



POLITÉCNICA



**UNIVERSIDAD POLITÉCNICA DE MADRID  
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA  
AERONÁUTICA Y DEL ESPACIO  
GRADO EN INGENIERÍA AEROESPACIAL**

**TRABAJO FIN DE GRADO**

**Detección y análisis de impactos sobre estructuras de  
material compuesto con sensores integrados**

**AUTOR: Daniel DEL RÍO VELILLA**

**ESPECIALIDAD: Ciencias y Tecnologías Aeroespaciales**

**TUTOR DEL TRABAJO: Antonio FERNÁNDEZ LÓPEZ**

**Julio de 2020**



# Agradecimientos

Me gustaría agradecer el apoyo de todas a las personas que me han ayudado a llegar hasta aquí.

A Antonio por haber confiado en mí desde el primer día hasta ahora.

A todos aquellos que durante estos años habéis pasado a ser mi familia y aún en este complicado año habéis estado a mi lado en la distancia.

A mis padres, mi hermano y al resto de mi familia por estar conmigo tanto en los momentos complicados como en los felices.



# Índice general

<b>1. Introducción y motivación</b>	<b>1</b>
<b>2. Estado del arte</b>	<b>3</b>
2.1. Structural Health Monitoring . . . . .	3
2.2. Monitorización de estructuras aeronáuticas . . . . .	5
2.2.1. Estructuras aeronáuticas . . . . .	5
2.2.2. Daños en estructuras aeronáuticas . . . . .	6
2.2.3. Característica sensibles al daño . . . . .	9
2.2.4. Tecnología SHM aplicable para una DSF . . . . .	9
2.3. Estado del arte de la IA . . . . .	12
2.3.1. Representación gráfica del funcionamiento de una Red Neuronal . . . . .	15
2.3.2. Software . . . . .	22
2.4. IA aplicada a la monitorización estructural . . . . .	24
2.4.1. SHM basado en vibraciones con DL . . . . .	27
2.4.2. SHM basado en visión con DL . . . . .	29
2.4.3. Aplicación de UAVs y teléfonos móviles para SHM con DL . . . . .	30
2.4.4. Transfer Learning (TL) a partir de modelos pre-entrenados . . . . .	32
2.5. IA aplicada a la monitorización de estructuras aeronáuticas . . . . .	33
<b>3. Aplicación de Deep Learning en SHM con sensores de deformación</b>	<b>36</b>
3.1. Proyecto INESASSE . . . . .	36
3.1.1. Estructura y red de sensores . . . . .	36
3.1.2. Planteamiento de daños y descripción de ensayos . . . . .	39
3.1.3. Modelo de elementos finitos . . . . .	40
3.1.4. Pre-procesado de señales . . . . .	42
3.2. Aplicación de DL a la detección de daños mediante sensores de deformaciones . . . . .	49
3.2.1. Visualización del campo de deformaciones . . . . .	49
3.2.2. Arquitectura de la Red Neuronal . . . . .	54
3.2.3. Resultados de la clasificación . . . . .	58
3.3. Aplicación de DL a la detección de cargas térmicas . . . . .	62
3.3.1. Estructura y modelo de elementos finitos . . . . .	62
3.3.2. Definición de daños y red sensorial . . . . .	63
3.3.3. Pre-procesado de señales . . . . .	64
3.3.4. Visualización del campo de deformaciones . . . . .	67
3.3.5. Arquitectura de la red . . . . .	70
3.3.6. Resultados de la clasificación . . . . .	71

<b>4. Caracterización de impactos con Deep Learning</b>	<b>74</b>
4.1. Generative Adversarial Networks . . . . .	76
4.1.1. Modelo generativo de secuencias temporales . . . . .	78
4.1.2. Arquitectura TimeGAN . . . . .	79
4.1.3. Resultados de la generación de impactos con TimeGAN . . . . .	80
4.2. Implementación del localizador de impactos . . . . .	88
4.2.1. Arquitectura del clasificador . . . . .	88
4.2.2. Resultados . . . . .	89
<b>5. Conclusiones y trabajos futuros</b>	<b>94</b>
5.1. Conclusiones . . . . .	94
5.2. Trabajos futuros . . . . .	97

# Índice de figuras

2.1.	SHM como sistema realimentado . . . . .	4
2.2.	Problemas asociados a las estructuras de una aeronave . . . . .	5
2.3.	Criterios de diseño [22] . . . . .	6
2.4.	Sección típica de un laminado ( $0^\circ/90^\circ$ ) <sub>8</sub> , fracción de en volumen de vacíos 0.31 %. [3] . . . . .	7
2.5.	Micrografía de la rotura de matriz en un laminado multidireccional [34] . . . . .	8
2.6.	Micrografía de delaminación en un laminado multidireccional [34] . . . . .	8
2.7.	Micrografía de la fractura de fibra en un laminado multidireccional [34] . . . . .	9
2.8.	Conjunto de daños por impacto en un laminado multidireccional [3] . . . . .	9
2.9.	Tabla de daños y DSF . . . . .	10
2.10.	Morfología de redes sensoriales de fibra óptica [37] . . . . .	11
2.11.	Representación esquemática de la estructura atómica del PZT antes y después de ser polarizado [2] . . . . .	11
2.12.	Ejemplo de Inteligencia Artificial débil . . . . .	12
2.13.	Descripción del aprendizaje supervisado . . . . .	13
2.14.	Descripción del aprendizaje no supervisado . . . . .	14
2.15.	Descripción del aprendizaje reforzado . . . . .	15
2.16.	Representación de las dos clases diferentes en dos colores . . . . .	16
2.17.	Representación del <i>perceptron</i> . . . . .	16
2.18.	Clasificación de datos con una sola neurona . . . . .	17
2.19.	Representación de las capas de una red . . . . .	17
2.20.	Red con dos <i>Hidden layers</i> y dos neuronas en cada capa . . . . .	18
2.21.	Efecto de la función de activación sobre el output de una neurona . . . . .	18
2.22.	Funciones de activación más populares . . . . .	19
2.23.	Combinación de funciones de activación . . . . .	19
2.24.	Combinación de funciones de activación . . . . .	20
2.25.	Resultado final de la clasificación . . . . .	20
2.26.	Influencia de un parámetro en la red . . . . .	21
2.27.	primer pase de backpropagation . . . . .	21
2.28.	Segundo pase . . . . .	22
2.29.	Tercer pase . . . . .	22
2.30.	Cuarto pase . . . . .	22
2.31.	Lista de lenguajes más usados para ML en GitHub . . . . .	23
2.32.	Paquetes de ML . . . . .	23
2.33.	Comparación de técnicas diferentes de SHM [5] . . . . .	25
2.34.	Mapa mental para la selección de algoritmos de Machine Learning [5] . . . . .	26
2.35.	Estructura bajo estudio con la posición de los acelerómetros [9] . . . . .	27
2.36.	Acelerómetro usado para medir las vibraciones en la estructura [9] . . . . .	28
2.37.	Proceso de detección de anomalías [12] . . . . .	28

2.38. Estructura estudiada en [12] . . . . .	29
2.39. Combinación de SURF (ML) y CNN (DL) para la extracción de características de imágenes con grietas [16] . . . . .	30
2.40. Resumen de la técnica desarrollada en [13] . . . . .	31
2.41. Resumen del experimento propuesto en [44] . . . . .	31
2.42. Resultados de la detección de grietas [44] . . . . .	32
2.43. Resultados de la detección de grietas [8] . . . . .	33
2.44. Arquitectura de la red clasificadora [8] . . . . .	33
2.45. Arquitectura de la red clasificadora [18] . . . . .	34
2.46. A310 Boom demostrador [15] . . . . .	34
2.47. Comparación de deformaciones del fuselaje medidas y generadas por la red [15] . . . . .	35
 3.1. UAV INTA Milano . . . . .	37
3.2. Fuselaje posterior . . . . .	37
3.3. Red de sensores FBG . . . . .	38
3.4. Red de sensores OBR . . . . .	38
3.5. Esquema de los daños realizados a la estructura . . . . .	39
3.6. Realización del daño por extracción de remaches . . . . .	40
3.7. Vista general de la configuración de los ensayos . . . . .	41
3.8. Modelo de elementos finitos del fuselaje posterior del UAV Milano . . . . .	41
3.9. Medida de todos los sensores durante el ensayo del Daño 1 con un remache extraído . . . . .	42
3.10. Detección de los escalones producidos por la carga vertical . . . . .	43
3.11. Medidas de los sensores FBG en el ensayo preprocesadas . . . . .	43
3.12. Histograma del escalón 4 kN, 100 % de carga . . . . .	44
3.13. Campo de deformaciones medido por los sensores FBG para todos los daños	45
3.14. Diferencia de los campos de deformaciones referenciado con el estado sin daño (FBG) . . . . .	45
3.15. Campo de deformaciones provocado por el daño 1 con 3 remaches extraídos para todos los casos de carga . . . . .	46
3.16. Aplicación del filtro Savitzky-Golay a una linea de sensor OBR . . . . .	46
3.17. Distribución de ruido gausiano con amplitud 1.5 deformaciones . . . . .	47
3.18. Detalle de las variaciones provocadas por el ruido gausiano . . . . .	47
3.19. Campo de deformaciones medido por el sensor OBR para todos los daños .	48
3.20. Diferencia de los campos de deformaciones referenciado con el estado sin daño (OBR) . . . . .	48
3.21. Visualización de los distintos tipos de daño . . . . .	50
3.22. Visualización de los distintos tamaños de daño . . . . .	51
3.23. Visualización de los distintos niveles de carga . . . . .	51
3.24. Visualización de los distintos tipos de daño . . . . .	52
3.25. Visualización de los distintos tamaños de daño . . . . .	52
3.26. Visualización de los distintos niveles de carga . . . . .	53
3.27. Posibles combinaciones de secuencias con RNN [38] . . . . .	54
3.28. RNN “unrolled”[39] . . . . .	54
3.29. Configuración interna de la red LSTM [42] . . . . .	55
3.30. Arquitecturas utilizadas en las redes de los sensores FBG . . . . .	56
3.31. Arquitecturas utilizadas en las redes de los sensores OBR . . . . .	57
3.32. Matriz de confusión: Tamaños de daño FBG . . . . .	58

3.33. Matriz de confusión: Nivel de carga FBG . . . . .	59
3.34. Matriz de confusión: Tamaños de daño OBR . . . . .	60
3.35. Matriz de confusión: Nivel de carga OBR . . . . .	61
3.36. Diagramas de estructuras aeronáuticas . . . . .	62
3.37. Imagen de la viga que se va a estudiar . . . . .	63
3.38. Viga con condiciones de contorno aplicadas . . . . .	64
3.39. EJEMPLOS DE LOS DAÑOS . . . . .	64
3.40. Tiras de sensores de los que se van a extraer las deformaciones . . . . .	65
3.41. Campos de deformaciones producidos por la carga vertical y temperatura . .	66
3.42. Campos de deformaciones producidos por la carga vertical y temperatura .	66
3.43. Detalle de la linea de sensor 4 . . . . .	67
3.44. Visualización de los diferentes daños . . . . .	68
3.45. Visualización de los diferentes tamaños de daño . . . . .	68
3.46. Visualización de las diferentes cagás . . . . .	69
3.47. Visualización de las diferentes temperaturas . . . . .	69
3.48. Arquitecturas utilizadas en las redes de los sensores en el FEM . . . . .	71
3.49. Matriz de confusión: nivel de carga . . . . .	72
3.50. Matriz de confusión: temperatura distribuida . . . . .	72
3.51. Matriz de confusión: tamaño de daño . . . . .	73
4.1. Costillas en una semiala del Airbus A380 . . . . .	75
4.2. Estructura en estudio instrumentada . . . . .	75
4.3. Discretización de la superficie de la costilla . . . . .	76
4.4. Representación simplificada de la arquitectura de una red GAN . . . . .	77
4.5. Diferentes ejemplos de impactos . . . . .	78
4.6. Diagrama de bloques del funcionamiento de TimeGAN . . . . .	81
4.7. Diferentes ejemplos de impactos generados sintéticamente con TimeGAN .	82
4.8. t-SNE de las celdas 11, 12, 13, 14 . . . . .	83
4.9. t-SNE de las celdas 21, 22, 23, 24 . . . . .	83
4.10. t-SNE de las celdas 31, 32, 33, 34 . . . . .	84
4.11. t-SNE de las celdas 41, 42, 43, 44 . . . . .	84
4.12. t-SNE de las celdas 55, 56, 57, 58 . . . . .	85
4.13. t-SNE de las celdas 65, 66, 67, 68 . . . . .	85
4.14. t-SNE de las celdas 75, 76, 77, 78 . . . . .	86
4.15. t-SNE de las celdas 85, 86, 87, 88 . . . . .	86
4.16. t-SNE de las celdas 95, 96, 97, 98 . . . . .	87
4.17. Arquitectura utilizada para el localizador de impactos . . . . .	88
4.18. Matriz de confusión de impactos sintéticos 1 . . . . .	90
4.19. Matriz de confusión de impactos sintéticos 2 . . . . .	91
4.20. Matriz de confusión de impactos reales 1 . . . . .	92
4.21. Matriz de confusión de impactos sintéticos 2 . . . . .	93

# Índice de tablas

3.1. Definición de daños en el fuselaje . . . . .	39
3.2. Laminados . . . . .	63

# Capítulo 1

## Introducción y motivación

Tanto en el sector aeronáutico como espacial, las aeronaves y sistemas son diseñados siguiendo un método basado en admisibles y reglas de diseño muy conservativas. Con esto se busca que, bajo una larga lista de condiciones de diseño que varían dependiendo del tipo de aeronave, se garantice la integridad estructural y su correcto funcionamiento durante el transcurso de cada de misión.

Estos admisibles de diseño pueden ir desde defectos estructurales complejos de gran importancia en la seguridad del conjunto de la aeronave (pérdida de un motor), hasta defectos con una pequeña influencia estructural pero que, con el tiempo, pueden convertirse en catastróficos (rotura de un pequeño remache).

La pérdida de un motor tiene influencia global y de una magnitud apreciable, siendo así fácilmente detectable. Sin embargo, cuando un remache se rompe, su influencia es local y de pequeña magnitud.

Para poder detectar o cuantificar el tamaño de un defecto en una estructura se comparan determinadas magnitudes entre la estructura dañada y la sana. La forma más sencilla de hacer esta comparación es realizando una inspección visual, pero este procedimiento solo permite detectar defectos que han producido cambios evidentes sobre la estructura quedando ocultos aquellos que no se detectan a simple vista.

Los mantenimientos preventivos son necesarios debido a la dificultad de detectar aquellos defectos que no son evidentes. Durante los mantenimientos se utilizan técnicas de inspección no destructivas (NDI) con las que se recogen datos que son procesados para asegurar la integridad estructural de la aeronave.

Sin embargo, durante los últimos años y, como una evolución de las técnicas NDI, ha habido un gran desarrollo de sistemas cuya función es evaluar el estado de las estructuras de una aeronave o vehículo espacial durante la operación y a tiempo real mediante el uso de sensores integrados en las propias estructuras. Estos son los llamados *Structural Health Monitoring Systems* (SHMS).

La utilización de estos sistemas tiene ventajas importantes frente al uso de NDI. Permiten la monitorización de las estructuras sin la intervención de inspectores y sin tener que desensamblarla para analizarla. Esto hace que el tiempo entre la detección de un defecto su reparación sea muy corto, haciendo que este no aumente y evitando que termine provocando un fallo catastrófico. Entonces, si el SHMS no detecta ningún tipo de defecto, se llega a la conclusión de que la aeronave no necesita ser revisada en profundidad con tanta frecuencia por lo que los períodos de mantenimiento pueden ser separados en el tiempo y,

mientras tanto, la aeronave sigue operando cumpliendo con los admisibles de diseño.

Por otra parte, la tendencia del sector aeroespacial desde hace varias décadas ha sido aumentar el porcentaje en peso del avión fabricado con materiales compuestos. Los materiales compuestos tienen unas propiedades específicas muy superiores a metales como el aluminio, por otro lado, presentan modos de fallo más complejos, variados y más difíciles de detectar comparados a los que sufren los materiales metálicos.

El aumento del uso de materiales compuestos, junto con el potencial de los SHMS ha sido el impulsor de este Trabajo Fin de Grado. El objetivo es desarrollar herramientas fiables basadas en algoritmos de Inteligencia Artificial (IA) para poder detectar defectos en estructuras aeronáuticas, más concretamente, en estructuras aeronáuticas con geometría compleja fabricadas con material compuesto.

Grandes empresas como Google, Facebook, Microsoft o Nvidia han invertido mucho esfuerzo en el campo de la IA durante los últimos años y han conseguido sorprendentes resultados. Por ejemplo, el NVidia RTX Voice, un cancelador de ruido basado en algoritmos de Deep Learning (DL).

Siguiendo esta línea de investigación en DL, se va a explorar la posibilidad de utilizar este tipo de algoritmos para desarrollar dos herramientas de SHM.

La primera tendrá que ser capaz de clasificar diferentes estados de una estructura basándose en medidas de deformación. A su vez, se pretende detectar el nivel de carga al que está sometida la estructura y a la temperatura que se encuentra.

Por otra parte, se desarrollará una segunda herramienta para localizar impactos en estructuras complejas.

El lado negativo del uso de IA para SHM es que se requiere un conjunto de datos (Data Set, DS) muy grande para poder entrenar los algoritmos y conseguir un nivel de confianza elevado en los resultados.

Usando de ejemplo la herramienta localizadora de impactos y teniendo en cuenta que se divide en una malla de  $10 \times 10$  con 100 impactos por celda, se llegarían a tener una cantidad mínima de  $10^5$  impactos. Esta cantidad es tan elevada que hace imposible la realización de los impactos de forma manual, por lo que su automatización es necesaria.

Para evitar esta tarea se va a proponer el uso de otro tipo de algoritmo de DL llamado GAN. Con este tipo de redes se intentará reducir la cantidad de impactos que se tengan que realizar de forma manual manteniendo la cantidad deseada de impactos totales que alimentarán la herramienta de localización de impactos.

# Capítulo 2

## Estado del arte

En este capítulo se va a realizar una introducción teórica a los pilares en los que se asienta este trabajo, SHM e IA, y sus aplicaciones en la industria aeroespacial.

### 2.1. Structural Health Monitoring

Structural Helath Monitoring (SHM) es el proceso de identificar daños en estructuras de forma no destructiva mediante el uso de sensores integrados [36]. Por lo tanto, es necesario tener una definición clara del concepto de daño. Se puede definir un daño como los cambios en el material y/o en las propiedades geométricas de un sistema estructural, incluyendo cambios en las condiciones de contorno y conectividad del mismo, que pueden afectar de manera adversa al funcionamiento presente y futuro del sistema [31].

La gran ventaja de este proceso frente a las NDI de las que evoluciona es que pueden ser aplicadas en tiempo real, incluyendo durante la operación del vehículo. Con esto se consigue, no solo tener un conocimiento total de la salud estructural, sino que también se puede llegar a monitorizar la magnitud de las cargas a las que las estructuras instrumentadas están sometidas.

El SHM se realiza en cuatro pasos principales:

1. **Evacuación operacional:** para aplicar la técnica de SHM, la evaluación operativa es el paso fundamental. Estudia los problemas y sus consecuencias que conducen a monitorizar las estructuras. La evaluación estructural trata de buscar solución a varias preguntas, tales como:
  - ¿qué tipo de daño se busca en la estructura monitorizada?
  - ¿a qué condiciones ambientales y operativas está sometido la estructura monitorizada mientras se lleva a cabo el proceso de SHM?
  - ¿Cuáles son las restricciones de la transferencia de datos a través del sistema SHM a causa de la operación del vehículo?
  - ¿cuáles son las ventajas del SHM en la vida segura del vehículo?
2. **Adquisición de datos:** el proceso de adquisición de datos es una parte crucial en SHM y se realiza a través de sensores integrados en la estructura. El tipo de sensores que se utilizarán está ligado a la técnica de SHM usada. En esta fase se estudia el número de sensores usados, su distribución en la estructura y el tipo de magnitud medida.

3. **Procesado de señales y extracción de características debidas al daño:** la parte más crítica del SHM es extraer las inferencias a partir de las señales recogidas por los sensores. Este proceso se basa en descubrir qué propiedades y áreas son afectadas por el daño junto con como varían a medida que el daño evoluciona.
4. **Modelado estadístico para evaluación de las características:** el desarrollo de un modelo estadístico es el paso final del SHM. Este proceso incluye dos categorías:
  - Cuando los datos recopilados pertenecen tanto a partes dañadas como a no dañadas, el modelado estadístico hace una clasificación general también conocida como aprendizaje supervisado, como puede ser el análisis de regresión y clasificación grupal.
  - Cuando los datos proceden de estructuras en las que no se han caracterizado los daños, se habla de aprendizaje no supervisado.

Desde un punto de vista matemático, los sistemas SHM se pueden considerar como sistemas de control realimentado, diferenciándose de éstos en los tiempos característicos y el modo de actuación sobre la planta.

Las escalas temporales dependen de lo que el sistema busca controlar. En el caso de una grieta, los tiempos característicos son del orden de la vida operativa de la planta (estructura), en cambio, si se quiere un sistema de localización y detección de impactos, el tiempo característico se reduce a decenas de milisegundos.

A su vez, también cambian las características espaciales, dependiendo fundamentalmente de la criticidad del daño. La longitud característica dependerá de la tolerancia al daño de la estructura [21].

Para funcionar, un sistema SHM medirá todas las entradas y salidas que afectan a la planta monitorizada, entendiéndose como planta aquella cuya función sea soportar o transmitir cargas, siendo fija o móvil. Los parámetros de observación serían los parámetros característicos de operación propios y los relativos al entorno. La situación anómala será aquella que comprometa el funcionamiento presente y futuro de la planta, en su conjunto, dentro de las condiciones de diseño [24].

Así pues, tenemos que el SHM es en esencia un sistema autónomo para inspeccionar y detectar daños en estructuras con una mínima intervención humana. En la Figura 2.1 se puede ver representada esta idea [21].

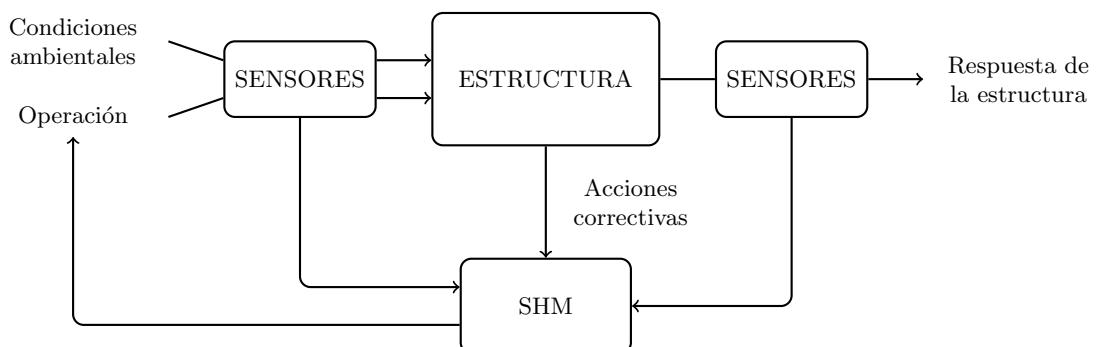


Figura 2.1: SHM como sistema realimentado

## 2.2. Monitorización de estructuras aeronáuticas

Dentro de la gran variedad de estructuras en las que se puede aplicar el SHM, este trabajo se va a centrar en estructuras aeronáuticas de geometría compleja fabricadas en material compuesto. Como ya se ha introducido en el primer paso del SHM, *Evaluación operacional*, hay una serie de preguntas que se necesitan responder para elegir el sistema óptimo con el que monitorizar una estructura determinada dentro de todas las que componen una aeronave.

