se alcanza algún otro criterio de convergencia.

*K-means* tiene como ventajas ser un algoritmo muy rápido, simple y computacionalmente eficiente, sin embargo es difícil determinar el valor de *k* (número de *clusters*) para obtener los mejores resultados; es sensible a *outliers* (datos fuera de rango) y la posición inicial de los centroides afecta al mínimo local que se alcanza, por lo que no existe una única solución del algoritmo para un número *k* determinado [11, 22, 24, 27–31].

### ***Correlation Clustering***

*Correlation Clustering* (CC) [32] es una técnica de *clustering* motivada por la necesidad de resolver problemas de clasificación con grandes conjuntos de datos cuando no se conoce o no se quiere establecer el número de *clusters* en los que dividir los datos.

Mientras que los algoritmos de *clustering* más comúnmente usados, como *k-means*, requieren conocer *a priori* el número de *clusters* en los que se quiere dividir los datos, existen ciertos tipos de problemas, como la clasificación de documentos web, en los que obtener el número de *clusters* no es una tarea trivial. *Correlation Clustering* constituye un método para clasificar un conjunto de objetos en el número óptimo de *clusters* sin la necesidad de especificar ese número de antemano, basándose en la similitud entre los elementos del conjunto de datos [33].

### ***Principal Component Analysis***

Cada vez son más comunes los grandes conjuntos de datos y estos a menudo resultan difíciles de interpretar. *Principal Component Analysis* (PCA o análisis de componentes principales) es una técnica de aprendizaje no supervisado empleada para reducir la dimensionalidad de estos conjuntos de datos, facilitando su interpretación pero al mismo tiempo minimizando la pérdida de información. Esto se consigue creando nuevas variables que maximicen la varianza y no estén correlacionadas entre ellas. Estas nuevas variables son funciones lineales de las variables originales del conjunto de datos [34].

PCA consiste en un proceso estadístico que emplea una transformación ortogonal para convertir un conjunto de datos de posibles variables correlacionadas en un conjunto de datos de variables no correlacionadas, que reciben el nombre de componentes principales. En este nuevo conjunto, las dimensiones de los datos se ven reducidas, lo que implica una computación más rápida y sencilla. PCA se emplea como técnica de reducción de dimensionalidad [22].



### Factorial Analysis

Además del análisis de componentes principales, otro de los métodos clásicos de reducción de dimensionalidad es *Factorial Analysis* (FA o análisis factorial). En este caso también se reducen las dimensiones del conjunto de datos a través de la creación de nuevas variables, llamadas factores, que son combinaciones lineales de las originales. Sin embargo, a diferencia del método PCA, donde el objetivo es encontrar una representación del conjunto de datos de menor dimensión teniendo en cuenta la varianza de los datos; en el análisis factorial se persigue el mismo objetivo pero teniendo en cuenta la correlación entre las variables [35].

#### 2.1.3 Aprendizaje por Refuerzo

El aprendizaje por refuerzo es un paradigma dentro del campo del *machine learning*, un conjunto de algoritmos cuyo objetivo es que un agente aprenda cómo comportarse en un entorno, donde la única realimentación consiste en una señal de recompensa [36]. En este tipo de problemas, no se le dice al agente (aprendiz) qué acciones debe tomar, sino que es él quien debe descubrir qué acciones producen una mayor recompensa mediante prueba y error. En algunos casos, las acciones pueden afectar no sólo a la recompensa inmediata sino también a situaciones futuras y por tanto a subsecuentes recompensas. En estos casos el agente debe tratar de maximizar la recompensa a largo plazo. Así, en los algoritmos de aprendizaje por refuerzo el agente aprende por interacción con el entorno, basándose en su propia experiencia [37].

Además del agente y el entorno, un sistema de aprendizaje de refuerzo cuenta con los siguientes elementos [37]:

- Una política que define el comportamiento del agente en un momento dado. La política es el núcleo del agente en el aprendizaje por refuerzo, ya que determina su comportamiento. En general, las políticas pueden ser estocásticas, asignando probabilidades a cada acción.
- Una señal de recompensa que define el objetivo del problema de aprendizaje por refuerzo. En cada estado, el entorno envía al agente un único número (señal) llamado recompensa. El objetivo del agente es maximizar la recompensa total en el largo plazo. Por tanto, la señal de recompensa define lo que son buenos y malos eventos para el agente, y son la principal fuente de alteración de la política del mismo.
- Mientras que una señal de recompensa indica qué es bueno o malo en un sentido inmediato, se emplea una función de valor para determinar esto mismo a largo plazo. El valor de un estado se define como la recompensa total que el agente puede esperar acumular en el futuro, empezando por dicho estado. Por ejemplo, un estado puede provocar una señal de recompensa inmediata baja pero aun así tener un gran valor porque suele ir seguido de otros estados futuros que impliquen grandes recompensas.
- Un modelo del entorno, que permite estimar el comportamiento del mismo. Dado un estado y una acción, el modelo del entorno puede predecir el siguiente estado resultante y la señal de recompensa.

Para formalizar matemáticamente los problemas de aprendizaje reforzado se emplean los llamados procesos de decisión de Markov. Los procesos de decisión de Markov son una formalización clásica de toma de decisiones secuenciales, donde las acciones no sólo tienen influencia en las recompensas inmediatas, sino también en situaciones o estados subsecuentes, y por tanto en recompensas futuras. Para que un proceso sea un proceso de decisión de Markov, las probabilidades de los estados futuros deben depender únicamente del estado presente, esto es, el estado presente debe incluir información de los estados anteriores. Si esto se cumple, se dice que el estado posee la propiedad de Markov [37].

Las técnicas de aprendizaje reforzado son también de aplicación en problemas en los que existen varios agentes inteligentes. Estos sistemas reciben el nombre de sistemas multi-agente, en los que un grupo de agentes inteligentes débilmente conectados actúan en un mismo entorno para lograr

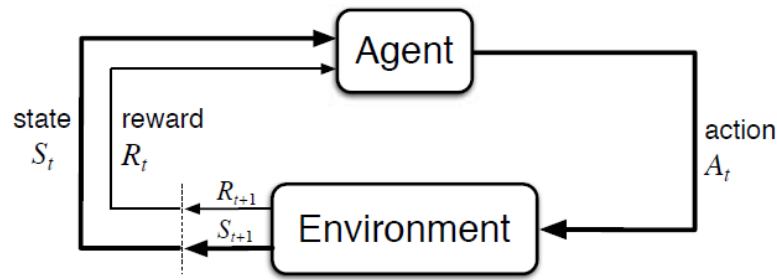


Figura 2.4 Interacción agente-entorno en un proceso de decisión de Markov [37].

una meta común. Esto puede lograrse mediante cooperación o competición y compartiendo o no información entre ellos [38].

Los métodos de aprendizaje reforzado constituyen una herramienta computacional práctica para el diseño de sistemas autónomos que mejoren su desempeño a través de su propia experiencia. Sus aplicaciones van desde el campo de el control de sistemas, la robótica y la fabricación industrial; hasta los problemas de optimización combinatoria como los juegos de mesa o videojuegos [39].

#### 2.1.4 Aprendizaje Semisupervisado

El aprendizaje semisupervisado se emplea para las mismas aplicaciones que el aprendizaje supervisado. En este caso el entrenamiento se hace usando datos tanto etiquetados como no etiquetados, típicamente una pequeña cantidad de datos etiquetados junto con un gran conjunto de datos no etiquetados. Esto último se debe a que la obtención de datos no etiquetados requiere de menos esfuerzo.

Este tipo de aprendizaje se puede aplicar en métodos como clasificación, regresión y predicción. Los algoritmos de aprendizaje semisupervisado son de utilidad cuando el coste asociado con el etiquetado de datos es muy alto como para disponer de un conjunto de datos completamente etiquetado [4].

## 2.2 Deep Learning

Las técnicas de *machine learning* convencionales se encuentran limitadas en su habilidad de procesar datos naturales en bruto. Durante décadas, construir un sistema de reconocimiento de patrones o *machine learning* requería de ingeniería de detalle y un dominio experimentado en la extracción de características, *features*, con el objetivo de transformar los datos en bruto de manera adecuada para que sirvieran de entrada a los subsistemas de aprendizaje, a menudo clasificadores, de manera que estos pudieran detectar o clasificar patrones [40]. Una mala presentación de los datos a menudo conlleva un peor desempeño de los algoritmos de *machine learning*.

En este contexto, surge el concepto de aprendizaje profundo o *deep learning*. En comparación a lo anterior, los algoritmos de *deep learning* pueden llevar a cabo la extracción de *features* de manera automática, lo que supone que los desarrolladores no necesiten ser expertos en el dominio del problema en cuestión y reduce al mínimo el esfuerzo humano necesario [41,42]. Así, las técnicas de *deep learning* son un subconjunto de métodos de *machine learning* que pueden aprender *features* de manera jerárquica, lo que permite obtener conceptos complejos a partir de otros más simples, construyendo arquitecturas profundas [43].

Estas arquitecturas se estructuran en capas de distintos niveles de representación del conocimiento, donde las *features* de capas de niveles más altos se extraen a partir de las de las capas de niveles más bajos. Este tipo de arquitecturas se inspiran originalmente en el funcionamiento del cerebro

humano [44], dando lugar a una familia de métodos que recibe el nombre de redes neuronales artificiales.

El cerebro humano está compuesto por un gran número de unidades de procesamiento, llamadas neuronas, y cada neurona está conectada a otras muchas neuronas a través de conexiones llamadas sinapsis. Las neuronas operan en paralelo y se transfieren información entre ellas a través de sinapsis. El procesamiento de la información es llevado a cabo por las neuronas y la memoria se encuentra en las sinapsis, esto es, en la forma en la que las neuronas están conectadas y se influyen unas a otras. Los modelos de redes neuronales persiguen ser una analogía computacional de estos procesos [10].

El objetivo de las redes neuronales artificiales es imitar el cerebro humano y sus neuronas, a través de una unidad fundamental: el perceptrón o neurona. Típicamente, el perceptrón cuenta con muchas entradas y una única salida [45, 46]. Cada entrada tiene asociada un peso, que multiplica a la señal de entrada, ponderando su valor. Estas entradas ponderadas se suman junto con un término, llamado *bias* o sesgo. La salida de la neurona se calcula aplicando a la suma anterior una función de activación [9] que añade una no linealidad al resultado. En la Figura 2.5 se muestra la estructura básica de una neurona artificial.

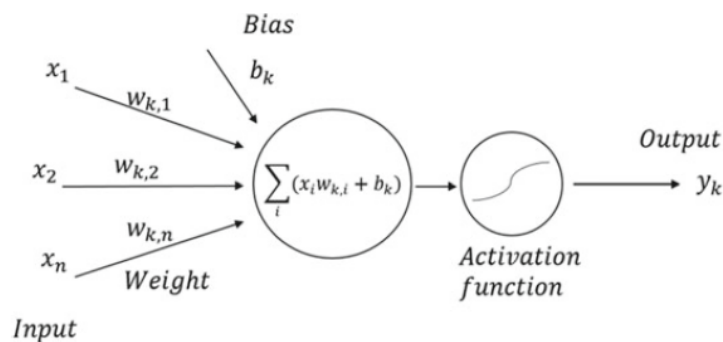
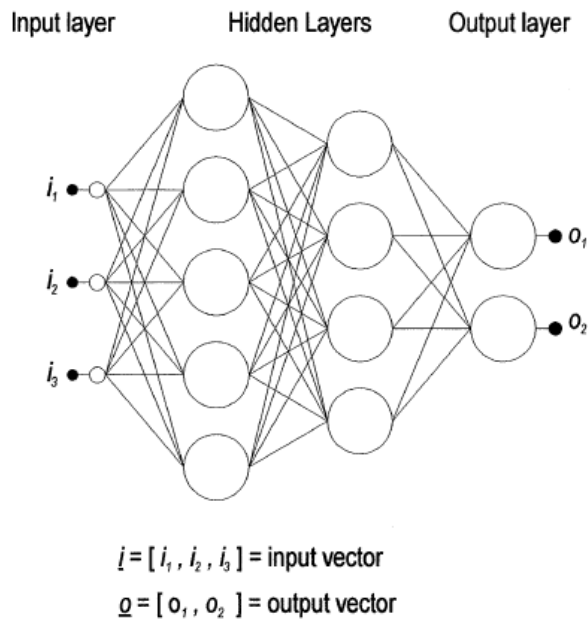


Figura 2.5 Estructura de una neurona artificial [9].

Las neuronas pueden disponerse en paralelo formando capas, donde todas las neuronas de una capa toman como entrada las salidas de las neuronas de la capa previa, calculan sus valores de salida en paralelo, y con estos valores alimentan a todas las neuronas de la siguiente capa. Esta arquitectura recibe el nombre de perceptrón multicapa [10] o red neuronal *feedforward* [44]. La Figura 2.6 muestra un ejemplo de perceptrón multicapa formado por una capa de entrada, dos capas intermedias, llamadas "capas ocultas"; y una de salida. La superposición de muchas funciones simples no lineales permite al perceptrón multicapa aproximar funciones extremadamente no lineales [47], lo que lo convierte en una excelente herramienta en tareas de detección de patrones, clasificación y regresión.

Para entrenar a la red neuronal, se alimenta a ésta con un conjunto de datos de entrenamiento. Durante el entrenamiento, las salidas producidas por la red neuronal se comparan con los valores deseados, calculándose una serie de errores. Evaluando estos errores se pueden ajustar los pesos de las neuronas hasta que la red neuronal produzca unas salidas con unos márgenes de error aceptables [9]. Los algoritmos más extendidos para entrenar redes neuronales son el descenso del gradiente y *backpropagation*:

- El descenso del gradiente es un método iterativo en el que el objetivo es minimizar una función de coste utilizando las derivadas parciales de la misma para "descender" hacia el mínimo de la función, haciendo uso de un tamaño de paso, o learning rate, para acelerar este descenso [24].
- El algoritmo *backpropagation* consiste en propagar los errores de las neuronas desde las capas más externas hacia las más cercanas a la entrada, esto es, "hacia atrás". Este método se utiliza para calcular el gradiente de la función de coste en una red neuronal para así, haciendo



**Figura 2.6** Ejemplo de perceptrón multicapa [9].

uso del descenso del gradiente, ajustar los pesos de las conexiones de la red, o dicho de otra manera, entrenar la red neuronal.

Además del perceptrón multicapa, existen otros tipos de redes neuronales que se construyen formando arquitecturas de diversa topología e introducen distintos algoritmos para alcanzar sus objetivos, como las redes neuronales probabilísticas [48] o de función de base radial [49], entre otras. A continuación se describen dos de las arquitecturas cuyo uso está más extendido y que serán de aplicación en capítulos posteriores.

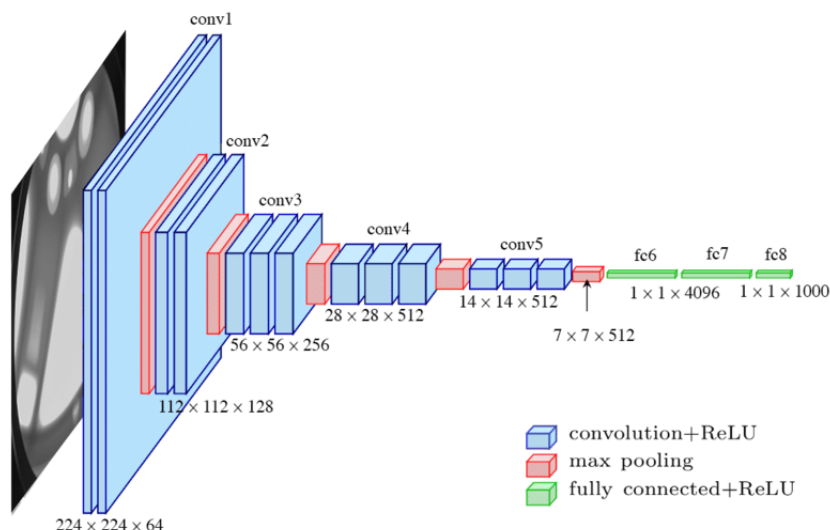
### 2.2.1 Redes Neuronales Convolucionales

Las redes neuronales convolucionales (CNN, del inglés Convolutional Neural Network) están pensadas para procesar datos contenidos en matrices de múltiples dimensiones, como por ejemplo imágenes a color [40] (una imagen a color se compone de tres matrices de dos dimensiones superpuestas, una por cada canal RGB). Este tipo de redes neuronales deben su nombre al empleo de la operación matemática de convolución. Las CNNs son redes neuronales que aplican convolución en al menos una de sus capas [50].

La salida de las capas de convolución pasa por una función de activación no lineal y después suele pasar por una capa de *pooling* (o capa de reducción), que reemplaza valores de ciertas localizaciones por los obtenidos en salidas cercanas. La función de *pooling* permite hacer que las *features* aprendidas sean invariantes a pequeñas traslaciones o rotaciones en los datos de entrada. Esto permite que los problemas a resolver por CNNs no tengan que ser espacialmente dependientes. En otras palabras, por ejemplo, en una aplicación de detección de rostros, no hace falta prestar atención a en qué punto de la imagen están localizadas las caras, la única preocupación debe ser la de encontrarlas independientemente de su posición en las imágenes de entrada.

Uno de los mayores beneficios que aporta el empleo de CNNs es la reducción del número de parámetros en comparación con las redes neuronales tradicionales. Esto permite a los desarrolladores construir modelos más extensos para resolver problemas más complejos. Otro de los grandes aspectos de las CNNs es la obtención de *features* abstractas a medida que la información se propaga a lo largo de las capas más profundas de la red. Por ejemplo, cuando se está tratando con imágenes, las

capas más cercanas a la entrada aprenden representaciones de *features* de bajo nivel (como bordes y esquinas) mientras que las capas más profundas aprenden representaciones de más alto nivel (como contornos y partes de objetos). Una vez se han aprendido las *features* de interés, sus activaciones se emplean en las capas finales, que suelen ser capas completamente conectadas (como las de perceptrones multicapa), para clasificar los datos de entrada o realizar procesos de regresión con ellos [44, 51–54]. La Figura 2.7 muestra una de las arquitecturas de CNN más populares, llamada VGG-16, ampliamente empleada en problemas de clasificación de imágenes.



**Figura 2.7** Arquitectura de red neuronal convolucional VGG-16 [55, 56].

Las CNNs han sido extensamente utilizadas en distintas aplicaciones como visión por ordenador [57], procesamiento del lenguaje natural [58] y procesamiento de voz [59], entre otros [41].

## 2.2.2 Redes Neuronales Recurrentes

Las redes neuronales recurrentes (RNN, del inglés *Recurrent Neural Network*) son modelos en los que la salida es función no sólo de las entradas actuales, sino también de las entradas previas, que se codifican como estados ocultos. Esto hace que las RNNs, en contraste con las redes neuronales tradicionales, puedan trabajar con datos secuenciales, en los que la información en un momento dado guarda correlación con estados anteriores. Ejemplos de este tipo de información podrían ser secuencias de texto, audio, vídeo o estructuras de datos de tamaño variable [40, 44, 50].

Uno de los principales inconvenientes de empleo de RNNs es su sensibilidad al desvanecimiento y explosión del gradiente [60]. En otras palabras, los gradientes pueden decaer o crecer exponencialmente (explotar) debido a las multiplicaciones de pequeños o grandes términos derivativos durante el entrenamiento de la red. Esta sensibilidad se reduce con el tiempo, lo que implica que la red "olvide" las entradas iniciales con la llegada de nuevas entradas. Para resolver esto, se emplea una arquitectura de RNNs llamada *Long Short-Term Memory* (LSTM) [61] que proporciona a la red bloques de memoria en sus conexiones recurrentes. Cada bloque incluye celdas de memoria que almacenan los estados temporales de la red e incluye puertas de control que controlan el flujo de información [41].

## 2.3 Algoritmos Bioinspirados

Los algoritmos bioinspirados son un conjunto de algoritmos de optimización que encuentran su inspiración en procesos que ocurren en la naturaleza. Estos procesos van desde los procesos evolutivos, hasta el comportamiento de las bandadas de aves, pasando por un amplio espectro de procesos y comportamientos que pueden hallarse en la naturaleza. Se trata de un conjunto de algoritmos de búsqueda que, partiendo de una solución o un conjunto de soluciones, emplean técnicas inspiradas en la naturaleza con el objetivo de alcanzar el óptimo global de una función objetivo. El gran potencial de estos algoritmos reside en parte en el hecho de que no es necesario que la función objetivo que define el problema de optimización presente una formulación matemática exacta, sino que esta puede ser aproximada o incluso inexistente. El otro pilar fundamental es la gran eficiencia en cuanto a coste computacional en el que incurren estos algoritmos.

A continuación se presenta una taxonomía de los diferentes algoritmos bioinspirados, cuya aplicación al campo objeto de estudio se verá más adelante en capítulos posteriores.

### 2.3.1 Algoritmos Evolutivos

Si bien la mayoría de esfuerzos puestos en desarrollar inteligencia artificial se centran en imitar el comportamiento y la inteligencia humana, resulta evidente que los procesos evolutivos que tienen lugar en la naturaleza presentan comportamientos inteligentes. Estos comportamientos pueden ser simulados y empleados en la creación de máquinas inteligentes [62].

Esta es la tesis que sostiene la computación evolutiva, una rama de las ciencias de la computación que aplica ideas de la evolución biológica para resolver problemas computacionales. La evolución biológica resulta ser una buena fuente de inspiración para abordar problemas computacionales difíciles. La evolución es, en efecto, un método de búsqueda entre un enorme número de posibilidades (por ejemplo, un conjunto de posibles secuencias genéticas) que suponen "soluciones" que permiten a los organismos sobrevivir y reproducirse en sus entornos. También puede verse la evolución como un método de adaptación a entornos cambiantes. Visto desde una perspectiva de alto nivel, las "reglas" evolutivas son bastante simples: las especies evolucionan a través de variaciones aleatorias (vía mutación, recombinación y otros procesos), seguido de un proceso de selección natural en el cual las especies más aptas tienden a sobrevivir y reproducirse, y de esta forma propagar su material genético hacia futuras generaciones. Aunque simples, estas reglas son responsables de una extraordinaria variedad y complejidad que podemos ver en la biosfera [63].

Existen múltiples técnicas dentro del campo de la computación evolutiva, que reciben el nombre de algoritmos evolutivos. La idea común que subyace detrás de todas estas técnicas es la misma: dada una población de individuos dentro de un entorno con recursos limitados, la competición por dichos recursos provoca la selección natural (supervivencia de los más aptos). Esto a su vez implica una subida en la aptitud de la población. Dada una función a maximizar, se puede crear un conjunto aleatorio de soluciones, que son elementos del dominio de la función. A continuación, se aplica dicha función a estos elementos como una medida de la aptitud de los mismos (cuanto más alto el valor, más aptos). Basándose en estos valores de aptitud, algunos de los mejores candidatos son elegidos para germinar la siguiente generación. Esto se hace aplicando recombinaciones y/o mutaciones sobre estos. La recombinación es una operación que se aplica a dos o más candidatos (llamados padres) para producir uno o varios nuevos candidatos (hijos). Las mutaciones se aplican sobre un candidato para producir otro nuevo. De esta forma la aplicación de las operaciones de recombinación y mutación sobre los padres conlleva la creación de un nuevo conjunto de candidatos (la descendencia). Estos nuevos candidatos compiten entre ellos y con las generaciones pasadas por un puesto en la próxima generación. Este proceso puede ser iterado hasta encontrar un candidato con suficiente calidad (una solución) o hasta toparse con una limitación computacional [64].