Será necesario dar respuesta a las siguientes cuestiones:

1. ¿Qué estructura se quiere monitorizar?
2. ¿Qué tipo de daño afecta a la integridad de dicha estructura?
3. ¿Cuál va a ser la característica sensitiva al daño, o Damage Sensitive Feature (DSF), la que se va a usar? El DSF tendrá que ser alguna característica o parámetro medible de la estructura que sea modificado por la presencia del daño.
4. ¿De qué modo el daño influye en la DSF?

Una vez que todas estas preguntas tengan respuesta, se tendrá toda la información necesaria para elegir una estrategia global de SHM apropiada para el objetivo que se ha fijado. A continuación, se van a responder estas cuestiones para el caso de estructuras aeronáuticas.

### 2.2.1. Estructuras aeronáuticas

No todas las estructuras que componen una aeronave están sometidas al mismo tipo de cargas y, por lo tanto, no presentan los mismos problemas ni se usan los mismos criterios de diseño para toda la aeronave.

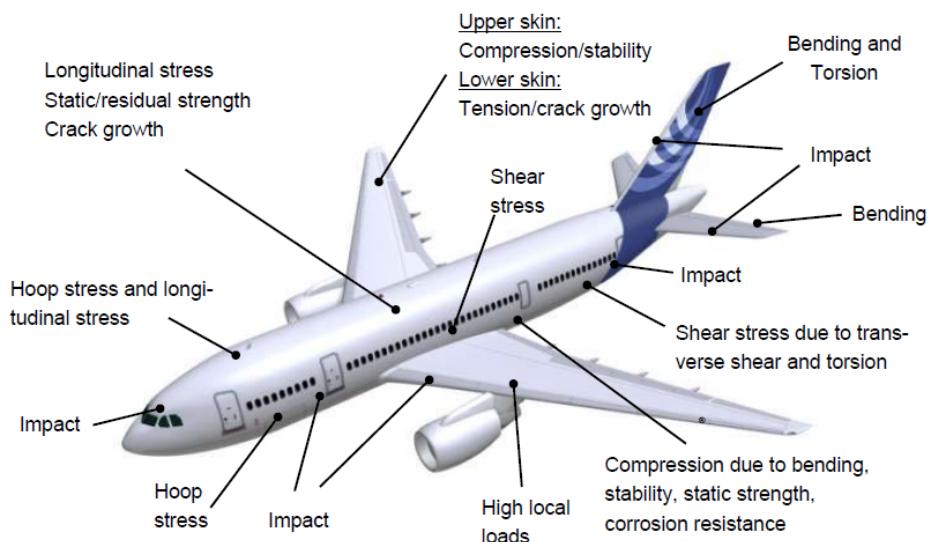


Figura 2.2: Problemas asociados a las estructuras de una aeronave

En aeronáutica, la inmensa mayoría de partes se diseñan a fatiga. Esto quiere decir que, para un número determinado de ciclos y con un espectro determinado de cargas, la pieza o estructura no compromete la integridad global de la aeronave.

Dentro del diseño a fatiga hay otras dos tendencias, el diseño a vida segura o a tolerancia al daño. En las estructuras diseñadas bajo el criterio de vida segura granizan que, mientras no se supere el espectro de carga, la integridad estructural no peligra. En cambio, el diseño de tolerancia al daño tiene en cuenta que en el propio proceso de fabricación se generan grietas en el material y asume que estas grietas van a crecer, por lo que toma medidas para asegurar que no se produzcan fallos catastróficos [22].

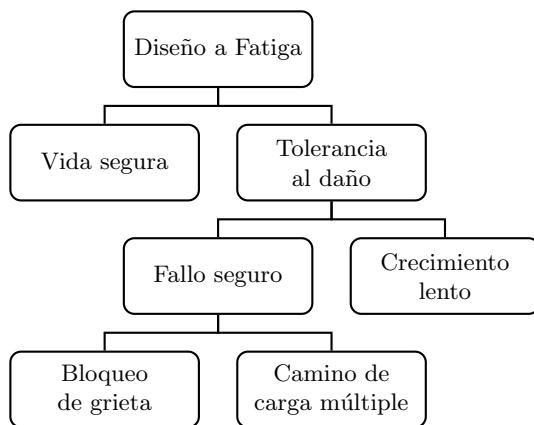


Figura 2.3: Criterios de diseño [22]

Usando las Figuras 2.2 y 2.3 se puede tener una idea general de la estructura bajo estudio y responder a la primera pregunta. El siguiente paso será diferenciar entre los distintos daños que pueden sufrir estas estructuras.

### 2.2.2. Daños en estructuras aeronáuticas

Como se ha definido anteriormente, se define un daño como los cambios en el material y/o en las propiedades geométricas de un sistema estructural, incluyendo cambios en las condiciones de contorno y conectividad del mismo, que pueden afectar de manera adversa al funcionamiento presente y futuro del sistema.

Por otro lado, llama material compuesto a aquel que está formado por dos o más componentes, de forma que las propiedades del nuevo material sean superiores a las de los componentes por separado.

Los materiales individuales se pueden agrupar en dos grupos:

- Matriz: configura geométricamente la pieza, da cohesión al material y transmite los esfuerzos de unas fibras a otras.
- Refuerzo: aporta rigidez y resistencia.

En aeronáutica la combinación de materiales más común que forma el material compuesto es la resina Epoxy como matriz y la fibra de carbono como refuerzo. Esta combinación tiene una densidad que ronda los  $1,5 \text{ g/cm}^3$  con un módulo elástico entre 120 y 400 GPa y unas propiedades específicas superiores a las de aleaciones ligeras como aluminio o titanio.

Los daños que se producen en los materiales metálicos son diferentes a los de material compuesto. Por lo que en esta sección solo se van a comentar algunos de los daños más característicos que sufre este tipo de material.

Durante el propio proceso de fabricación del material pueden aparecer defectos, al igual que ocurre en los materiales metálicos. Los defectos más comunes suelen ser poros o impurezas que se quedan atrapadas entre las capas en el momento de ser apiladas.

En la Figura 2.4 se puede apreciar que los poros (V) tienden a estar localizados entre las pieles del compuesto, cerca de las zonas ricas en resina (R). Poros muy pequeños pueden estar presente dentro de las hebras de fibra (B). En las zonas ricas en resina los poros son más grandes (macroporos, 30-600  $\mu\text{m}$ ) que los que aparecen en las hebras de fibra (microporo, 7-30  $\mu\text{m}$ ).

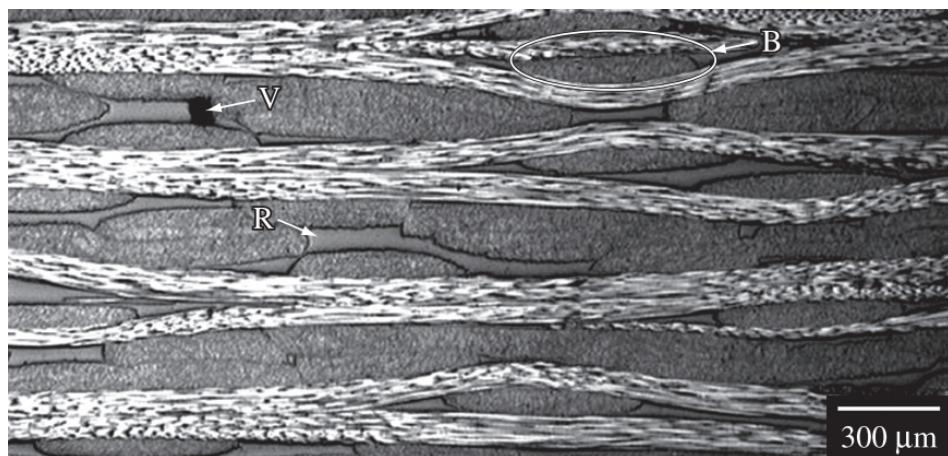


Figura 2.4: Sección típica de un laminado  $(0^\circ/90^\circ)_8$ , fracción de en volumen de vacíos 0.31 %. [3]

Son muy variadas las causas que pueden originar un daño durante la operación de una estructura de material compuesto, sin embargo, el impacto de baja velocidad es reconocido por diversos autores [7] como la condición de carga más severa para los materiales compuestos de tipo laminado.

Los laminados de material compuesto son susceptibles a dañarse por impacto debido a varias razones. Una de ellas es que normalmente no llevan refuerzo en la dirección del espesor. Por lo tanto, en esta dirección el comportamiento del material depende exclusivamente del comportamiento de la matriz. Al ser el espesor de resina entre las capas del laminado muy pequeño, hace que un impacto produzca delaminaciones con facilidad [34].

Debido a la fragilidad de la matriz y la ausencia de mecanismos de deformación plástica, los materiales compuestos de carbono-epoxy son más susceptibles al daño por impacto que los metales.

Los daños en materiales compuestos causados por impacto suelen ocurrir por uno o la combinación de los siguientes modos de fallo descritos a continuación:

- **Rotura de la matriz :** normalmente ocurre por un impacto de baja velocidad, seguido de tensión, compresión y cizallamiento de la matriz. Ejemplos de agrietamiento de la matriz son el agrietamiento por cizallamiento (inclinación de 45°) y el agrietamiento por flexión (inclinación vertical). Después de un impacto, el agrietamiento de la matriz es el primer daño en los materiales compuestos que no se puede ver a simple vista. El agrietamiento de la matriz reducirá la fuerza de compresión y el cizallamiento interlaminar de la interfaz del material [6]. Se puede este modo de fallo en la Figura 2.5

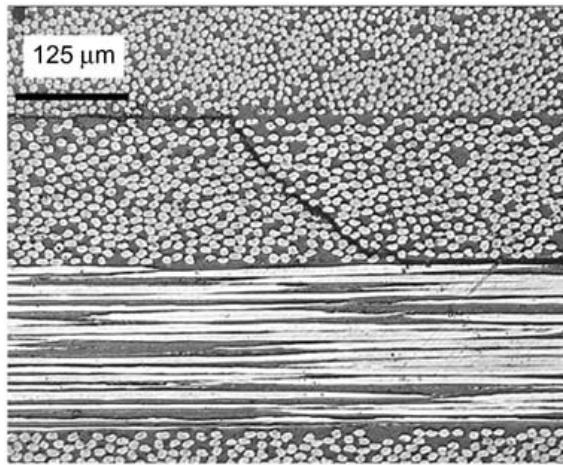


Figura 2.5: Micrografía de la rotura de matriz en un laminado multidireccional [34]

- **Delaminación**: la delaminación es el daño más crucial en un impacto y se produce en las intercapas que conforman el laminado del material compuesto. La delaminación es causada por el agrietamiento de la matriz, las grietas de flexión o las grietas de cizallamiento. La delaminación afecta a la resistencia en compresión del laminado y lentamente hará que el compuesto experimente un fallo de pandeo.

La tensión experimentada por el material inicia las grietas y causa la delaminación entre las capas. Esto se debe a que cuando un proyectil impacta, se producen ondas de compresión que viajan a través del espesor. Cuando rebotan sobre la cara opuesta, se convierten en ondas de expansión y generan el daño. Se puede ver una micrografía en la Figura 2.6

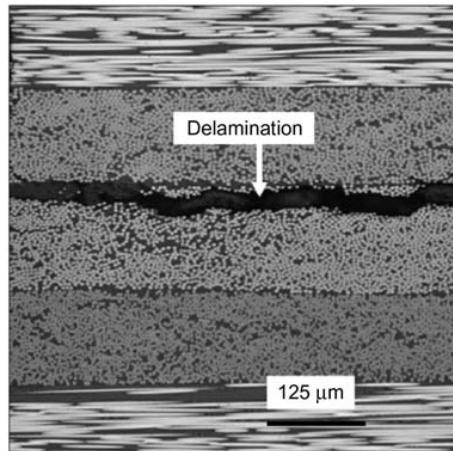


Figura 2.6: Micrografía de delaminación en un laminado multidireccional [34]

- **Rotura de fibras**: la rotura de la fibra ocurre debido al campo de alta tensión que se genera alrededor de la abolladura. El proyectil induce una fuerza cortante y altas tensiones de flexión en el lado no impactado del material. La rotura de la fibra es uno de los pasos hacia el fallo final del compuesto. En un material compuesto, el fallo final bajo carga de tracción ocurre cuando un cierto número de fibras se rompen. En

la Figura 2.7 se puede ver una micrografía de este fenómeno.

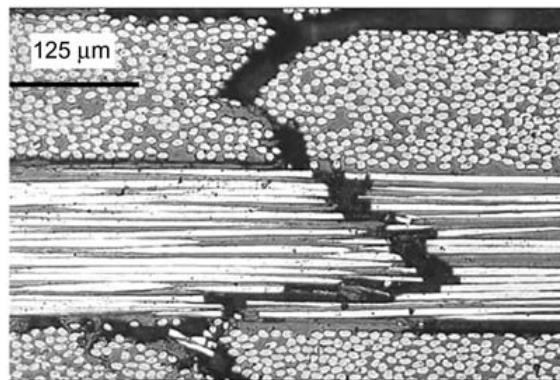


Figura 2.7: Micrografía de la fractura de fibra en un laminado multidireccional [34]

Finalmente, en la Figura 2.8 se puede ver todas los daños descritos anteriormente actuando de forma simultáneamente tras un impacto.

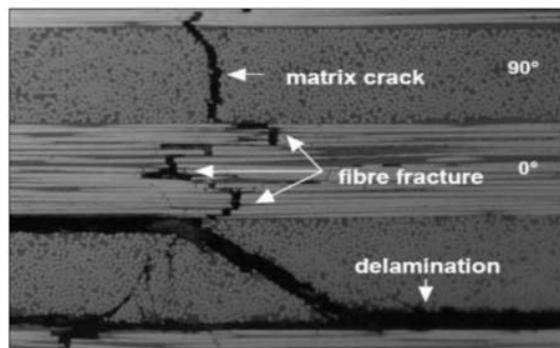


Figura 2.8: Conjunto de daños por impacto en un laminado multidireccional [3]

### 2.2.3. Característica sensibles al daño

No solo interesa saber el tipo de daño que se va a monitorizar, sino también la característica sensible al daño que permite monitorizarlos.

En la Tabla 2.9 se tienen varios ejemplos de daños con la característica sensible al daño a la que afectan.

### 2.2.4. Tecnología SHM aplicable para una DSF

Además de la DSF hay que tener en cuenta como la presencia de dicho daño influye en dicha característica. Esto se denomina la firma de daño. Dicha firma puede ser un cambio en la medida o señal que se obtiene de la DSF.

Por lo tanto, con la combinación de DSF y la firma de daño se puede elegir la tecnología SHM a utilizar.

DAÑOS TÍPICOS	CARACTERÍSTICAS SENSIBLES AL DAÑO					
	Forma estructural	Deformación estructural puntual	Campo de deformación	Interferencia electromagnética	Deformación-desplazamiento por Vibraciones	Deformación-desplazamiento por Ondas Elásticas
Grietas fatiga	NO	SI	SI	SI	SI	SI
Corrosión	NO	NO	NO	SI	NO	SI
Sobrecarga	SI	SI	SI	NO	SI	NO
Delaminación	NO	SI	SI	SI	SI	SI
Despegado	SI	SI	SI	SI	SI	SI
Widespread Fatigue Damage	NO	SI	SI	NO	SI	SI
Impactos	SI	SI	NO	SI	NO	SI
Abolladuras	NO	NO	SI	NO	NO	SI
Impacto de rayo (*)	NO	NO	NO	NO	NO	NO
Pérdida remache	NO	SI	SI	NO	SI	SI
Porosidad	NO	NO	NO	SI	NO	SI
Tenso-corrosión	NO	SI	SI	SI	NO	SI
Erosión	SI	SI	SI	SI	SI	SI
Pandeos locales	SI	SI	NO	NO	SI	NO
Deformaciones plásticas	SI	SI	SI	SI	NO	NO

Figura 2.9: Tabla de daños y DSF

En este trabajo se van a estudiar dos tipos de daño diferentes: despegues e impactos. En el caso de los despegues, la característica sensible al daño que se va a medir será el campo de deformaciones y la firma de daño será la variación de este campo de deformaciones respecto al estado estructural sin daño (referencia). La tecnología elegida ha sido sensores de deformación de fibra óptica, más concretamente del tipo FBG y OBR.

La utilización de sensores de fibra óptica para medir deformaciones tiene ventajas frente a la utilización de otro tipos de sensores, como las galgas extensiométricas. Estos sensores tienen un tamaño y peso reducido que junto con su naturaleza no eléctrica y capacidad de multiplexación los hacen ideales para ser integradas en estructuras de material compuesto.

A continuación se va a hacer una breve explicación de los sensores usados.

- Fibre Bragg Grating (FBG): tienen un comportamiento similar a las galgas extensiométricas pero con la ventaja de poder instalar varios sensores en una misma fibra lo que y se pueden tomar medidas a muy alta frecuencia. A día de hoy solo se han integrado en estructuras aeronáuticas en estructuras aeronáuticas a nivel de laboratorio y para su fabricación se requieren altas inversiones y personal cualificado.
- Sensores de medida distribuida (OBR): estos sensores permiten establecer millones de puntos de medida a lo largo de una fibra óptica, pudiendo llegar hasta varios kilómetros de longitud. A diferencia de los FBGs, estos sensores solo pueden tomar medidas estáticas ya que requieren varios segundos para procesar una señal completa.

Se puede ver una comparación de los sensores en la Figura 2.10.

Por otro lado, la DSF de los impactos va a ser la deformación por ondas elásticas (Lamb) que produce el impacto y la firma de daño será la variación de estas ondas.

La tecnología elegida para medir la DSF en este daño han sido los sensores piezoelectrónicos.

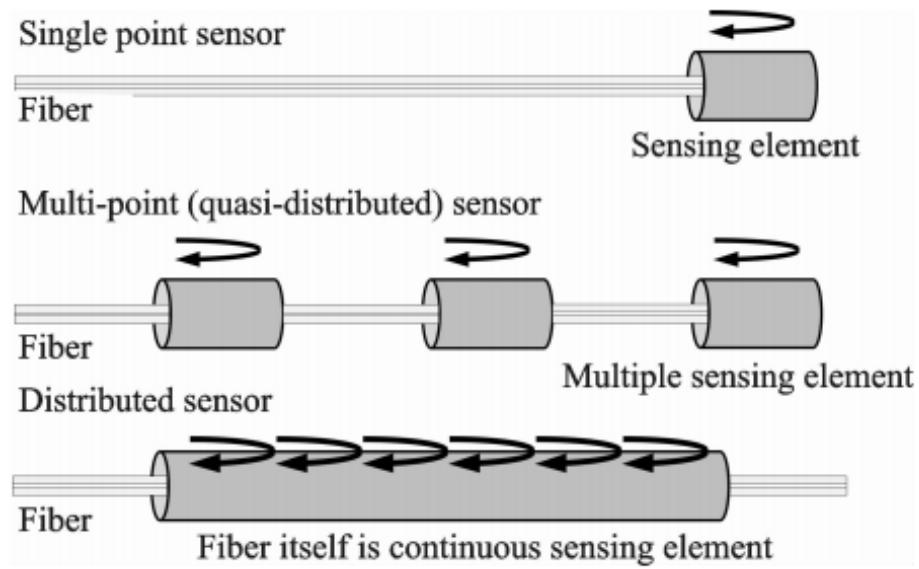


Figura 2.10: Morfología de redes sensoriales de fibra óptica [37]

Los sensores piezoeléctricos más usados son los PZT. Están fabricados de titanato zirconato de plomo que tienen un fuerte efecto piezoeléctrico, lo cual los hacen ideales para la lectura de óndas mecánicas. El como modifica la estructura atómica del material una onda al llegar al PZT se puede ver en la Figura 2.11.

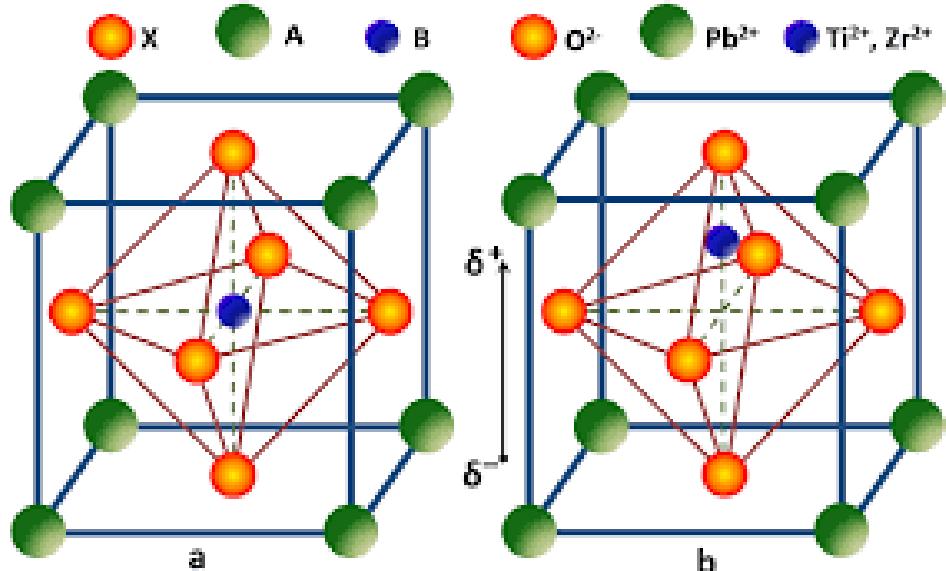


Figura 2.11: Representación esquemática de la estructura atómica del PZT antes y después de ser polarizado [2]

## 2.3. Estado del arte de la IA

El término *Inteligencia Artificial* fue acuñado por John McCarthy en 1956 cuando celebró la primera conferencia académica sobre el tema. McCarthy la definió como "la subdisciplina del campo de la informática que busca la creación de máquinas que puedan imitar comportamientos inteligentes", pero el camino para entender si las máquinas de verdad son capaces de pensar comenzó mucho antes. En la obra *As We May Think* de Vannevar Bush se propuso un sistema que amplificaba el conocimiento y capacidad de entender de la gente. Cinco años más tarde Alan Turing escribió un artículo sobre la noción de que las máquinas son capaces de simular a los seres humanos y la capacidad de realizar tareas inteligentes, como jugar al ajedrez [27].

La Inteligencia Artificial puede ser dividida en dos grandes grupos [23]:

- **IAs fuertes:** hacen referencia a aquellas inteligencias artificiales que son capaces de enfrentarse a una gran variedad de problemas y dominios diferentes
- **IAs débiles:** en este grupo se encuentran aquellos sistemas que solo pueden llevar a cabo un conjunto muy limitado de tareas.

A día de hoy, todas las IAs existentes solo se clasifican en el segundo grupo.

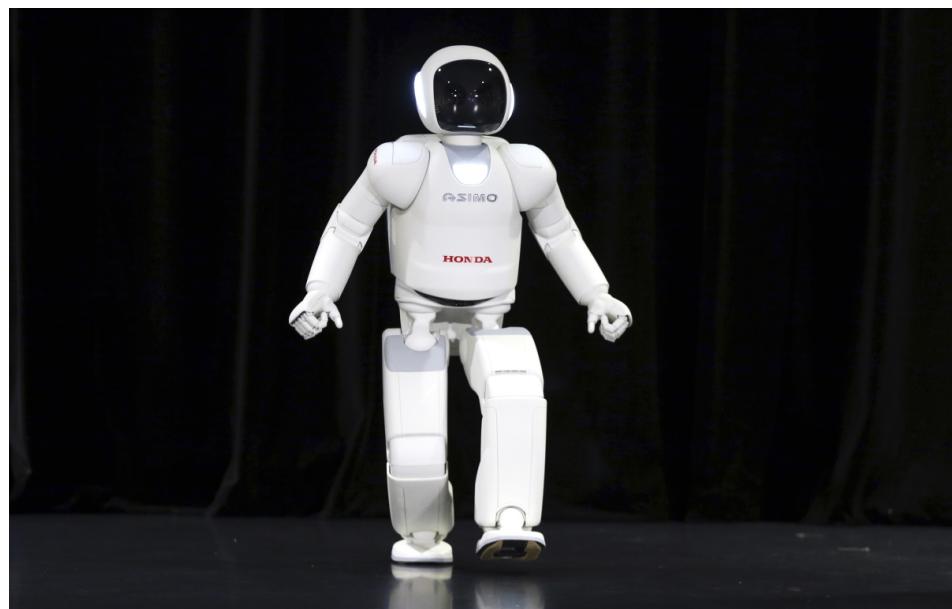


Figura 2.12: Ejemplo de Inteligencia Artificial débil

En la definición dada por McCarthy es muy importante la palabra *imitar*. Imitar no significa que dicho comportamiento sea un comportamiento cognitivo.

En la Figura 2.12 se puede ver un robot que imita una serie de movimientos o acciones de manera similar a un humano. Aunque el movimiento que realiza el robot haya sido programado por un humano, entra dentro de la lógica de la definición, ya que se puede considerar un movimiento aparentemente inteligente.

Dentro del campo de la inteligencia artificial podemos encontrarnos varias subcategorías que responden a diferentes comportamientos inteligentes.

Siguiendo con el ejemplo anterior, si nos centramos en las capacidades de moverse y adaptarse al entorno tratamos con el campo de la robótica. También está la capacidad de entender el lenguaje (NLP [20]) o la capacidad de hablar [19].

Sin embargo, la capacidad que define al ser humano como agentes inteligentes es la capacidad de aprender. Y esta habilidad también ha sido transferida a las máquinas.

El Machine Learning, o Aprendizaje Automático, es la rama de la inteligencia Artificial que busca como dotar a las máquinas de capacidad de aprendizaje, entendido este como la generalización de conocimiento a partir de un conjunto de experiencias.

Dentro del Machine Learning hay tres paradigmas de aprendizaje diferentes [32]:

1. **Aprendizaje supervisado:** es una tarea de aprendizaje en la que una función opera entre los datos de entrada y salida utilizando como referencia unas etiquetas asignadas a cada entrada.

En este tipo de aprendizaje es necesario dar tanto la entrada como la salida al ordenador para que aprenda. Lo que hace el ordenador es generar una función basada en estos datos, que puede ser desde una simple línea, hasta una función compleja, dependiendo de los datos proporcionados.

El aprendizaje supervisado es el tipo más básico de paradigma de aprendizaje y la mayoría de los algoritmos utilizados hoy en día se basan en este tipo de patrón. Algunos ejemplos de esto son:

- Regresión lineal
- Regresión logística
- Deep Learning

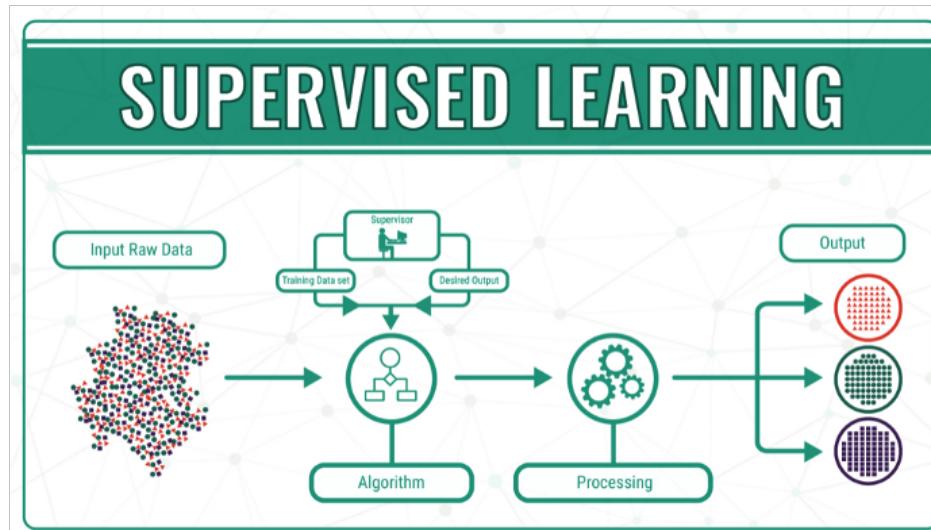


Figura 2.13: Descripción del aprendizaje supervisado

2. **Aprendizaje no supervisado:** en este tipo de paradigma el ordenador sólo utiliza la entrada para desarrollar un patrón de aprendizaje, básicamente aprender sin resultados.

Esto significa que el ordenador tiene que reconocer un patrón en la entrada dada y desarrollar un algoritmo de aprendizaje en consecuencia. Así que concluimos que "la máquina aprende a través de la observación y encuentra estructuras en los datos". Este es todavía un campo muy inexplorado del Machine Learning por lo que los grandes gigantes de la tecnología como Google y Microsoft están actualmente investigando sobre él.

Algunos ejemplos son:

- Clusterización
- Asociación

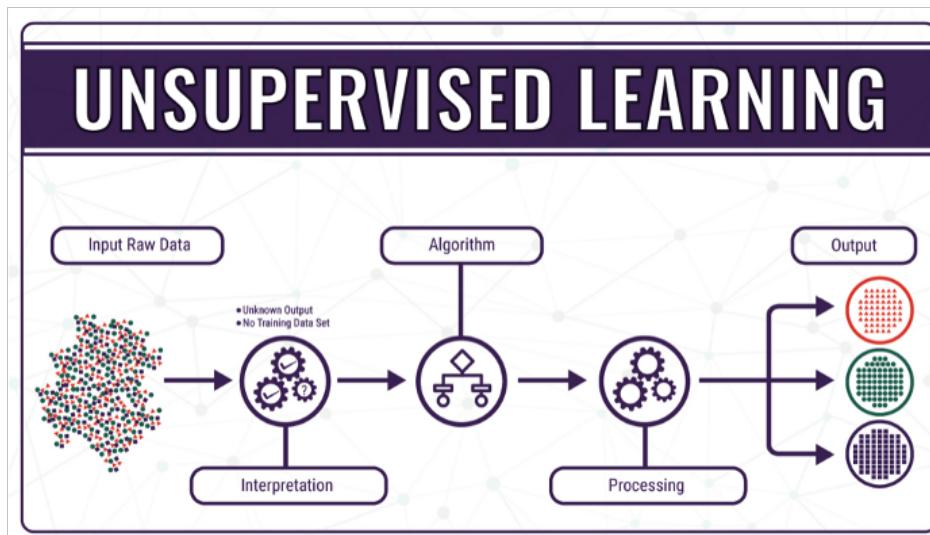


Figura 2.14: Descripción del aprendizaje no supervisado

3. **Aprendizaje reforzado** El Aprendizaje por Refuerzo permite a las máquinas y software determinar automáticamente el comportamiento ideal dentro de un contexto específico, para maximizar su rendimiento.

Hay una gran variedad de problemas diferentes que pueden ser resueltos usando el aprendizaje reforzado. Dado que los agentes de RL pueden aprender sin la supervisión de un experto, el tipo de problemas que mejor se adaptan a RL son los problemas complejos en los que parece no haber una solución obvia o fácilmente programable. Dos de los principales son:

- Videojuegos
- Problemas de control

En resumen, el Machine Learning es un componente nuclear de la Inteligencia Artificial que ha generado un cambio de paradigma. Se ha pasado de programar una máquina para imitar un comportamiento inteligente a programar una máquina para que aprenda a imitar ese comportamiento inteligente.

Entre los diferentes algoritmos de Machine Learning, el que más fama le ha dado en la última década ha sido sin duda las *Redes Neuronales* y su versión más potente, el *Deep Learning*.

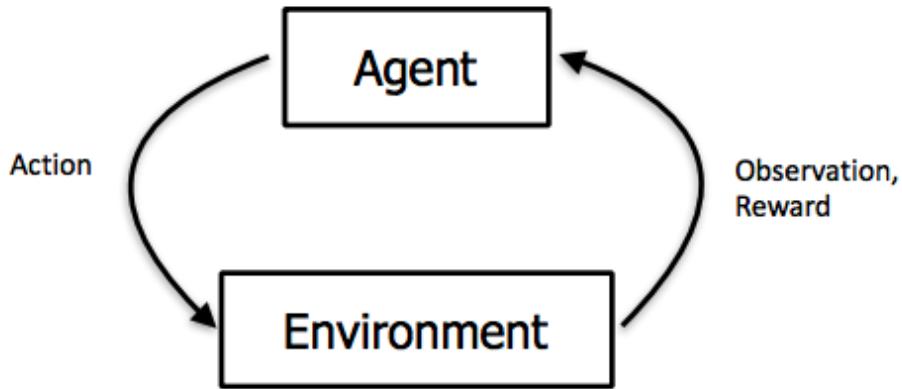


Figura 2.15: Descripción del aprendizaje reforzado

A continuación se va a explicar en que consisten estas redes y una representación gráfica de su funcionamiento.

### 2.3.1. Representación gráfica del funcionamiento de una Red Neuronal

Aunque el campo del DL lleva unos años en auge, para la mayoría de usuarios sigue siendo una simple herramienta, es desconocido cómo funciona y mucho menos se tiene una representación gráfica de qué hace y cómo llega a los resultados.

Dado que este trabajo se va a centrar en tecnología de DL, es conveniente hacer una breve explicación de cómo una NN es capaz de clasificar diferentes grupos de datos. Para ello, esta explicación se ha basado en el gran trabajo realizado por el equipo de *TensorFlow - playground* [35]. Un entorno perfecto para la visualización del comportamiento de los diferentes parámetros que componen una Red Neuronal.

Lo primero de todo es saber la forma de los argumentos de entrada que, como para casi todo algoritmo, sus *inputs* son números. Dependiendo del problema que se busque resolver estos datos tendrán una forma u otra, más complejos o más simples, pero en esta introducción se va a utilizar un dataset de 2 dimensiones ( $x_1, x_2$ ) y dos clases diferentes (azul y naranja).

Los datos que se van a intentar clasificar están representados en la Figura 2.16

Una vez se conocen los dos grupos que se busca clasificar se va a comenzar la descripción de los elementos que componen una NN y como interactúan entre ellos. Estos elementos son:

- **Neurona**

Es la unidad básica de procesamiento que compone una Red Neuronal. Estas neuronas tienen conexiones a través de las que reciben los valores de entrada y realizan una suma ponderada con ellos. Cada una de las entradas es multiplicada por un valor, llamado peso, que definirá a cual de los valores de entrada se le da más importancia. A este cálculo se le añadirá un sesgo o *bias*, dicho de otro modo, se le sumará un escalar.

Si se observa detenidamente la Figura 2.17 (primera neurona llamada *perceptron*) se puede ver que, matemáticamente, una neurona es equivalente a una regresión lineal

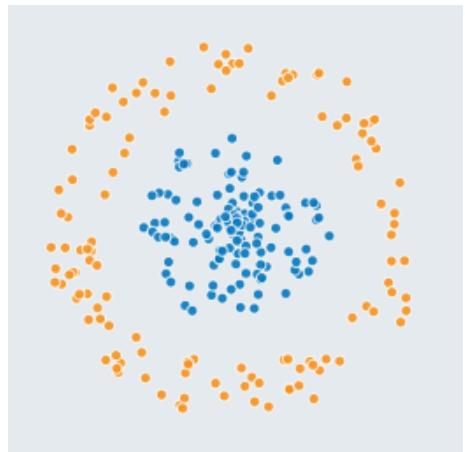


Figura 2.16: Representación de las dos clases diferentes en dos colores

$$(w_1 \cdot x_1 + w_2 \cdot x_2 + b = 0).$$

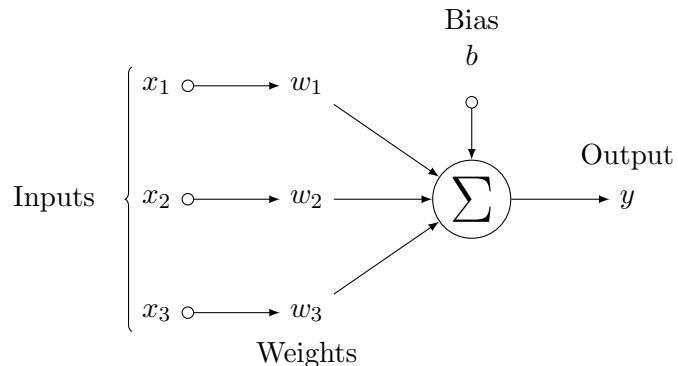


Figura 2.17: Representación del *perceptron*

En una regresión lineal bidimensional, ajustando los pesos  $(w_1, w_2)$  se modifica la recta que separa el plano  $(x_1, x_2)$  en dos regiones. En el ejemplo que se está resolviendo, por muy bien que se ajusten estos pesos, con una neurona solo se puede llegar a separar el plano en dos regiones, como se ve en la Figura 2.18.

Para definir que región del plano es asignada a la clase azul o naranja se utiliza la siguiente expresión:

$$w_1 \cdot x_1 + w_2 \cdot x_2 + b = y \quad \begin{cases} y \geq 0, & \text{naranja} \\ y < 0, & \text{azul} \end{cases} \quad (2.1)$$

Por lo tanto, la neurona se puede sintetizar un una función. Como una neurona no consigue separar las dos regiones de puntos de forma efectiva, hace que sea necesario dar un paso más.

#### • Capas

Para modelizar conocimiento complejo no es suficiente con utilizar una única neurona, es necesario concatenar varias de ellas. Una forma de colocar las neuronas sería una debajo de otra en forma de columna, lo que se define como, en la misma capa.

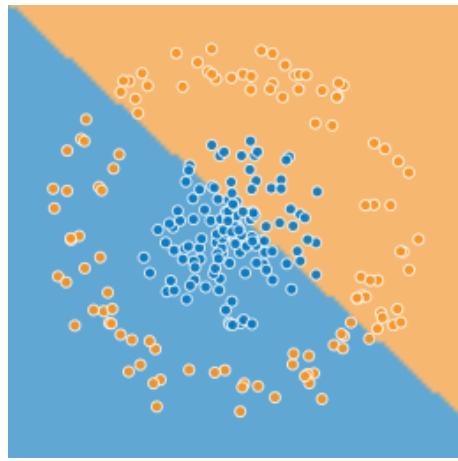


Figura 2.18: Clasificación de datos con una sola neurona

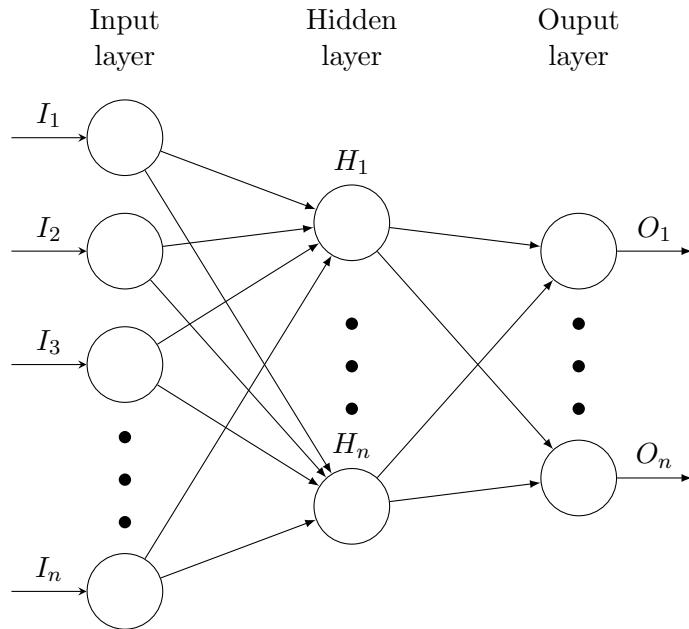


Figura 2.19: Representación de las capas de una red

Como se representa en la Figura 2.19, las capas se pueden concatenar de manera secuencial, una detrás de otra y dependiendo de dónde estén colocadas tendrán un nombre u otro. La capa que recibe primero los argumentos de entrada se llama *Input layer*, la que calcula el argumento de salida se llama *Output layer* y todas las capas que estén entre estas dos, se llamarán *Hidden layers*.

La concatenación secuencial de las capas hace que la red sea capaz de aprender conocimiento jerarquizado y es lo que da nombre al Deep Learning (Aprendizaje Profundo). Dicho de otro modo, si a la red se le introduce una imagen de un coche, es capaz de aprender que un coche no es un objeto único, una capa sintetiza lo que es una rueda, la siguiente lo que es una puerta y las últimas capas son capaces de juntar toda esa información para construir la idea de coche como un conjunto de objetos.

Las neuronas dentro de la misma capa lo que permiten es diversificar el conocimiento. Cada neurona se especializa en una parte de la información ajustando los pesos para dar más importancia a un canal de entrada u otro.

Juntando lo explicado anteriormente se entiende mejor este concepto de aprendizaje. Una red parte de una cantidad grande de datos, los píxeles de una imagen, por ejemplo. La capa de entrada será la que primero reciba la información, por ello necesitará muchas neuronas para procesarla y extraer de ella qué zonas de la imagen son más importantes. A medida que se va avanzando, cada una de las capas tiene que desglosar cada vez más la información sintetizada que le ha llegado de la capa anterior, por lo que necesitará menos neuronas, su nivel de abstracción de conocimiento aumenta.

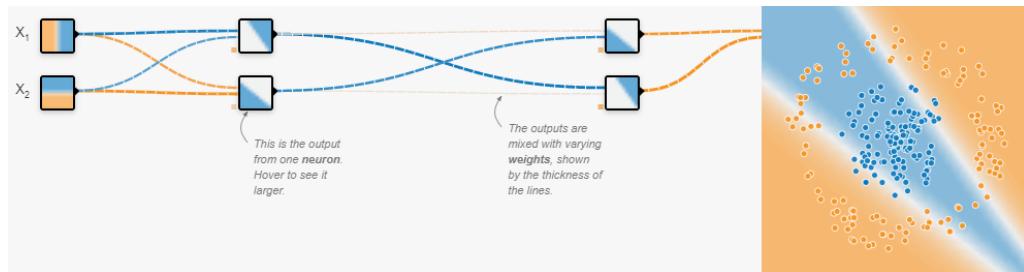


Figura 2.20: Red con dos *Hidden layers* y dos neuronas en cada capa

La aplicación de lo explicado anteriormente se representa en la Figura 2.20. Se puede observar en la caja que representa cada neurona que, en esencia, todas hacen una regresión lineal y que no pueden crear una frontera circular para separar el espacio. Esto se debe a que la operación matemática de concatenación de regresiones lineales termina colapsando en una regresión lineal, no puede dibujar curvas. Para conseguirlo, es necesario añadir un elemento más.

#### • Función de activación

La necesidad de aplicar una distorsión no lineal a la salida de las neuronas convierte en indispensables a las funciones de activación. Con ellas se podrá encadenar de forma efectiva la computación de varias neuronas.

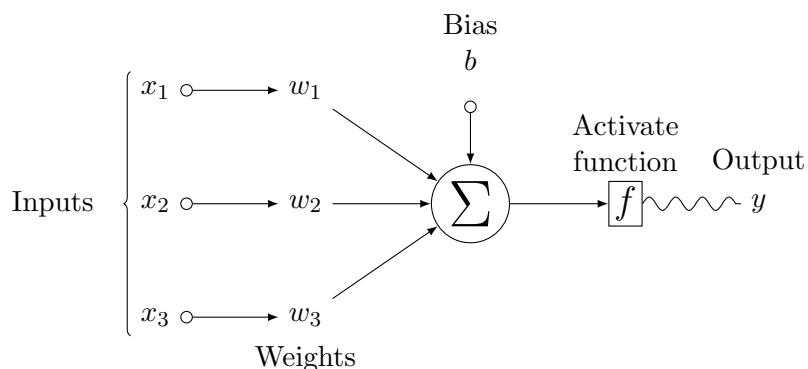


Figura 2.21: Efecto de la función de activación sobre el output de una neurona

En cierto modo ya se estaba usando una función de activación antes de nombrarlas. En la Ecuación 2.1 se había usado la función escalón para separar las dos regiones

del plano, pero esta función no provoca no linealidades y no es muy efectiva cuando se requieren fronteras complejas.

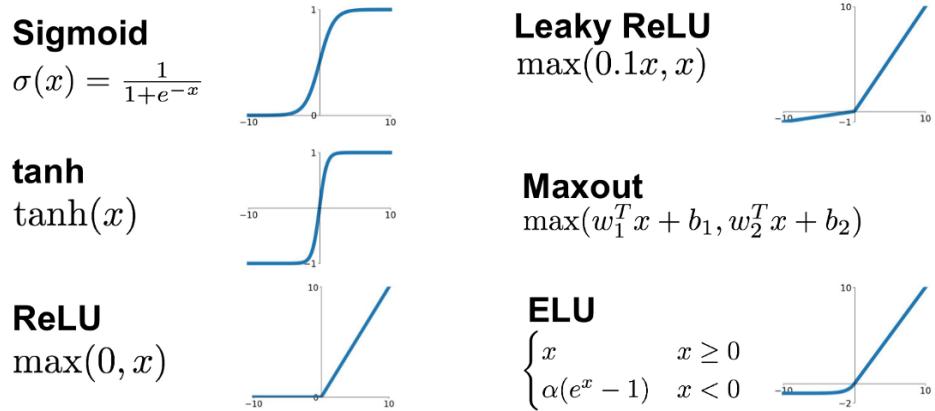


Figura 2.22: Funciones de activación más populares

Las funciones de activación más conocidas están representadas en la Figura 2.22. Su función principal es hacer que los valores muy grandes o muy pequeños se saturen a determinado valor, dependiendo de la función elegida.

Para ver el efecto de distorsión geométrica que tienen estas funciones sobre el *output* de las neuronas se usa de ejemplo la Figura 2.23. La nueva función de salida se convierte en una superficie, continua y derivable, que para cada *input*  $(x_1, x_2)$  le corresponde una altura  $y$  entre 0 y 1.

Tal y como se ha hecho anteriormente, se selecciona un valor umbral de  $y$  a partir del cual se clasificarán los puntos como azules o naranjas. En la Figura 2.23 este valor umbral se representa como un plano a una altura de 0,5, por lo tanto, los puntos  $(x_1, x_2)$  que tras ser evaluados en la neurona saquen un valor de  $y$  inferior a 0,5 se clasificarán como *naranjas* y los que den un valor superior o igual a 0,5 se clasificarán como *azules*. Esto recuerda a la Figura 2.18.