A continuación se describen dos de los algoritmos evolutivos más populares y que serán de aplicación en siguientes capítulos.

### Algoritmo Genético

El algoritmo genético (GA, *Genetic Algorithm*) es el algoritmo evolutivo más ampliamente conocido [64]. Este método se inspira en la teoría de la evolución de Darwin [65, 66], y en él se simula la supervivencia de los individuos más aptos y sus genes.

GA es un algoritmo de búsqueda probabilístico basado en las mecánicas de selección natural y genética natural. El algoritmo comienza con un conjunto de soluciones, llamado población. Cada solución se corresponde con un cromosoma y cada parámetro representa un gen. GA evalúa la aptitud de cada individuo de la población usando una función de aptitud (función objetivo). Esta evaluación se lleva a cabo en cada generación, y los cromosomas de la siguiente generación son seleccionados de manera probabilística en base a sus valores de aptitud. Algunos de los cromosomas seleccionados mutan y/o se recombinan con otros de manera aleatoria. Dado que los cromosomas más aptos tienen mayor probabilidad de ser seleccionados, los cromosomas de las siguientes generaciones tendrán de media un mayor nivel de aptitud. El proceso de evolución se repite hasta que se satisface un criterio de convergencia o de parada.

La forma en la que se lleva a cabo la selección de las siguientes generaciones permite escoger las mejores soluciones al mismo tiempo que evita caer en óptimos locales, ya que también se escogen soluciones menos aptas. Por esto, GA es un algoritmo fiable y capaz de estimar el óptimo global para un problema, dado que mantiene las mejores soluciones en cada iteración y las usa para mejorar otras soluciones.

GA se emplea para resolver problemas de optimización y para modelar sistemas en los que cierto grado de aleatoriedad está presente [67–69].

### Evolución Diferencial

Evolución diferencial (DE, *Differential Evolution*) es un algoritmo evolutivo similar a GA, en el que una población de individuos se emplea para buscar una solución óptima. La principal diferencia es que, mientras que en GA la mutación es el resultado de pequeñas perturbaciones en los genes de un individuo, en DE esta operación consiste en realizar combinaciones aritméticas de individuos. Al principio del proceso evolutivo, la operación de mutación favorece la exploración. A medida que el proceso avanza, esta misma operación favorece la explotación. Es por esto que el algoritmo DE adapta automáticamente los incrementos en las operaciones de mutación al mejor valor de acuerdo a la fase del proceso evolutivo. Por tanto, en DE la mutación no está basada en una función de densidad de probabilidad predefinida [70, 71].

Una ventaja de DE sobre GA es que, mientras que en el último las posibles soluciones son tratadas como cadenas de caracteres, en DE estas se tratan como cadenas de números reales, por lo que no son necesarios procesos de codificación y decodificación [72].

#### 2.3.2 Optimización por Enjambre de Partículas

Optimización por enjambre de partículas (PSO, *Particle Swarm Optimization*) es un algoritmo computacional inteligente, estocástico, de optimización global basado en población. Está inspirado en el comportamiento social de una bandada de aves en búsqueda de comida [71, 73]. Durante el "vuelo", cada partícula ajusta su posición de acuerdo a su propia experiencia y a la de sus vecinos, moviéndose hacia la mejor posición encontrada por ella misma o por sus vecinos. De esta forma, PSO combina métodos de búsqueda local con métodos de búsqueda global [74, 75].

En la práctica, el algoritmo PSO se inicializa con una población de soluciones candidatas aleatorias (partículas). El estado de cada partícula en el espacio de búsqueda se caracteriza mediante dos factores: su posición y su velocidad. El desempeño de cada partícula se evalúa de acuerdo a una función de aptitud (como una función de coste). Se asigna una velocidad aleatoria a cada partícula

y estas se mueven de manera iterativa por el espacio de búsqueda. Las partículas son atraídas hacia el punto de mayor aptitud encontrado hasta el momento por ellas mismas y por sus vecinos [76].

PSO se diferencia de los algoritmos evolutivos en que, a pesar de inicializar un conjunto de soluciones e ir actualizando distintas generaciones, no emplea operaciones evolutivas como recombinación y mutación. En el algoritmo PSO, las partículas tratan de alcanzar el óptimo siguiendo el óptimo global de cada momento, en lugar de empleando dichas operaciones evolutivas. Dado que en PSO no se descartan las peores soluciones al avanzar en las sucesivas generaciones, como se hacía en GA, este muestra una mayor eficiencia a la hora de mantener la diversidad en el conjunto de soluciones [71].

### 2.3.3 Algoritmo Colonia de Abejas Artificiales

El algoritmo colonia de abejas artificiales (ABC, *Artificial Bee Colony*) está inspirado en el comportamiento de las abejas en una colonia buscando fuentes de comida y comunicándose entre ellas. En este modelo, las abejas artificiales se agrupan en tres categorías: empleadas, espectadoras y exploradoras. Las soluciones potenciales del problema de optimización son "fuentes de comida", con la función de aptitud de la solución modelada como "cantidad de néctar". Las abejas exploradoras identifican nuevas fuentes de comida, las empleadas evalúan la calidad de las fuentes de comida existentes, y las espectadoras obtienen información de las empleadas acerca de la calidad de las fuentes de comida que estas últimas han chequeado. Más tarde las abejas espectadoras evalúan las fuentes de comida frente a sus vecinas de manera que las soluciones puedan ser descartadas y reemplazadas por las descubiertas por las exploradoras.

El método funciona de la siguiente forma: inicialmente se asigna a cada abeja empleada una fuente de comida (una solución). Cada empleada evalúa su fuente de comida y la modifica. Si la nueva fuente es más óptima que la anterior, la empleada se olvida de la antigua y se queda con la nueva. Tras esto se comunican con las abejas espectadoras. Estas últimas escogen una fuente de comida dependiendo de la aptitud comunicada por todas las empleadas y producen una modificación. Nuevamente, si la modificación supone una mejora, esta reemplaza a la fuente de comida original. Las fuentes abandonadas son reemplazadas por nuevas fuentes por las abejas exploradoras.

ABC ha demostrado ser un algoritmo robusto, simple y flexible en comparación con otros algoritmos basados en población [77–80].

### 2.3.4 *Cuckoo Search*

El algoritmo *Cuckoo search* (CS) es un algoritmo meta-heurístico de optimización desarrollado recientemente (en 2009). Es un método bioinspirado, basado en el parasitismo de puesta de algunas especies de cuco, junto con vuelos de Lévy, un tipo de paseo aleatorio.

Algunas especies de cuco se reproducen de forma parásita depositando sus huevos en nidos de otras especies de pájaro. Algunos cucos se involucran de tal manera que la hembra parasitaria puede incluso imitar varios colores y patrones de los huevos de algunas especies huéspedes. Esto reduce la probabilidad de que los huevos sean abandonados y por tanto aumenta la reproductividad. Si un pájaro huésped descubre que los huevos no son suyos, este tirará el huevo parásito o bien abandonará el nido para construir otros. A la hora de parasitar un nido, los cucos suelen escoger aquel en el que el pájaro huésped acaba de poner sus huevos. En general, los huevos de cuco eclosionan algo antes que los del nido huésped. El primer instinto de un cuco al nacer es el de echar del nido al resto de huevos. Esta acción resulta en un incremento de la cuota de comida de la cría de cuco [81]. El comportamiento reproductivo del cuco puede ser aplicado a varios problemas de optimización.

Para mejorar el desempeño del algoritmo CS se emplean vuelos de Lévy en lugar de simples paseos aleatorios. Un vuelo de Lévy es un paseo aleatorio en el que la longitud de los pasos son calculados de acuerdo a una distribución de probabilidad de cola pesada. Varios estudios han



demostrado que el comportamiento de los vuelos de muchos animales e insectos demuestran las características típicas de los vuelos de Lévy.

En el algoritmo CS, cada huevo en un nido representa una solución, y un huevo de cuco una nueva solución. El objetivo es emplear las soluciones nuevas y potencialmente mejores (huevos de cuco) para reemplazar a las soluciones no tan buenas de los nidos. En la versión más simple del algoritmo cada nido tiene un huevo. El método puede extenderse a casos más complicados en el que cada nido contenga múltiples huevos representando un conjunto de soluciones. El algoritmo CS se basa en tres reglas:

- Cada cuco deposita un huevo cada vez, y lo suelta en un nido elegido aleatoriamente.
- Los nidos con los huevos de mejor calidad (soluciones) pasarán a las siguientes generaciones.
- El número de nidos disponibles es fijo, y un huésped puede descubrir un huevo parásito con una cierta probabilidad. En tal caso, el pájaro huésped puede o bien tirar el huevo de cuco o bien abandonar el nido para construir otro completamente nuevo en una nueva localización.

CS es un algoritmo basado en población, como GA y PSO. En él, el empleo de vuelos de Lévy hace que la aleatoriedad generada sea más eficiente. Además, el número de parámetros a ajustar es menor que en GA y PSO, lo que implica que el algoritmo sea más genérico y se pueda adaptar a una mayor variedad de problemas de optimización [82, 83].

### 2.3.5 Algoritmo de Recocido Simulado

El algoritmo de recocido simulado (SA, *Simulated Annealing*) es una técnica de búsqueda global basada en trayectoria, empleada para resolver problemas de optimización.

SA está motivado por una analogía de las mecánicas estadísticas del recocido de sólidos. En metalurgia, el recocido es el proceso empleado para templar o endurecer metales y cristales, calentándolos a altas temperaturas para después enfriarlos gradualmente, permitiendo así que el material alcance un estado cristalino de baja energía [2]. Un estado de baja energía suele implicar un estado de alta ordenación. SA emplea un conjunto de operaciones análogas a un "enfriamiento controlado" en problemas de optimización con el objetivo de transformar soluciones malas, "desordenadas", en soluciones óptimas.

La metodología se plantea como la simulación de un modelo en la que el objetivo es llevar la energía del sistema a su mínimo (o "congelarla"), esto es, cuando la temperatura es reducida hasta su valor límite. En la simulación, el material es considerado como un sistema de átomos. Para cada posible configuración de los átomos la energía del sistema puede ser calculada. En cada iteración, un átomo está sujeto a pequeños desvíos y la variación de energía se calcula. Si esta última decrece, la nueva configuración es aceptada automáticamente. En caso contrario, esta puede ser aceptada con una cierta probabilidad. El algoritmo simula el proceso de recocido reduciendo la temperatura del sistema repetidamente hasta que este alcance un estado de equilibrio, en el que el sistema queda congelado en su estado fundamental (de mínima energía) [76, 84–86].

## 2.4 Lógica difusa

La lógica difusa fue definida por primera vez en 1965 por el matemático e ingeniero eléctrico Lofti A. Zadeh, según el cual un conjunto difuso es aquel en el que los objetos tienen un grado de pertenencia a un conjunto. Estos conjuntos están caracterizados por una función de membresía la cual asigna a cada objeto un grado de pertenencia que varía entre 0 y 1. Los conjuntos de "números mucho mayores que 1" o de "personas altas" constituirían ejemplos de conjuntos difusos que no se definen en los términos matemáticos habituales. La definición de este tipo de clases juega un importante papel en el pensamiento humano, en particular en el dominio del reconocimiento de patrones, comunicación de información y abstracción [87].

La teoría de conjuntos clásica se construye en base al concepto fundamental de "conjunto" en el que un elemento es un miembro de un conjunto o no lo es. En esta teoría, las distinciones entre miembros y no miembros es nítida e inequívoca y los conjuntos están muy bien definidos, con límites claros que indican si un elemento pertenece o no a ellos. La respuesta a la pregunta de si un elemento pertenece a un conjunto puede ser "sí" o "no". Esto es así tanto en casos deterministas como estocásticos. En estos últimos, la pregunta sería más bien "¿qué probabilidad hay que este elemento pertenezca al conjunto?" y, aunque la respuesta sería del tipo "hay un 90% de probabilidad de que el elemento pertenezca al conjunto", el resultado final sigue siendo "pertenece al conjunto" o "no pertenece al conjunto". La afirmación de que la probabilidad de un elemento de pertenecer a un conjunto es del 90% no significa que el elemento pertenezca a ese conjunto en un 90% y tenga un 10% de no pertenencia al mismo. En resumen, en la teoría de conjuntos clásica, no está permitido que un elemento esté y no esté en un conjunto al mismo tiempo. Por lo tanto, muchos problemas del mundo real no pueden ser descritos y resueltos empleando esta teoría, incluyendo todos aquellos en los que existen elementos que pertenecen parcialmente a un conjunto. En contraste a lo anterior, la teoría de conjuntos difusos acepta pertenencias parciales y, por tanto, constituye una generalización de la teoría de conjuntos clásica [88].

La lógica difusa constituye un método de razonamiento a partir de expresiones lógicas que describen grados de pertenencias a conjuntos difusos [2]. Esta lógica se construye en base a una serie de conceptos como variables lingüísticas, funciones de membresía, sistema difuso, reglas difusas, fuzzificación, defuzzificación e inferencia difusa.

Una variable cuyos valores están definidos con conjuntos difusos recibe el nombre de variable lingüística. Por ejemplo, una variable lingüística empleada para representar la temperatura podría definirse con los valores "fría", "agradable" y "cálida", cada uno de ellos definidos como un conjunto difuso. Estas etiquetas lingüísticas, que son imprecisas por su propia naturaleza, quedan definidas de manera muy precisa con el uso de conjuntos difusos [19].

Como se indicaba anteriormente, en la teoría de conjuntos difusos un elemento forma parte de un conjunto difuso con un determinado grado de pertenencia. Para determinar el grado de pertenencia de un elemento a un conjunto difuso se emplean funciones de membresía. Así, la función de membresía o función de pertenencia de un conjunto difuso es aquella que toma como entrada un valor y devuelve como salida su grado de pertenencia a dicho conjunto difuso [89].

Los sistemas basados en lógica difusa poseen dos características principales que les confieren un mejor rendimiento en aplicaciones específicas:

- Los sistemas difusos son apropiados para el razonamiento incierto o aproximado, especialmente en los sistemas cuyo modelo matemático es difícil de inferir.
- La lógica difusa permite la toma de decisiones a partir de valores estimados bajo condiciones de falta de información o incertidumbre en los datos.

A continuación se presentan los dos paradigmas de empleo de lógica difusa más extendidos en su uso, que a su vez encuentran aplicación en capítulos posteriores.

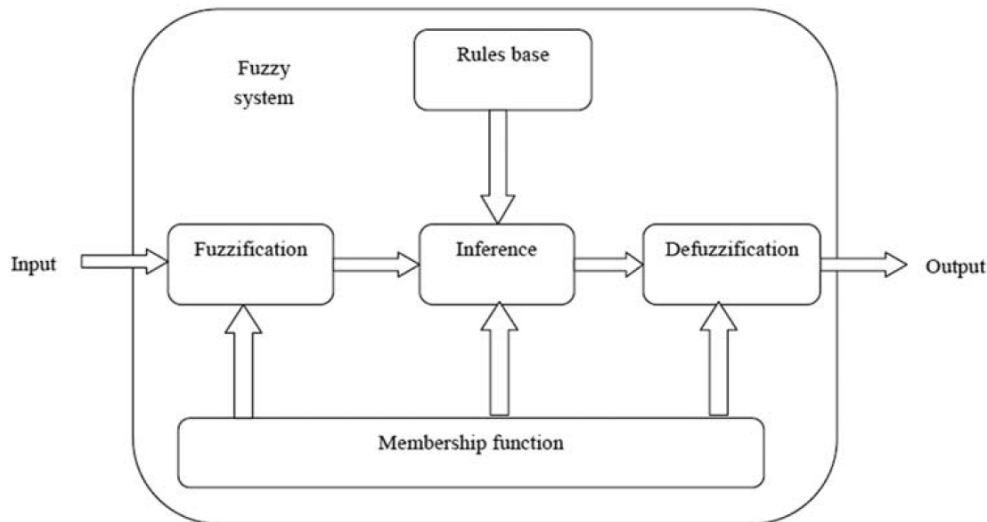
### 2.4.1 Control Difuso

El uso más extendido de la lógica difusa se da en el área de control. Los controladores difusos son capaces de tomar decisiones de control inteligentes en entornos a menudo volátiles y muy cambiantes [90].

Los sistemas basados en reglas difusas, sistemas difusos, o controladores difusos constituyen una extensión de los sistemas basados en reglas que hacen uso de la lógica clásica puesto que emplean reglas del tipo "SI-ENTONCES" (*IF-THEN*) en las que los antecedentes y consecuentes están compuestos por proposiciones difusas en lugar de proposiciones clásicas. Por ejemplo, dado un invernadero en el que se tiene una válvula que se puede abrir o cerrar para modificar la temperatura

del interior, un ejemplo de regla difusa sería: "SI la Temperatura es Alta ENTONCES Abrir la válvula Poco" [89], donde la temperatura y el nivel de apertura de la válvula son variables lingüísticas, y "Alta" y "Poco" representarían conjuntos difusos.

En un sistema difuso típico pueden diferenciarse cuatro partes principales: un fuzzificador, una base de conocimiento, un mecanismo de inferencia y un defuzzificador [91]. En la Figura 2.8 se muestra el esquema típico de un sistema difuso.



**Figura 2.8** Componentes de un sistema difuso [92].

La base de conocimiento consiste en una base de reglas difusas definidas y una base de datos que contiene las definiciones de los términos lingüísticos para cada variable lingüística de entrada y de salida. El fuzzificador o interfaz de fuzzificación transforma los valores numéricos de entrada en valores difusos, aplicando las funciones de pertenencia de estos a cada término lingüístico del correspondiente dominio de entrada. El mecanismo de inferencia ejecuta los procesos de inferencia difusa, lo que consiste en calcular el grado de activación de la salida de cada regla difusa. El defuzzificador o interfaz de defuzzificación calcula los resultados numéricos combinando las salidas de las reglas y aplicando transformaciones específicas [19].

Los sistemas difusos se pueden clasificar en distintas categorías. Los más empleados son el modelo Mamdani [93] y el modelo Takagi-Sugeno [94]. En un sistema difuso tipo Mamdani las salidas son descritas como variables lingüísticas mientras que en un sistema Takagi-Sugeno estas se obtienen como combinaciones lineales de las variables de entrada. La principal fortaleza de los sistemas difusos reside en su habilidad de representar conceptos imprecisos y establecer relaciones entre ellos [19].

### 2.4.2 Clasificación Difusa

Además de en los sistemas de control, los principios de la lógica difusa sirven también de aplicación en problemas de clasificación. Las técnicas de *fuzzy clustering*, también llamadas *soft clustering*, son beneficiosas cuando se tiene un conjunto de datos en el que existen subgrupos cuyos límites son borrosos y existe superposición entre *clusters* [92].

Las técnicas tradicionales de *clustering* (*hard clustering*) imponen como restricción que cada punto del conjunto de datos debe pertenecer a un único *cluster* [95]. Las técnicas de clasificación difusa, sin embargo, asignan a cada elemento un grado de pertenencia a cada uno de los *clusters* [96].

El método más popular de *clustering* difuso es *Fuzzy C-means* (FCM) [97–101]. FCM consiste en una versión difusa del algoritmo *K-means* [96], que permite que los elementos del conjunto de datos pertenezcan a dos o más *clusters*. Este algoritmo se encarga de distribuir en *clusters* los elementos basándose en grados de pertenencia, usando la distancia entre los datos y los centros de los *clusters*. Si un elemento está más cerca del centro de un *cluster*, el grado de pertenencia del elemento aumentará conforme disminuya la distancia a dicho centro. El objetivo de FCM es evaluar los grados de pertenencia de los elementos para reducir las distancias dentro de los *clusters* y aumentar la distancia entre estos [92].

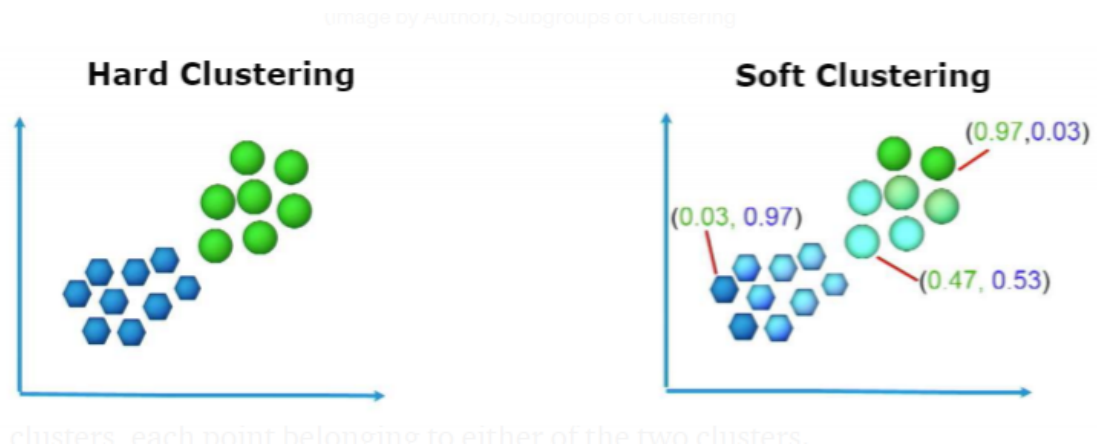


Figura 2.9 Clasificación tradicional frente a clasificación difusa [102].

## 2.5 Procesamiento del Lenguaje Natural

El procesamiento del lenguaje natural (NLP, *Natural Language Processing*) es una serie de algoritmos y técnicas enfocadas a enseñar a las máquinas a entender el lenguaje humano. El desarrollo de NLP supone un gran desafío dada la complejidad y ambigüedad de la estructura del lenguaje humano. Además, el NLP es altamente contextual, dado que el significado de una palabra puede cambiar según la forma de esta, el empleo de sarcasmo y el dominio específico.

La mayoría de modelos de NLP siguen la misma fase de preprocesado: primero el texto de entrada es segmentado en unidades de texto más pequeñas (normalmente palabras), que reciben el nombre de *tokens*, y después estos *tokens* se representan en forma de vectores. Representar palabras en un espacio de menores dimensiones es importante de cara a crear una percepción precisa del grado de similitud y diferenciación entre las palabras [41]. Este último proceso recibe el nombre de *embedding*.

Distintos métodos de *deep learning* han demostrado ser capaces de alcanzar altos niveles de precisión en tareas de NLP. Algunas de las aplicaciones del procesamiento del lenguaje natural más extendidas son clasificación de documentos, análisis de sentimiento, traducción, resumen de textos y respuesta a preguntas [41].

Existen múltiples definiciones del concepto *Big Data*. En [103] se propone la siguiente definición: "*Big Data* es el activo de información que se caracteriza por un volumen, una velocidad y una variedad tan altos que requiere tecnología y métodos analíticos específicos para su transformación en valor". En general, *Big Data* se refiere a conjuntos de datos que no pueden ser percibidos, adquiridos, gestionados y procesados por tecnologías de la información ni software/hardware tradicionales en un tiempo razonable [104].

A diferencia de los datos tradicionales, el término *Big Data* hace referencia a grandes volúmenes de datos que incluyen formatos heterogéneos: datos estructurados, no estructurados y semi-estructurados. Por su compleja naturaleza, el *Big Data* requiere de potentes tecnologías y algoritmos avanzados [105].

El término *Big Data* nació cuando los sistemas de bases de datos relacionales tradicionales no eran capaces de manejar datos no estructurados (blogs, vídeos, fotos, información de redes sociales, etc.) generadas por organizaciones, medios de comunicación y otras fuentes diversas. Se trata en parte de un término de marketing, como la nube o los negocios sostenibles [106].