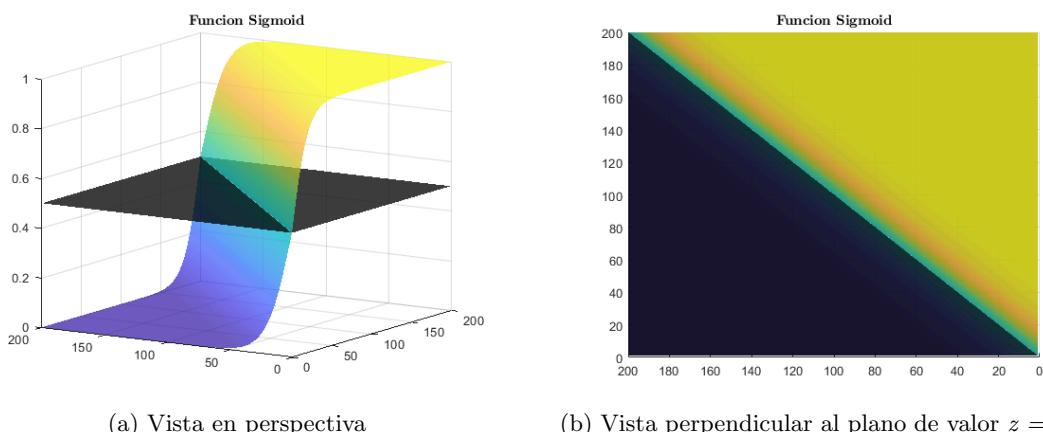


Figura 2.23: Combinación de funciones de activación

Ahora ya se puede utilizar una *hidden layer* con varias neuronas de forma efectiva.

La combinación de varias neuronas hace que se sumen superficies (funciones) como la representada en la Figura 2.23, pero con diferentes orientaciones, generando superficies más complejas, Figura 2.24. Igual que antes, los puntos  $(x_1, x_2)$  que tras ser evaluados tienen un  $y$  mayor o igual a 0,5 pertenecerán a la clase azul y los menores a la naranja.

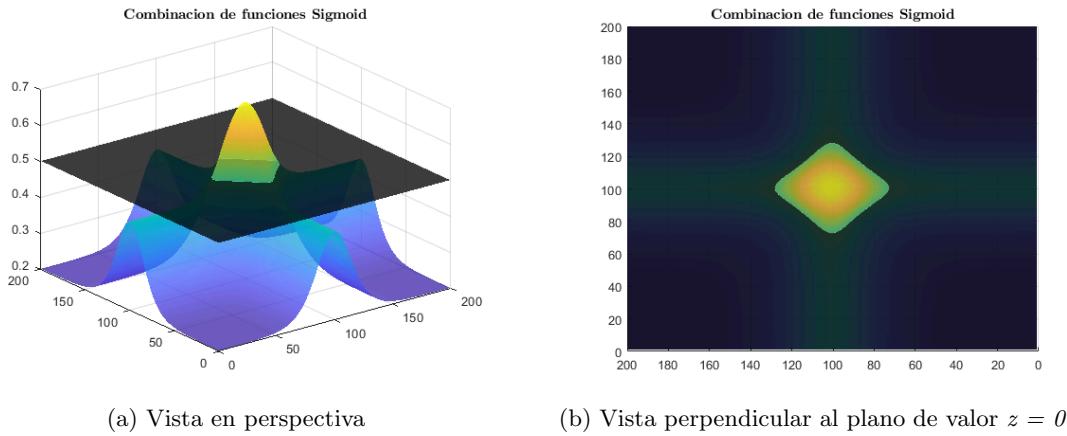


Figura 2.24: Combinación de funciones de activación

Finalmente se puede ver en la Figura 2.25 la aplicación de todo lo explicado. Gracias a las funciones de activación se consiguen las no linealidades necesarias para generar la superficie que separa exitosamente las dos clases de datos diferentes.

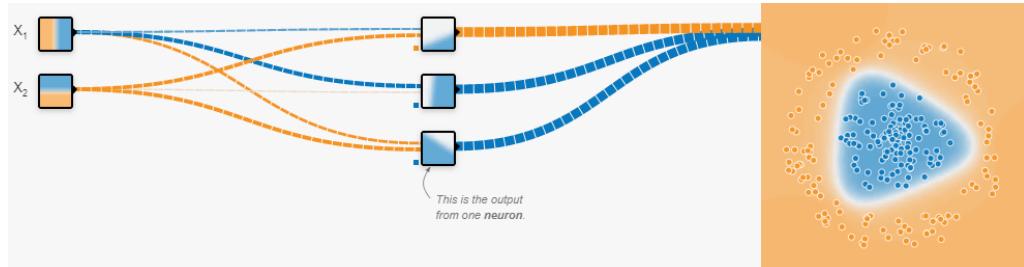


Figura 2.25: Resultado final de la clasificación

Las neuronas de la Figura 2.25 están unidas mediante unas líneas discontinuas de diferente color y grosor. Esto representa qué pesos son mayores y que zona perturban más. Los pesos son los elementos que ajustan las funciones de activación para conseguir la superficie final que mejor realice la clasificación. Pero, ¿cómo se realiza el ajuste de los pesos?

#### • Entrenamiento

Se llama entrenamiento al proceso de aprendizaje automático que utilizan las redes neuronales para ajustar sus parámetros internos. El algoritmo que hizo posible que la optimización de parámetros en DL fuera eficiente se llama *backpropagation* y es el responsable del final del *Invierno de la Inteligencia Artificial*

Para llevar a cabo el proceso de optimización es necesario calcular las derivadas parciales de cada uno de los parámetros de la red con respecto a la función de error.

Una vez se consigan las derivadas, se utilizará el algoritmo de descenso del gradiente para variar los parámetros y minimizar la función de error.

La función de error ( $E$ ) puede ser, por ejemplo, la suma de las distancias de cada punto a la frontera de su clase. Volviendo a la Figura 2.18, los puntos azules que se encuentran en la región naranja aumentarán mucho el valor de la función de error, mientras que los que están en la región azul no aumentarán casi su valor. La variación de la función de error respecto a un pequeño cambio los parámetros se representa matemáticamente como el gradiente gradiente:  $\frac{\partial E}{\partial w}$ .

Como se ve en la Figura 2.26, el peso asociado a una de las primeras conexiones afecta a prácticamente todas las conexiones posteriores y por lo tanto, también influirán en su gradiente. Esto hace que el aprendizaje de este primer peso sea especialmente complicado y convierte en esencial la utilización del *backpropagation* para el proceso de optimización.

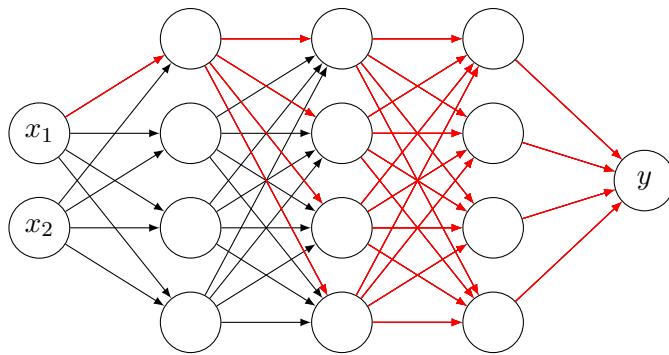


Figura 2.26: Influencia de un parámetro en la red

Cuando la red clasifica erróneamente un *input* genera un valor alto de la función de error. En una NN, el error de las últimas capas depende directamente de las capas anteriores, es decir, si se detecta que una neurona no tiene influencia sobre el error, las capas posteriores a ella tampoco lo tendrán. En esto consiste el *backpropagation*, en la retropropagación de errores, lo que hace muy eficiente el proceso. En la Figura 2.27 se puede ver el inicio del algoritmo partiendo de la última capa.

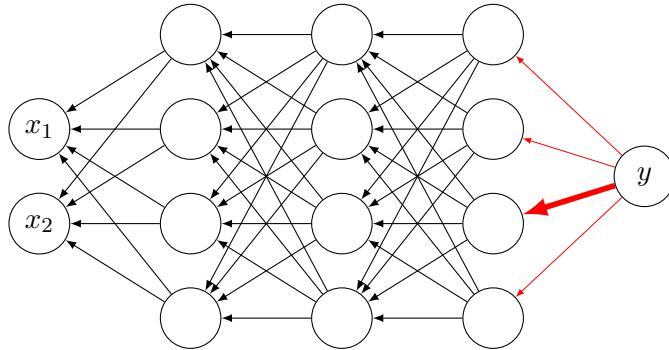


Figura 2.27: primer pase de backpropagation

Esta distribución de errores se utiliza para actualizar los pesos y bias de cada una de las neuronas.

Ahora que se han imputado los errores a las neuronas de la última capa se puede proceder a repetir el mismo proceso de antes como si este fuera el error final de la red,

como si esta fuera ahora la última capa. Se va avanzando capa tras capa moviendo el error hacia atrás.

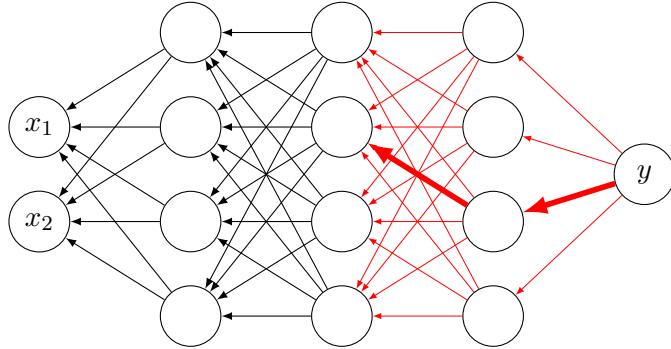


Figura 2.28: Segundo pase

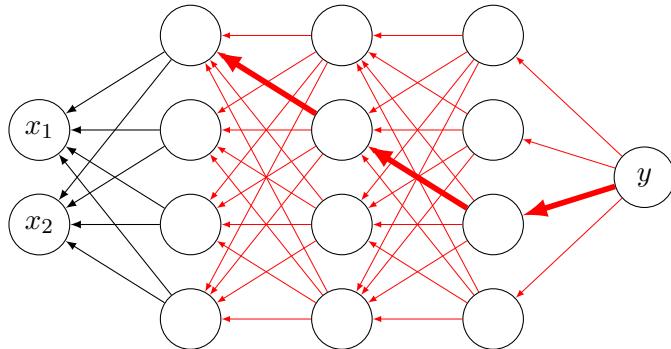


Figura 2.29: Tercer pase

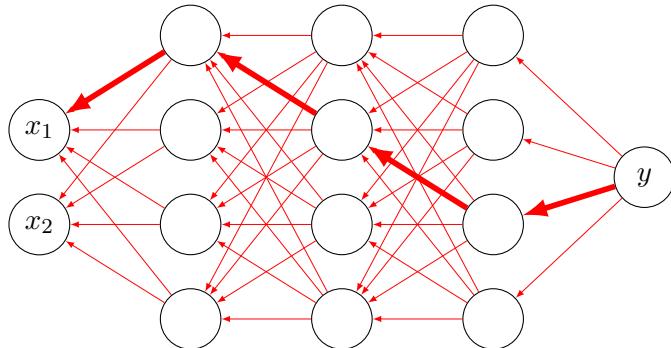


Figura 2.30: Cuarto pase

### 2.3.2. Software

Como se ha dicho anteriormente, el Machine Learning consiste en programar un ordenador para aprender una tarea de apariencia inteligente. Pero para escribir un programa hay una grán variedad de lenguajes de programación que se pueden elegir.

Generalmente, cada lenguaje está orientado a una tarea en concreto, en la que es más eficiente que otros. C se suele usar en microcontroladores, C++ es útil para programar servicios web, mientras que Matlab y Python son más eficientes en el campo del Data Science.

En el año 2018, la web GitHub [41] publicó una lista de los lenguajes más usados dentro del mundo de Inteligencia Artificial y Machine Learning. En esta lista (Figura 2.31) la primera posición la ocupa Python.

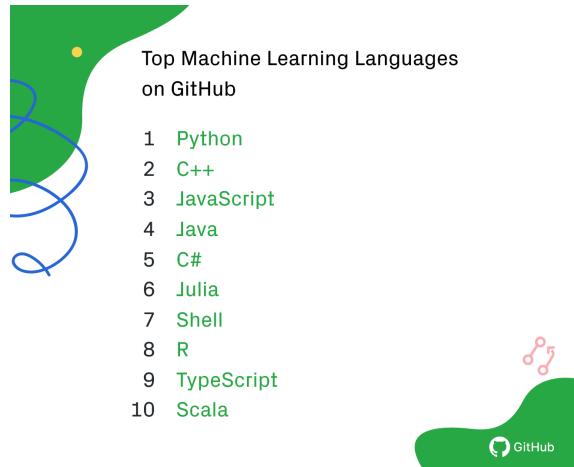


Figura 2.31: Lista de lenguajes más usados para ML en GitHub

En los últimos años, Python ha crecido dentro del mundo del ML principalmente por su filosofía Open Source y una gran comunidad de desarrolladores que hacen público la mayoría de sus trabajos.

En la Figura 2.32 se pueden ver múltiples librerías de Python usadas en ML.



Figura 2.32: Paquetes de ML

En [41] se afirma que la librería en la que más están colaborando sus usuarios es *PyTorch*, por lo que en este trabajo se ha decidido apoyar el desarrollo de esta comunidad utilizando la librería PyTorch como herramienta para el estudio de SHM.

## 2.4. IA aplicada a la monitorización estructural

Las limitaciones de las capacidades de medida de los sensores y los desafíos a la hora de desplegar grandes redes de sensores causados tanto por los requerimientos de energía, como el manejo de una elevada cantidad de datos, ha obstaculizado históricamente este tipo de redes.

Actualmente, ya se pueden obtener una gran cantidad de datos heterogéneos procedentes de diversos tipos de sensores. Sin embargo, los métodos de SHM convencionales basados en modelos físicos validados no son apropiados para una monitorización estructural eficaz utilizando una gran cantidad de datos. A su vez, los avances en las técnicas de tratamiento de datos han revolucionado su adquisición e interpretación.

A diferencia de los modelos de SHM tradicionales basados en la física, los modelos basados en datos incluyen la detección de daños y estimación de la vida restante. Además, los modelos tradicionales requieren un nivel de ruido pequeño en los datos, lo que es difícil de conseguir teniendo en cuenta el tipo de estructuras monitorizadas y sensores usados.

Por lo tanto, los modelos basados en datos han demostrado versatilidad y se han convertido en uno de los enfoques más atractivos para el SHM [5].

El uso de Deep Learning para el procesado de grandes cantidades de dato está en auge. Los modelos de DL pueden capturar y aprender información que está oculta en los datos para predecir diversos patrones gracias a la concatenación de capas y funciones de activación.

Los modelos DL han sido exitosamente usados en varias aplicaciones como *computer vision*, reconocimiento de audio o SHM. Las principales razones para su popularización en SHM son las siguientes:

- Avances en *Big Data* y computación en la nube: el coste de los sensores ha caído en los últimos años a pesar de que el precio de sus materiales aumentaba. Esto ha hecho posible la integración de una gran cantidad de sensores en las estructuras monitorizadas y transferir los datos inalámbrica a ordenadores en la nube.
- Avances en hardware y software de ordenadores: los procesadores multi núcleo han mejorado drásticamente durante la última década y, consecuentemente, se ha prestado atención al potencial de los procesadores gráficos (GPU) para entrenar a los algoritmos de DL.
- Avances en *Data Science*: los datos son el componente nuclear de cualquier aplicación de SHM. Hace unas décadas, el término *Data Science* no existía y ahora está a la vanguardia en las aplicaciones basadas en los datos, incluyendo el SHM. Con los recientes avances en algoritmos de Machine Learning como en la adquisición y transmisión de datos , los ingenieros de datos pueden interpretarlos, detectar anomalías o recuperar información dañada.
- Avances en *Transfer Learning*: ii bien la investigación y las aplicaciones basadas en DL se están generalizando de manera efectiva, las redes previamente entrenadas, como VGG, AlexNet y ResNet, han recibido una mayor atención para las aplicaciones relacionadas con SHM, que a menudo se ha demostrado que es muy eficaz y ahorra tiempo.

Como se ha dicho en el apartado anterior, el Deep Learning es una subcategoría perteneciente al Machine Learning. Las limitaciones en cuanto a rendimiento dentro de los

métodos tradicionales de ML como son el diseño y pre-selección de características puede ser eliminado con DL.

Para crear un modelo convencional se requiere conocimiento y experiencia en la selección de DSF para una aplicación concreta de SHM, lo cual no es práctico conforme los modelos aumentan de complejidad.

Por otro lado, los métodos SHM basados en DL tienen como objetivo desarrollar mecanismos de representación jerárquica y extracción de DSF totalmente automatizados a partir de los datos de entrada sin procesar.

Cada capa que forma una *Deep Neural Network* es responsable de aprender una nueva representación de los datos de entrada y, por lo tanto, el SHM basado en DL es un sistema de cerrado que no requiere la intervención humana para diseñar características. Por lo tanto, todos los parámetros de extracción de DSF, clasificación de patrones y regresión se diseñan simultáneamente, lo que hace que la SHM basado en DL sea aplicable a una amplia gama de problemas con un conocimiento mínimo sobre las características específicas.

En la Figura 2.33 se puede ver la comparación entre los modelos convencionales y los basados en DL.

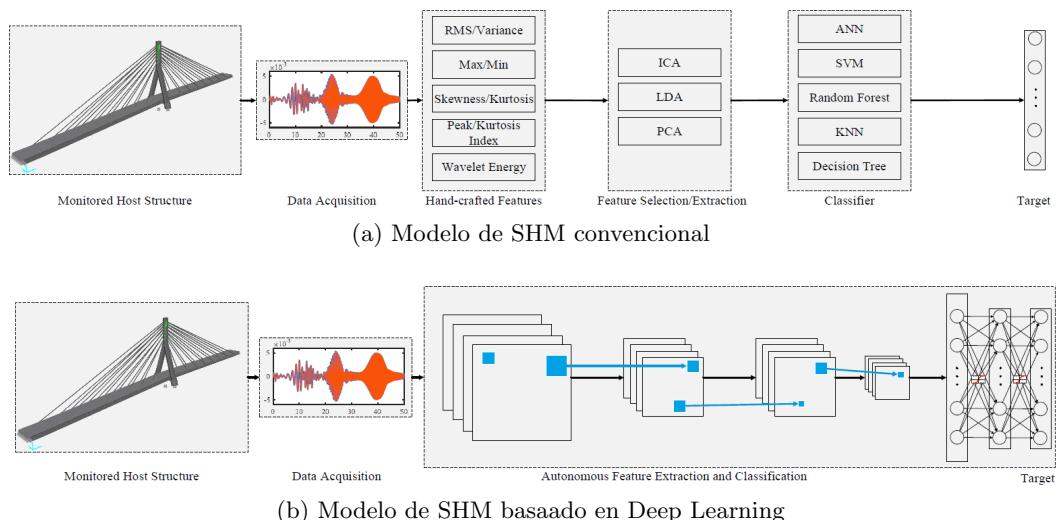


Figura 2.33: Comparación de técnicas diferentes de SHM [5]

Varios modelos de DL están siendo utilizados, como las redes convolucionales profundas (DCNN), Deep Boltzman Machines, Deep Belief Network, redes neuronales recurrentes (RNN), redes generativas enfrentadas (GANs), etc.

Como el número de algoritmos de ML y DL aumenta día tras día, en la Figura 2.34 representa un mapa mental de los algoritmos de ML más usados.

Las siguientes secciones se van a centrar en algunas de las aplicaciones más recientes del DL en SHM.

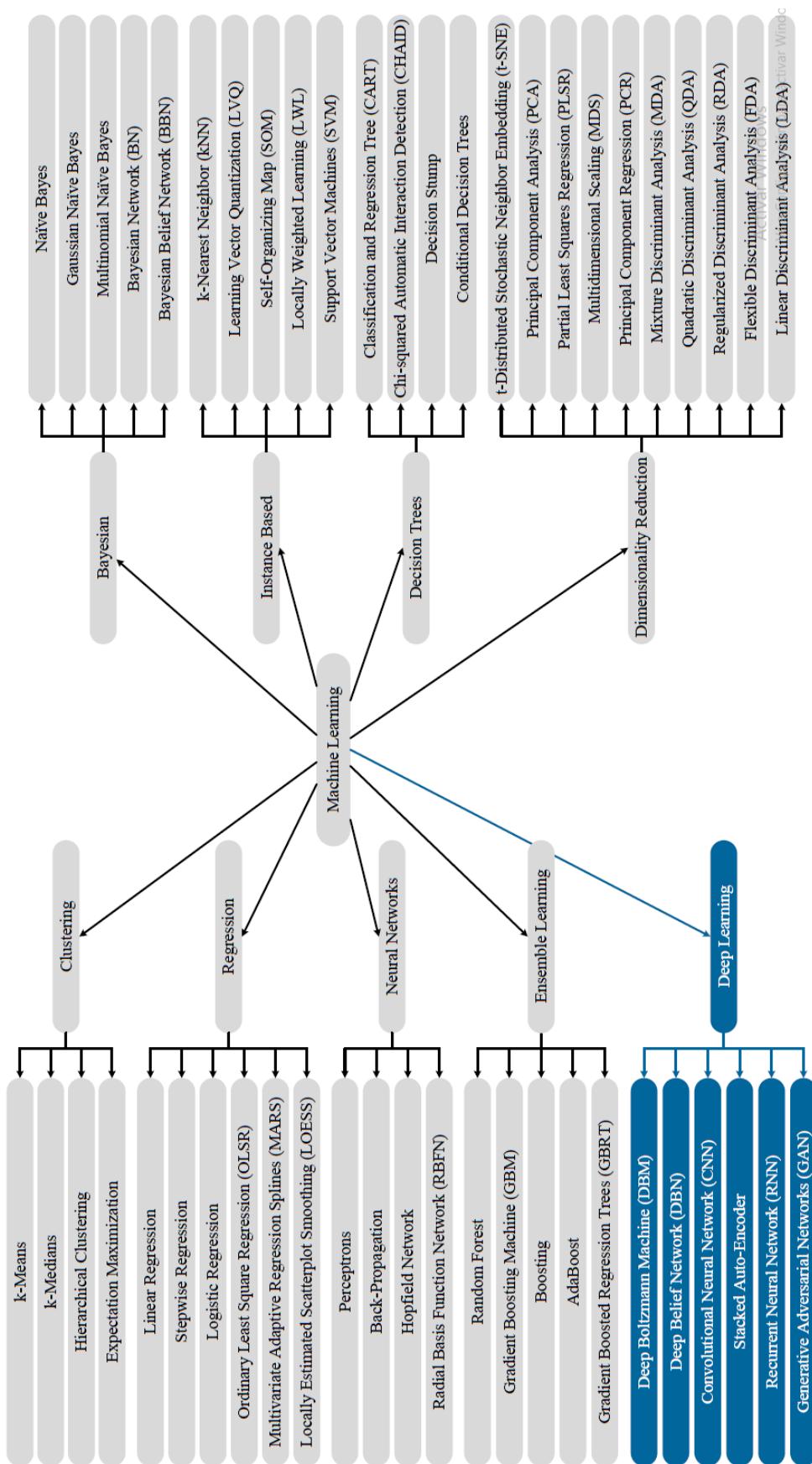


Figura 2.34: Mapa mental para la selección de algoritmos de Machine Learning [5]

#### 2.4.1. SHM basado en vibraciones con DL

Se han desarrollado numerosos métodos de evaluación de daños para shm basados en vibraciones utilizando algoritmos de ML. El DL ha introducido nuevos horizontes en estructuras de gran escala y ha facilitado la adquisición y el procesado de grandes conjuntos de datos de diferentes tipos de sensores.

Mientras que la mayoría de métodos convencionales de localización de daños, como radiografía o ultrasonidos, requieren un conocimiento previo de la localización aproximada del daño. Sin embargo, los modelos basados en vibraciones parten de la premisa que los daños modifican las características de vibración (formas modales, frecuencias) y localizan el daño partiendo del cambio en la señal medida.

Los algoritmos de ML se han utilizado tanto para extraer características de los métodos no basados en parámetros para evitar que sean extraídas manualmente, como en los procesos de clasificación utilizando estas características.

A continuación se van a comentar varios ejemplos de trabajos que han utilizado métodos de DL para SHM basado en vibraciones [5].

En [9], Abdeljaber et al. realizaron un proceso de localización de daños basado en redes convolucionales (CNN) y sensores inalámbricos.

La estructura estudiada tiene unas dimensiones en planta de 4,2 x 4,2 metros (Figura 2.35) y tiene integrados 10 acelerómetros inalámbricos como el mostrado en la Figura 2.36.

Los diferentes daños en la estructura se realizan soltando tornillos que unen las vigas entre sí.

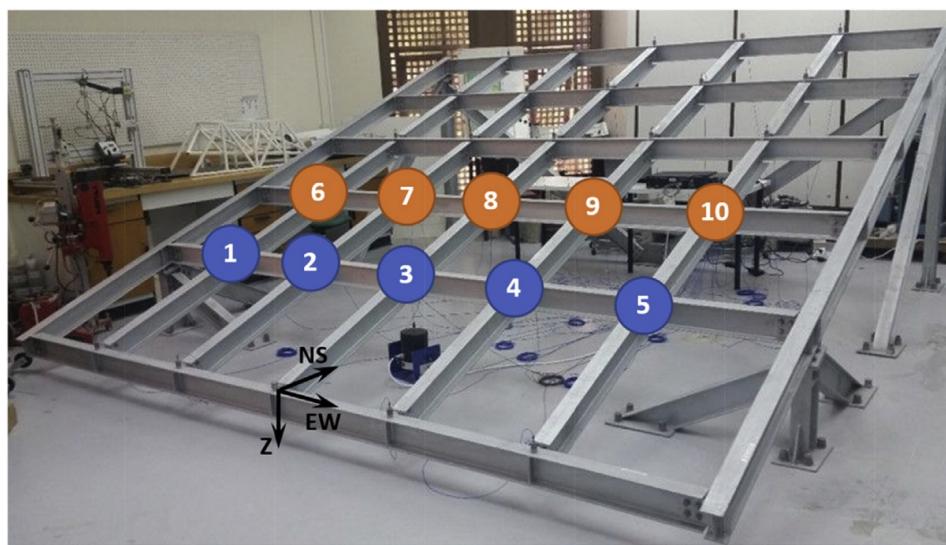


Figura 2.35: Estructura bajo estudio con la posición de los acelerómetros [9]

Tras entrenar a la red convolucional con todos los daños, se creó un índice al que llamaron Probabilidad de Daño (número de imágenes clasificadas como dañadas dividido por el número total de imágenes) que logró hacer una clssificación del estado de la estructura con un error del 0,54 %.

Por otro lado, Bao et al. en [12] utiliza las redes convolucionales de una manera diferente



Figura 2.36: Acelerómetro usado para medir las vibraciones en la estructura [9]

a [9].

Las redes convolucionales son el estado del arte en el campo del reconocimiento de imágenes con DL. Por ello, en vez de tratar los datos de los sensores sin procesado previo, se van a convertir en imágenes con escala de grises y a cada imagen se le asociará una etiqueta correspondiente al daño al que pertenece.

Una vez que se han pre-procesado los datos de los sensores, se utiliza una red convolucional profunda para clasificar los diferentes estados de daño. En la Figura 2.37

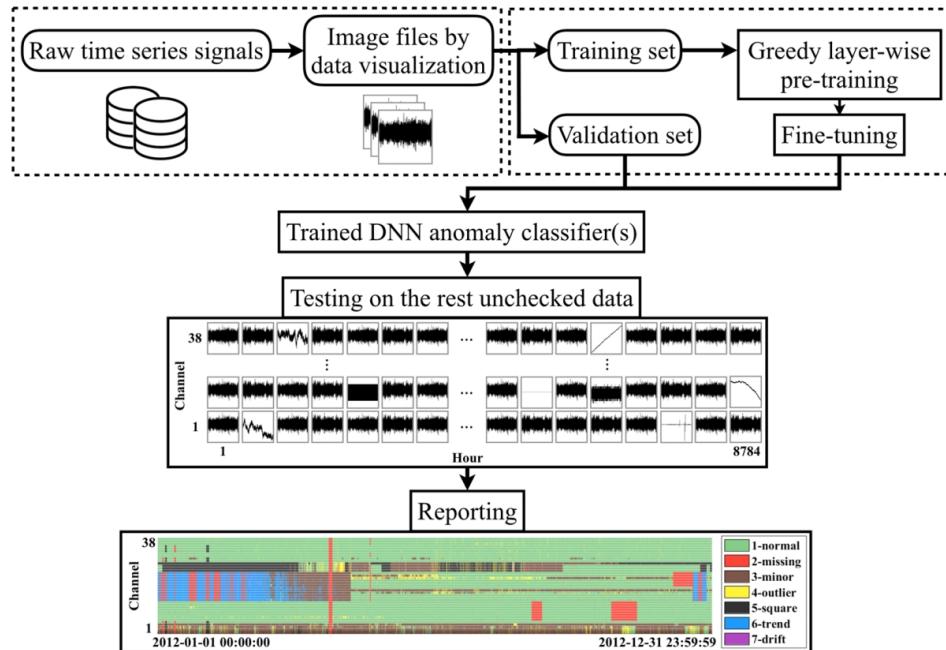


Figura 2.37: Proceso de detección de anomalías [12]

Bao et al. se centraron en el estudio de un puente localizado en China representado en la Figura 2.38, en la que se ven las posiciones de los diversos sensores.

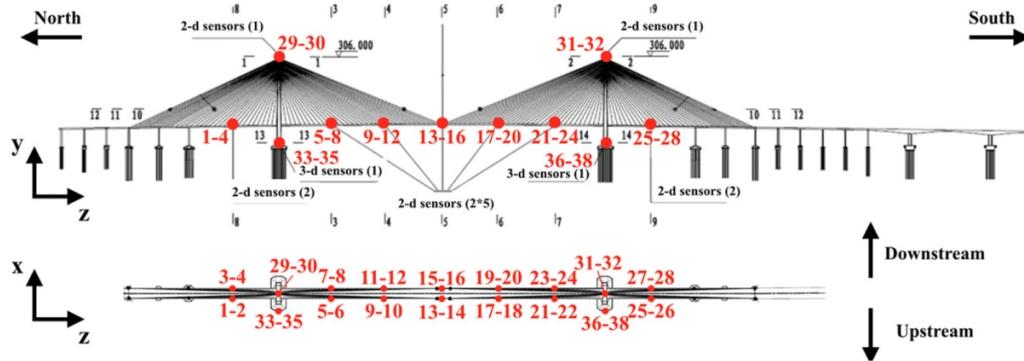


Figura 2.38: Estructura estudiada en [12]

Tras el de etiquetado y procesado de las señales consiguieron alcanzar una precisión de un 87 % en la clasificación de los diversos daños utilizando datos reales de un año completo.

#### 2.4.2. SHM basado en visión con DL

La mayoría de las técnicas de inspección de infraestructuras se basan en evaluaciones visuales que dependen de la percepción y experiencia de un experto, lo que no es siempre fiable.

Los procesos de inspección basados en imágenes se realizan estudiando imágenes sin procesar o aplicando filtros básicos para ser capaces de detectar mejor los bordes y acelerar el proceso. Para automatizar esto se requieren algoritmos de ML.

La mayor parte de los esfuerzos en visión computacional se concentran en desarrollar algoritmos de aprendizaje basados en DL, particularmente a través de Deep CNN que han sido capaces de lograr más del 95 % de precisión en problemas de clasificación basados en imágenes. Además, la aplicación de las CNN se ha extendido a la clasificación a nivel de píxel dentro de una imagen para detectar y localizar diferentes objetos de interés utilizando filtros no lineales y mapas de características.

Los avances recientes en visión computacional han atraído la atención a estas tecnologías en SHM como una de las herramientas más efectivas en la evaluación de estructuras por medio de imágenes para la detección de corrosión, desconchado del hormigón, grietas en hormigón y pavimento, detección de fatiga y daños superficiales y subterráneos.

Las técnicas de DL en SHM basadas en visión se pueden separar en las dos clases siguientes:

- **Detección de grietas con sistemas de visión de DL**

Las infraestructuras, especialmente las estructuras de hormigón envejecidas, son propensas a la formación de grietas debido a las condiciones cambiantes de carga, corrosión, etc. Las grietas en el hormigón o pavimentos de carreteras suelen aparecer como líneas con orientaciones e intensidad aleatorias. Por lo general, estas líneas son más oscuras y están conectadas, y se puede realizar una simple detección de grietas utilizando los umbrales de contraste correctamente elegidos.

Los investigadores en el campo del visión computacional han desarrollado consistentemente las Deep CNN. Varios estudios adaptaron CNN para detectar grietas superficiales y subterráneas en pavimentos y cemento. Estudios anteriores utilizaron

DCNN para clasificar superficies de hormigón o pavimento mediante el método de ventana deslizante. Este enfoque se ha empleado para abordar problemas de clasificación complejos en diferentes campos, incluido el SHM.

Recientemente, Kim et al. [16] propusieron un modelo rápido basado en ML y CNN para determinar la ubicación de zonas con y sin grietas sobre superficies fotografiadas. En la Figura 2.39 se puede ver el un diagrama del proceso.

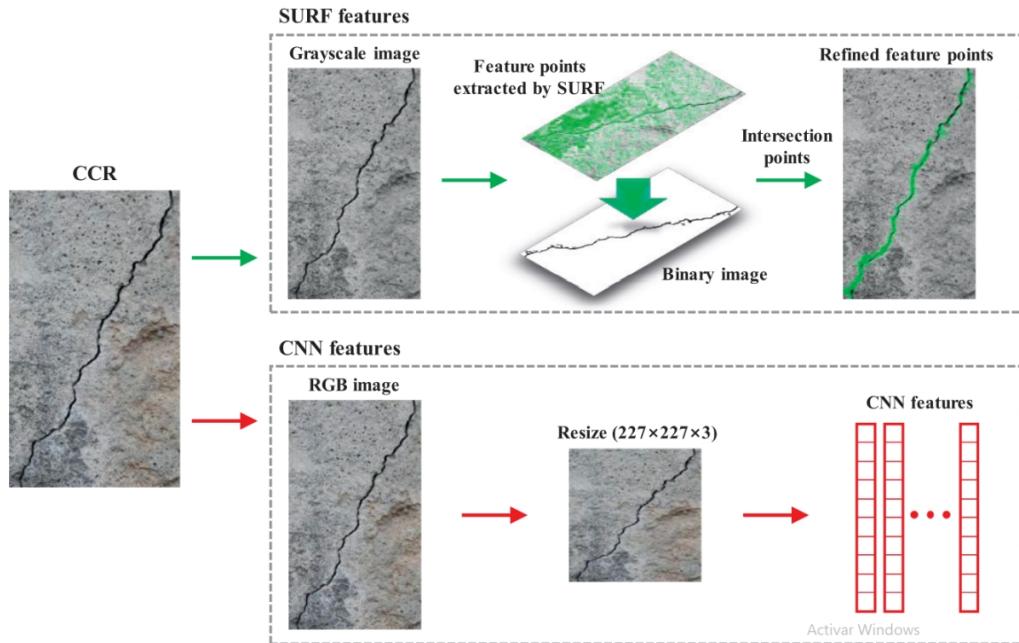


Figura 2.39: Combinación de SURF (ML) y CNN (DL) para la extracción de características de imágenes con grietas [16]

- **Reconocimiento de componentes estructurales y detección de cambios a través de DL basado en la visión**

Es esencial realizar una inspección a nivel global, así como el proceso de reconocimiento de componentes estructurales, para comprender la relación entre daños y seguridad de las estructuras. Sin embargo, estudios recientes en SHM basados en DL no han abordado completamente esta preocupación. Incluso los modelos de detección de grietas basados en vídeo no interpretan el impacto de los daños en un contexto global.

Para ello, Yhum et al. en [13] proponen un sistema basado en la combinación de redes convolucionales y UAV para obtener y clasificar imágenes en estructuras cubiles de grandes dimensiones. Una vez que se haya realizado una primera clasificación de a qué zona de la estructura pertenece cada imagen, se utilizará otra red convolucional para buscar posibles daños. Se puede ver la idea de forma visual en la Figura 4.7d

#### 2.4.3. Aplicación de UAVs y teléfonos móviles para SHM con DL

Las redes de DL facilitaron la automatización del proceso de detección de daños además de lograr niveles aceptables de precisión. Sin embargo, en algunos casos, los inspectores no tienen acceso a todas las partes de las estructuras para adquirir imágenes (para enfoques basados en visión) o datos de sensores (para métodos basados en vibraciones). Ésta es una

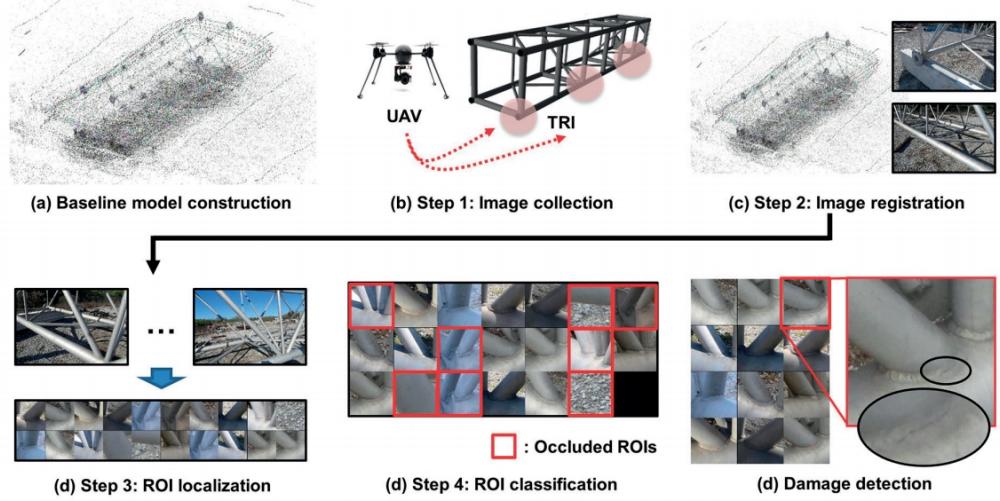


Figura 2.40: Resumen de la técnica desarrollada en [13]

de las principales dificultades en las estructuras como edificios altos, puentes y estructuras patrimoniales.

Los drones se propusieron como herramientas para inspeccionar tales estructuras a fin de superar estas dificultades. Los drones, los vehículos aéreos no tripulados (UAV) o los sistemas aéreos no tripulados (UAS) minimizan la necesidad de trabajo físico, además de ahorrar tiempo, ser rentables, seguras, disponibles y precisos. En los últimos años, se han realizado diferentes estudios con el fin de proporcionar un marco para el uso de UAV, mostrar su aplicabilidad y abordar algunas de sus desventajas.

Junto con los avances en las técnicas de transmisión de datos inalámbricas, se han realizado varios estudios que utilizaron tecnologías de UAS para ampliar la inspección basada en visión en SHM, así como técnicas basadas en vibraciones.

Kang y Cha [44] desarrollaron un sistema UAV autónomo para SHM mientras usaban balizas ultrasónicas para reemplazar la función del GPS que funciona mal en lugares parcialmente cubiertos, como debajo de cubiertas de puentes. El experimento se puede ver en la Figura 2.41

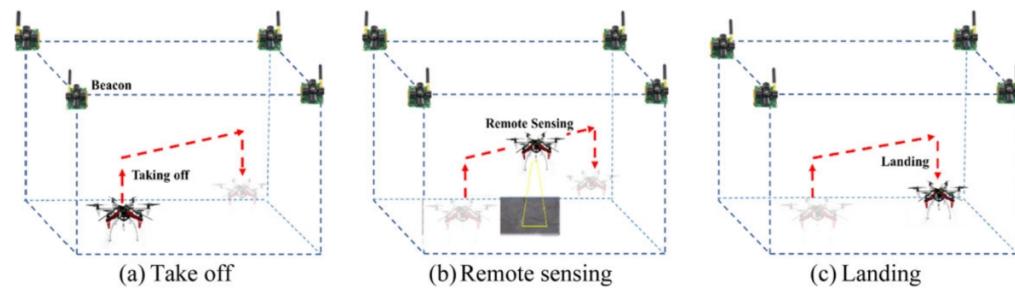


Figura 2.41: Resumen del experimento propuesto en [44]

Una vez obtenidas las fotografías de la estructura, se utilizó una red DCNN para localizar las grietas con unos resultados positivos. Estos resultados se pueden ver en la Figura 2.42

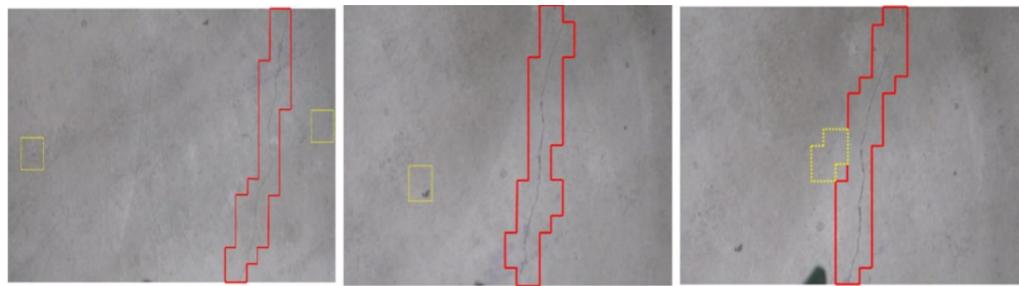


Figura 2.42: Resultados de la detección de grietas [44]

#### 2.4.4. Transfer Learning (TL) a partir de modelos pre-entrenados

Cuando el dataset es relativamente pequeño y hay alguna red previamente entrenada con un dataset más grande, una forma de actuar eficiente es ajustar la red ya existente para la nueva tarea de clasificación. Usando técnicas de aprendizaje por transferencia (TL) el tiempo de entrenamiento se puede minimizar transfiriendo los coeficientes del modelo base en lugar de comenzar con pesos asignados al azar.

La profundidad de la red tiene una relación directa con el número de parámetros de entrenamiento. Por lo tanto, las redes de DL requieren un tiempo considerablemente largo y una gran cantidad de datos para ser entrenadas. El TL puede aliviar este problema proporcionando conocimientos previos que se obtuvieron de un problema similar; por lo tanto, el ajuste final se puede realizar con un menor costo computacional y menos muestras de datos.

Para un modelo CNN con capas convolucionales en serie, seguidas de capas completamente conectadas (FC), una práctica común es ajustar las últimas capas FC o reemplazarlas por otras nuevas. Por lo tanto, las capas convolucionales se congelan durante el entrenamiento.

## 2.5. IA aplicada a la monitorización de estructuras aeronáuticas

A continuación, se van a exponer varios estudios en los que se aplicaron técnicas de ML y DL al SHM a estructuras aeronáuticas. Estos estudios se centran sobre todo en estructuras fabricadas en material compuesto.

La tecnología de Deep Learning también está habiéndose paso en el sector aeronáutico. Como se ha visto en los ejemplos anteriores, sus aplicaciones son variadas tanto en el procesado de imágenes, como en la detección de patrones en grupos de datos no lineales. Estas características las hacen ideales para SHM aplicado a la aeronáutica.

En [8], Khan et al. proponen el uso de redes convolucionales para la clasificación y predicción de varios tipos de delaminaciones en laminados de material compuesto utilizando las vibraciones de la estructura como dato de entrada a la red.

Para ello, preprocessan la señal recogida por los acelerómetros y pasan al dominio de la frecuencia como se ve en la Figura 2.43 y se obtiene una imagen.

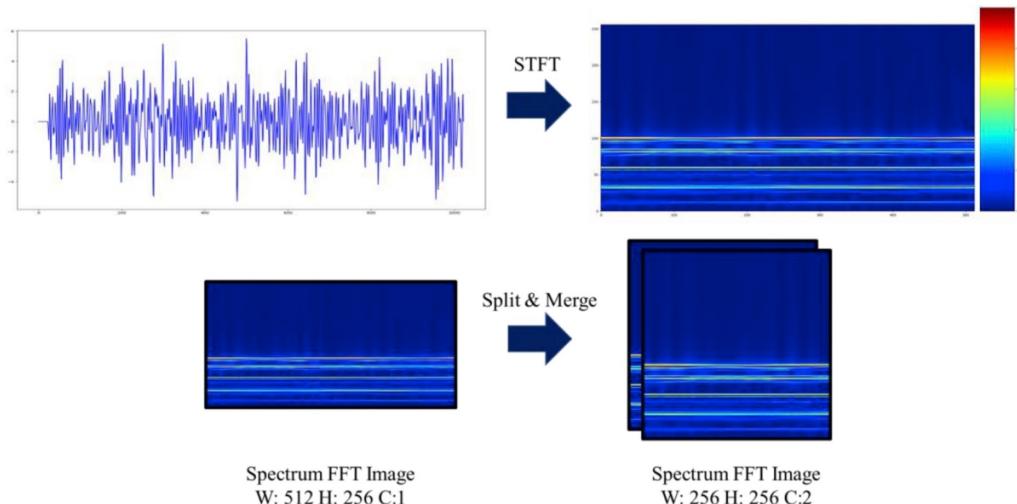


Figura 2.43: Resultados de la detección de grietas [8]

Una vez que se tiene la imagen con su etiqueta correspondiente se entrena a la red clasificadora (Figura 2.44)

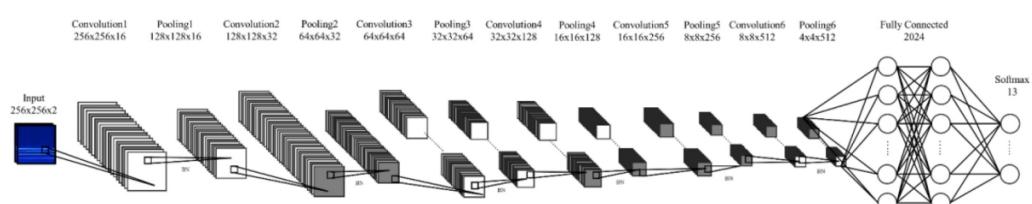


Figura 2.44: Arquitectura de la red clasificadora [8]

Por otro lado, Vincentius Ewald et al. en [18] presentan un marco novedoso llamado DeepSHM que implica el aumento de datos de las señales de sensores capturadas y formaliza un método genérico de DL para SHM.

El caso de estudio se limita a ondas guiadas ultrasónicas SHM. La respuesta de la señal del sensor de un modelo de elementos finitos (FEM) se procesa previamente a través de

transformadas matemáticas para obtener la matriz de coeficientes de ondas (WCM), que luego alimentará a la CNN para ser entrenada. Este proceso se puede ver en la Figura 2.45.

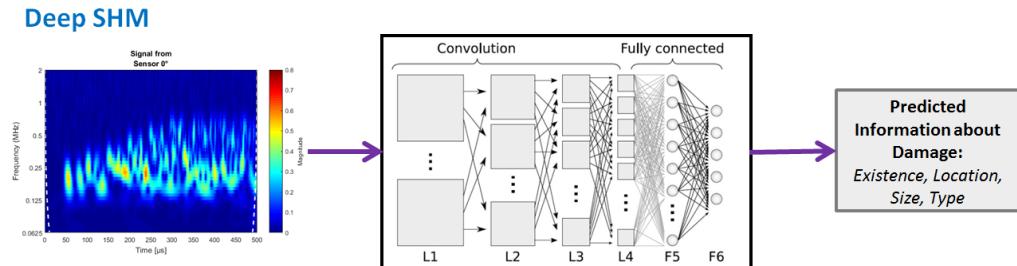


Figura 2.45: Arquitectura de la red clasificadora [18]

Otro ejemplo es el trabajo realizado por Javier Gómez-Escaloniella et al. en [15]. Este trabajo consistió en la utilización de varios parámetros obtenidos en ensayos en vuelo para predecir deformaciones en puntos concretos de la aeronave A330 Multirole Tanker Transport (MRTT). Este proyecto se encuentra dentro del desarrollo del sistema de monitorización de fatiga llamado Structural Helath Monitoring System (SHMS).