A continuación se definen las características que definen el *Big Data*, conocidas como las cinco Vs:

- **Volumen:** Grandes volúmenes de datos digitales se generan continuamente desde millones de dispositivos y aplicaciones (TICs, *smartphones*, redes sociales, etc.) [105]. Esta cantidad de información crece diariamente de manera exponencial y es tal que no puede ser tratada, gestionada ni controlada usando métodos tradicionales de gestión de datos [107]. Según el portal de estadística alemán Statista, se estima que alrededor de 2 zettabytes ( $10^{12}$  gigabytes) fueron creados, capturados, copiados y consumidos en el mundo en 2010. En 2020 esta cifra alcanzó los 59 zettabytes y se estima que llegue a 149 zettabytes en 2024 [108].
- **Velocidad:** Los datos se generan de forma rápida y deben ser procesados en tiempo real, casi en tiempo real o en lotes [109].
- **Variedad:** Los datos se generan desde múltiples plataformas y diversas fuentes (vídeos, documentos, comentarios, etc.). Grandes conjuntos de datos están compuestos de datos estructurados y no estructurados, públicos o privados, locales o remotos, compartidos o confidenciales, completos o incompletos, etc. [105]. Esta variedad implica nuevas necesidades de almacenamiento de datos y de diseño de bases de datos que se adapten de manera dinámica a los distintos formatos de datos [109].
- **Veracidad:** Incluye dos aspectos:
  - Consistencia de los datos, que puede definirse como su fiabilidad estadística.
  - Integridad de los datos, que se define mediante varios factores que incluyen el origen de los datos, su recolección y los métodos de procesamiento [109]. En este sentido, se hace

necesario verificar la precisión de los datos eliminando el ruido que puedan contener a través de la aplicación de distintas metodologías [110].

- Valor: Hace referencia a la utilidad de los datos en relación al proceso al que se destinen, y podrá implicar que estos reciban un tratamiento u otro [107], [109]. Este aspecto también se refiere al coste económico que se le puede asignar a los datos, haciendo de estos una "mercancía" que puede ser vendida a terceros para obtener beneficios [110].

Además de las anteriores, se pueden atribuir más características al *Big Data*, tales como exhaustividad, resolución, relacionalidad o extensionalidad [111].

Cuando se hace referencia a tecnologías de *Big Data*, estas suponen un conjunto de métodos destinados a la obtención, procesamiento y transformación en valor de grandes cantidades de datos. Dada la enorme cantidad de datos, resulta difícil procesarlos en un único sistema, ya que los tiempos serían inasumibles e incluso colapsaría el sistema [106]. Por esto, en el ámbito de *Big Data* se emplean métodos de almacenamiento distribuido y computación en paralelo: El almacenamiento distribuido basado en computación en la nube puede gestionar *Big Data* de manera efectiva mientras que la computación en paralelo, en virtud de la computación en la nube, es capaz de mejorar la eficiencia en la adquisición y análisis de *Big Data* [104]. En este contexto, *Hadoop* junto con *MapReduce* [112] y más recientemente *Apache Spark* han emergido como entornos de trabajo (*frameworks*) populares en el análisis de datos a gran escala en la nube [113–115].

*Big Data* e inteligencia artificial son dos campos que pueden formar una simbiosis beneficiosa para ambas partes. Por un lado, los algoritmos de inteligencia artificial, en especial los de *machine learning*, resultan útiles en tareas de clasificación, *clustering*, detección de *outliers* y filtrado de grandes cantidades de datos [116]. De igual manera, las nuevas tecnologías de adquisición, gestión y procesamiento de enormes cantidades de datos dotan a los algoritmos de inteligencia artificial, en especial a los de *deep learning*, de nuevas oportunidades en la resolución de problemas hasta entonces inaccesibles [113, 117].

Existen múltiples aplicaciones del *Big Data* relacionadas con la amplia mayoría de los ámbitos: ciencias de la computación, ingeniería, matemáticas, finanzas, contabilidad, física, astronomía, biología, bioquímica, medicina, ciencias sociales, etc. [110]. Este trabajo se centrará en las aplicaciones de *Big Data* en el ámbito de la gestión óptima de parques fotovoltaicos, en concreto en las aplicaciones propuestas en el Capítulo 1.

## 3 Modelado

---

El modelado constituye la fase inicial que ha de preceder cualquier aplicación de dimensionamiento, identificación o análisis de desempeño de los sistemas de generación fotovoltaica (FV). Los sistemas de generación FV son considerados sistemas complejos no lineales. Debido a la influencia de la variación de las condiciones climáticas en la operación de los mismos, se hace difícil determinar modelos analíticos o numéricos que representen el comportamiento de este tipo de plantas [90, 118].

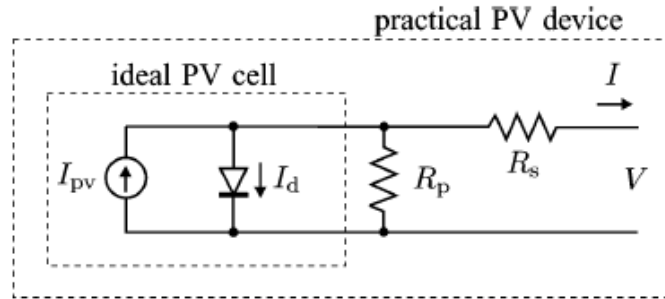
Los métodos de modelado tradicionales están limitados por la pequeña muestra de datos, y se hace difícil considerar la influencia de muchos factores no lineales. Esto hace necesario el uso de técnicas de inteligencia artificial para mejorar el modelado de los sistemas de generación FV, y construir modelos modulares y reutilizables [119, 120]. Además, aplicando estas técnicas se pueden obtener modelos que impliquen una importante reducción en los tiempos de simulación y que incluso puedan servir para desarrollar la aplicación de técnicas de IA en tareas adicionales al propio modelado equivalente.

En este capítulo se lleva a cabo una revisión del estado del arte de la aplicación de técnicas de IA a problemas relacionados con el modelado de plantas FV conectadas a red. En general, las técnicas a revisar aplicarán tanto a la representación de modelos en régimen permanente como ante distintos eventos en régimen transitorio. Se seguirá un orden de menor a mayor nivel de agregación de componentes de plantas FV. Se comenzará por revisar las aplicaciones de IA en la identificación de parámetros a nivel de módulo FV, tras lo cual se abordará la misma tarea a nivel de inversor FV para terminar por estudiar los avances de IA en el modelado a nivel de parque FV conectado a red.

### 3.1 Modelado a nivel de módulo FV

Se entiende por módulo FV un conjunto de paneles FV conectados en serie o en paralelo, que al mismo tiempo están compuestos por células FV las cuales constituyen el dispositivo elemental de un sistema FV. Este conjunto de dispositivos presentan una característica  $I - V$  no lineal con una serie de parámetros que necesitan ser ajustados a través de datos obtenidos experimentalmente [121].

Para caracterizar este comportamiento, el modelo de diodo único (SDM, *single-diode model*) ofrece un buen compromiso entre simplicidad y precisión [122]. Los cinco parámetros que definen el circuito equivalente del modelo SDM, mostrado en la Figura 3.1, son la intensidad fotovoltaica ( $I_{pv}$ ), resistencia serie ( $R_s$ ), resistencia paralelo ( $R_p$ ), corriente de saturación del diodo ( $I_d$ ) y factor de idealidad ( $n$ ) [123]. Estos parámetros pueden ser identificados a partir de las medidas de tensión y corriente de los módulos FV a través de técnicas de identificación de parámetros. La precisa identificación de estos parámetros es importante para reflejar el comportamiento de los módulos FV. Esta tarea puede ser resultado por métodos de optimización basados en IA [124].



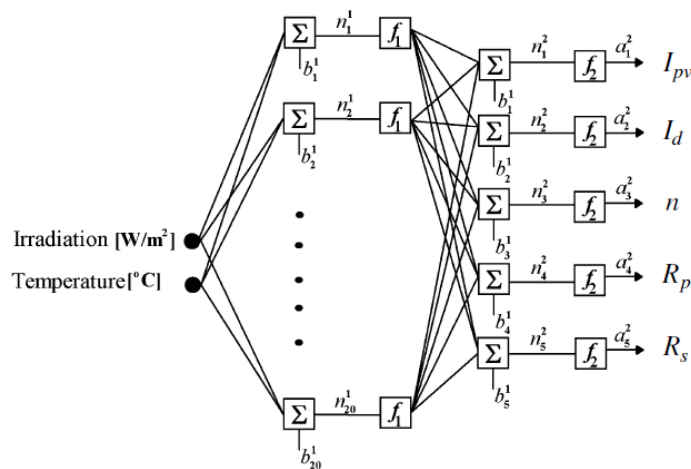
**Figura 3.1** Modelo de diodo único de la célula FV teórica y circuito equivalente de un dispositivo FV incluyendo resistencia serie y paralelo [121].

Con el objetivo de identificar los parámetros del modelo SDM, en [72] se lleva a cabo una comparativa del desempeño de varios algoritmos bioinspirados:

- Algoritmo Genético.
- Evolución Diferencial.
- Optimización por Enjambre de Partículas.
- Algoritmo Colonia de Abejas Artificiales.
- *Cuckoo Search*.

Tras evaluar la identificación de parámetros del modelo SDM aplicando los métodos citados, el estudio concluye que el algoritmo *Cuckoo Search* obtiene los mejores resultados desde el punto de vista estadístico, mostrando mayor rapidez de convergencia y robustez de búsqueda.

En [123] emplea una red neuronal artificial para mejorar la precisión del modelo de circuito equivalente de módulo FV. La ANN toma como valores de entrada la radiación solar y la temperatura, y es entrenada usando curvas  $I - V$  medidas, para así estimar los parámetros del modelo SDM, con los que posteriormente se pueden obtener los valores de intensidad y tensión del modelo. La configuración de la ANN y el módulo FV se muestran en la Figura 3.2 y Figura 3.3. El modelo propuesto mejora en precisión y coste computacional al comparar sus resultados con los obtenidos con otros modelos convencionales.



**Figura 3.2** Configuración de la red neuronal artificial [123].



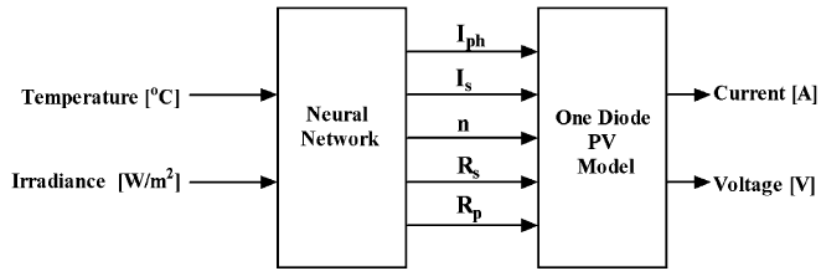


Figura 3.3 Configuración básica del modelo FV [123].

## 3.2 Modelado a nivel de inversor

El inversor fotovoltaico es el componente más crítico de una planta de generación FV, la cual a su vez juega un importante papel en las características dinámicas del sistema de generación en su conjunto. Por tanto, obtener los parámetros del inversor FV de manera precisa es la base para analizar el impacto de la conexión a red de plantas FV [125].

En [126] se propone un método para identificar los parámetros de control proporcional e integral del inversor FV y su inductancia de filtrado, que combina un algoritmo de recocido simulado [84] y optimización por enjambre de partículas [73]. El primero proporciona una búsqueda global más robusta y puede evitar caer en óptimos locales y el segundo implica una mayor velocidad de convergencia, simplicidad y bajo requerimiento de datos de origen.

El método propuesto compara los parámetros calculados con los previamente obtenidos en campo y obtiene menores errores relativos en comparación con la aplicación de otros algoritmos bioinspirados. La Tabla 3.1 muestra estos errores para los distintos parámetros de control e inductancia de filtrado empleando diferentes algoritmos bioinspirados.

**Tabla 3.1** Comparativa de errores relativos (%) obtenidos en identificación de parámetros empleando distintos algoritmos bioinspirados [126].

Parámetro	PSO	SA	GA	GA + PSO	SA + PSO
$k_{PU}$	0.2143	0.1886	10.2157	0.1043	0.0014
$k_{IU}$	0.6857	11.3249	6.8140	0.0821	0.0867
$k_{PI}$	0.1000	150.1000	333.6000	0.0333	0.0000
$k_{II}$	0.2080	8.1015	3.1045	0.0440	0.0360
$L$	9.1775	63.2397	55.3159	2.3069	0.3511

En [125] se propone un algoritmo mejorado que combina algoritmo genético y optimización por enjambre de partículas. Este método reúne las ventajas de GA, que destaca por una gran capacidad de búsqueda global; y PSO, a saber: rapidez de búsqueda, alta eficiencia de convergencia, algoritmo simple y bajo requerimiento de datos de entrada. A diferencia de [126], en [125] se identifican los parámetros de control de los ejes  $d$  y  $q$  por separado en lugar de suponerse idénticos. Aún con ello los errores relativos obtenidos resultan ser superiores a los mostrados para el método anterior.

## 3.3 Modelado a nivel de planta

Las grandes plantas FV están normalmente compuestas por docenas o incluso cientos de unidades FV. Debido a el gran volumen de datos de modelos, estructuras complejas, lenta velocidad de simulación y gran cantidad de cálculos, modelar el esquema detallado de la planta no resulta práctico. Por tanto,

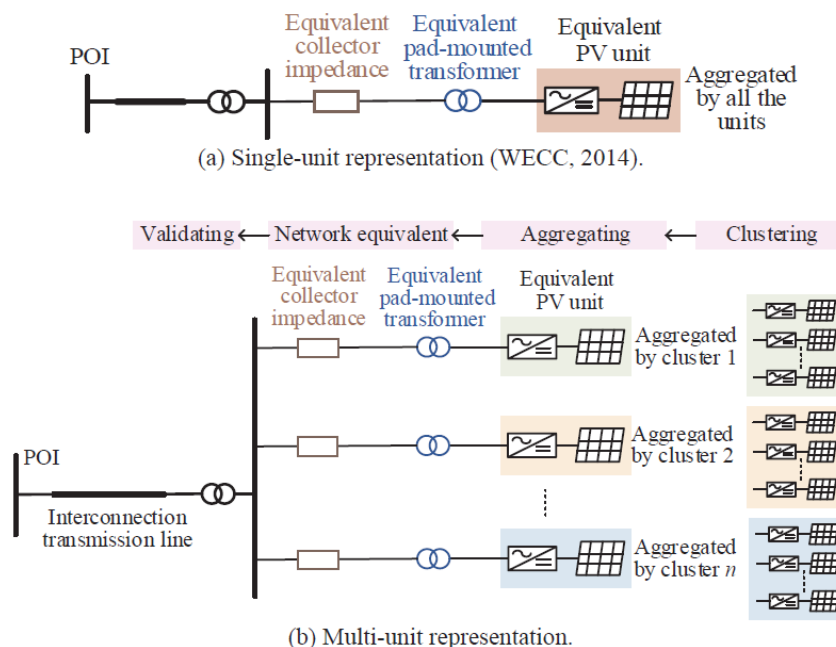
es de gran interés el estudio y establecimiento de modelos equivalentes de planta FV en lugar de modelos en detalle, para su uso en simulación y análisis con fines prácticos [127–129].

Establecer un modelo equivalente de planta FV consiste en simplificar el modelo en detalle de la planta con múltiples unidades FV en un modelo equivalente con solo una o unas pocas unidades FV [130]. Las características de la respuesta dinámica de la planta en el punto de conexión a red son consistentes antes y después del modelado equivalente. El tradicional modelo de máquina única equivalente equipara todas las unidades FV en una sola, lo que implica ignorar las distintas características de operación de cada unidad FV de la planta y resulta en un error equivalente considerable [131].

En plantas FV de gran escala, los tipos de inversores pueden variar. Además de esto, las condiciones del entorno de las unidades FV pueden ser diferentes. Aún cuando los tipos y parámetros de los inversores coinciden, las condiciones de operación pueden ser distintas [130]. Por tanto, es necesario agrupar todas las unidades FV de acuerdo a las diferencias internas de la planta y establecer un modelo multi-máquina para satisfacer los requerimientos de rapidez de simulación y precisión. El núcleo del modelado equivalente multi-máquina es la selección de los índices de agrupamiento y el empleo de los métodos de *clustering* [132].

Los pasos a seguir para el modelado equivalente de una planta FV se pueden resumir en cuatro puntos [133]:

- Agrupar las unidades FV. Para ello se deben definir índices de agrupamiento razonables para identificar la coherencia entre unidades FV, y emplear métodos de *clustering* para dividir estas unidades en diferentes grupos basados en los índices definidos.
- Agregar las distintas unidades FV pertenecientes a un mismo grupo, para lo cual deben identificarse los parámetros y la estructura del modelo agregado.
- Asignar el sistema colector de cada unidad FV equivalente.
- Validar el modelo equivalente para probar su efectividad.



**Figura 3.4** Modelo equivalente de planta FV de gran escala conectada a red [133].

Esta sección se centrará en el empleo de técnicas de IA para llevar a cabo los dos primeros puntos.

En [134] se estudia el agrupamiento de grandes plantas FV conectadas a red aplicando el método de *clustering k-means*. Este agrupamiento se realiza a nivel de unidad de generación FV (Figura 3.5) de acuerdo a los parámetros de irradiancia, temperatura y potencia activa generada en tiempo real de cada unidad. Dado un número de *clusters*  $k$ , el algoritmo agrupa las distintas unidades de generación FV en  $k$  unidades equivalentes, tal y como se muestra en la Figura 3.6), calculando los parámetros equivalentes de todos los elementos que componen estos grupos y asignando valores de desviación para el comportamiento dinámico de estos frente al comportamiento de la planta en detalle.

Finalmente se evalúa mediante simulación el comportamiento dinámico de los modelos equivalentes resultantes para distinto número de *clusters* ante distintos eventos, comparándolos con el comportamiento del modelo en detalle de la planta. En la Figura 3.7 se muestra el resultado de esta comparativa ante un evento de caída de tensión.

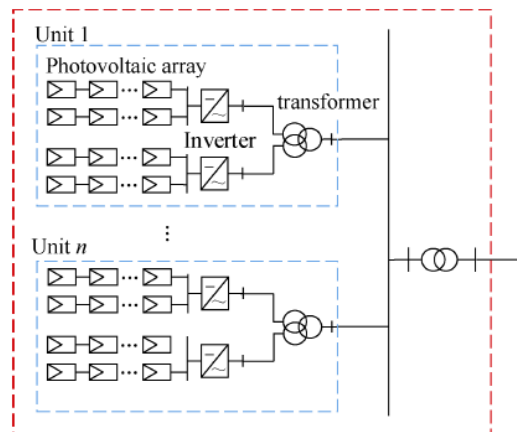


Figura 3.5 Estructura típica de planta de generación FV [27].

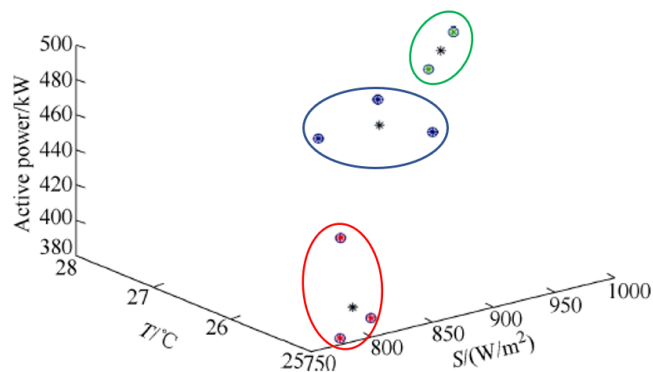
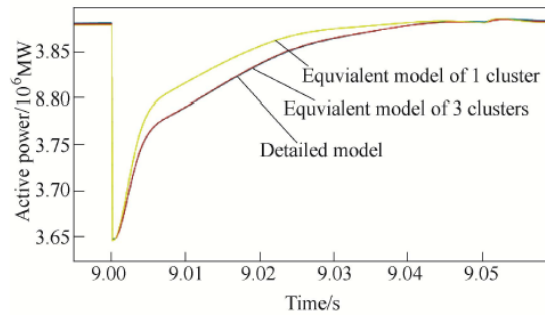


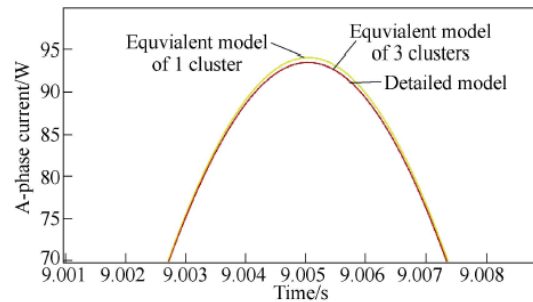
Figura 3.6 Mapa tridimensional de parámetros para el agrupamiento de *k-means* [27].

Siguiendo esta línea, en [135] se propone un modelado equivalente de grandes plantas FV conectadas a red por medio de agrupamientos a nivel de inversor. En esta ocasión se toman como índices de agrupamiento los parámetros de control de los inversores y se se lleva a cabo nuevamente un algoritmo de clasificación de *k-means*. Sin embargo, en este estudio se actualiza la ejecución del algoritmo de clasificación en caso de que varíe la generación de potencia por parte de cualesquiera de los paneles FV que componen la planta (*clusterización online*).

En la Figura 3.8 se compara la respuesta dinámica ante una falta trifásica en el punto de conexión a red del modelo equivalente con respecto al original antes (azul) y después (rojo) de actualizar el



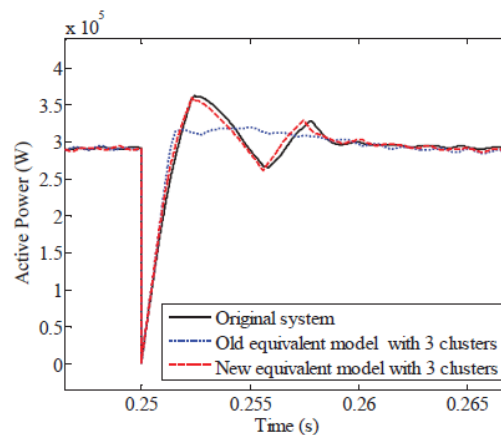
(a) Comparison of active power in PCC



(b) Comparison of a-phase current in PCC

**Figura 3.7** Comparativa del comportamiento dinámico de los modelos equivalentes ante caída de tensión [27].

resultado del algoritmo de clasificación tras variar la generación de potencia activa de los paneles FV, modelando un cambio en la irradiancia de los mismos.



**Figura 3.8** Respuesta dinámica de los modelos equivalentes antes y después de actualizar la clasificación [135].

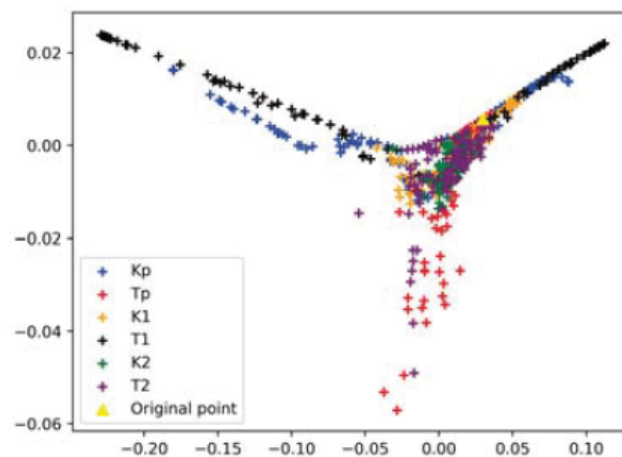
En [136] se propone otro ejemplo de aplicación del algoritmo de clasificación *k-means online*, pero esta vez tomando como índice de clasificación los parámetros de control de *low voltage ride-through* (LVRT) de los inversores.