Figura 2.46: A310 Boom demostrador [15]

La red utilizada en el proyecto es diferente a las que se han comentado hasta ahora. A diferencia de las que se han mencionado en este capítulo, la red de [15] no tiene como objetivo realizar una clasificación, sino que, a partir de una serie de datos de entrada tiene que predecir un valor (deformación).

Dicho de otro modo, durante el proceso de entrenamiento, los pesos no se ajustan para crear fronteras entre los datos, lo que hacen es crear una función  $f(x_1, \dots; x_n)$ , donde  $x_1, \dots, x_n$  son los distintos parámetros medidos durante el vuelo, que mejor ajuste la curva de deformaciones del sensor elegido.

Los resultados que fueron obtenidos en el fuselaje se pueden ver en la Figura 2.47. Utilizando una red sin optimizar se consiguió alcanzar un error menor al 3,5 %, lo que hace ver el potencial de esta tecnología en el campo del SHM aeronáutico.

También es importante recalcar el trabajo que realizaron C. Aguilar [2] y J. García Ramírez [37] en este campo. Sin embargo, como se va a hablar de sus proyectos en pro-

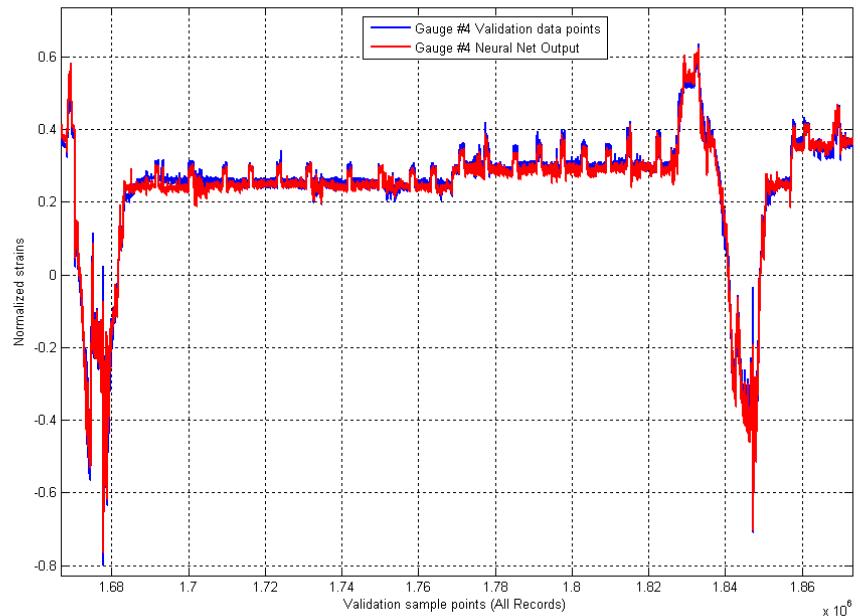


Figura 2.47: Comparación de deformaciones del fuselaje medidas y generadas por la red [15]

fundidad más adelante, en este apartado solo se van a ser nombrados.

## Capítulo 3

# Aplicación de Deep Learning en SHM con sensores de deformación

### 3.1. Proyecto INESASSE

El Proyecto INESASSE (Integración y Explotación de Sistemas de Autodiagnóstico y Supervisión de Salud Estructural en aviones no tripulados) tiene como objetivo principal investigar la capacidad de implantación, en condiciones operativas, de sistemas de supervisión estructural (SSE), empleando redes de sensores de fibra óptica.

Este proyecto de investigación fue realizado por el INTA (Instituto Nacional de Técnica Aeronáutica), en conjunto con la ETSIAE (Escuela Técnica Superior de Ingeniería Aeronáutica y del Espacio), y consta de 3 fases:

1. Elaboración de un modelo de elementos finitos (FEM) del sistema estructural, a través del cual se establece la morfología de la red de sensores. Sobre este FEM se simularán distintos daños producidos sobre la estructura, a fin de analizarlos mediante distintas técnicas SHM y determinar la posición óptima de la red sensorial.
2. Fabricación de dos estructuras idénticas de CFRP que se instrumentaran con la red de sensores definida en la fase 1, integrables en el UAV en cuestión. Tras esto, a una de estas estructuras se le realizará una campaña de ensayos en tierra a fin de validar los FEM, además de comparar los resultados obtenidos mediante herramientas SHM sobre el modelo teórico y el real. Al igual que en el FEM, se realizarán los mismos daños estructurales controlados.
3. En la última fase del proyecto, la estructura fabricada anteriormente que no fue ensayada, se integrará en un UAV en operaciones de vuelo reales, procesando datos de los sensores integrados en la fase 2, e investigando las capacidades del sistema SHM seleccionado para el vuelo.

El vehículo aéreo no tripulado que protagoniza este proyecto es el UAV Milano, con una masa en servicio de 900 kg y una autonomía de hasta 20 horas. En la Figura 3.1 se puede apreciar una imagen del mismo.

#### 3.1.1. Estructura y red de sensores

La estructura de la aeronave sobre la que se ha realizado el FEM y se ha fabricado después es el fuselaje posterior del UAV Milano. Esta estructura está compuesta por:



Figura 3.1: UAV INTA Milano

- Revestimiento inferior con un larguero longitudinal en T dividido en dos y refuerzos sobre las pestañas. Fabricado con cinta UD fuera de autoclave.
- Revestimiento superior con dos largueros en L y refuerzos sobre las pestañas. Fabricado con cinta UD fuera de autoclave.
- Cuatro cuadernas con refuerzos sobre las mismas. Fabricadas con cinta UD fuera de autoclave.
- Dimensiones aproximadas de 2600x850x890 mm.

En la Figura 3.2 se puede ver el diseño explosionado de los componentes descritos.

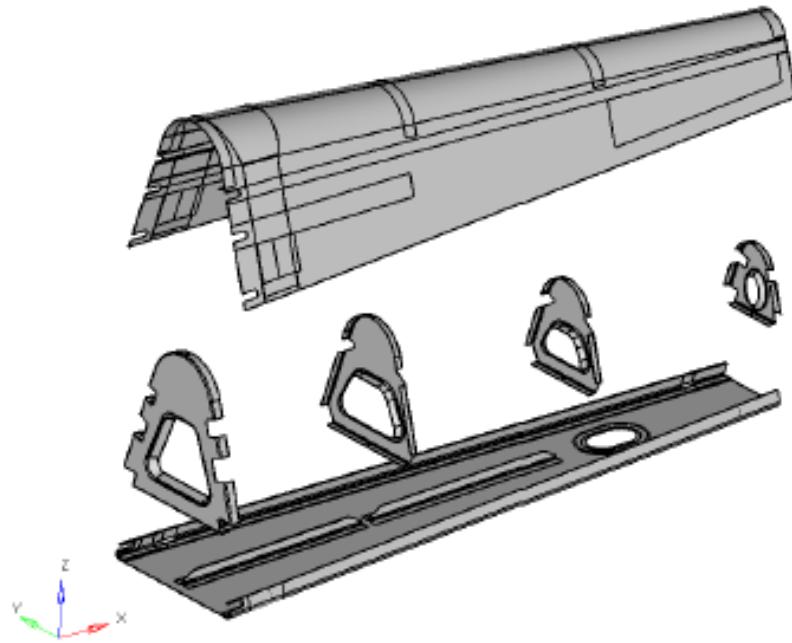


Figura 3.2: Fuselaje posterior

Sobre esta estructura se van a distribuir dos redes de sensores de fibra óptica, uno formado por sensores FBG y otro por OBR. Con ellos se recogerá el campo de deformaciones de la estructura que se procesará para obtener la clasificación de los daños. En las

Figuras 3.3 y 3.4 se tiene una representación gráfica de la distribución de estas redes sobre la estructura.

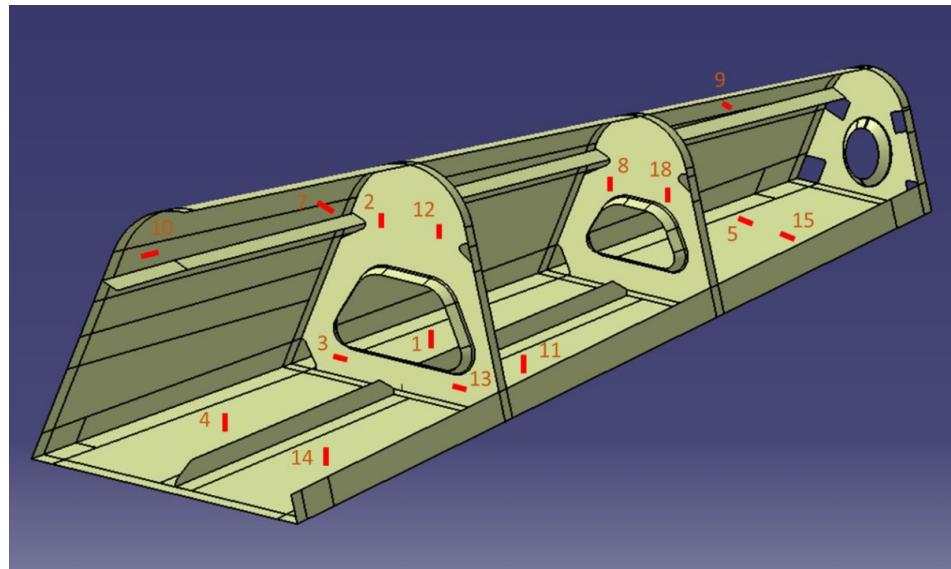


Figura 3.3: Red de sensores FBG

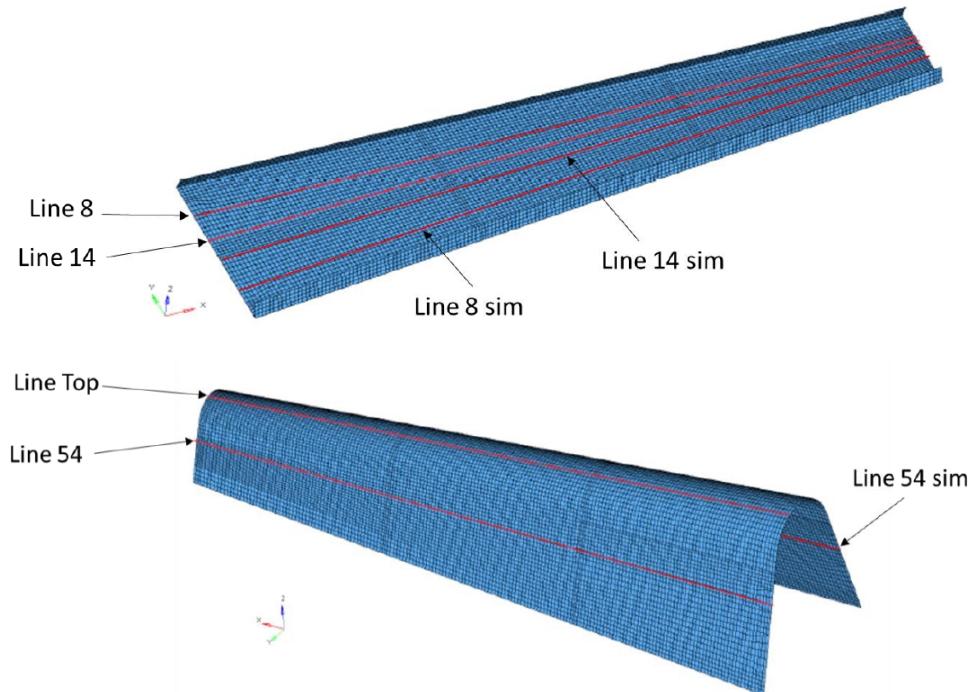


Figura 3.4: Red de sensores OBR

Es importante indicar que, puesto que los sensores de medida distribuida son unidireccionales, se han obtenido las deformaciones en la dirección longitudinal de la fibra (tracción/compresión).

### 3.1.2. Planteamiento de daños y descripción de ensayos

El objetivo de este proyecto es tener un sistema SHM para detectar daños en un fuselaje de material compuesto durante su operación, por lo tanto, se necesita definir la localización de estos daños y sus tamaños.

Los daños corresponden con desencolados parciales o totales en las zonas que se definirán a continuación. En la Figura 3.5 se puede ver la localización de cada daño en la estructura y en la Tabla 3.1 se describen sus características.

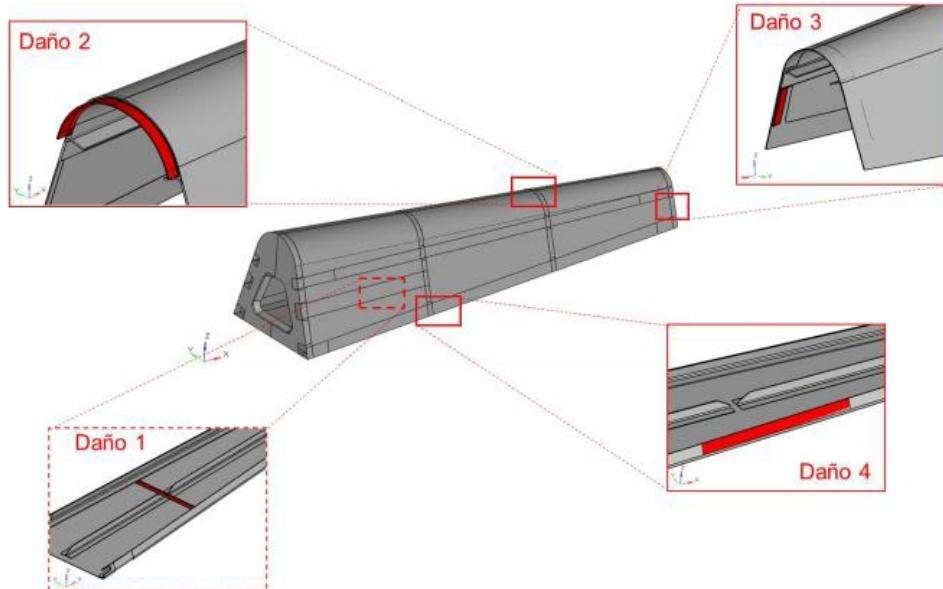


Figura 3.5: Esquema de los daños realizados a la estructura

DAÑO	TIPO DE DAÑO	ELEMENTOS AFECTADOS	LONGITUD MÁXIMA	INCREMENTO
D1	Desencolado parcial del pie de la cuaderna	Cuaderna 2, revestimiento inferior	450 mm	30 mm
D2	Desencolado parcial de la pestaña superior de la cuaderna	Cuaderna 3, revestimiento superior	350 mm	25 mm
D3	Desencolado parcial de la pestaña lateral de la cuaderna	Cuaderna 4, revestimiento superior	180 mm	30 mm
D4	Desencolado parcial entre el revestimiento inferior y el revestimiento superior	Revestimiento inferior, revestimiento superior	900 mm	30 mm

Tabla 3.1: Definición de daños en el fuselaje

Es importante resaltar que, aunque el ensayo estructural y las simulaciones se realizaron para los cuatro daños, se llegó a la conclusión de que las diferencias en el campo de deformaciones que producía el daño 3 respecto al de la estructura sin daño era inapreciable

tras ver los resultados obtenidos. Por lo tanto, dicho daño fue descartado.

Para facilitar el trabajo a los operarios que realizaron los ensayos del fuselaje, se sustituyó la unión adhesiva por uniones remachadas. Esto permitió que fuera más sencillo controlar la progresión de los daños y también su reparación tal y como se ve en la Figura 3.6.

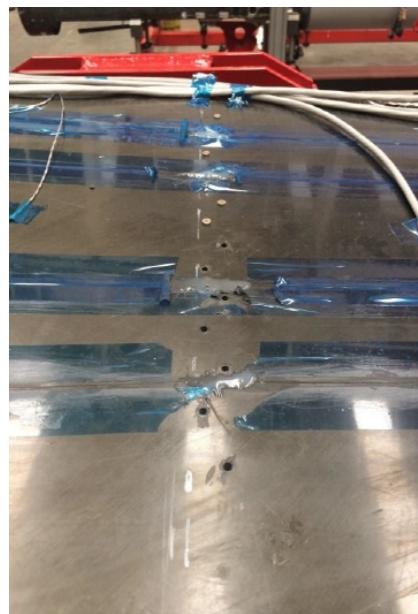


Figura 3.6: Realización del daño por extracción de remaches

Los ensayos sobre la estructura real se simularon también siguiendo la misma configuración en el FEM. Las condiciones de contorno a las que está sometido el fuselaje son las siguientes:

- Empotramiento en la sección de unión con el fuselaje anterior
- Carga vertical en la sección de unión con la cola

El ensayo comienza con la estructura descargada y se aplica la carga de forma escalonada. La carga va aumentando 400 N en 10 escalones hasta llegar a un máximo de 4 kN.

La configuración del banco de ensayos se puede ver en la Figura 3.7

### 3.1.3. Modelo de elementos finitos

Se va a hacer un breve resumen de las características del FEM utilizado. Este modelo consta de 62802 grados de libertad, el tamaño medio de sus elementos es de 15 mm, y está formado por:

- 26181 nodos
- 34080 elementos quad4 (Ratio máximo de 4.768)
- 25 elementos tria3 (Ratio máximo de 6.219)

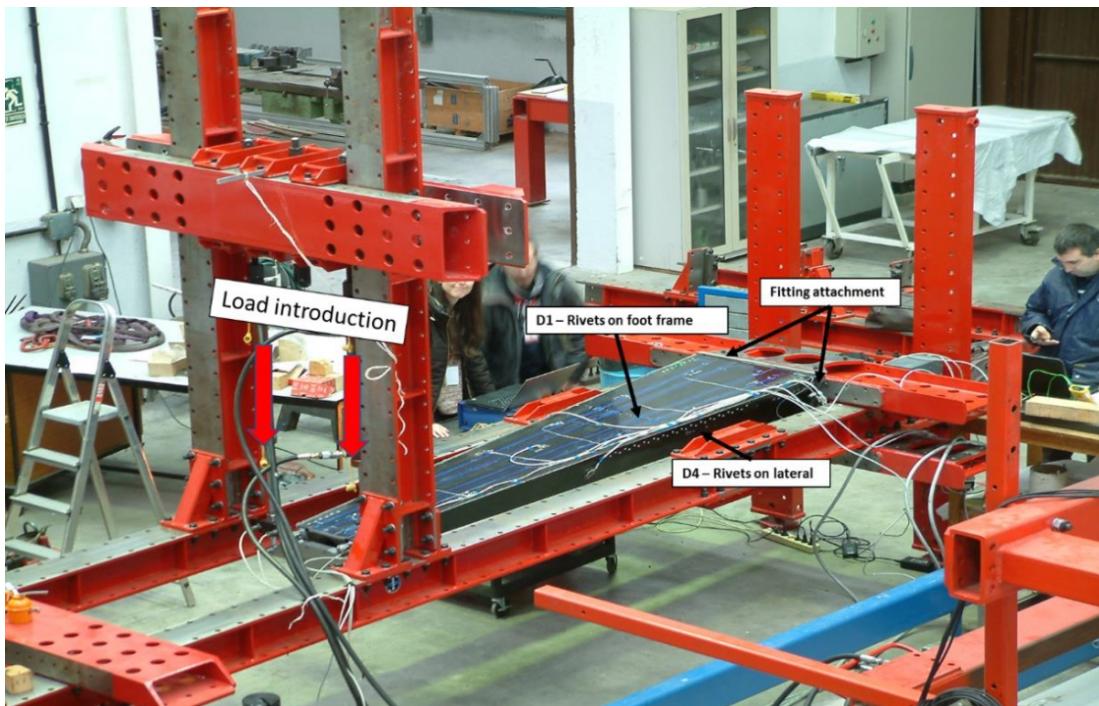


Figura 3.7: Vista general de la configuración de los ensayos

Los elementos tria3 del modelo se encuentran principalmente en las cuatro cuadernas debido a que su geometría es más complicada de mallar que la del resto del fuselaje.

En la Figura 3.8 se pueden ver los distintos elementos del modelo y en cuales de ellos se introducen las condiciones de contorno.

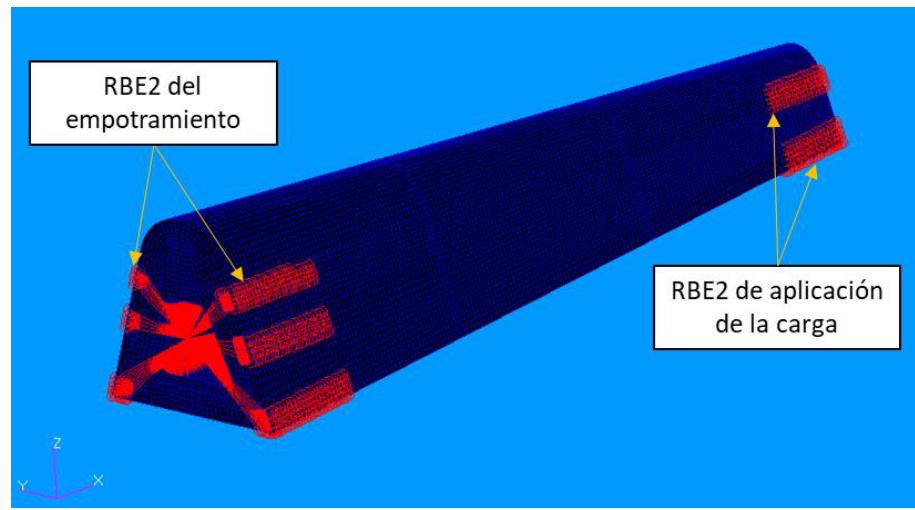


Figura 3.8: Modelo de elementos finitos del fuselaje posterior del UAV Milano

### 3.1.4. Pre-procesado de señales

En este trabajo solo se van a utilizar los datos obtenidos de los ensayos sobre la estructura real debido a que el tiempo disponible es limitado.

Una vez que se han realizado los ensayos y simulaciones, es necesario pre-procesar los datos recogidos por las redes sensoriales para ajustarlos en contenido y forma para que las redes neuronales puedan realizar una clasificación correcta.

#### 3.1.4.1. Sensores FBG

Durante el ensayo completo, la red de sensores FBG registra las deformaciones de forma ininterrumpida, dando así una curva de deformación frente al tiempo para cada uno de los 20 sensores distribuidos en la estructura, tal y como se representa a continuación en la Figura 3.9.

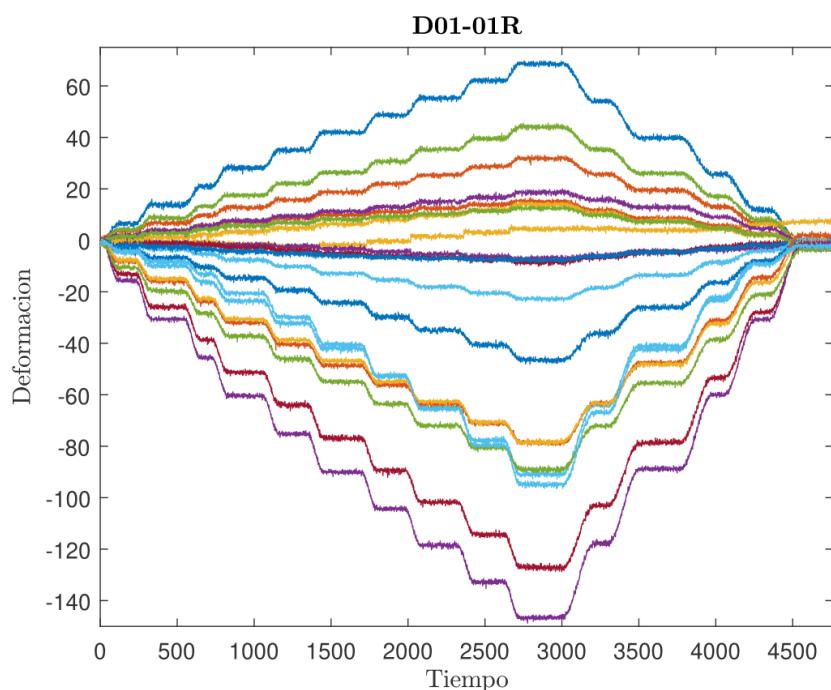


Figura 3.9: Medida de todos los sensores durante el ensayo del Daño 1 con un remache extraído

Cuando los actuadores llegan a un nivel de carga paran durante un tiempo antes de aumentar o disminuir la carga aplicada con el fin de estabilizar la carga. Por lo tanto, para separar las medidas en diferentes niveles de carga hay que detectar cuando el actuador comienza a moverse. La detección de el comienzo y final de aplicación de carga está representada en las Figuras 3.10 y 3.11.

Finalmente se eliminan las medidas de las transiciones entre escalones usando como referencia las líneas naranjas de la Figura 3.10 y se fija la carga mínima que se va a usar durante el estudio en un 30 % de la carga máxima.

Una vez que se han aislado los escalones se dibuja un histograma para ver la variabilidad de los datos obtenidos. Como se ve en la Figura 3.12, los valores de las medidas

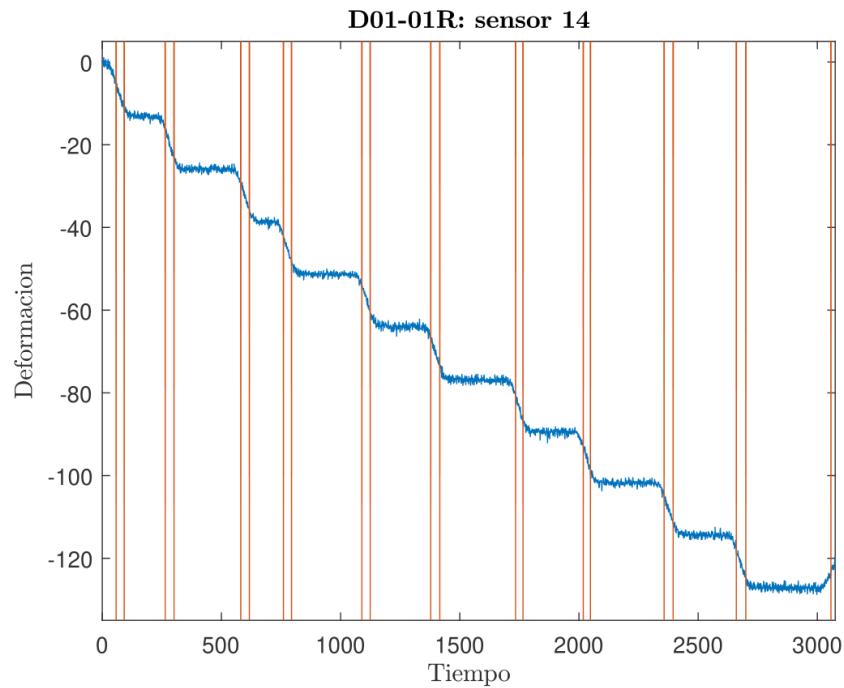


Figura 3.10: Detección de los escalones producidos por las carga vertical

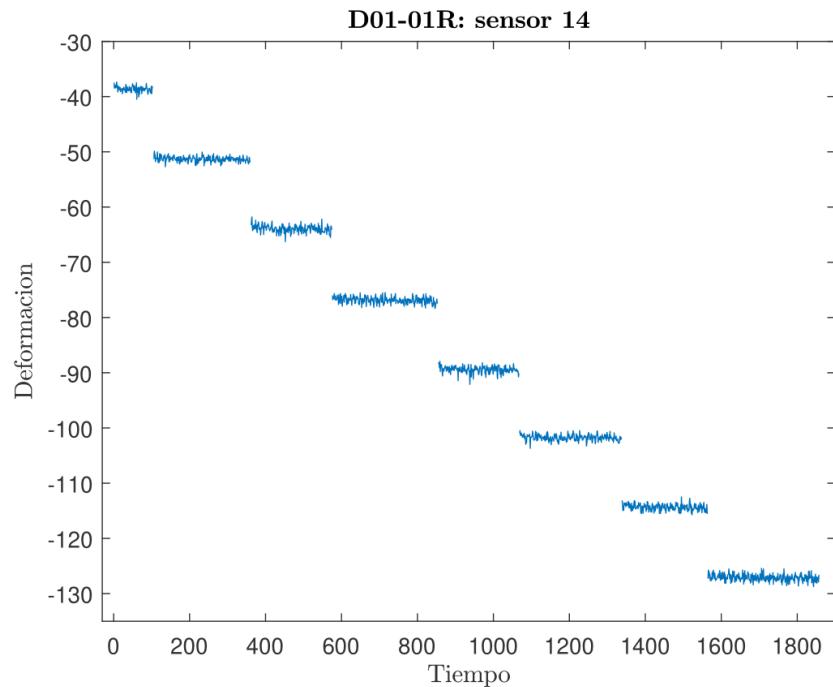


Figura 3.11: Medidas de los sensores FBG en el ensayo preprocesadas

se asemejan a una distribución normal con moda -127.75 deformaciones. Esto quiere decir que el ruido introducido por los sensores FBG hace que cada medida que se realiza en un tiempo diferente pueda ser considerada como un ensayo independiente. De un único ciclo de carga se obtienen una gran cantidad de datos para entrenar a la red.

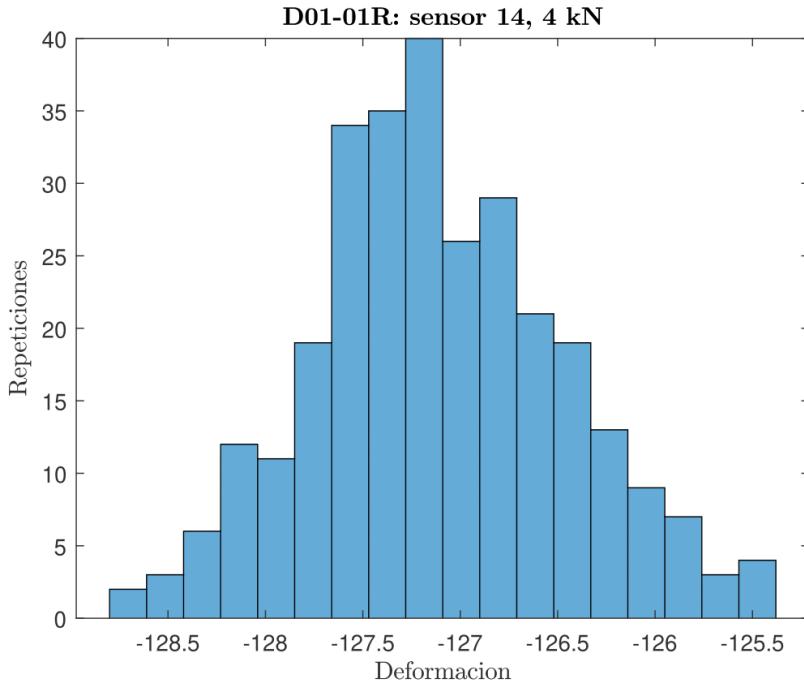


Figura 3.12: Histograma del escalón 4 kN, 100 % de carga

Tras pre-procesar las señales se puede visualizar el campo de deformaciones medido por los sensores FBG para cada estado de daño y el estado sin daño o de referencia en la Figura 3.13. Al estar cada daño localizado en una zona diferente de la estructura, los cambios en el campo de deformaciones son locales y en la Figura 3.14 se aprecia que sensores están más cerca de la zona dañada ya que sufren una mayor variación respecto a las medidas obtenidas de la estructura sin daño.

### 3.1.4.2. Sensores OBR

Por otro lado, los sensores OBR realizan una captura instantánea del campo de deformaciones. Esto quiere decir que se tiene una única muestra para cada tipo de daño y nivel de carga (ejemplo en la Figura 3.15), lo cual es insuficiente para conseguir unos buenos resultados con la red.

Siguiendo el mismo razonamiento usado con los sensores FBGs, si se añade una distribución de ruido gaussiano a un valor del campo de deformaciones, se pueden generar nuevos ensayos independientes con los que entrenar a la red. Sin embargo, el campo de deformaciones de la Figura 3.15 ya contiene ruido asociado a la medida.

Para conseguir un campo de deformaciones que usar como referencia, al que añadir posteriormente ruido, se va a utilizar el filtro *Savitzky-Golay*. Este filtro digital es comúnmente usado para suavizar series de datos equiespaciados [43].

La diferencia entre la medida sin filtrar y filtrada se aprecia claramente en la Figura 3.16.

Ahora que se tiene una medida de referencia, ya se puede añadir la distribución de ruido de amplitud 1.5 deformaciones (Figura 3.17) y generar más muestras.

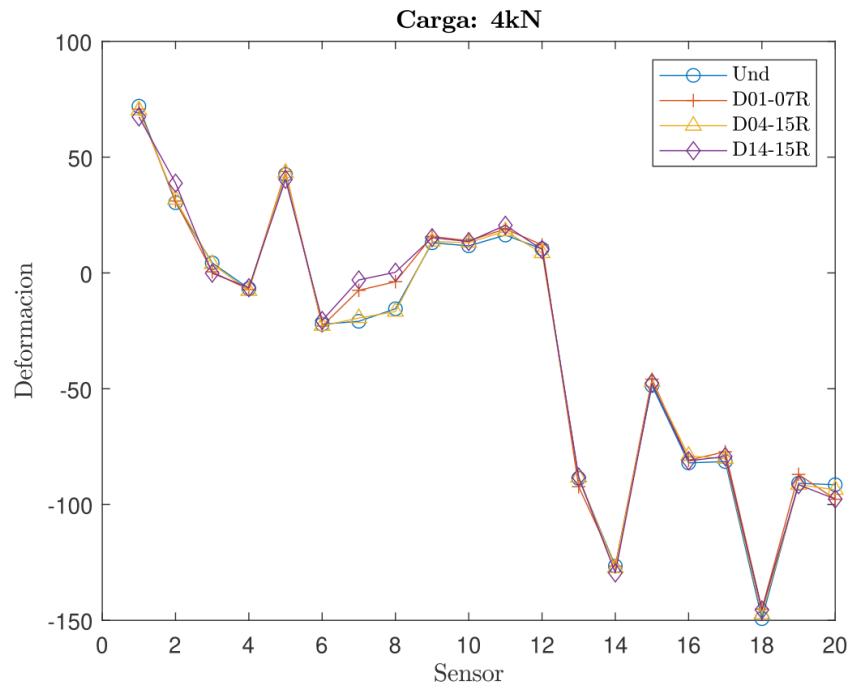


Figura 3.13: Campo de deformaciones medida por los sensores FBG para todos los daños

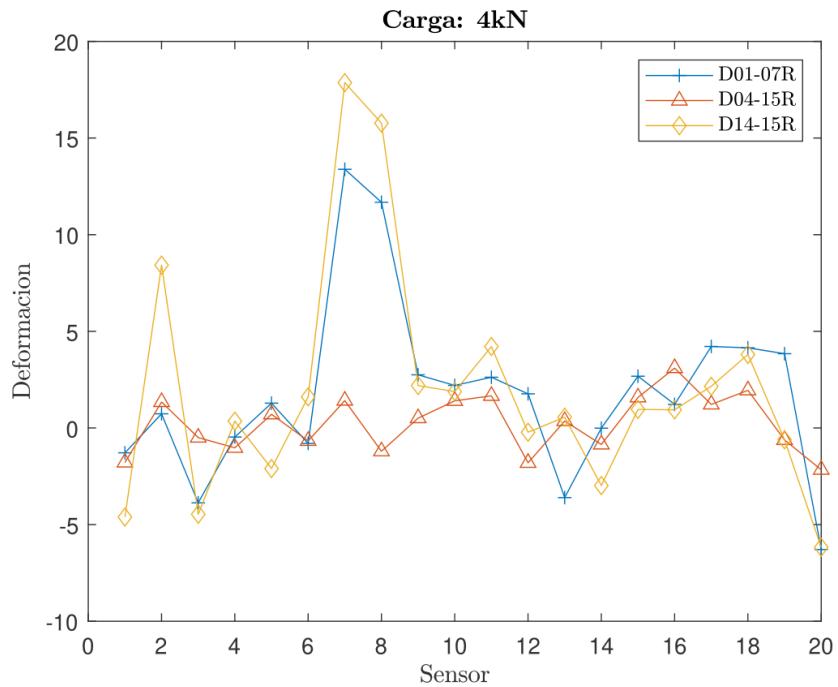


Figura 3.14: Diferencia de los campos de deformaciones referenciado con el estado sin daño (FBG)

En la Figura 3.18 se puede ver un detalle del caso D01-R03 en el que se ha ampliado las deformaciones medidas por una línea de sensor. Aquí se aprecia de forma más clara

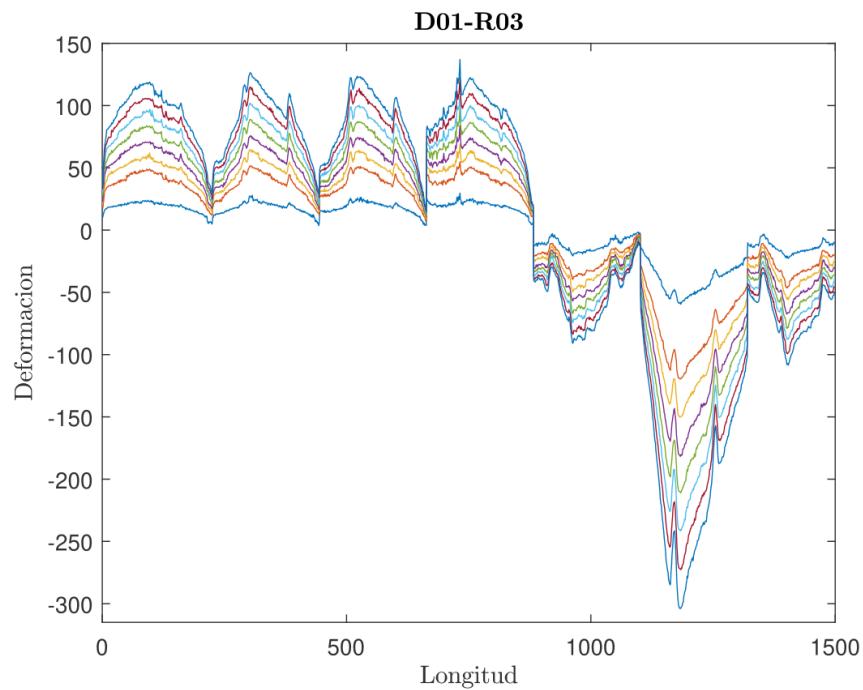


Figura 3.15: Campo de deformaciones provocado por el daño 1 con 3 remaches extraídos para todos los casos de carga

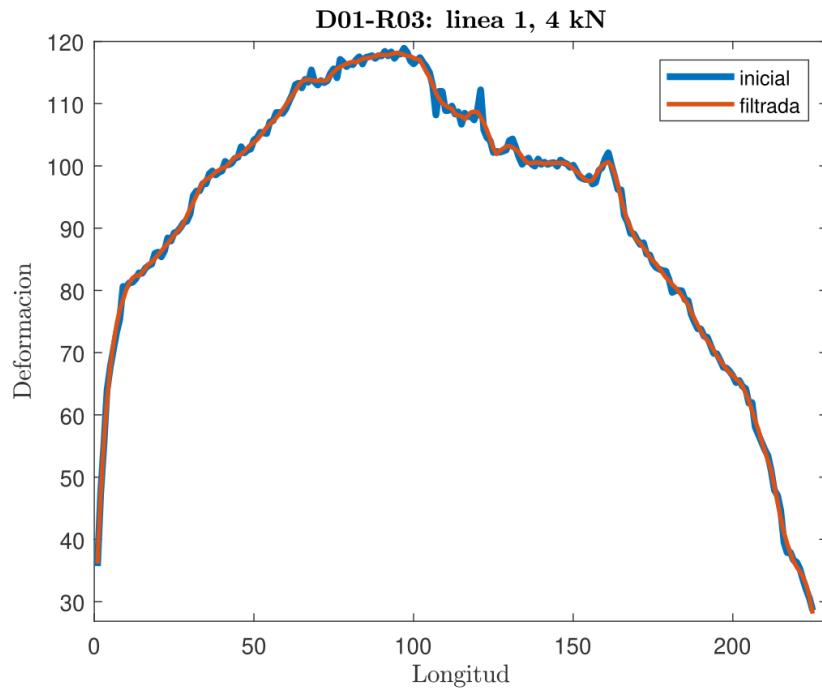


Figura 3.16: Aplicación del filtro Savitzky-Golay a una linea de sensor OBR

que cada una de estas muestras se pueda considerar como un ensayo independiente.

Igual que se ha hecho con los FBGs, en las Figuras 3.19 y 3.20 se pueden ver los campos

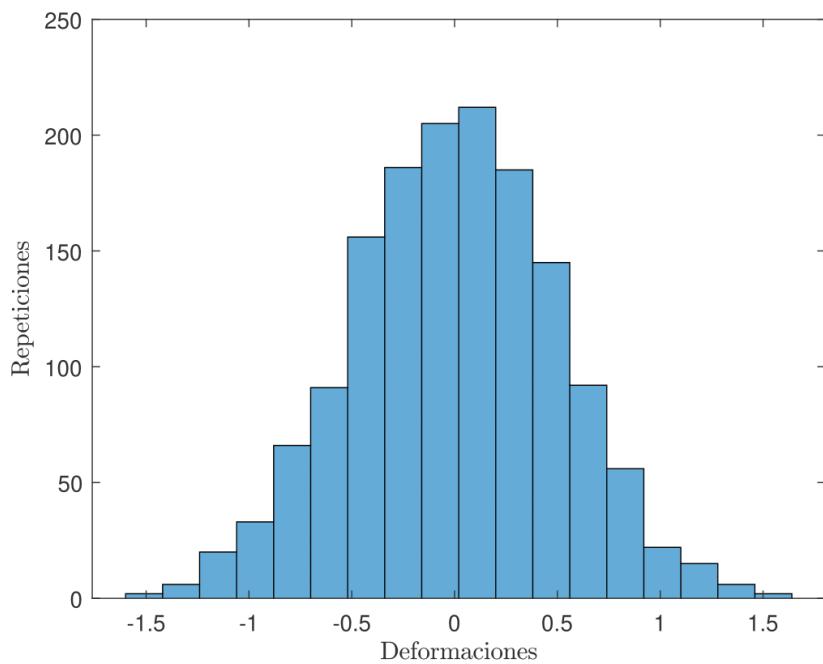


Figura 3.17: Distribución de ruido gausiano con amplitud 1.5 deformaciones

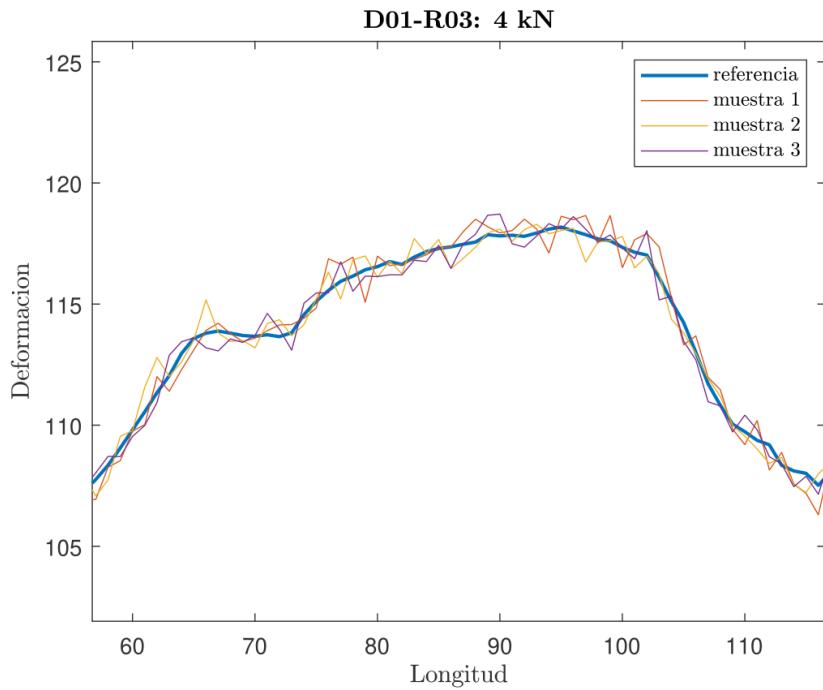


Figura 3.18: Detalle de las variaciones provocadas por el ruido gausiano

de deformaciones provocados por los distintos daños con la carga máxima y su diferencia respecto al estado sin daño.

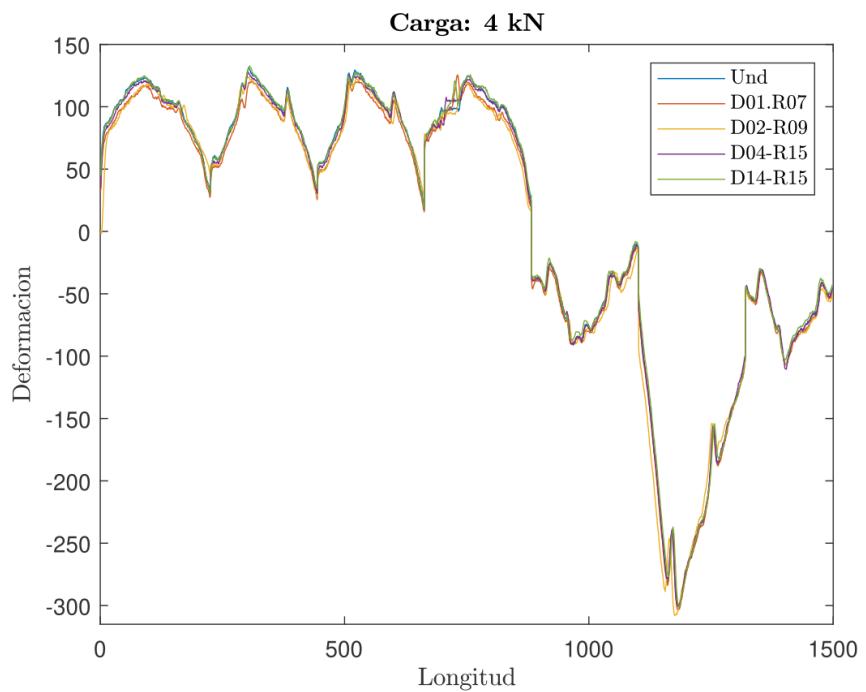


Figura 3.19: Campo de deformaciones medido por el sensor OBR para todos los daños

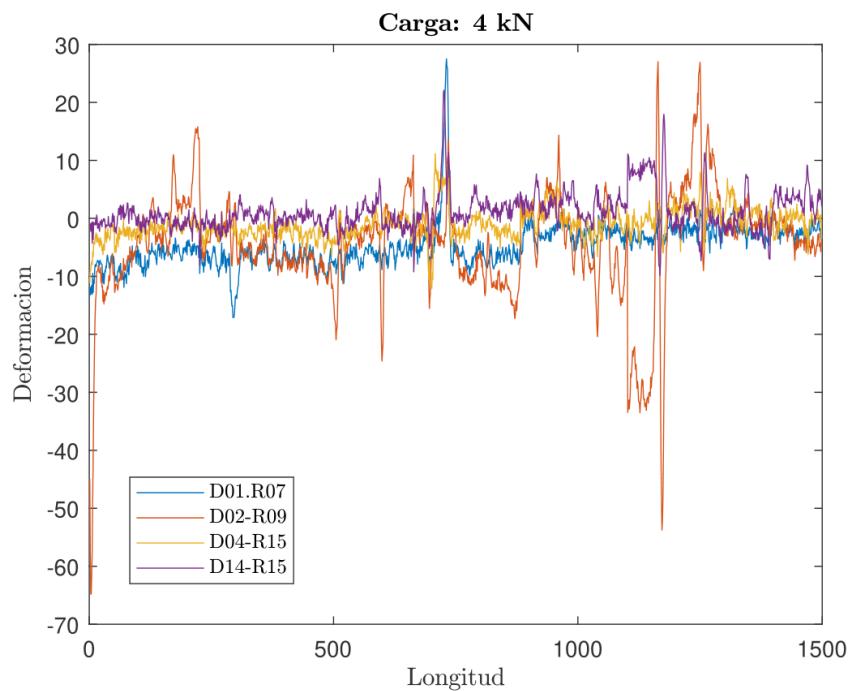


Figura 3.20: Diferencia de los campos de deformaciones referenciado con el estado sin daño (OBR)

## 3.2. Aplicación de DL al a la detección de daños mediante sensores de deformaciones

### 3.2.1. Visualización del campo de deformaciones

Una vez que se han procesado las deformaciones se tiene un DS para los sensores FBG en el que cada muestra es de dimensión 20 (20 sensores) mientras que en el dataset de los sensores OBR las muestras son de dimensión 1499 (equivalente a tener 1499 sensores).