En [137] se pone de manifiesto la necesidad de preprocesar los datos de entrada al algoritmo de clasificación a fin de evitar óptimos locales y reducir el coste computacional. A este fin, aplica un algoritmo *Canopy* de pre-clasificación para obtener el número de *clusters* necesarios y sus centros iniciales. *Canopy* es un algoritmo no supervisado de *pre-clustering* cuyo objetivo es acelerar las operaciones de *clustering* en problemas con grandes conjuntos de datos. Esto se consigue dividiendo

los datos iniciales en subgrupos superpuestos [138, 139]. Tras pre-clasificar los datos empleando algoritmo *Canopy*, se aplica un algoritmo de clasificación *Fuzzy C-Means* [101]. Los resultados de tiempos de simulación y errores en potencia activa, reactiva y tensión ante distintos eventos en el punto de conexión a red revelan una gran reducción en tiempo de simulación y pequeños márgenes de error al aplicar algoritmos de clusterización para modelado equivalente de planta, y la mejora que supone en los mismos el preprocesado de los datos mediante algoritmo *Canopy*.

Siguiendo esta misma filosofía, en [132] se aplica un análisis factorial previo al algoritmo de clasificación con objeto de reducir la dimensión de las variables iniciales empleadas como índices de clusterización. Tras este análisis, se ejecuta un algoritmo de *Correlation Clustering* y test de significación estadística, este último para evaluar estadísticamente los resultados obtenidos en el proceso de *clustering*. Los resultados obtenidos en cuanto a errores con respecto al modelo en detalle y tiempos de simulación se comparan con el modelo de máquina única y se observa que la metodología propuesta disminuye ambos indicadores en varios órdenes de magnitud. Además, se compara el desempeño de esta estrategia antes y después de aplicar análisis factorial, comprobándose que su aplicación implica un gran descenso en los errores de modelado a cambio de un pequeño incremento en el tiempo de simulación.

Con objeto de identificar los parámetros clave para el modelado de grandes plantas fotovoltaicas, [140] aplica un algoritmo de optimización por enjambre de partículas. Previo a ello y con el fin de determinar cuáles de los numerosos parámetros que definen el comportamiento de una planta FV tienen una mayor influencia sobre el mismo y con esto reducir la dimensión de los datos del problema, se lleva a cabo un análisis de componentes principales. La visualización de este análisis se representa en la Figura 3.9, donde se observa que ante una pequeña perturbación la modificación en los valores de ciertos parámetros de control alejan al sistema del estado inicial en mayor medida que otros, lo que los define como parámetros clave a identificar para el modelado de la planta. En el caso mostrado en la figura, se puede ver que la constante de tiempo integral del control de potencia activa ( $T_p$ ) y la de el control de la componente activa del control de intensidad ( $T_1$ ) influyen en el comportamiento del sistema en mayor medida que el resto.



**Figura 3.9** Visualización de los resultados de la aplicación de PCA para la reducción de la dimensión de datos [140].

### 3.4 Resumen

En este capítulo se han revisado las aplicaciones de técnicas de IA al modelado de plantas FV a distintos niveles de agregación.

A nivel de módulo e inversor FV las tareas de modelado consisten en identificar los parámetros que definen estos elementos, minimizando los errores con respecto a los valores reales de campo y teniendo presente el coste computacional en el que se incurre al aplicar los diversos métodos estudiadas. En este área predomina el empleo de algoritmos bioinspirados, que obtienen una alta precisión en la identificación de parámetros de módulo e inversor FV.

Para la obtención de modelos equivalentes de planta FV, destaca el empleo de técnicas de *clustering*. Estos modelos equivalentes son clave para la reducción de tiempos de simulación, lo que puede servir de base para la construcción de herramientas más complejas que requieran de la ejecución de un número elevado de simulaciones en intervalos de tiempo reducidos. Se ha visto además cómo las técnicas de preprocesado de datos mejoran la precisión de los modelos obtenidos. Algunos de los trabajos revisados proponen métodos *online* que son actualizados ante cambios en las condiciones de la planta, si bien no todos presenta esta característica.

A continuación, en la tabla Tabla 3.2 se resumen las aplicaciones de técnicas de IA al modelado de plantas FV a distintos niveles de agregación revisadas en este capítulo:

**Tabla 3.2** Resumen de aplicaciones de técnicas de IA a modelado de plantas FV.

<b>Método</b>	<b>Modelado a nivel de módulo</b>	<b>Modelado a nivel de inversor</b>	<b>Modelado a nivel de planta</b>
GA	[72]		
DE	[72]		
PSO	[72]	[126], [125]	[140]
ABC	[72]		
ANN	[123]		
SA		[126]	
GA		[126], [125]	
k-means			[134], [135], [136]
FCM			[137]
FA			[132]
CC			[132]
PCA			[140]

## 4 Control

---

En este capítulo se describe el estado del arte de la aplicación de técnicas de IA a problemas relacionados con el control de plantas FV conectadas a red. Se han abordado cuatro paradigmas de control que tienen como objetivo contribuir al correcto funcionamiento de la red externa a la que se conecta la planta, es decir, qué estrategias pueden seguir las plantas FV para controlar sus aportaciones a la red en su punto de conexión.

### 4.1 Limitación de Potencia Activa

En los últimos años, con el aumento de penetración de los sistemas FV, han aumentado significativamente las fluctuaciones de potencia provocadas por la intermitencia y aleatoriedad propias de la generación FV. Esto afecta negativamente la calidad e incluso la estabilidad del sistema eléctrico [141].

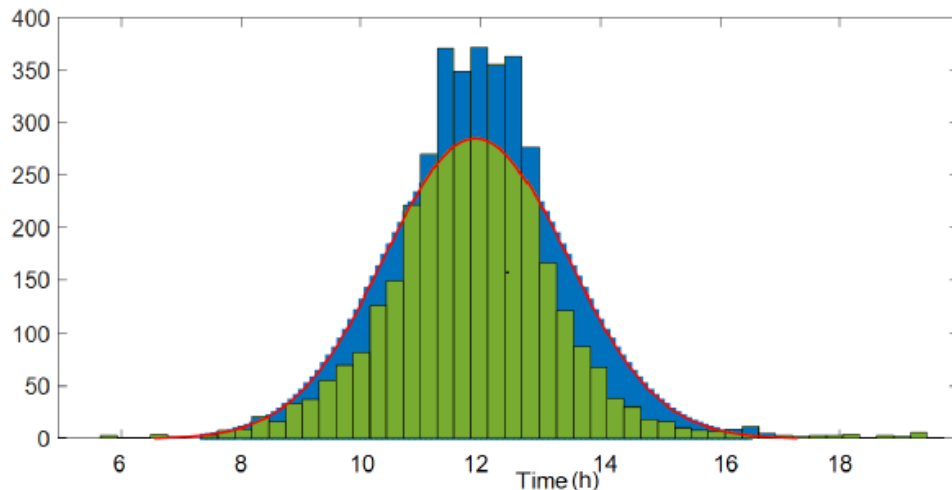
Limitar la potencia activa suministrada por las plantas FV ayuda a mitigar los problemas de fluctuaciones de frecuencia, tensiones y potencia que ocasiona la gran penetración de generación FV en la red. Para ejecutar un control directo sobre la inyección de potencia activa de una planta FV, esta debe estar equipada con un sistema de almacenamiento de energía eléctrica [142]. En este contexto, con el empleo de sistemas de almacenamiento junto con algoritmos de control basados en IA es posible suavizar la inyección de potencia activa que las plantas FV inyectan a la red [143].

En [141] se propone un método de suavizado de inyección de potencia activa para plantas FV con volante de inercia basado en control difuso. Los resultados obtenidos en simulación muestran que el método basado en lógica difusa reduce en un 80 % el nivel de fluctuación de potencia activa en la red en comparación con el método de control directo sobre el sistema de almacenamiento tradicional y en un 95 % con respecto al caso sin almacenamiento de energía.

Por su parte, [143] analiza la viabilidad técnica del empleo de redes neuronales artificiales para suavizar la inyección de potencia activa de una planta FV con almacenamiento conectada a red. La ANN recibe como entradas datos de temperatura, velocidad de viento, presión, radiación y humedad y predice la generación FV para un día completo. En base a esta predicción, el sistema de almacenamiento absorbe e inyecta energía para suavizar la inyección de potencia activa a la red. En este estudio se analizan distintas topologías de ANN, siendo mejores los resultados cuanto menor es la complejidad de ésta. La Figura 4.1 muestra los resultados obtenidos empleando una ANN formada por 10 neuronas y 1 capa oculta.

### 4.2 Control de frecuencia

En los sistemas de potencia tradicionales, la frecuencia está regulada principalmente por los generadores síncronos, que poseen inercia mecánica y regulación primaria para lidiar con eventos



**Figura 4.1** Resultados de suavizado de inyección de potencia. En verde, la predicción de generación FV de la ANN, en rojo la curva óptima de generación y en azul la inyección/absorción requerida al sistema de almacenamiento [143].

de frecuencia [144]. Cuando se produce un gran aumento o disminución del nivel de carga, los generadores síncronos regulan sus rotores para disminuir el desequilibrio entre generación y consumo de la red, y de esta forma mitigar la desviación de la frecuencia del sistema. Sin embargo, cuando los generadores síncronos son reemplazados por sistemas de generación distribuida basados en electrónica de potencia (como las plantas FV), si no se toman medidas adicionales, estos últimos no responderán a los desequilibrios de carga como los primeros, y por tanto el sistema puede perder su capacidad de reducir las oscilaciones de frecuencia. Es por esto que resulta imprescindible que los sistemas de generación basados en electrónica de potencia, incluidas las plantas FV, contribuyan al control potencia-frecuencia a través de sus sistemas de control [145].

[146] presenta una estrategia de control basada en lógica difusa [147] para ajustar apropiadamente la inyección total de potencia activa a la red local por parte de una planta FV de gran escala con almacenamiento de energía, con el fin de contribuir en la regulación de frecuencia de la red.

Esta estrategia consiste en un controlador central de planta y una serie de controladores locales a nivel de agente FV. En ella, el controlador central emplea un módulo de regulación de frecuencia basado en lógica difusa para determinar el valor referencia de la inyección total de potencia activa de la planta y coordinar los valores de las referencias individuales de los controladores locales de cada agente FV. Al mismo tiempo, cada controlador local dirige todos los componentes de electrónica de potencia y convertidor de batería instalado en cada agente FV para suministrar potencia de acuerdo a la consigna mandada por el controlador central.

En dicho estudio, el controlador basado en lógica difusa tiene tres objetivos principales a lograr:

- La potencia activa total evacuada por la planta debe estar ajustada para asegurar que la frecuencia de la red se mantenga dentro de límites aceptables ( $\pm 0.05 Hz$  en régimen permanente y  $\pm 0.2 Hz$  en régimen transitorio).
- En cada agente FV local, el estado de carga del banco de baterías debe permanecer en el rango seguro  $[0.2 - 0.8]$  para mejorar la durabilidad de los mismos.
- Los dos objetivos anteriores deben cumplirse incluso en caso de variación abrupta de carga de la red o radiación solar.

Los resultados obtenidos en simulación muestran una eficacia sobresaliente de la estrategia de control propuesta, incluso en el caso de variación repentina de carga de la red o radiación solar.

Sin embargo, la instalación y mantenimiento de sistemas de almacenamiento puede suponer un coste económico que supere los beneficios que implica la regulación de frecuencia. Además, los



elevados ciclos de carga y descarga pueden acortar la vida útil de estos sistemas [148]. Teniendo esto en cuenta, [149] propone una estrategia de control basada en lógica difusa para sistema FV conectado a red, que permite participar en la regulación de frecuencia sin necesidad de contar con sistema de almacenamiento. En los períodos de alta generación FV, el sistema FV cambia de modo de control *Maximum Power Point Tracker* (MPPT) a operar por debajo de su punto máximo para poder dar respuesta primaria ante variaciones de frecuencia. En esos períodos, el controlador difuso recibe como entradas la desviación de frecuencia y variación de inercia del sistema, y calcula el punto de operación para dar respuesta primaria.

La estrategia anterior es evaluada en la red del IEEE de 14 nudos dividida en tres áreas, mostrado en la Figura 4.2, que incluye distintos tipos de generación además de FV. La Figura 4.3 muestra la variación de frecuencia del sistema ante un escalón de carga de  $0.05 pu$  en el área 1, comparando la estrategia propuesta en este estudio con la desarrollada en [150], en la que se emplea un controlador proporcional-integral para ajustar el punto de operación FV. Se comprueba que el controlador basado en lógica difusa es capaz de dotar a las plantas FV de capacidad para aportar respuesta primaria sin necesidad de instalar sistemas de almacenamiento.

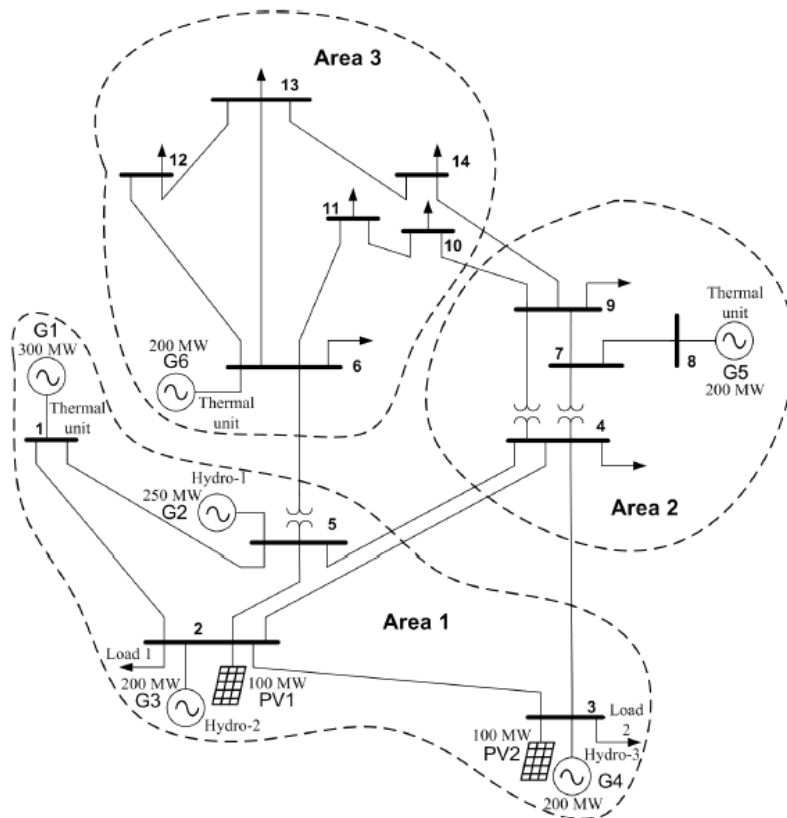
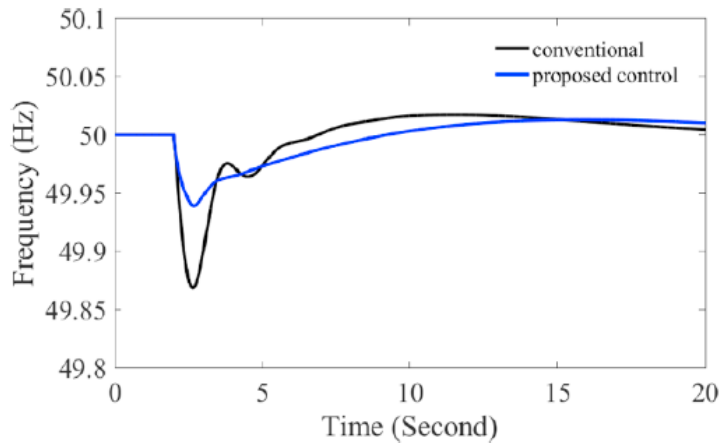


Figura 4.2 Red de 14 nudos del IEEE empleada en [149].

Además de lo anterior, se lleva a cabo un análisis económico de la estrategia propuesta, que demuestra una mayor viabilidad económica en comparación con otros métodos que requieren una disminución en la generación FV a lo largo del día completo en lugar de exclusivamente en los momentos de mayor generación [151–153].



**Figura 4.3** Variación de frecuencia ante escalón de carga para distintos modos de operación PV [149].

### 4.3 Control de tensiones

El gran incremento de la penetración de la generación distribuida variable, como la generación FV, experimentada en los últimos años, puede causar problemas de tensión en los sistemas de distribución [154]. Generalmente, los operadores de la red de distribución confían en cambiadores de tomas en carga y bancos de condensadores fijos o conmutables para mantener perfiles de tensión adecuados a lo largo de la red. Sin embargo, estas tecnologías están limitadas por su número y velocidad de operaciones, y son insuficientes para adaptarse a la gran variabilidad de la producción FV y su impacto en las tensiones.

El estándar IEEE 1547 [155] permite a la generación FV con inversores inteligentes (*Smart Inverters*, SI) participar en la regulación de tensión mediante diversas funcionalidades inteligentes. Éstas incluyen reducción de la inyección de potencia activa (Volt-Watt) e inyección y absorción de reactiva (Volt-Var).

Comparado con las tecnologías clásicas de regulación de tensión, los SIs proporcionan una respuesta más rápida al cambio de condiciones en la red para la regulación de tensión. Sin embargo, las funcionalidad Volt-Var, comúnmente usada a este fin, está basada en una curva de droop de tensión  $Q - V$  local en la cual cada SI absorbe o inyecta potencia reactiva basándose en la tensión de su nudo de regulación local. Esto puede resultar en una regulación de tensión subóptima dado que no existe coordinación entre los diferentes SIs [156].

Con objeto de mejorar la regulación de tensión en redes de distribución con alta penetración de plantas FV, [157] y [158] proponen métodos multi-agente basados en aprendizaje profundo reforzado (*multi-agent deep reinforcement learning*, MADRL). En ellos, los inversores FV (agentes) pueden aprender estrategias de control coordinadas a partir de datos históricos.

[158] supone una mejora a la metodología aplicada en [157] al presentar una mejor escalabilidad del algoritmo a un mayor número de agentes.

La Tabla 4.1 muestra una comparativa de los valores medios y máximos de desviación de tensión obtenidos aplicando distintas estrategias de regulación de tensión sobre la red de 126 nudos del IEEE [159]. Los resultados muestran que las estrategias basadas en IA suponen una mejora con respecto a otras como control  $Q - V$  (droop de tensión) clásico. El método de control centralizado representa el límite teórico del problema, ya que asume condiciones de comunicación perfectas y que los datos de demanda y generación de potencia activa de los generadores FV son conocidos en todo momento.

Otra ventaja de la metodología propuesta en [158] es que puede alcanzar el control coordinado usando únicamente información local, lo que reduce el coste en enlaces de comunicación.

**Tabla 4.1** Comparativa de desviaciones de tensión (%) obtenidos empleando distintas estrategias de regulación de tensión [158].

Método	Media	Máxima subida (%)	Máxima caída (%)	Tiempo de cálculo (s)
Original (sin control)	1.89	7.92	4.33	—
<i>Droop</i>	1.26	3.79	4.67	0.0002
Propuesto en [157]	0.58	1.63	3.85	0.0012
Propuesto en [158]	0.53	1.58	3.76	0.0012
Centralizado	0.42	1.52	3.61	1.32

Por su parte, [156] aplica una estrategia de coordinación entre SIs basada en aprendizaje profundo por refuerzo aplicada a la regulación de tensión en redes de distribución. Los resultados de aplicar esta metodología sobre la red de 37 nudos del IEEE [160] se comparan con los de la estrategia de control Volt-Var convencional en la Tabla 4.2 para un período de un año. Se puede apreciar que el método propuesto evita sub y sobretensiones en la red al mismo tiempo que reduce las pérdidas del sistema y la necesidad de reducir la producción de energía.

**Tabla 4.2** Resultados de aplicación de metodología propuesta en [156] para la regulación de tensión.

Método	Nº de subtensiones	Nº de sobretensiones	Reducción de energía FV (kWh)	Pérdidas del sistema (kWh)
Original (sin control)	630	38201	0	394131
Volt-Var	0	0	14610	440187
Propuesto en [156]	0	0	3330	396015

## 4.4 Low-voltage Ride Through

Convencionalmente, de acuerdo a los distintos códigos de red, ante huecos de tensión de cierta profundidad provocados en condiciones de falta se requiere que los sistemas FV se desconecten de forma inmediata para reducir su impacto en la red [161]. Sin embargo, dada la gran capacidad de energías renovables en la red, la desconexión o aislamiento de las plantas FV puede afectar a la estabilidad y fiabilidad del sistema. Además, la desconexión de los sistemas FV no es viable económicamente e implica grandes pérdidas de potencia. Es por esto que muchos países han establecido nuevos requerimientos en sus códigos de red con el objetivo de que las plantas FV permanezcan conectadas a red durante eventos de falta. Esta habilidad de las plantas FV de permanecer conectadas ante huecos de tensión recibe el nombre de capacidad *low voltage ride-through* (LVRT) [162]. Entre los distintos códigos de red, la capacidad de respuesta LVRT es de particular importancia junto con el apoyo a la dinámica de tensiones a través del aporte de potencia reactiva requerida ante eventos de falta en la red [163, 164].

En este contexto, [165] emplea una técnica basada en PSO sobre el MPPT de un inversor FV para mejorar la capacidad LVRT de una planta de pequeña escala conectada a red. Los resultados obtenidos en simulación demuestran una respuesta dinámica más rápida y una mejora del régimen permanente ante faltas severas en la red en comparación con técnicas tradicionales.

Por su parte, [166] presenta una metodología de control avanzado para mejorar la respuesta ante LVRT de plantas FV de gran escala conectadas a red. En ella, se implementa un controlador basado en lógica difusa junto con el MPPT convencional [167], para la inyección controlada de potencia activa en caso de falta en la red. La respuesta obtenida mejora a la estrategia de control convencional, destacando los siguientes puntos:

- La estrategia presentada limita la subida de corriente de falta transitoria.
- Se suministra potencia reactiva a la red durante el período de subtensión, lo cual es de gran importancia para la dinámica de tensiones.
- El inversor FV es capaz de inyectar corriente de manera constante, lo que puede evitar el disparo de protecciones de subtensión de la planta.

Pese a lo anterior, dicha metodología precisa de mejora en la limitación de la tensión del bus de continua, al marcarse el límite en  $1.1 pu$ . [168] propone una estrategia de control de potencia activa y reactiva basada en lógica difusa, que obtiene una mejora de los resultados de control de corriente y sobretensión.

## 4.5 Resumen

En este capítulo se han revisado las aplicaciones de técnicas de IA a cuatro paradigmas de control de plantas FV cuyo objetivo es contribuir al correcto funcionamiento de la red externa. Se ha podido comprobar que estas técnicas obtienen muy buenos resultados a la hora de resolver los problemas que se presentan en estas tareas a pesar de la gran complejidad que éstos pueden llegar a suponer.

Destaca el uso de estrategias de control basadas en lógica difusa, que encuentran aplicación en problemas de limitación de potencia activa, control de frecuencia y respuesta LVRT con una eficacia superior a los métodos hasta ahora más extendidos para la resolución de estos problemas. Por su parte, las técnicas de aprendizaje reforzado para sistemas multi-agente cobran protagonismo en los problemas de control de tensiones, dadas las características propias de este paradigma de control, mejorando también los resultados en comparación con los métodos de tradicionales. Estas mejoras se traducen en reducción de pérdidas del sistema, mayor viabilidad económica de las soluciones propuestas, y, en general, mejora de la integración de las plantas fotovoltaicas en las redes de transporte y distribución.

A continuación, en la tabla Tabla 4.3 se resumen las aplicaciones de técnicas de IA a los cuatro paradigmas de control de plantas FV estudiados:

**Tabla 4.3** Resumen de aplicaciones de técnicas de IA a control de plantas FV.

<b>Método</b>	<b>Limitación de potencia activa</b>	<b>Control de frecuencia</b>	<b>Control de tensiones</b>	<b>LVRT</b>
Control difuso	[141]	[146], [149]		[166], [168]
ANN	[143]			
MADRL			[157], [158], [156]	
PSO				[165]

# 5 Detección y diagnóstico de falta.

## Mantenimiento predictivo

---

En este capítulo se describe el estado del arte de la aplicación de técnicas de IA y *Big data* en la detección y diagnóstico de falta y en el mantenimiento predictivo en plantas FV.

### 5.1 Detección y diagnóstico de falta

Los sistemas fotovoltaicos están expuestos a diferentes tipos de fallos, que pueden producirse en células, módulos, diodos de protección, conexiones entre módulos, *strings* e inversores [118]. Las faltas producidas en cualquier componente de un sistema FV pueden afectar seriamente a la eficiencia, la producción de energía, la seguridad y la fiabilidad de toda la planta, en caso de no ser detectadas y corregidas rápidamente. Existe además riesgo de fuego en caso de que ciertos tipos de faltas (como las faltas de arco o las de línea-línea) persistan en el tiempo [169]. En el año 2018, las pérdidas de potencia solar debidas a la aparición de faltas registraron un 17.5 % del total de la producción de potencia [170].

La detección y diagnóstico de falta en plantas FV conectadas a red constituye una tarea fundamental para proteger los componentes del sistema FV (módulos, baterías e inversores) de sufrir daños y eliminar posibles riesgos de fuego [171, 172]. La principal función de la detección de faltas en sistemas FV consiste en comparar la diferencia entre los valores medidos y calculados de ciertos parámetros del sistema con valores de referencia, con el fin de verificar la ocurrencia de cualquier falta. Por su parte, el diagnóstico de falta se centra en identificar el tipo de falta y localizarla basándose en conocimiento *a priori* o empleando técnicas de búsqueda [173]. La localización de faltas sigue suponiendo un gran reto, particularmente en plantas FV de gran escala [174].

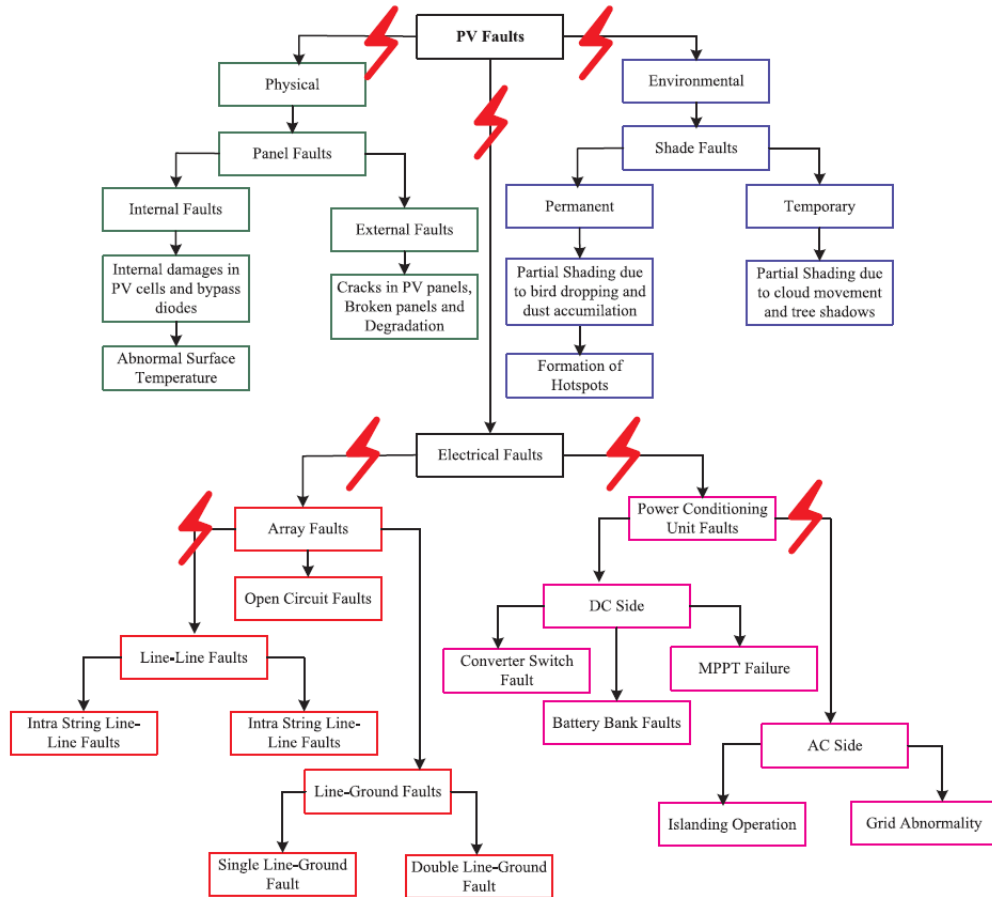
Con objeto de operar una planta FV de manera fiable, eficiente y segura, es indispensable contar con un sistema de monitorización que incluya algún método de diagnóstico automático [175]. La principal tarea del sistema de monitorización incluye la monitorización de datos meteorológicos y eléctricos, a fin de controlar y supervisar la planta FV en tiempo real. Los sistemas de monitorización integran técnicas de Internet de las Cosas (IoT, *Internet of Things*), para comprobar el comportamiento y la evolución del sistema en tiempo real [9].

Las faltas producidas en sistemas FV pueden clasificarse atendiendo a distintos criterios:

- Según su severidad puede distinguirse entre faltas agudas (faltas de cortocircuito o de arco) y crónicas (como las debidas al sombreado o la degradación de componentes) [176].
- Atendiendo a si son permanentes (requieren el reemplazo del módulo FV fallido, como en caso de delaminación) o temporales (pueden ser eliminadas por operadores sin necesidad de reemplazar el equipo, como en caso de ensuciamiento) [9].

- Según su naturaleza pueden distinguirse faltas físicas, ambientales o eléctricas [175, 177].
- Según si ocurren en el lado de corriente continua o alterna del sistema FV, las faltas se pueden categorizar según su localización y estructura. [178, 179]

La Figura 5.1 muestra una posible clasificación de los distintos tipos de falta que pueden producirse en un sistema FV [180]:



**Figura 5.1** Clasificación de faltas en sistemas FV [180].

Numerosos métodos para la detección y diagnóstico de faltas han sido propuestos en la literatura. Las principales propiedades que caracterizan a estos métodos son: velocidad para detectar averías, datos de entrada necesarios (climatológicos y eléctricos) y selectividad (habilidad para distinguir entre diferentes tipos de faltas) [169]. De manera global, estos métodos pueden clasificarse en dos categorías principales [181]:

- Métodos eléctricos: se pueden emplear en la detección y diagnósticos de módulos FV, *strings* y *arrays* fallidos incluyendo faltas por arco eléctrico, faltas línea-línea, faltas a tierra, en diodos, etc.
- Métodos visuales y térmicos: usados para detectar decoloración, bronceado, ensuciamiento de superficie, puntos calientes, rotura y delaminación.

### 5.1.1 Métodos eléctricos

La mayor parte de los métodos eléctricos de detección y diagnóstico de falta se valen de un modelo de sistema FV para detectar varios tipos de falta. A su vez, estos métodos pueden clasificarse en tres grupos [9]:

- Métodos estadísticos y de procesamiento de señales: se basan principalmente en el análisis de las señales de forma de onda.
- Métodos basados en el análisis de la característica  $I - V$ : se centran en el análisis de las curvas  $I - V$  de los *arrays* FV.
- Métodos basados en técnicas de inteligencia artificial: En ellos resulta crucial disponer de un sistema de monitorización para coleccionar distintos datos, como información meteorológica y datos eléctricos de los sistemas FV.

En esta subsección se presentarán distintas aplicaciones de técnicas de IA para la detección y diagnóstico de faltas en sistemas FV conectados a red empleando métodos eléctricos.

Analizando cambios en las curvas  $I - V$ , [182] desarrolla una metodología de diagnóstico de falta para detectar sombreado, degradación producida por potencial e incremento de las pérdidas por resistencia serie de *strings* FV. En ella, se emplea lógica difusa para evaluar las reglas de diagnóstico, lo que aporta robustez y flexibilidad a la misma. En concreto, se aplican tres clasificadores difusos [183, 184], uno por cada tipo de falta a evaluar.

[185] compara el desempeño de cuatro redes neuronales de función de base radial con distintas estructuras para detectar faltas en módulos FV dentro de un mismo *string*, y hace lo mismo con dos tipos de sistemas de clasificación de faltas basados en lógica difusa: Mamdani [93] y Sugeno [94].

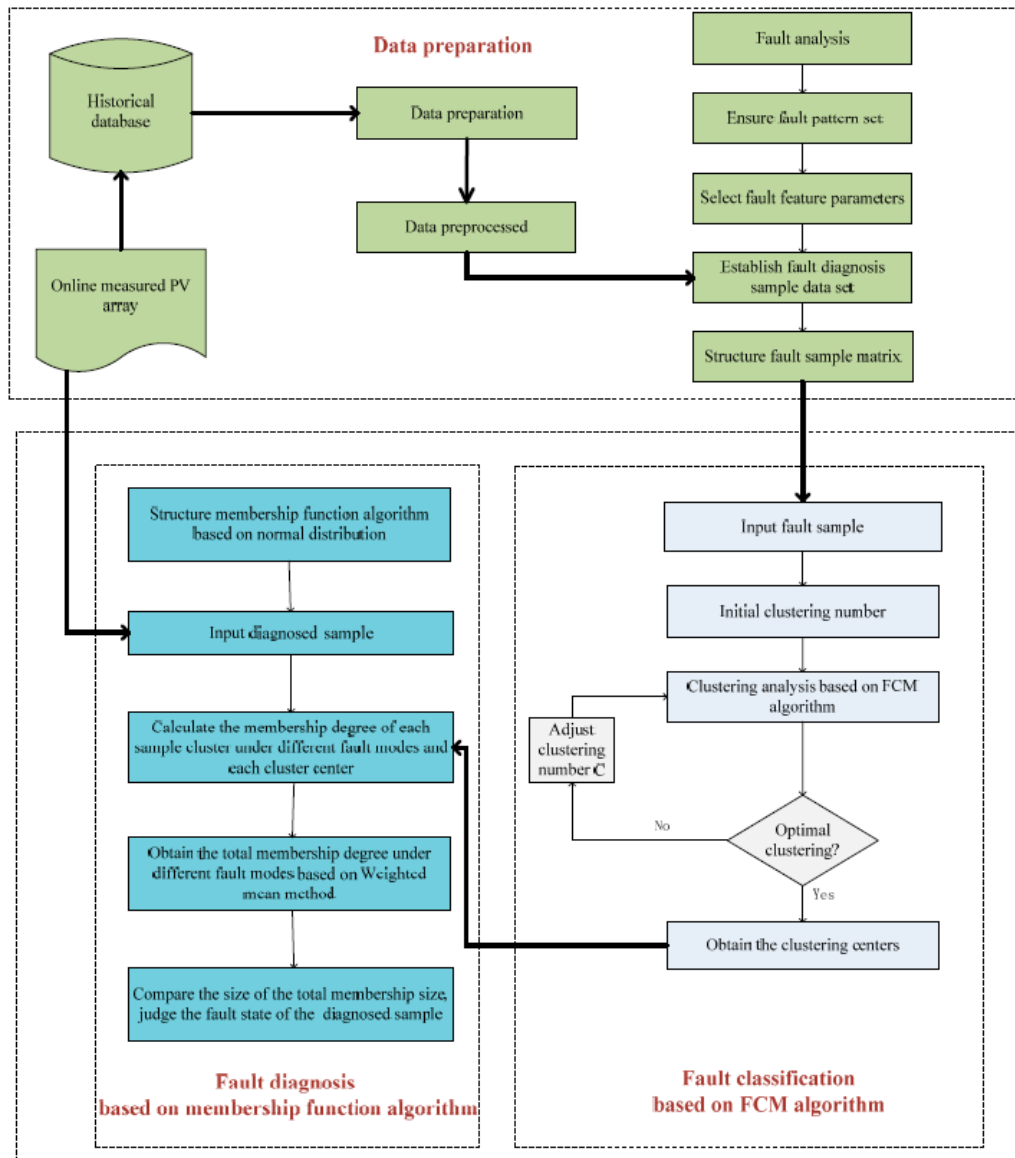
Una de las faltas más problemáticas que afectan a los sistemas FV son los puntos calientes, donde una célula o grupo de células FV se calienta significativamente en comparación con células adyacentes, provocando una disminución de la generación de potencia. Con objeto de detectar estas faltas, en [186] se lleva a cabo un análisis sobre 2580 módulos FV afectados por distintos tipos de puntos calientes y que son operados bajo distintas condiciones ambientales, distribuidos a lo largo del Reino Unido. El modelo de detección de faltas consiste en una sistema de inferencia difusa que usa un controlador difuso tipo Mamdani.

[187] propone un método que combina las ventajas de un algoritmo *Fuzzy C-means* (excelente habilidad de clasificación y bajo coste computacional) con algoritmo para el cálculo de membresía difusa (muy buena capacidad de computación de distancias), mejorando la precisión en el diagnóstico. Este método es capaz de detectar faltas de cortocircuito, circuito abierto y sombreado parcial entre otras. La Figura 5.2 muestra la arquitectura del método propuesto, distinguiendo entre las fases de adquisición y preparación de datos, la fase de clasificación usando FCM y el diagnóstico basado en membresía difusa. Análisis en simulación demostraron una efectividad del 96 % del método propuesto, que a su vez fue testado en campo obteniendo igualmente buenos resultados.

Además de la lógica difusa, otra de las técnicas de IA más extendidas en la detección y diagnóstico de faltas en sistemas FV son las redes neuronales artificiales [9]. Estas son capaces de detectar todos los tipos de faltas más comunes en el lado de corriente continua con una precisión superior al 90 % en la mayoría de los casos [188].

[189] desarrolla un modelo inteligente para detección y clasificación de faltas de circuito abierto y línea-línea para sistemas FV. El modelo usa una red neuronal probabilística enfocada a aprendizaje supervisado, y es entrenado usando un gran set de datos históricos de un sistema FV en operación normal y en estado fallido, distinguiendo entre condiciones ambientales de verano e invierno. La Figura 5.3 muestra la arquitectura del sistema incluyendo las fases de adquisición y preprocesado de datos. Los resultados experimentales revelan una precisión del 100 % en la detección y clasificación de las faltas citadas, incluso en casos en los que dadas las condiciones es difícil distinguir de operación normal y falta línea-línea.

[190] propone un método basado en red neuronal probabilística [48] para detectar y clasificar faltas de circuito abierto y línea-línea en tiempo real. La Figura 5.4 muestra la matriz de confusión de los resultados obtenidos en simulación, para los cuales se ha empleado una propuesta de modelado de sistemas FV que únicamente requiere de información del *datasheet* del fabricante en condiciones



**Figura 5.2** Arquitectura del sistema de detección y diagnóstico de falta desarrollado en [187].

normales de operación y test de condiciones estándar de operación. La precisión del método en el diagnóstico de faltas es del 98.53%.

Ante el problema de la falta de datos de módulos FV bajo condiciones de falta, [191] desarrolla un método de detección y diagnóstico de falta que emplea un modelo de predicción de la generación FV basada en red neuronal LSTM para el diagnóstico de falta. La red primero es entrenada en condiciones normales de operación, y luego se emplea una pequeña cantidad de datos en condiciones de falta para afinar sus parámetros (esta técnica recibe el nombre de *transfer learning* [192]). Los resultados obtenidos en fase experimental muestran que el método propuesto es eficaz en la detección y diagnóstico de falta incluso en condiciones climáticas adversas y resulta útil en la operación y mantenimiento inteligentes de sistemas FV.

En [193], se emplea un mecanismo de procesamiento de datos *MapReduce* para construir un modelo de diagnóstico de falta en plantas fotovoltaicas basado en reflectometría [194]. En este trabajo se propone una arquitectura de sistema de diagnóstico de falta basado en computación en la nube, compuesto por múltiples servidores que tiene por objeto la adquisición, procesamiento y análisis de los



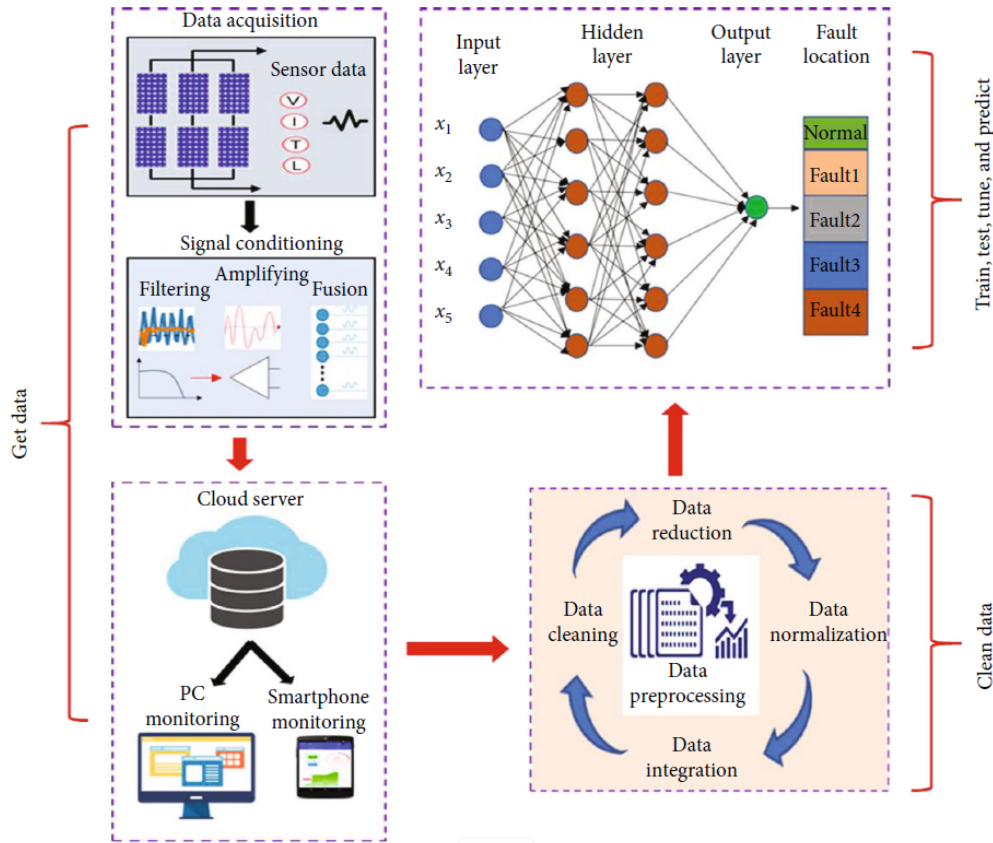


Figura 5.3 Arquitectura del sistema de detección y diagnóstico de falta desarrollado en [189].

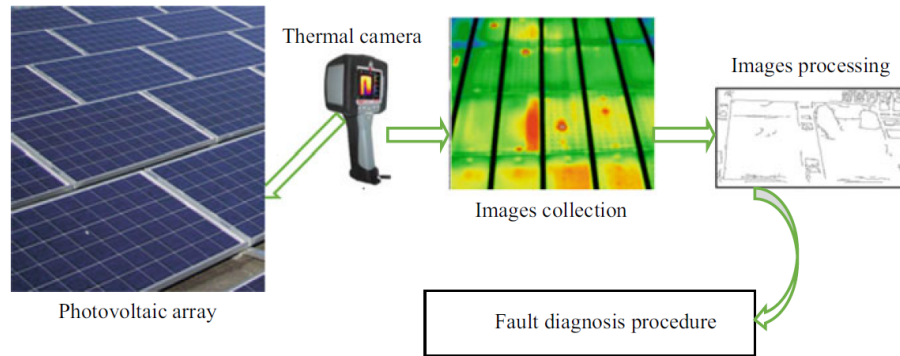
Output (obtained) Class	1	85 25.00%	2 0.588	0 0.00%	0 0.00%	97.70% 2.30%
	2	0 0.00%	83 24.41%	3 0.88	0 0.00%	96.51% 3.49%
	3	0 0.00%	0 0.00%	82 24.12%	0 0.00%	100.00% 0.00%
	4	0 0.00%	0 0.00%	0 0.00%	85 25.00%	100.00% 0.00%
		100.00% 0.00%	97.65% 2.35%	96.47% 3.53%	100.00% 0.00%	98.53% 1.47%
	1	2	3	4		
	Target(actual) Class					

Figura 5.4 Matriz de confusión de resultados [190]. Clase 1: Operación normal, Clase 2: Circuito abierto en 50% de los módulos o menos. Clase 3: Circuito abierto en 75% de los módulos o menos. Clase 4: Falta línea-línea.

datos recogidos de los distintos módulos fotovoltaicos de una planta de generación de gran escala. El procesamiento de los datos se hace a través de computación en paralelo empleando *MapReduce*.

### 5.1.2 Métodos visuales y térmicos

Los métodos visuales y térmicos se basan en el procesamiento de imágenes para detectar las faltas más comunes que se dan en módulos FV, como decoloración, bronceado, ensuciamiento de superficie, puntos calientes, rotura y delaminación. Estas técnicas son capaces de identificar y localizar las faltas, aunque para ello requieren de equipos o instrumentos sofisticados como cámaras térmicas y drones. En la Figura 5.5 se muestra un diagrama simplificado de un método de diagnóstico de falta basado en procesamiento de imagen térmica [9].



**Figura 5.5** Diagrama simplificado de diagnóstico de falta basado en procesamiento de imagen térmica [9].

[195] presenta un método basado en *deep learning* para detectar faltas en plantas FV de gran escala usando imágenes aéreas obtenidas por vehículos aéreos no tripulados. Para ello emplea una red neuronal convolucional [196] entrenada con un conjunto de 8400 imágenes aéreas de módulos FV bajo distintas condiciones. El método propuesto es capaz de detectar y clasificar múltiples tipos de faltas, incluyendo sombreado por polvo, delaminación, corrosión de cuadrícula, estela de caracol y amarilleo; siendo posible su implementación en tiempo real una vez entrenado el modelo. La Tabla 5.1 muestra la precisión del método en el diagnóstico de distintos tipos de falta en comparación con un algoritmo convencional de reconocimiento de patrones [197]. En la Tabla 5.2 se muestra la influencia del tamaño del set de datos en la precisión del modelo.

**Tabla 5.1** Comparativa de la precisión del método propuesto en [195] frente a método convencional [197].

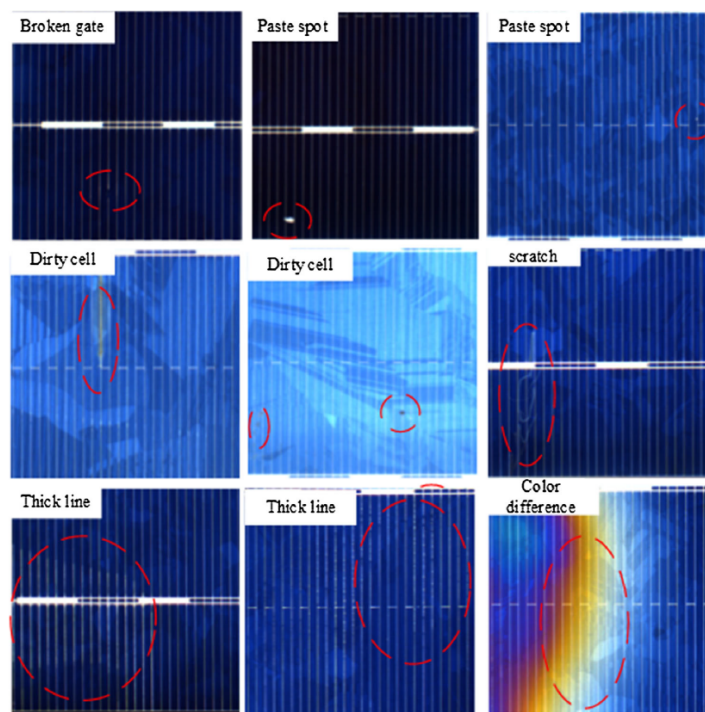
Falta	Método convencional [197]	Método propuesto en [195]
Sin falta	86.8 %	98.5 %
Sombreado por polvo	89.7 %	97.8 %
Delaminación	77.5 %	99.5 %
Corrosión de cuadrícula	47.9 %	98.0 %
Estela de caracol	48.3 %	97.8 %
Amarilleo	96.8 %	98.1 %

Empleando también imágenes aéreas de vehículos aéreos no tripulados, [198] desarrolla un método de detección y diagnóstico de faltas en módulos FV basado en *deep learning*. En concreto, emplea un modelo VGG16 pre-entrenado [56] para el reconocimiento de patrones y clasificación de faltas a partir de imágenes. El modelo es capaz de detectar y clasificar faltas de quemaduras, delaminación, decoloración, rotura de cristal y estela de caracol con una precisión del 95.4 %.

**Tabla 5.2** Precisión del método propuesto en [195] según el tamaño del set de datos.

Nº de muestras	600	1200	1800	2400	3000	3600	4200
<b>Precisión</b>	15.33%	16.25%	15.84%	18.1%	21.17%	17.8%	19.4%
Nº de muestras	4800	5400	6000	6600	7200	8400	4200
<b>Precisión</b>	26.5%	91.2%	97.3%	98.5%	98.1%	98.6%	98.7%

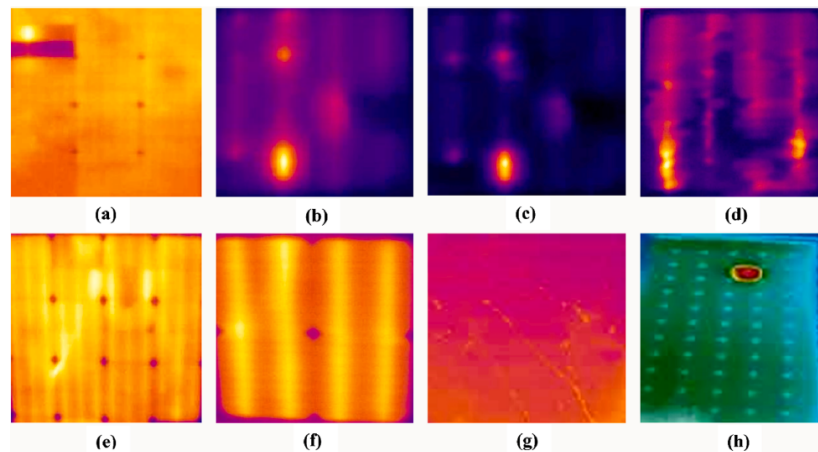
Empleando un conjunto de 21245 imágenes de células FV para entrenar una red neuronal convolucional, [199] desarrolla un método visual de detección y clasificación de faltas que alcanza una precisión del 94.3%. El tiempo consumido por el modelo para detectar defectos en 100 imágenes de células FV es de 3.66 segundos, lo que satisface los requerimientos de detección de faltas en tiempo real. La Figura 5.6 muestra algunos ejemplos de imágenes de defectos en células FV que este método es capaz de detectar.

**Figura 5.6** Varios defectos en la superficie de células solares [199].

En [200] se desarrolla un método de clasificación de faltas en módulos FV basado en análisis de imágenes térmicas empleando una red neuronal artificial. Al ser validado, el modelo alcanza una precisión del 91.7%.

[201] emplea red neuronal convolucional para detectar faltas en módulos FV a través del análisis de imágenes infrarrojas. Este estudio compara el desempeño de la CNN tanto al ser entrenada de manera aislada como al aplicar *transfer learning*. Ambas técnicas alcanzan unas precisiones del 98.67% y 99.23%, respectivamente, al mismo tiempo que mantienen una velocidad de predicción compatible con su implementación en tiempo real, de entorno a 1.3 segundos por cada 100 imágenes analizadas. La Figura 5.7 muestra algunos ejemplos de imágenes infrarrojas de módulos FV fallidos que son detectados por estas técnicas.

Aplicando un algoritmo basado en red neuronal convolucional, [202] alcanza precisiones cercanas al 100% en la detección de puntos calientes en paneles FV a partir de imágenes termográficas



**Figura 5.7** Defectos mostrados en imágenes infrarrojas de módulos FV [201]: (a) interconexión fallida en célula (b) agrietamiento (c) grietas que aíslan partes de célula (d) uniones de soldaduras fallidas (e) derivación localizada en célula (f) Alta densidad de corriente en nudo (g) Rotura de cristal de módulo (h) célula fallida en imagen infrarroja exterior.

obtenidas por vehículos no tripulados y operadores a nivel de suelo. El método llega incluso a alcanzar el 100 % de efectividad al aplicar distintas técnicas de preprocesado y para conjuntos de datos de 1000 imágenes.

## 5.2 Mantenimiento predictivo

Históricamente se han empleado tres maneras de resolver los problemas relativos al mantenimiento: correctivo, preventivo y predictivo. El primer tipo se lleva a cabo sólo cuando el equipo falla, lo que resulta costoso y conlleva problemas relacionados con la seguridad de las personas y el entorno. El mantenimiento preventivo se basa en el reemplazamiento periódico de los componentes de los equipos. Una estimación aproximada de la vida útil de los componentes incurre en un uso no óptimo de los mismos, fallos inesperados, incremento del tiempo fuera de servicio y reducción de la seguridad. En el caso del mantenimiento predictivo, las condiciones de los parámetros de los equipos se recogen y analizan automáticamente en tiempo real, y se planean las acciones de mantenimiento en base a los resultados de estos análisis. La constante monitorización de los componentes de las máquinas provee un pronóstico fiable y preciso de la vida útil y permite la completa automatización e implementación de las tareas de mantenimiento de manera económica [203].

Junto con los avances en la detección y diagnóstico de faltas en sistemas FV, que predicen faltas que están ocurriendo en el momento, existen también estudios en el campo del pronóstico de faltas, que consiste en la predicción del rendimiento del sistema FV durante su vida útil restante (RUL, *remaining useful life*) y, en consecuencia, predecir posibles faltas futuras empleando técnicas de regresión basadas en *machine learning* [204]. La RUL de un sistema FV se determina estimando el ratio de degradación de los módulos FV empleando modelos de degradación que tienen en cuenta radiación ultravioleta, temperatura, humedad, modelos de estado de salud y de final de vida [205–207].

[206] desarrolla un modelo de pronóstico de la RUL combinando monitorización en tiempo real, que permite la supervisión de parámetros críticos de los módulos FV; y algoritmo *relevance vector machine*, empleado para predecir degradaciones en los mismos. La Figura 5.8 muestra los resultados de predicción de la RUL en comparación con los obtenidos al aplicar regresión de proceso Gaussiano y modelo autorregresivo integrado de promedio móvil (ARIMA, por sus siglas en inglés), mejorando a ambos métodos.

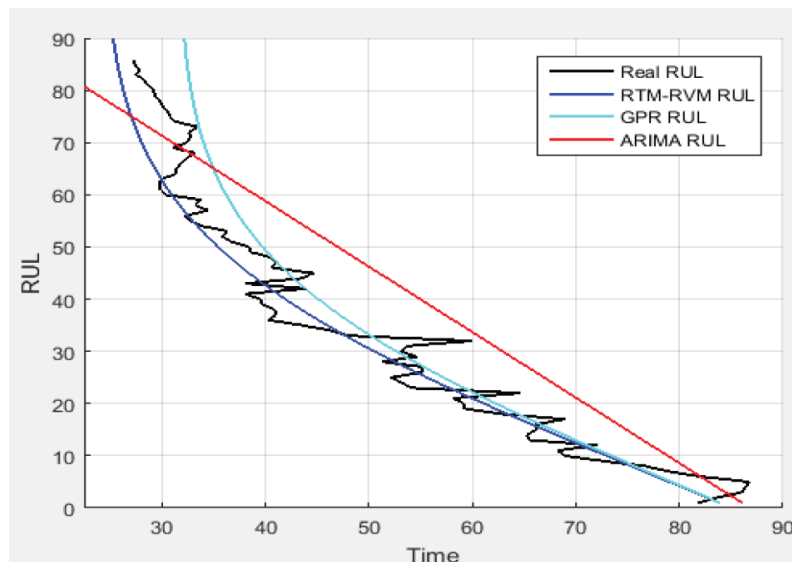


Figura 5.8 Comparativa de la RUL real y estimada por distintos métodos [206].

Entre los distintos componentes de un sistema FV, los inversores destacan por su impacto en la fiabilidad del sistema, ya que son responsables de buena parte de las faltas que pueden tener lugar y en consecuencia representan una gran cantidad de pérdidas de energía del sistema [208,209]. Es por esto que [210] analiza un gran set de datos de registros de mantenimiento en inversores, compuesto por 55000 registros procedentes de 880 plantas FV, con el fin de establecer estrategias de operación y mantenimiento. Dado que la fuente de datos consiste en información basada en textos, para este análisis se emplean técnicas de *machine learning* y procesamiento del lenguaje natural [211,212].

[213] desarrolla un algoritmo inteligente que permite predecir faltas de polaridad inversa en sistemas FV. El algoritmo consiste en un híbrido de *support vector regression* y *k-nearest neighbor*.

Con el objetivo de detectar anomalías en sistemas FV y generar alertas predictivas de carácter diario para los operadores, [214] implementa un algoritmo de relativamente bajo nivel de complejidad para dar soporte en tareas de mantenimiento predictivo. Para ello entrena una red neuronal con datos de generación, irradiancia y temperatura de una planta FV de gran escala (6.22 MW de potencia nominal, compuesta por 20384 módulos y 120 inversores) en una ventana temporal de 2 años. Este modelo predice la generación de la planta y en base al residuo de la predicción con respecto al valor medido genera alertas sobre posibles faltas en el sistema. Este algoritmo es capaz de predecir faltas con antelación suficiente (incluso semanas antes de que la falta ocurra) para planificar tareas de mantenimiento y así evitar las pérdidas que supondría la aparición de dichas faltas. Resultados experimentales demostraron una precisión en la detección de anomalías superior al 90%.

## 5.3 Resumen

En este capítulo se han revisado las aplicaciones de técnicas de IA y *Big Data* a tareas de detección y diagnóstico de faltas y mantenimiento predictivo de plantas FV.

Las técnicas de IA muestran una excelente capacidad para detectar anomalías en grandes conjuntos de datos y clasificarla. Es por esto que es posible encontrar un gran número de aplicaciones de estas técnicas a tareas de detección y diagnóstico de faltas en plantas FV. En este sentido, los métodos basados en lógica difusa destacan por su flexibilidad en la detección y clasificación de anomalías, lo que permite abandonar el uso de umbrales fijos en la detección de faltas y evita falsas alarmas. Esto se traduce en una disminución del tiempo fuera de servicio de las plantas FV. Los modelos de

detección y clasificación de faltas basados en redes neuronales son también capaces de abordar esta tarea de manera precisa y eficiente.

Dentro del campo del *deep learning*, destacan las CNNs por su habilidad de análisis de imágenes. Estas herramientas, junto con el empleo de vehículos aéreos no tripulados, se muestran como una excelente solución para la detección y diagnóstico de faltas a través de métodos visuales y térmicos, obteniendo elevadas tasas de acierto.

Por su parte, los métodos basados en *machine learning* son capaces de estimar la vida útil de módulos FV, lo que supone una tarea clave en el mantenimiento predictivo de plantas FV. En este ámbito también encuentra aplicación el procesamiento del lenguaje natural.

A continuación, en la tabla Tabla 5.3 se resumen las aplicaciones de técnicas de IA y *Big Data* a las tareas objeto de estudio de este capítulo:

**Tabla 5.3** Resumen de aplicaciones de técnicas de IA y *Big Data* a detección y diagnóstico de falta y mantenimiento predictivo.

Método	Detección y diagnóstico de falta		
	Métodos eléctricos	Métodos visuales y térmicos	Mantenimiento predictivo
Clasificación difusa	[182], [185], [186], [187]		
ANN	[185], [189], [190]	[200]	[214]
LSTM	[191]		
<i>MapReduce</i>	[193]		
CNN		[196], [198], [199], [201], [202]	
RVM			[206]
NLP			[210]
SVR			[213]
<i>k</i> -NN			[213]

## 6 Conclusiones

---

En el presente trabajo se ha llevado a cabo una revisión del estado del arte del empleo de técnicas de inteligencia artificial y *Big Data* en distintos ámbitos de aplicación dentro de las plantas FV conectadas a red.

Para el modelado de plantas FV se distinguen dos tareas principales: identificación de parámetros y modelado equivalente.

La identificación de parámetros toma mayor relevancia a nivel de módulo e inversor FV. En esta tarea cobran protagonismo los algoritmos bioinspirados, que mejoran a los métodos tradicionales en cuanto a precisión de los parámetros identificados. Una posible mejora en estos métodos sería la actualización de los parámetros identificados ante cambios en las condiciones de la planta FV, dado que hasta el momento estos parámetros se identifican offline y se consideran invariables en el tiempo.

Para el modelado equivalente de plantas FV se emplean técnicas de aprendizaje no supervisado de clustering. Gracias al empleo de estas técnicas se pueden construir modelos equivalentes que reducen en gran medida los tiempos de simulación. Además, la precisión de los modelos puede ser mejorada gracias al empleo de técnicas de preprocesado de datos. Algunos de los trabajos revisados proponen modelos online que son actualizados ante cambios en las condiciones de la planta, si bien no todos presentan esta característica. Se echa en falta el empleo de técnicas de IA en la identificación de los parámetros de los modelos resultantes de los algoritmos de clustering, lo que mejoraría aún más la precisión de los mismos.

Dada la complejidad que supone el control de plantas de generación en sistemas de energía eléctrica, y en especial las solares fotovoltaicas, las técnicas de IA suponen una herramienta capaz de resolver con solvencia los problemas que se presentan en este ámbito, mejorando incluso a los métodos convencionales.

De entre estas técnicas, los sistemas de control basados en lógica difusa son capaces de resolver problemas de limitación de potencia activa, control de frecuencia y respuesta LVRT con una eficacia superior a los métodos hasta ahora más extendidos para la resolución de estos problemas. Para el control de tensiones, las técnicas de aprendizaje reforzado para sistemas multi-agente cobran protagonismo dadas las características propias de este paradigma de control, mejorando también los resultados en comparación con los métodos de tradicionales. Estas mejoras se traducen en reducción de pérdidas del sistema, mayor viabilidad económica de las soluciones propuestas, y, en general, mejora de la integración de las plantas fotovoltaicas en las redes de transporte y distribución.

Al realizar esta revisión del estado del arte se han detectado dos posibles líneas a seguir en cuanto al empleo de técnicas de IA aplicadas al control de plantas FV. Por una parte, el desarrollo de estas herramientas precisa de modelos de planta en los que simular el desempeño de los sistemas de

control presentados. En este aspecto, se echa en falta el empleo de los modelos estudiados en el Capítulo 3, que implicaría una mejora en la precisión de los datos de entrenamiento y una reducción en el tiempo de simulación en esta fase. Por otra parte, los métodos propuestos han sido probados en entornos de simulación, echándose en falta la implementación de los mismos en pruebas en campo con el fin de evaluar el desempeño de estas técnicas en condiciones reales de operación.

La detección y diagnóstico de faltas es el campo donde se ha podido encontrar una mayor cantidad de artículos que aplican técnicas de IA para resolver estos problemas. Esto se explica dada la habilidad de estas técnicas para analizar grandes conjuntos de datos y detectar e identificar anomalías en los mismos.

El empleo de IA elimina la necesidad de fijar valores límites para distintas magnitudes a la hora de implementar un método de detección de faltas, lo que aporta flexibilidad a la tarea y evita falsas alarmas, reduciendo el tiempo fuera de servicio de las plantas FV. En este aspecto la técnicas de lógica difusa son ideales para escapar del empleo de umbrales rígidos de detección. Las redes neuronales artificiales también demuestran una excelente habilidad de detección de anomalías, como las que implican faltas en un sistema FV.

Una vez detectada una falta, resulta imprescindible clasificar la misma e identificar el elemento fallido y sus causas. En este sentido, los clasificadores basados en redes neuronales y lógica difusa muestran una alta precisión a la hora de resolver esta tarea. Estas técnicas permiten el diagnóstico cuando no es posible establecer una relación matemática entre los datos de entrada y las faltas que se producen.

Los avances del deep learning en el campo del análisis de imágenes hacen de las redes neuronales convolucionales una excelente herramienta para detectar y diagnosticar faltas en plantas FV, con elevadas tasas de acierto. La implantación de estos métodos suele ir de la mano del empleo de vehículos aéreos no tripulados capaces de obtener imágenes de los elementos de la planta.

Las técnicas de IA también encuentran aplicación en el ámbito del mantenimiento predictivo, siendo capaces de estimar la vida útil de módulos FV, predecir faltas e incluso analizar datos basados en texto para establecer estrategias de operación y mantenimiento.

Como inconvenientes del empleo de estas técnicas en el campo de la detección y diagnóstico de falta y mantenimiento predictivo, destaca la dificultad que supone obtener datos de plantas FV en estado fallido, imprescindibles en el entrenamiento de modelos. En este sentido las técnicas de transfer learning se presentan como una solución para reducir el volumen de datos en estado fallido necesario para ajustar los modelos.

Existe una gran diversidad de tipos de falta que pueden producirse en un sistema FV, y se ha comprobado como los distintos métodos revisados se especializaban en distintos subconjuntos de las mismas. El empleo conjunto de métodos basados en mediciones de magnitudes eléctricas y climatológicas y métodos basados en análisis de imagen podría ser la hoja de ruta a seguir para la implantación de soluciones capaces de detectar y clasificar una mayor variedad de tipos de faltas en plantas FV.

Tras esta revisión del estado del arte se concluye que las técnicas de IA representan una potente herramienta a la hora de resolver los problemas estudiados, mejorando en buena medida a los métodos convencionales e incluso encontrando soluciones que por el momento solo son posibles a través del empleo de estas técnicas.

En cuanto a la aplicación de *Big Data*, se ha podido comprobar que esta se encuentra en una fase preliminar y que apenas existe literatura en los campos objeto de estudio de este trabajo. Entre ellos, la detección y diagnóstico de falta y el mantenimiento predictivo constituyen el único ámbito donde, ligado a la monitorización del sistema, se pueden encontrar algunos trabajos que introducen tecnologías de *Big Data* para la adquisición y procesado de datos.



El concepto de *Big Data* es amplio y algo difuso. Es un término que hace referencia a cantidades masivas de datos, tecnologías para tratar con estos datos, arquitecturas destinadas a su manejo, y que, además de lo anterior, es un término de marketing. El impacto del empleo del término *Big Data* ligado a la ambigüedad del mismo provoca que se puedan encontrar en la literatura artículos que, haciendo mención al *Big Data* en sus títulos, no reúnan los anteriores requisitos para que los métodos que emplean puedan ser considerados como técnicas de *Big Data* en los términos definidos en este trabajo. Esto se ha podido poner de manifiesto a la hora de realizar esta revisión del estado del arte, durante la cual múltiples artículos han sido descartados por este motivo.

La mayoría de artículos que tratan el *Big Data* en el ámbito cubierto por este trabajo, hacen referencia a extensos sistemas de distribución y transporte de energía eléctrica en el que existen multitud de agentes que generan grandes volúmenes de datos a gran velocidad. Estos enfoques se escapan del alcance de esta revisión del estado del arte, ya que en ellos los parques FV, tanto de menor como de mayor escala, son únicamente un agente más y el protagonismo recae en el sistema en su conjunto. Como ejemplo, se pueden encontrar trabajos acerca de la aplicación de *Big Data* al control de grandes smart grids, pero no tanto así aplicados a parques FV como agentes diferenciados del resto.

Lo anterior no implica que no sea posible aplicar técnicas de *Big Data* confinando el alcance a una planta FV. Los parques FV están compuestos por una gran multitud de elementos que, gracias a los sistemas de monitorización y comunicaciones, pueden generar grandes cantidades de datos a una frecuencia elevada. Sin embargo, durante esta revisión del estado del arte no se han encontrado artículos con este enfoque, lo que por otro lado deja abierta la puerta a futuras aplicaciones de *Big Data* en este contexto.

Además de lo anterior, cabe tener presente que en las distintas aplicaciones estudiadas, el papel del *Big Data* es el de tratar con datos para que estos sirvan de entrada a otras herramientas para la consecución de los objetivos expuestos en capítulos anteriores. Como ejemplo para la detección de faltas en plantas fotovoltaicas, podría plantearse un método que combinara *Big Data* con red neuronal, en el que el papel del *Big Data* es el de adquirir, procesar y suministrar los datos que la red neuronal empleará para detectar las faltas que se produzcan. Hay que entender *Big Data* como un conjunto de herramientas orientadas a los datos, y que sirve para mejorar en las aplicaciones que tienen como herramientas técnicas de inteligencia artificial, entre otras. En la búsqueda de artículos que cubran la aplicación de *Big Data* en estos términos no se ha encontrado literatura que desarrolle aplicaciones que hagan uso de *Big Data* junto con técnicas de IA, por lo que se puede concluir que existe una posibilidad de aplicación de ambos paradigmas conjuntamente, de gran interés y aún por explorar en el campo cubierto por este trabajo.

En los últimos años se está viviendo una auténtica revolución en los campos de la inteligencia artificial y el *Big Data*, lo que junto al auge de la instalación de plantas fotovoltaicas y la digitalización de los sistemas de energía eléctrica, conforman un escenario en el que estas técnicas tendrán un papel protagonista en la obtención de soluciones más inteligentes, viables, eficaces, precisas, ...mejores.



# Índice de Figuras

---

2.1	Relación entre inteligencia artificial, machine learning y deep learning	4
2.2	Tipos de aprendizaje automático	5
2.3	SVM. Ejemplos de hiperplanos separadores [19]	7
2.4	Interacción agente-entorno en un proceso de decisión de Markov [37]	10
2.5	Estructura de una neurona artificial [9]	11
2.6	Ejemplo de perceptrón multicapa [9]	12
2.7	Arquitectura de red neuronal convolucional VGG-16 [55, 56]	13
2.8	Componentes de un sistema difuso [92]	19
2.9	Clasificación tradicional frente a clasificación difusa [102]	20
3.1	Modelo de diodo único de la célula FV teórica y circuito equivalente de un dispositivo FV incluyendo resistencia serie y paralelo [121]	24
3.2	Configuración de la red neuronal artificial [123]	24
3.3	Configuración básica del modelo FV [123]	25
3.4	Modelo equivalente de planta FV de gran escala conectada a red [133]	26
3.5	Estructura típica de planta de generación FV [27]	27
3.6	Mapa tridimensional de parámetros para el agrupamiento de $k$ -means [27]	27
3.7	Comparativa del comportamiento dinámico de los modelos equivalentes ante caída de tensión [27]	28
3.8	Respuesta dinámica de los modelos equivalentes antes y después de actualizar la clasificación [135]	28
3.9	Visualización de los resultados de la aplicación de PCA para la reducción de la dimensión de datos [140]	29
4.1	Resultados de suavizado de inyección de potencia. En verde, la predicción de generación FV de la ANN, en rojo la curva óptima de generación y en azul la inyección/absorción requerida al sistema de almacenamiento [143]	32
4.2	Red de 14 nudos del IEEE empleada en [149]	33
4.3	Variación de frecuencia ante escalón de carga para distintos modos de operación PV [149]	34
5.1	Clasificación de faltas en sistemas FV [180]	38
5.2	Arquitectura del sistema de detección y diagnóstico de falta desarrollado en [187]	40
5.3	Arquitectura del sistema de detección y diagnóstico de falta desarrollado en [189]	41

---

5.4	Matriz de confusión de resultados [190]. Clase 1: Operación normal, Clase 2: Circuito abierto en 50% de los módulos o menos. Clase 3: Circuito abierto en 75% de los módulos o menos. Clase 4: Falta línea-línea	41
5.5	Diagrama simplificado de diagnóstico de falta basado en procesamiento de imagen térmica [9]	42
5.6	Varios defectos en la superficie de células solares [199]	43
5.7	Defectos mostrados en imágenes infrarrojas de módulos FV [201]: (a) interconexión fallida en célula (b) agrietamiento (c) grietas que aíslan partes de célula (d) uniones de soldaduras fallidas (e) derivación localizada en célula (f) Alta densidad de corriente en nudo (g) Rotura de cristal de módulo (h) célula fallida en imagen infrarroja exterior	44
5.8	Comparativa de la RUL real y estimada por distintos métodos [206]	45

# Índice de Tablas

---

3.1	Comparativa de errores relativos (%) obtenidos en identificación de parámetros empleando distintos algoritmos bioinspirados [126]	25
3.2	Resumen de aplicaciones de técnicas de IA a modelado de plantas FV	30
4.1	Comparativa de desviaciones de tensión (%) obtenidos empleando distintas estrategias de regulación de tensión [158]	35
4.2	Resultados de aplicación de metodología propuesta en [156] para la regulación de tensión	35
4.3	Resumen de aplicaciones de técnicas de IA a control de plantas FV	36
5.1	Comparativa de la precisión del método propuesto en [195] frente a método convencional [197]	42
5.2	Precisión del método propuesto en [195] según el tamaño del set de datos	43
5.3	Resumen de aplicaciones de técnicas de IA y Big Data a detección y diagnóstico de falta y mantenimiento predictivo	46



# Bibliografía

---

- [1] J. N. Kok, E. J. Boers, W. A. Kusters, P. Van der Putten, and M. Poel, “Artificial intelligence: definition, trends, techniques, and cases,” *Artificial intelligence*, vol. 1, pp. 270–299, 2009.
- [2] S. Russell and P. Norvig, “Artificial intelligence: a modern approach,” 2002.
- [3] D. Poole, A. Mackworth, and R. Goebel, “Computational intelligence,” 1998.
- [4] P. Ongsulee, “Artificial intelligence, machine learning and deep learning,” in *2017 15th International Conference on ICT and Knowledge Engineering (ICT&KE)*, pp. 1–6, IEEE, 2017.
- [5] A. Munoz, “Machine learning and optimization,”
- [6] A. L. Samuel, “Some studies in machine learning using the game of checkers,” *IBM Journal of research and development*, vol. 3, no. 3, pp. 210–229, 1959.
- [7] Z. Ghahramani, “Unsupervised learning,” in *Summer School on Machine Learning*, pp. 72–112, Springer, 2003.
- [8] I. El Naqa and M. J. Murphy, “What is machine learning?,” in *machine learning in radiation oncology*, pp. 3–11, Springer, 2015.
- [9] A. Mellit, “Recent applications of artificial intelligence in fault diagnosis of photovoltaic systems,” in *A Practical Guide for Advanced Methods in Solar Photovoltaic Systems*, pp. 257–271, Springer, 2020.
- [10] E. Alpaydin, *Machine learning: the new AI*. MIT press, 2016.
- [11] Y. Zhang, *New advances in machine learning*. BoD–Books on Demand, 2010.
- [12] C. J. Burges, “A tutorial on support vector machines for pattern recognition,” *Data mining and knowledge discovery*, vol. 2, no. 2, pp. 121–167, 1998.
- [13] V. Cherkassky and F. M. Mulier, *Learning from data: concepts, theory, and methods*. John Wiley & Sons, 2007.
- [14] C. Cortes and V. Vapnik, “Support-vector networks,” *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.
- [15] N. Cristianini, J. Shawe-Taylor, *et al.*, *An introduction to support vector machines and other kernel-based learning methods*. Cambridge university press, 2000.
- [16] V. Vapnik, *The nature of statistical learning theory*. Springer science & business media, 2013.

- [17] V. N. Vapnik, "An overview of statistical learning theory," *IEEE transactions on neural networks*, vol. 10, no. 5, pp. 988–999, 1999.
- [18] A. S. Lampropoulos and G. A. Tsihrintzis, "Machine learning paradigms," *Applications In Recommender Systems. Switzerland: Springer International Publishing*, 2015.
- [19] C. Sammut and G. I. Webb, *Encyclopedia of machine learning*. Springer Science & Business Media, 2011.
- [20] H. Drucker, C. J. Burges, L. Kaufman, A. Smola, V. Vapnik, *et al.*, "Support vector regression machines," *Advances in neural information processing systems*, vol. 9, pp. 155–161, 1997.
- [21] M. E. Tipping, "The relevance vector machine," in *Advances in neural information processing systems*, pp. 652–658, 2000.
- [22] B. Mahesh, "Machine learning algorithms-a review," *International Journal of Science and Research (IJSR).[Internet]*, vol. 9, pp. 381–386, 2020.
- [23] A. C. Lorena, L. F. Jacintho, M. F. Siqueira, R. De Giovanni, L. G. Lohmann, A. C. De Carvalho, and M. Yamamoto, "Comparing machine learning classifiers in potential distribution modelling," *Expert Systems with Applications*, vol. 38, no. 5, pp. 5268–5275, 2011.
- [24] S. Ray, "A quick review of machine learning algorithms," in *2019 International conference on machine learning, big data, cloud and parallel computing (COMITCon)*, pp. 35–39, IEEE, 2019.
- [25] A. Singh, N. Thakur, and A. Sharma, "A review of supervised machine learning algorithms," in *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*, pp. 1310–1315, Ieee, 2016.
- [26] Z. Huang, "Extensions to the k-means algorithm for clustering large data sets with categorical values," *Data mining and knowledge discovery*, vol. 2, no. 3, pp. 283–304, 1998.
- [27] S. Na, L. Xumin, and G. Yong, "Research on k-means clustering algorithm: An improved k-means clustering algorithm," in *2010 Third International Symposium on intelligent information technology and security informatics*, pp. 63–67, Ieee, 2010.
- [28] J. MacQueen *et al.*, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, vol. 1, pp. 281–297, Oakland, CA, USA, 1967.
- [29] A. Fahim, A. Salem, F. A. Torkey, and M. Ramadan, "An efficient enhanced k-means clustering algorithm," *Journal of Zhejiang University-Science A*, vol. 7, no. 10, pp. 1626–1633, 2006.
- [30] C. M. Bishop *et al.*, *Neural networks for pattern recognition*. Oxford university press, 1995.
- [31] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, "A local search approximation algorithm for k-means clustering," *Computational Geometry*, vol. 28, no. 2-3, pp. 89–112, 2004.
- [32] N. Bansal, A. Blum, and S. Chawla, "Correlation clustering," *Machine learning*, vol. 56, no. 1, pp. 89–113, 2004.
- [33] H. Becker, "A survey of correlation clustering," *Advanced Topics in Computational Learning Theory*, pp. 1–10, 2005.
- [34] I. T. Jolliffe and J. Cadima, "Principal component analysis: a review and recent developments," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 374, no. 2065, p. 20150202, 2016.



- [35] R. O. Duda, P. E. Hart, *et al.*, *Pattern classification and scene analysis*, vol. 3. Wiley New York, 1973.
- [36] M. Van Otterlo and M. Wiering, “Reinforcement learning and markov decision processes,” in *Reinforcement learning*, pp. 3–42, Springer, 2012.
- [37] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.
- [38] P. Balaji and D. Srinivasan, “An introduction to multi-agent systems,” in *Innovations in multi-agent systems and applications-1*, pp. 1–27, Springer, 2010.
- [39] L. P. Kaelbling, M. L. Littman, and A. W. Moore, “Reinforcement learning: A survey,” *Journal of artificial intelligence research*, vol. 4, pp. 237–285, 1996.
- [40] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [41] S. Pouyanfar, S. Sadiq, Y. Yan, H. Tian, Y. Tao, M. P. Reyes, M.-L. Shyu, S.-C. Chen, and S. S. Iyengar, “A survey on deep learning: Algorithms, techniques, and applications,” *ACM Computing Surveys (CSUR)*, vol. 51, no. 5, pp. 1–36, 2018.
- [42] M. M. Najafabadi, F. Villanustre, T. M. Khoshgoftaar, N. Seliya, R. Wald, and E. Muharemagic, “Deep learning applications and challenges in big data analytics,” *Journal of big data*, vol. 2, no. 1, pp. 1–21, 2015.
- [43] R. O. S. Juan and J. Kim, “Utilization of artificial intelligence techniques for photovoltaic applications,” *Current Photovoltaic Research*, vol. 7, no. 4, pp. 85–96, 2019.
- [44] A. Carrio, C. Sampedro, A. Rodriguez-Ramos, and P. Campoy, “A review of deep learning methods and applications for unmanned aerial vehicles,” *Journal of Sensors*, vol. 2017, 2017.
- [45] F. Rosenblatt, “Principles of neurodynamics. perceptrons and the theory of brain mechanisms,” tech. rep., Cornell Aeronautical Lab Inc Buffalo NY, 1961.
- [46] B. Kröse, B. Krose, P. van der Smagt, and P. Smagt, “An introduction to neural networks,” 1993.
- [47] M. W. Gardner and S. Dorling, “Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences,” *Atmospheric environment*, vol. 32, no. 14-15, pp. 2627–2636, 1998.
- [48] D. F. Specht, “Probabilistic neural networks for classification, mapping, or associative memory.,” in *ICNN*, pp. 525–532, 1988.
- [49] A. G. Bors, “Introduction of the radial basis function (rbf) networks,” in *Online symposium for electronics engineers*, vol. 1, pp. 1–7, 2001.
- [50] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.
- [51] S. Albawi, T. A. Mohammed, and S. Al-Zawi, “Understanding of a convolutional neural network,” in *2017 International Conference on Engineering and Technology (ICET)*, pp. 1–6, Ieee, 2017.
- [52] O. Abdel-Hamid, L. Deng, and D. Yu, “Exploring convolutional neural network structures and optimization techniques for speech recognition.,” in *Interspeech*, vol. 11, pp. 73–5, Citeseer, 2013.
- [53] V. Dumoulin and F. Visin, “A guide to convolution arithmetic for deep learning,” *arXiv preprint arXiv:1603.07285*, 2016.
- [54] “An intuitive explanation of convolutional neural networks – the data science blog.” <https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/>. (Accessed on 08/23/2021).

- [55] M. Ferguson, R. Ak, Y.-T. T. Lee, and K. H. Law, "Automatic localization of casting defects with convolutional neural networks," in *2017 IEEE international conference on big data (big data)*, pp. 1726–1735, IEEE, 2017.
- [56] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.
- [57] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, pp. 1097–1105, 2012.
- [58] Y. Wang, M. Huang, X. Zhu, and L. Zhao, "Attention-based lstm for aspect-level sentiment classification," in *Proceedings of the 2016 conference on empirical methods in natural language processing*, pp. 606–615, 2016.
- [59] G. E. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *IEEE Transactions on audio, speech, and language processing*, vol. 20, no. 1, pp. 30–42, 2011.
- [60] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 249–256, JMLR Workshop and Conference Proceedings, 2010.
- [61] X. Li and X. Wu, "Constructing long short-term memory based deep recurrent neural networks for large vocabulary speech recognition," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4520–4524, IEEE, 2015.
- [62] D. B. Fogel, *Evolutionary computation: toward a new philosophy of machine intelligence*, vol. 1. John Wiley & Sons, 2006.
- [63] M. Mitchell and C. E. Taylor, "Evolutionary computation: an overview," *Annual Review of Ecology and Systematics*, vol. 30, no. 1, pp. 593–616, 1999.
- [64] A. E. Eiben, J. E. Smith, *et al.*, *Introduction to evolutionary computing*, vol. 53. Springer, 2003.
- [65] D. E. Goldberg and J. H. Holland, "Genetic algorithms and machine learning," 1988.
- [66] J. H. Holland, "Genetic algorithms," *Scientific american*, vol. 267, no. 1, pp. 66–73, 1992.
- [67] S. Mirjalili, "Genetic algorithm," in *Evolutionary algorithms and neural networks*, pp. 43–55, Springer, 2019.
- [68] M. Kumar, M. Husain, N. Upreti, and D. Gupta, "Genetic algorithm: Review and application," *Available at SSRN 3529843*, 2010.
- [69] W. Lee and H.-Y. Kim, "Genetic algorithm implementation in python," in *Fourth Annual ACIS International Conference on Computer and Information Science (ICIS'05)*, pp. 8–11, IEEE, 2005.
- [70] R. Storn and K. Price, "Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces," *Journal of global optimization*, vol. 11, no. 4, pp. 341–359, 1997.
- [71] S. Binitha, S. S. Sathya, *et al.*, "A survey of bio inspired optimization algorithms," *International journal of soft computing and engineering*, vol. 2, no. 2, pp. 137–151, 2012.
- [72] J. Ma, Z. Bi, T. O. Ting, S. Hao, and W. Hao, "Comparative performance on photovoltaic model parameter identification via bio-inspired algorithms," *Solar Energy*, vol. 132, pp. 606–616, 2016.

- [73] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95-international conference on neural networks*, vol. 4, pp. 1942–1948, IEEE, 1995.
- [74] A. Salman, I. Ahmad, and S. Al-Madani, "Particle swarm optimization for task assignment problem," *Microprocessors and Microsystems*, vol. 26, no. 8, pp. 363–371, 2002.
- [75] I. C. Trelea, "The particle swarm optimization algorithm: convergence analysis and parameter selection," *Information processing letters*, vol. 85, no. 6, pp. 317–325, 2003.
- [76] S. Olariu and A. Y. Zomaya, *Handbook of bioinspired algorithms and applications*. CRC Press, 2005.
- [77] D. Karaboga, "An idea based on honey bee swarm for numerical optimization," tech. rep., Citeseer, 2005.
- [78] D. Karaboga and C. Ozturk, "A novel clustering approach: Artificial bee colony (abc) algorithm," *Applied soft computing*, vol. 11, no. 1, pp. 652–657, 2011.
- [79] X. Fan, W. Sayers, S. Zhang, Z. Han, L. Ren, and H. Chizari, "Review and classification of bio-inspired algorithms and their applications," *Journal of Bionic Engineering*, vol. 17, pp. 611–631, 2020.
- [80] X.-S. Yang, *Nature-inspired metaheuristic algorithms*. Luniver press, 2010.
- [81] R. B. Payne and M. D. Sorensen, *The cuckoos*, vol. 15. Oxford University Press, 2005.
- [82] X.-S. Yang and S. Deb, "Cuckoo search via lévy flights," in *2009 World congress on nature & biologically inspired computing (NaBIC)*, pp. 210–214, Ieee, 2009.
- [83] A. S. Joshi, O. Kulkarni, G. M. Kakandikar, and V. M. Nandedkar, "Cuckoo search optimization-a review," *Materials Today: Proceedings*, vol. 4, no. 8, pp. 7262–7269, 2017.
- [84] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *science*, vol. 220, no. 4598, pp. 671–680, 1983.
- [85] R. A. Rutenbar, "Simulated annealing algorithms: An overview," *IEEE Circuits and Devices magazine*, vol. 5, no. 1, pp. 19–26, 1989.
- [86] T. Bäck, J. N. Kok, and G. Rozenberg, *Handbook of natural computing*. Springer, Heidelberg, 2012.
- [87] L. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, no. 3, pp. 338–353, 1965.
- [88] G. Chen and T. T. Pham, *Introduction to fuzzy sets, fuzzy logic, and fuzzy control systems*. CRC press, 2000.
- [89] S. Diciembre Sanahuja, "Sistemas de control con lógica difusa: Métodos de mamdani y de takagi-sugeno-kang (tsk)," 2017.
- [90] A. Mellit and S. A. Kalogirou, "Artificial intelligence techniques for photovoltaic applications: A review," *Progress in energy and combustion science*, vol. 34, no. 5, pp. 574–632, 2008.
- [91] "A very brief introduction to fuzzy logic and fuzzy systems | by carmel gafa | towards data science." <https://towardsdatascience.com/a-very-brief-introduction-to-fuzzy-logic-and-fuzzy-systems-d68d14b3a3b8>. (Accessed on 08/28/2021).
- [92] D. Binu and B. Rajakumar, *Artificial Intelligence in Data Mining: Theories and Applications*. Academic Press, 2021.
- [93] E. H. Mamdani and S. Assilian, "An experiment in linguistic synthesis with a fuzzy logic controller," *International journal of man-machine studies*, vol. 7, no. 1, pp. 1–13, 1975.

- [94] T. Takagi and M. Sugeno, “Fuzzy identification of systems and its applications to modeling and control,” *IEEE transactions on systems, man, and cybernetics*, no. 1, pp. 116–132, 1985.
- [95] M.-S. Yang, “A survey of fuzzy clustering,” *Mathematical and Computer modelling*, vol. 18, no. 11, pp. 1–16, 1993.
- [96] M. G. Omran, A. P. Engelbrecht, and A. Salman, “An overview of clustering methods,” *Intelligent Data Analysis*, vol. 11, no. 6, pp. 583–605, 2007.
- [97] S. Miyamoto, “Fuzzy clustering—basic ideas and overview,” in *Springer Handbook of Computational Intelligence*, pp. 239–248, Springer, 2015.
- [98] J. C. Dunn, “A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters,” 1973.
- [99] J. C. Dunn, “Well-separated clusters and optimal fuzzy partitions,” *Journal of cybernetics*, vol. 4, no. 1, pp. 95–104, 1974.
- [100] J. C. Bezdek, *FUZZY-MATHEMATICS IN PATTERN CLASSIFICATION*. Cornell University, 1973.
- [101] J. C. Bezdek, *Pattern recognition with fuzzy objective function algorithms*. Springer Science & Business Media, 2013.
- [102] “Fuzzy c-means clustering — is it better than k-means clustering? | by satyam kumar | towards data science.” <https://towardsdatascience.com/fuzzy-c-means-clustering-is-it-better-than-k-means-clustering-448a0aba1ee7>. (Accessed on 08/29/2021).
- [103] A. De Mauro, M. Greco, and M. Grimaldi, “A formal definition of big data based on its essential features,” *Library Review*, 2016.
- [104] M. Chen, S. Mao, and Y. Liu, “Big data: A survey,” *Mobile networks and applications*, vol. 19, no. 2, pp. 171–209, 2014.
- [105] A. Oussous, F.-Z. Benjelloun, A. A. Lahcen, and S. Belfkih, “Big data technologies: A survey,” *Journal of King Saud University-Computer and Information Sciences*, vol. 30, no. 4, pp. 431–448, 2018.
- [106] S. Sharma and V. Mangat, “Technology and trends to handle big data: Survey,” in *2015 Fifth International Conference on Advanced Computing & Communication Technologies*, pp. 266–271, IEEE, 2015.
- [107] A. Gupta, B. P. Dubey, H. Sivaraman, and M. Lohani, “Big data analytics—analysis and comparison of various tools,” in *Advances in Information Communication Technology and Computing*, pp. 491–499, Springer, 2021.
- [108] Statista, “Volume of data/information created, captured, copied, and consumed worldwide from 2010 to 2024,” 2021.
- [109] Y. Demchenko, P. Grosso, C. De Laat, and P. Membrey, “Addressing big data issues in scientific data infrastructure,” in *2013 International Conference on Collaboration Technologies and Systems (CTS)*, pp. 48–55, IEEE, 2013.
- [110] L. Rodríguez-Mazahua, C.-A. Rodríguez-Enríquez, J. L. Sánchez-Cervantes, J. Cervantes, J. L. García-Alcaraz, and G. Alor-Hernández, “A general perspective of big data: applications, tools, challenges and trends,” *The Journal of Supercomputing*, vol. 72, no. 8, pp. 3073–3113, 2016.
- [111] R. Kitchin and G. McArdle, “What makes big data, big data? exploring the ontological characteristics of 26 datasets,” *Big Data & Society*, vol. 3, no. 1, p. 2053951716631130, 2016.

- [112] J. Dean and S. Ghemawat, "Mapreduce: Simplified data processing on large clusters," 2004.
- [113] D. Ostrowski, "Artificial intelligence with big data," in *2018 First International Conference on Artificial Intelligence for Industries (AI4I)*, pp. 125–126, IEEE, 2018.
- [114] J. G. Shanahan and L. Dai, "Large scale distributed data science using apache spark," in *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 2323–2324, 2015.
- [115] M. Zaharia, R. S. Xin, P. Wendell, T. Das, M. Armbrust, A. Dave, X. Meng, J. Rosen, S. Venkataraman, M. J. Franklin, *et al.*, "Apache spark: a unified engine for big data processing," *Communications of the ACM*, vol. 59, no. 11, pp. 56–65, 2016.
- [116] T. Erl, W. Khattak, and P. Buhler, *Big data fundamentals: concepts, drivers & techniques*, vol. 1. Prentice Hall Boston, 2016.
- [117] Y. Ma, H. Wu, L. Wang, B. Huang, R. Ranjan, A. Zomaya, and W. Jie, "Remote sensing big data computing: Challenges and opportunities," *Future Generation Computer Systems*, vol. 51, pp. 47–60, 2015.
- [118] A. Mellit and S. A. Kalogirou, "A survey on the application of artificial intelligence techniques for photovoltaic systems," in *McEvoy's Handbook of Photovoltaics*, pp. 735–761, Elsevier, 2018.
- [119] D. Zhao, L. Ge, M. Qian, D. Jiang, L. Qu, H. Han, Y. Liu, and X. Wang, "Review on modeling of photovoltaic power generation systems," in *2019 IEEE Innovative Smart Grid Technologies-Asia (ISGT Asia)*, pp. 1943–1946, IEEE, 2019.
- [120] Y. LIU, L. ZHANG, and Y. LAILI, "Study on model reuse for complex system simulation," *Scientia Sinica Informationis*, vol. 48, no. 7, pp. 743–766, 2018.
- [121] M. G. Villalva, J. R. Gazoli, and E. Ruppert Filho, "Comprehensive approach to modeling and simulation of photovoltaic arrays," *IEEE Transactions on power electronics*, vol. 24, no. 5, pp. 1198–1208, 2009.
- [122] C. Carrero, J. Amador, and S. Arnaltes, "A single procedure for helping pv designers to select silicon pv modules and evaluate the loss resistances," *Renewable Energy*, vol. 32, no. 15, pp. 2579–2589, 2007.
- [123] E. Karatepe, M. Boztepe, and M. Colak, "Neural network based solar cell model," *Energy conversion and management*, vol. 47, no. 9-10, pp. 1159–1178, 2006.
- [124] T. Kang, J. Yao, S. Yang, T. Duong, and X. Zhu, "Novel cuckoo search algorithm with quasi-oppositional population initialization strategy for solar cell parameters identification," in *2018 13th World Congress on Intelligent Control and Automation (WCICA)*, pp. 911–916, IEEE, 2018.
- [125] K. Dong, J. Yan, W. Shen, S. Li, X. Ma, and R. Jia, "Parameter identification of grid-connected photovoltaic inverter based on adaptive-improved gpsy algorithm," in *2019 IEEE 8th International Conference on Advanced Power System Automation and Protection (APAP)*, pp. 1536–1540, IEEE, 2019.
- [126] Y. Xu and W. Jin, "A parameter identification model for the photovoltaic grid-connected inverter," *Optik*, vol. 142, pp. 23–32, 2017.
- [127] X. Pan, Y. Zhang, P. Ju, Y. Jin, T. Liu, and P. Zeng, "Equivalent modeling for photovoltaic power station," *Power System Technology*, vol. 39, no. 5, pp. 1173–1178, 2015.
- [128] D. Remon, A. M. Cantarellas, and P. Rodriguez, "Equivalent model of large-scale synchronous photovoltaic power plants," *IEEE Transactions on Industry Applications*, vol. 52, no. 6,

- pp. 5029–5040, 2016.
- [129] A. Samadi, L. Söder, E. Shayesteh, and R. Eriksson, “Static equivalent of distribution grids with high penetration of pv systems,” *IEEE Transactions on Smart Grid*, vol. 6, no. 4, pp. 1763–1774, 2015.
- [130] P. Han, Z. Lin, L. Wang, G. Fan, and X. Zhang, “A survey on equivalence modeling for large-scale photovoltaic power plants,” *Energies*, vol. 11, no. 6, p. 1463, 2018.
- [131] H. Li, S. L. Pang, and J. D. Huang, “Equivalent modeling and simulation of large photovoltaic station,” in *Applied Mechanics and Materials*, vol. 615, pp. 27–30, Trans Tech Publ, 2014.
- [132] P. Han, Z. Lin, J. Zhang, Y. Xia, and L. Wang, “Equivalent modeling of photovoltaic power plant based on factor analysis and correlation clustering,” *IEEE Access*, vol. 7, pp. 56935–56946, 2019.
- [133] P. Chao, W. Li, X. Liang, Y. Shuai, F. Sun, and Y. Ge, “A comprehensive review on dynamic equivalent modeling of large photovoltaic power plants,” *Solar Energy*, vol. 210, pp. 87–100, 2020.
- [134] F. Li, Y. Huang, F. Wu, Y. Liu, and X. Zhang, “Research on clustering equivalent modeling of large-scale photovoltaic power plants,” *Chinese Journal of Electrical Engineering*, vol. 4, no. 4, pp. 80–85, 2018.
- [135] Z. Ma, J. Zheng, S. Zhu, X. Shen, L. Wei, X. Wang, and K. Men, “Online clustering modeling of large-scale photovoltaic power plants,” in *2015 IEEE Power & Energy Society General Meeting*, pp. 1–5, IEEE, 2015.
- [136] Z. Ma, J. Zheng, S. Zhu, L. Wei, X. Shen, Z. Wang, and X. Wang, “Online clustering modeling of photovoltaic power plant with lvrt control function,” in *2016 IEEE International Conference on Power System Technology (POWERCON)*, pp. 1–6, IEEE, 2016.
- [137] H. Wu, J. Zhang, C. Luo, and B. Xu, “Equivalent modeling of photovoltaic power station based on canopy-fcm clustering algorithm,” *IEEE Access*, vol. 7, pp. 102911–102920, 2019.
- [138] A. McCallum, K. Nigam, and L. H. Ungar, “Efficient clustering of high-dimensional data sets with application to reference matching,” in *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 169–178, 2000.
- [139] A. Kumar, Y. S. Ingle, A. Pande, and P. Dhule, “Canopy clustering: a review on pre-clustering approach to k-means clustering,” *Int. J. Innov. Adv. Comput. Sci.(IJIACS)*, vol. 3, no. 5, pp. 22–29, 2014.
- [140] W. Wei, S. Huang, H. Guan, R. Su, H. Fan, X. Sun, and L. Ding, “Parameter identification of photovoltaic power plant based on data dimension reduction,” in *2019 IEEE 3rd International Electrical and Energy Conference (CIEEC)*, pp. 1480–1484, IEEE, 2019.
- [141] Y. Zheng, G. Sun, Z. Wei, J. Ma, and Y. Sun, “An active power smoothing control for photovoltaic-flywheel system using fuzzy control,” in *2012 IEEE Fifth International Conference on Advanced Computational Intelligence (ICACI)*, pp. 1113–1118, IEEE, 2012.
- [142] D. Murillo-Yarce, J. Alarcón-Alarcón, M. Rivera, C. Restrepo, J. Muñoz, C. Baier, and P. Wheeler, “A review of control techniques in photovoltaic systems,” *Sustainability*, vol. 12, no. 24, p. 10598, 2020.
- [143] N. N. S. Torres, V. N. S. Diaz, O. H. Ando, and J. J. G. Ledesma, “Analysis of the technical feasibility of using artificial intelligence for smoothing active power in a photovoltaic system connected to the power system,” *Brazilian Archives of Biology and Technology*, vol. 64, 2021.

- [144] P. Tielens and D. Van Hertem, "The relevance of inertia in power systems," *Renewable and Sustainable Energy Reviews*, vol. 55, pp. 999–1009, 2016.
- [145] Q. Peng, A. Sangwongwanich, Y. Yang, and F. Blaabjerg, "Grid-friendly power control for smart photovoltaic systems," *Solar Energy*, vol. 210, pp. 115–127, 2020.
- [146] N. G. M. Thao and K. Uchida, "A novel fuzzy-based control strategy for grid-connected large-scale solar farm with supporting the grid-frequency regulation," in *2015 10th Asian Control Conference (ASCC)*, pp. 1–8, IEEE, 2015.
- [147] C.-C. Lee, "Fuzzy logic in control systems: fuzzy logic controller. i," *IEEE Transactions on systems, man, and cybernetics*, vol. 20, no. 2, pp. 404–418, 1990.
- [148] A. Oudalov, D. Chartouni, and C. Ohler, "Optimizing a battery energy storage system for primary frequency control," *IEEE Transactions on power systems*, vol. 22, no. 3, pp. 1259–1266, 2007.
- [149] R. Rajan and F. M. Fernandez, "Power control strategy of photovoltaic plants for frequency regulation in a hybrid power system," *International journal of electrical power & energy systems*, vol. 110, pp. 171–183, 2019.
- [150] P. Zarina, S. Mishra, and P. Sekhar, "Exploring frequency control capability of a pv system in a hybrid pv-rotating machine-without storage system," *International Journal of Electrical Power & Energy Systems*, vol. 60, pp. 258–267, 2014.
- [151] R. Rajan, F. M. Fernandez, and Y. Yang, "Primary frequency control techniques for large-scale pv-integrated power systems: A review," *Renewable and Sustainable Energy Reviews*, vol. 144, p. 110998, 2021.
- [152] H. Bevrani, F. Habibi, P. Babahajyani, M. Watanabe, and Y. Mitani, "Intelligent frequency control in an ac microgrid: Online pso-based fuzzy tuning approach," *IEEE transactions on smart grid*, vol. 3, no. 4, pp. 1935–1944, 2012.
- [153] A. Abazari, H. Monsef, and B. Wu, "Coordination strategies of distributed energy resources including fess, deg, fc and wtg in load frequency control (lfc) scheme of hybrid isolated micro-grid," *International Journal of Electrical Power & Energy Systems*, vol. 109, pp. 535–547, 2019.
- [154] C. Li, V. R. Disfani, Z. K. Pecenak, S. Mohajeryami, and J. Kleissl, "Optimal oltc voltage control scheme to enable high solar penetrations," *Electric Power Systems Research*, vol. 160, pp. 318–326, 2018.
- [155] D. G. Photovoltaics and E. Storage, "Ieee standard for interconnection and interoperability of distributed energy resources with associated electric power systems interfaces," *IEEE Std*, pp. 1547–2018, 2018.
- [156] C. Li, C. Jin, and R. Sharma, "Coordination of pv smart inverters using deep reinforcement learning for grid voltage regulation," in *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*, pp. 1930–1937, IEEE, 2019.
- [157] S. Wang, J. Duan, D. Shi, C. Xu, H. Li, R. Diao, and Z. Wang, "A data-driven multi-agent autonomous voltage control framework using deep reinforcement learning," *IEEE Transactions on Power Systems*, vol. 35, no. 6, pp. 4644–4654, 2020.
- [158] D. Cao, W. Hu, J. Zhao, Q. Huang, Z. Chen, and F. Blaabjerg, "A multi-agent deep reinforcement learning based voltage regulation using coordinated pv inverters," *IEEE Transactions on Power Systems*, vol. 35, no. 5, pp. 4120–4123, 2020.
- [159] H. J. Liu, W. Shi, and H. Zhu, "Distributed voltage control in distribution networks: Online

- and robust implementations,” *IEEE Transactions on Smart Grid*, vol. 9, no. 6, pp. 6106–6117, 2017.
- [160] D. T. Feeders, “Ieee pes distribution system analysis subcommittee,” *Online Available: <http://www.ewh.ieee.org/soc/pes/dsac/m/testfeeders/index.html>*, 2011.
- [161] X. Chen, Y. Cui, X. Wang, and S. Li, “Research of low voltage ride through control strategy in photovoltaic (pv) grid,” in *2017 Chinese Automation Congress (CAC)*, pp. 5146–5150, IEEE, 2017.
- [162] Y. Bae, T.-K. Vu, and R.-Y. Kim, “Implemental control strategy for grid stabilization of grid-connected pv system based on german grid code in symmetrical low-to-medium voltage network,” *IEEE Transactions on Energy Conversion*, vol. 28, no. 3, pp. 619–631, 2013.
- [163] M. Tsili and S. Papathanassiou, “A review of grid code technical requirements for wind farms,” *IET Renewable power generation*, vol. 3, no. 3, pp. 308–332, 2009.
- [164] S. I. Nanou and S. A. Papathanassiou, “Modeling of a pv system with grid code compatibility,” *Electric Power Systems Research*, vol. 116, pp. 301–310, 2014.
- [165] N. H. Saad, A. A. El-Sattar, and A. E.-A. M. Mansour, “Improved particle swarm optimization for photovoltaic system connected to the grid with low voltage ride through capability,” *Renewable Energy*, vol. 85, pp. 181–194, 2016.
- [166] M. K. Hossain and M. H. Ali, “Fuzzy logic controlled power balancing for low voltage ride-through capability enhancement of large-scale grid-connected pv plants,” in *2017 IEEE Texas Power and Energy Conference (TPEC)*, pp. 1–6, IEEE, 2017.
- [167] T. Eram and P. L. Chapman, “Comparison of photovoltaic array maximum power point tracking techniques,” *IEEE Transactions on energy conversion*, vol. 22, no. 2, pp. 439–449, 2007.
- [168] P. Chandran, R. S. Madhura, J. P. Roselyn, D. Devaraj, V. Gopal, and A. Ravi, “Development of intelligent fuzzy pq-frt control strategy for grid-connected solar pv system,” in *2019 IEEE International Conference on Intelligent Techniques in Control, Optimization and Signal Processing (INCOS)*, pp. 1–5, IEEE, 2019.
- [169] A. Mellit, G. M. Tina, and S. A. Kalogirou, “Fault detection and diagnosis methods for photovoltaic systems: A review,” *Renewable and Sustainable Energy Reviews*, vol. 91, pp. 1–17, 2018.
- [170] N. G. Dhere and N. S. Shiradkar, “Fire hazard and other safety concerns of photovoltaic systems,” *Journal of Photonics for Energy*, vol. 2, no. 1, p. 022006, 2012.
- [171] M. Falvo and S. Capparella, “Safety issues in pv systems: Design choices for a secure fault detection and for preventing fire risk,” *Case Studies in Fire Safety*, vol. 3, pp. 1–16, 2015.
- [172] J. Krueger and D. Smith, “A practical approach to fire hazard analysis for offshore structures,” *Journal of hazardous materials*, vol. 104, no. 1-3, pp. 107–122, 2003.
- [173] H. Al-Sheikh and N. Moubayed, “Fault detection and diagnosis of renewable energy systems: An overview,” in *2012 International Conference on Renewable Energies for Developing Countries (REDEC)*, pp. 1–7, IEEE, 2012.
- [174] G. D. Hasenfus, “Smart sensors for solar panels,” Oct. 23 2012. US Patent 8,294,451.
- [175] A. Triki-Lahiani, A. B.-B. Abdelghani, and I. Slama-Belkhodja, “Fault detection and monitoring systems for photovoltaic installations: A review,” *Renewable and Sustainable Energy Reviews*, vol. 82, pp. 2680–2692, 2018.
- [176] I. U. Khalil, A. Ul-Haq, Y. Mahmoud, M. Jalal, M. Aamir, M. U. Ahsan, and K. Mehmood,



- “Comparative analysis of photovoltaic faults and performance evaluation of its detection techniques,” *IEEE Access*, vol. 8, pp. 26676–26700, 2020.
- [177] Y. Hu and W. Cao, “Theoretical analysis and implementation of photovoltaic fault diagnosis,” in *Renewable Energy-Utilisation and System Integration*, IntechOpen, 2016.
- [178] A. Livera, M. Theristis, G. Makrides, and G. E. Georghiou, “Recent advances in failure diagnosis techniques based on performance data analysis for grid-connected photovoltaic systems,” *Renewable energy*, vol. 133, pp. 126–143, 2019.
- [179] M. Sabbaghpur Arani and M. A. Hejazi, “The comprehensive study of electrical faults in pv arrays,” *Journal of Electrical and Computer Engineering*, vol. 2016, 2016.
- [180] D. S. Pillai and N. Rajasekar, “A comprehensive review on protection challenges and fault diagnosis in pv systems,” *Renewable and Sustainable Energy Reviews*, vol. 91, pp. 18–40, 2018.
- [181] G. M. Tina, F. Cosentino, and C. Ventura, “Monitoring and diagnostics of photovoltaic power plants,” in *Renewable Energy in the Service of Mankind Vol II*, pp. 505–516, Springer, 2016.
- [182] S. Spataru, D. Sera, T. Kerekes, and R. Teodorescu, “Diagnostic method for photovoltaic systems based on light i–v measurements,” *Solar Energy*, vol. 119, pp. 29–44, 2015.
- [183] S. Abe, R. Thawonmas, and M. Kayama, “A fuzzy classifier with ellipsoidal regions for diagnosis problems,” *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 29, no. 1, pp. 140–148, 1999.
- [184] L. I. Kuncheva, “How good are fuzzy if-then classifiers?,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 30, no. 4, pp. 501–509, 2000.
- [185] M. Dhimish, V. Holmes, B. Mehrdadi, and M. Dales, “Comparing mamdani sugeno fuzzy logic and rbf ann network for pv fault detection,” *Renewable energy*, vol. 117, pp. 257–274, 2018.
- [186] M. Dhimish and G. Badran, “Photovoltaic hot-spots fault detection algorithm using fuzzy systems,” *IEEE Transactions on Device and Materials Reliability*, vol. 19, no. 4, pp. 671–679, 2019.
- [187] Q. Zhao, S. Shao, L. Lu, X. Liu, and H. Zhu, “A new pv array fault diagnosis method using fuzzy c-mean clustering and fuzzy membership algorithm,” *Energies*, vol. 11, no. 1, p. 238, 2018.
- [188] B. Li, C. Delpha, D. Diallo, and A. Migan-Dubois, “Application of artificial neural networks to photovoltaic fault detection and diagnosis: A review,” *Renewable and Sustainable Energy Reviews*, p. 110512, 2020.
- [189] B. Basnet, H. Chun, and J. Bang, “An intelligent fault detection model for fault detection in photovoltaic systems,” *Journal of Sensors*, vol. 2020, 2020.
- [190] M. N. Akram and S. Lotfifard, “Modeling and health monitoring of dc side of photovoltaic array,” *IEEE Transactions on Sustainable Energy*, vol. 6, no. 4, pp. 1245–1253, 2015.
- [191] S. Hu, F. Wang, H. Guo, and W. Zhu, “A fault diagnosis method for photovoltaic modules based on transfer long short-term memory neural network,” in *2021 6th Asia Conference on Power and Electrical Engineering (ACPEE)*, pp. 46–51, IEEE, 2021.
- [192] S. J. Pan and Q. Yang, “A survey on transfer learning,” *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2009.
- [193] L. Yang, S. Yao, Y. Teng, X. Yang, X. Li, and J. Ren, “Research on fault diagnosis of photovoltaic power generation system based on cloud computing data,” in *2016 International*

- Conference on Smart City and Systems Engineering (ICSCSE)*, pp. 594–597, IEEE, 2016.
- [194] Y. Xu, Y. Sun, J. Wan, X. Liu, and Z. Song, “Industrial big data for fault diagnosis: Taxonomy, review, and applications,” *IEEE Access*, vol. 5, pp. 17368–17380, 2017.
- [195] X. Li, Q. Yang, Z. Lou, and W. Yan, “Deep learning based module defect analysis for large-scale photovoltaic farms,” *IEEE Transactions on Energy Conversion*, vol. 34, no. 1, pp. 520–529, 2018.
- [196] H.-C. Shin, H. R. Roth, M. Gao, L. Lu, Z. Xu, I. Nogues, J. Yao, D. Mollura, and R. M. Summers, “Deep convolutional neural networks for computer-aided detection: Cnn architectures, dataset characteristics and transfer learning,” *IEEE transactions on medical imaging*, vol. 35, no. 5, pp. 1285–1298, 2016.
- [197] X. Li, Q. Yang, Z. Chen, X. Luo, and W. Yan, “Visible defects detection based on uav-based inspection in large-scale photovoltaic systems,” *IET Renewable Power Generation*, vol. 11, no. 10, pp. 1234–1244, 2017.
- [198] S. N. Venkatesh and V. Sugumaran, “Fault detection in aerial images of photovoltaic modules based on deep learning,” in *IOP Conference Series: Materials Science and Engineering*, vol. 1012, p. 012030, IOP Publishing, 2021.
- [199] H. Chen, Y. Pang, Q. Hu, and K. Liu, “Solar cell surface defect inspection based on multispectral convolutional neural network,” *Journal of Intelligent Manufacturing*, vol. 31, no. 2, pp. 453–468, 2020.
- [200] V. B. Kurukuru, A. Haque, M. A. Khan, and A. K. Tripathy, “Fault classification for photovoltaic modules using thermography and machine learning techniques,” in *2019 International Conference on Computer and Information Sciences (ICCIS)*, pp. 1–6, IEEE, 2019.
- [201] M. W. Akram, G. Li, Y. Jin, X. Chen, C. Zhu, and A. Ahmad, “Automatic detection of photovoltaic module defects in infrared images with isolated and develop-model transfer deep learning,” *Solar Energy*, vol. 198, pp. 175–186, 2020.
- [202] D. Manno, G. Cipriani, G. Ciulla, V. Di Dio, S. Guarino, and V. L. Brano, “Deep learning strategies for automatic fault diagnosis in photovoltaic systems by thermographic images,” *Energy Conversion and Management*, vol. 241, p. 114315, 2021.
- [203] D. Kovalev, I. Shanin, S. Stupnikov, and V. Zakharov, “Data mining methods and techniques for fault detection and predictive maintenance in housing and utility infrastructure,” in *2018 International Conference on Engineering Technologies and Computer Science (EnT)*, pp. 47–52, IEEE, 2018.
- [204] S. Rodrigues, H. G. Ramos, and F. Morgado-Dias, “Machine learning in pv fault detection, diagnostics and prognostics: A review,” in *2017 IEEE 44th Photovoltaic Specialist Conference (PVSC)*, pp. 3178–3183, IEEE, 2017.
- [205] A. Chokor, M. El Asmar, and S. V. Lokanath, “A review of photovoltaic dc systems prognostics and health management: Challenges and opportunities,” in *Proceedings of the Annual Conference of the Prognostics and Health Management Society, Denver, CO, USA*, pp. 3–6, 2016.
- [206] N. Laayouj, H. Jamouli, and M. El Hail, “New prognostic framework for degradation assessment and remaining useful life estimation of photovoltaic module,” in *2016 24th Mediterranean Conference on Control and Automation (MED)*, pp. 378–383, IEEE, 2016.
- [207] N. Clements, “Introduction to prognostics,” in *Annual Conference of the PHM Society Montreal, Canada*, 2011.

- [208] A. Golnas, “Pv system reliability: An operator’s perspective,” in *2012 IEEE 38th Photovoltaic Specialists Conference (PVSC) PART 2*, pp. 1–6, IEEE, 2012.
- [209] D. C. Jordan, B. Marion, C. Deline, T. Barnes, and M. Bolinger, “Pv field reliability status—analysis of 100 000 solar systems,” *Progress in Photovoltaics: Research and Applications*, vol. 28, no. 8, pp. 739–754, 2020.
- [210] T. Gunda, S. Hackett, L. Kraus, C. Downs, R. Jones, C. McNalley, M. Bolen, and A. Walker, “A machine learning evaluation of maintenance records for common failure modes in pv inverters,” *IEEE Access*, vol. 8, pp. 211610–211620, 2020.
- [211] N. Hardeniya, J. Perkins, D. Chopra, N. Joshi, and I. Mathur, *Natural language processing: python and NLTK*. Packt Publishing Ltd, 2016.
- [212] J. Lilleberg, Y. Zhu, and Y. Zhang, “Support vector machines and word2vec for text classification with semantic features,” in *2015 IEEE 14th International Conference on Cognitive Informatics & Cognitive Computing (ICCI\* CC)*, pp. 136–140, IEEE, 2015.
- [213] W. Rezgui, N. K. Mouss, L.-H. Mouss, M. D. Mouss, and M. Benbouzid, “A regression algorithm for the smart prognosis of a reversed polarity fault in a photovoltaic generator,” in *2014 First International Conference on Green Energy ICGE 2014*, pp. 134–138, IEEE, 2014.
- [214] M. De Benedetti, F. Leonardi, F. Messina, C. Santoro, and A. Vasilakos, “Anomaly detection and predictive maintenance for photovoltaic systems,” *Neurocomputing*, vol. 310, pp. 59–68, 2018.



# Glosario

---

- ABC** Artificial Bee Colony. 16
- ARIMA** Autoregressive Integrated Moving Average. 44
- CC** Correlation Clustering. 8
- CNN** Convolutional Neural Network. 12
- CS** Cuckoo Search. 16
- DE** Differential Evolution. 15
- FA** Factorial Analysis. 9
- FCM** Fuzzy C-Means. 20
- FV** Fotovoltaico/a. 23
- GA** Genetic Algorithm. 15
- IA** Inteligencia Artificial. 1
- IoT** Internet of Things. 37
- LSTM** Long Short-Term Memory. 13
- LVRT** Low voltage ride-through. 28
- MADRL** Multi-Agent Deep Reinforcement Learning. 34
- MPPT** Maximum Power Point Tracker. 33
- NLP** Natural Language Processing. 20
- PCA** Principal Component Analysis. 8
- PSO** Particle Swarm Optimization. 15
- RNN** Recurrent Neural Network. 13
- RUL** Remaining useful life. 44

**RVM** Relevance Vector Machine. 6

**SA** Simulated Annealing. 17

**SDM** Single-diode model. 23

**SI** Smart Inverter. 34

**SVM** Support Vector Machine. 6

**SVR** Support Vector Regression. 6