Un ser humano no puede visualizar un espacio con tantas dimensiones. Esto hace complicado tener una idea de la distribución espacial de las muestras y, mucho menos, imaginar las fronteras que genera la NN entre las distintas clases de datos (como se ha explicado en *Representación gráfica del funcionamiento una NN*).

Para reducir la dimensionalidad de los datasets y tener una representación gráfica bidimensional de ellos se va a utilizar un algoritmo de ML muy popular para la visualización de datos multidimensionales: **t-SNE (t-distributed Stochastic Neighbor Embedding)**.

A continuación se va a hacer una breve introducción al funcionamiento de este algoritmo extraído de [[t-sne](#)].

*Stochastic Neighbor Embedding (SNE)* comienza convirtiendo las distancias euclídeas entre puntos del DS de alta dimensionalidad en probabilidades condicionales que representan su similitud. La similitud del punto  $x_j$  con el punto  $x_i$  es la probabilidad condicional  $p_{j|i}$  y su proximidad se calcula usando una distribución de densidad de probabilidad gaussiana centrada en  $x_i$ . Para los puntos cercanos, el  $p_{j|i}$  es relativamente alto, mientras que para los puntos lejanos, el  $p_{j|i}$  será casi infinitesimal. Matemáticamente, el la probabilidad condicional  $p_{j|i}$  viene dada por la Ecuación 3.1.

$$p_{j|i} = \frac{\exp(-\|x_i - x_j\|^2 / 2\sigma_i^2)}{\sum_{k \neq i} (\exp(-\|x_i - x_k\|^2 / 2\sigma_i^2))} \quad (3.1)$$

En el espacio de baja dimensión los puntos  $y_i$  y  $y_j$  son la transformación de  $x_i$  y  $x_j$  del espacio con alta dimensionalidad. Es posible calcular una probabilidad condicional similar,  $q_{j|i}$  a la anterior. Para ello, fijamos la varianza del gaussiano que se emplea en el cálculo de las probabilidades condicionales  $q_{j|i}$  en  $\frac{1}{\sqrt{2}}$ . Por lo tanto, modelamos la similitud del punto  $y_j$  con  $y_i$  con la Ecuación 3.2

$$q_{j|i} = \frac{\exp(-\|y_i - y_j\|^2)}{\sum_{k \neq i} (\exp(-\|y_i - y_k\|^2))} \quad (3.2)$$

Si los puntos  $y_i$  y  $y_j$  del mapa de baja dimensionalidad modelan bien la similaridad entre los puntos  $x_i$  y  $x_j$  del espacio con alta dimensionalidad, las probabilidades  $p_{j|i}$  y  $q_{j|i}$  serán iguales. Por lo tanto, el objetivo de SNE es minimizar las discordancias entre  $p_{j|i}$  y  $q_{j|i}$ .

Igual que para las NN, es necesario tener una función de costes que represente el error que está cometiendo el algoritmo y así optimizarla por medio del descenso del gradiente. La función de costes elegida es la divergencia Kullback-Leibler individual entre las

distribuciones de probabilidad conjunta, P y Q (Equación 3.3).

$$C = KL(P||Q) = \sum_i \sum_j p_{ji} \log \frac{p_{ji}}{q_{ji}}$$
 (3.3)

Este algoritmo de visualización se va a aplicar a ambas redes sensoriales. Con esto se va a conseguir visualizar la similitud entre los distintos tipos de daño, su tamaño y la carga aplicada para entender mejor el rendimiento de las redes en su tarea de clasificación.

En las Figuras que se van a comentar a continuación se han representado los campos de deformaciones multidimensionales reducidos a dos dimensiones.

Con estos resultados se puede comprender el potencial de t-SNE. En las figuras que representan los tipos de daños (3.21, 3.22, 3.24 y 3.22) se ha utilizado el parámetro *perplexity* bajo, orden 50. Por otro lado, en las que representan la carga (3.23 y 3.23) se ha utilizado un *perplexity* de orden 100.

En las Figuras 3.23 y 3.23 se observa que las muestras que están sometidas a un mismo nivel de carga están próximas en el espacio y separadas de los otros grupos que forman el resto de cargas y lo mismo ocurre con los tipos y tamaños de daño. Por lo tanto, las muestras se separan en grupos de niveles de carga y, dentro de cada carga, se agrupan por tipo y tamaño de daño.

Viendo los resultados se llega a la conclusión que variando el parámetro *perplexity* se puede ajustar el nivel de abstracción del algoritmo.

#### • Sensores FBG

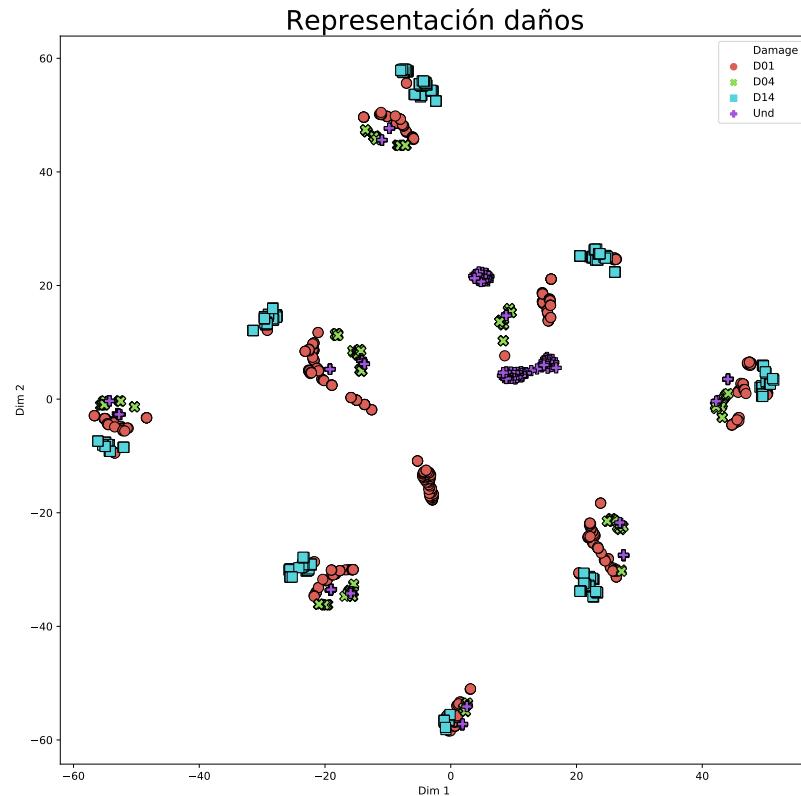


Figura 3.21: Visualización de los distintos tipos de daño

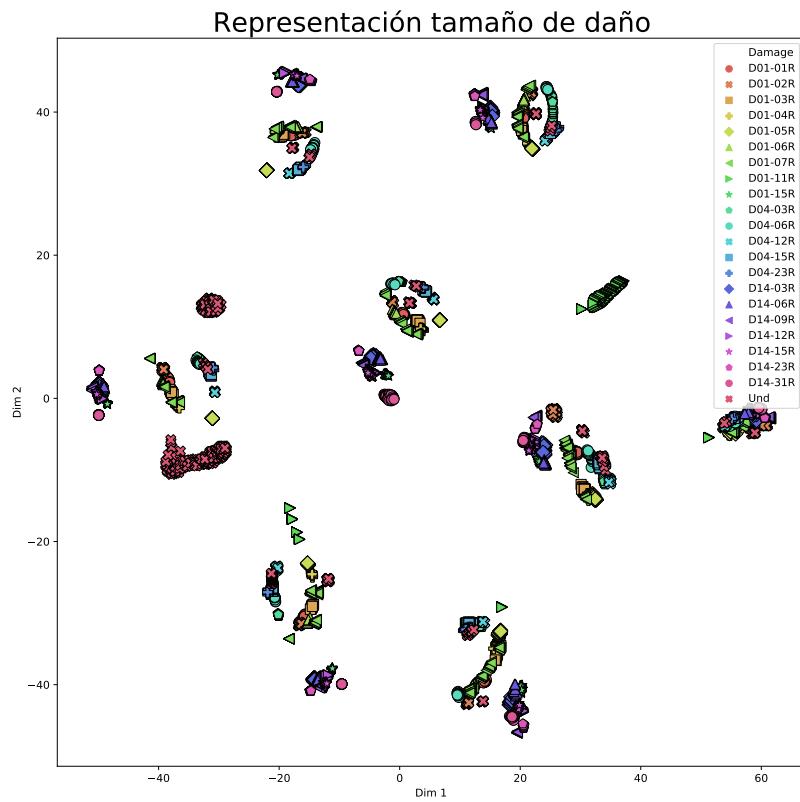


Figura 3.22: Visualización de los distintos tamaños de daño

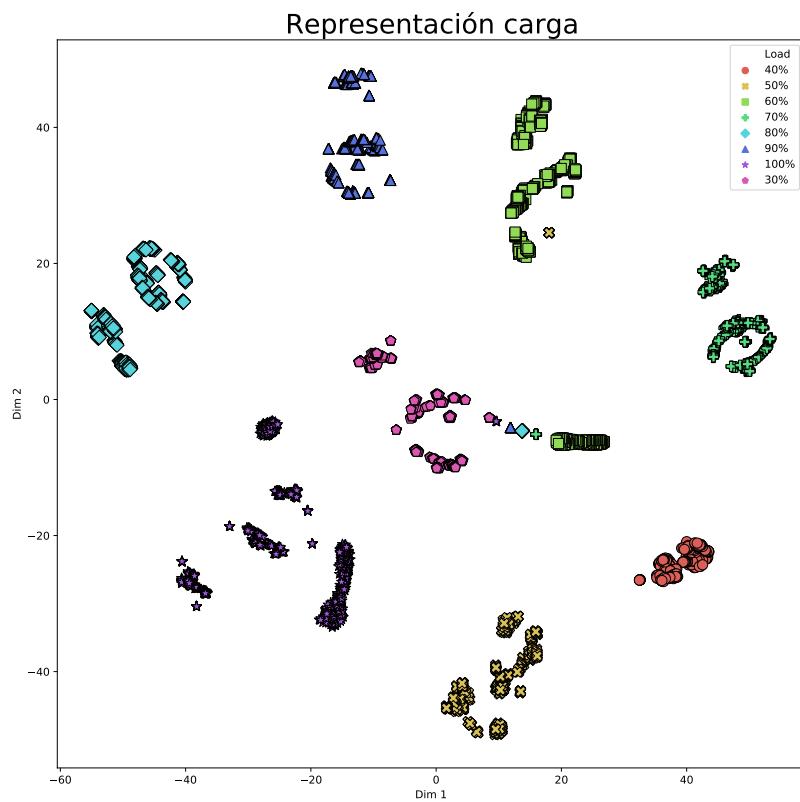


Figura 3.23: Visualización de los distintos niveles de carga

- Sensores OBR

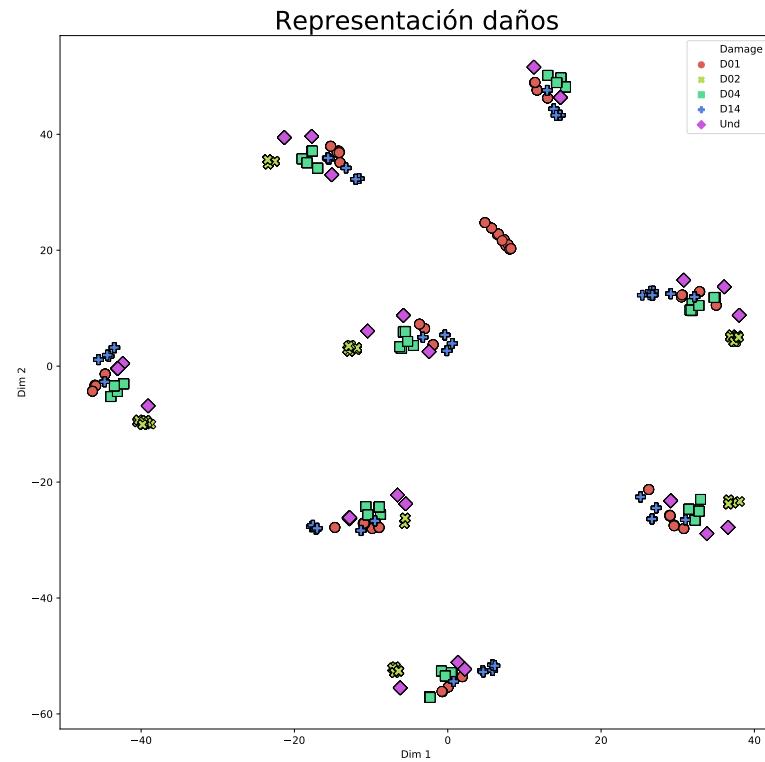


Figura 3.24: Visualización de los distintos tipos de daño

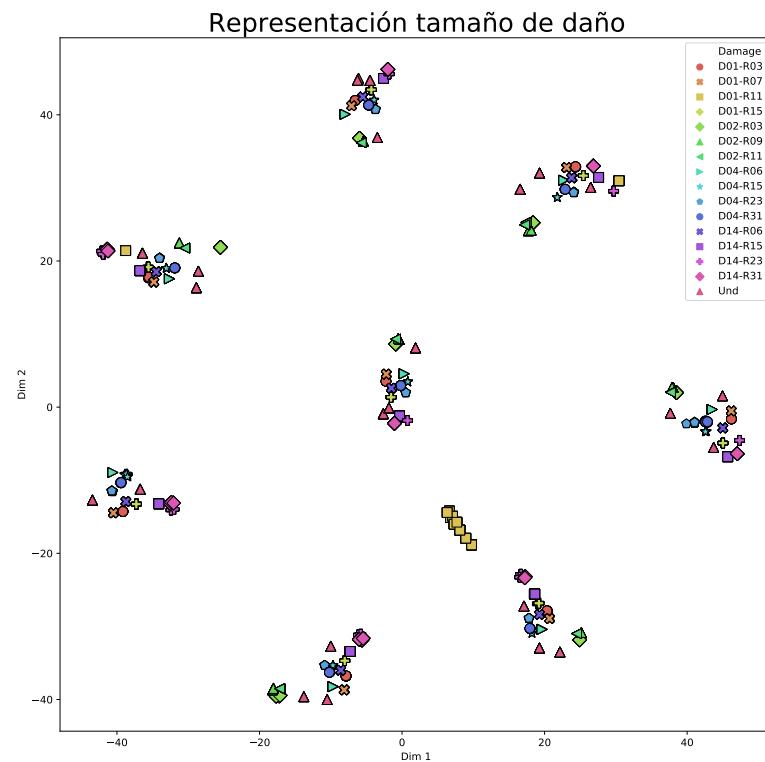


Figura 3.25: Visualización de los distintos tamaños de daño

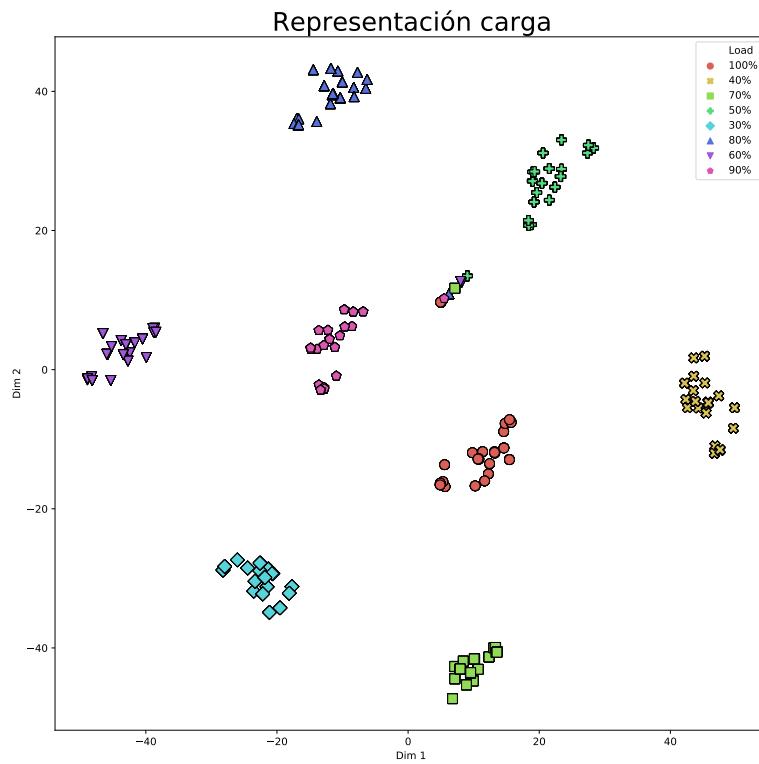


Figura 3.26: Visualización de los distintos niveles de carga

Analizando con más profundidad la Figura 3.21 se aprecia que los tipos de daño están claramente diferenciados unos de otros, por lo que la tarea de clasificación debería obtener una buena precisión. Si pasamos a la Figura 3.22 se aprecia que los diferentes tamaños del mismo daño se distribuyen en la misma forma que tenían en la Figura 3.21.

Como los grupos están visualmente separados se ha decidido realizar una única clasificación de tamaño de daño ya que integra en sí misma la de tipo de daño.

### 3.2.2. Arquitectura de la Red Neuronal

El tipo de celda principal elegida para este trabajo ha sido la Recurrent Neural Network (RNN). El principal motivo de esta elección es que este tipo de arquitecturas han obtenido buenos resultados a la hora de trabajar con secuencias de datos, más concretamente con secuencias de datos dependientes del tiempo.

En la Figure 3.27 se pueden ver las diversas configuraciones que se pueden realizar cuando se trabaja con estas arquitecturas. Se puede llegar a utilizar un único vector de entrada y tener un único vector de salida, hasta varios vectores de entrada y varios de salida; pasando por todas las combinaciones intermedias.

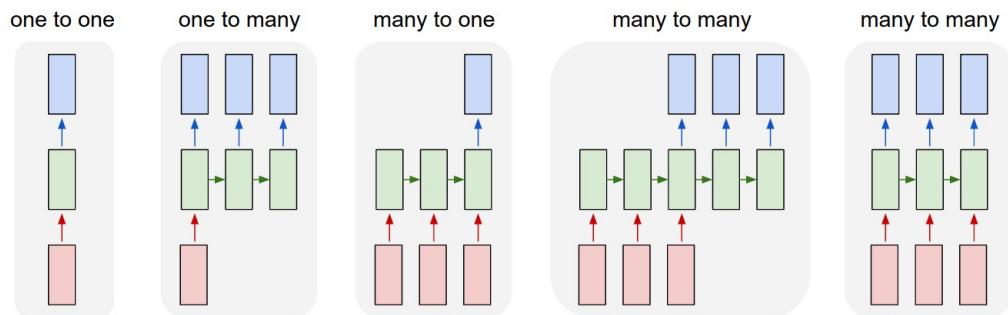


Figura 3.27: Posibles combinaciones de secuencias con RNN [38]

Otra de las características que ha hecho muy conocidas a estas arquitecturas es que, como su nombre indica, son recurrentes. Esto quiere decir que una vez la neurona ha generado una salida, esta información no solo viaja a la siguiente capa, sino que también se utiliza para realimentar la propia neurona. Cuando una neurona recibe un input, este es la combinación de información nueva y su salida anterior. Para entenderlo mejor se puede ver en la Figura 3.28 lo que se conoce como una RNN “unrolled”<sup>a</sup> lo largo del tiempo.

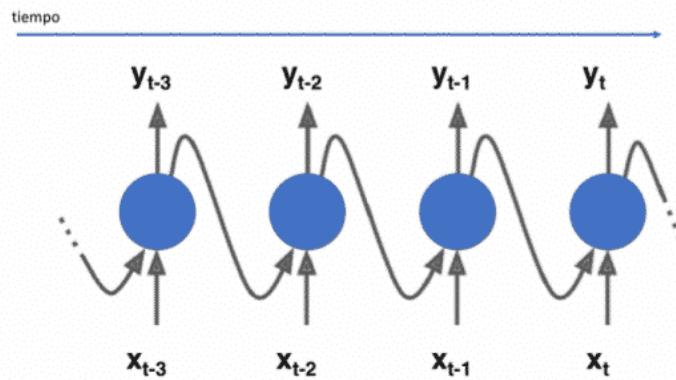


Figura 3.28: RNN “unrolled” [39]

Con esto se llega a la conclusión que la salida de la neurona recurrente es una función de las entradas de los tiempos anteriores, lo que hace a las RNN tener memoria.

Esta propiedad, junto a la variabilidad de tamaños de entradas que puede recibir, es la que hace a las RNN tan apropiadas para trabajar con datos secuenciales. Gracias a la memoria las RNN son capaces de relacionar destinas posiciones de una secuencia de forma efectiva aunque estén distantes en el tiempo [39].

A pesar de que las RNN, idealmente, son capaces de mantener en memoria toda la información asociada a la secuencia introducida, a la hora de ser aplicadas en una tarea no son capaces de conectar toda la información necesaria a lo largo de la secuencia de datos.

Se va a utilizar un ejemplo de autocompletado de palabras generando texto [17] para explicar mejor esta dificultad. Cuando el hueco a completar está cercano a las palabras que contienen el contexto de la oración las RNN funcionan bien. Sin embargo, cuando este hueco está alejado al grupo de palabras que contienen la el contexto las RNN no tienen buenos resultados.

Para resolver el problema de las dependencias temporales nacieron las redes Long Short Term Memory (LSTM). Las redes LSTM están específicamente diseñadas para guardar en memoria estas dependencias temporales alejadas unas de otras. Internamente tienen la configuración mostrada en la Figura ??.

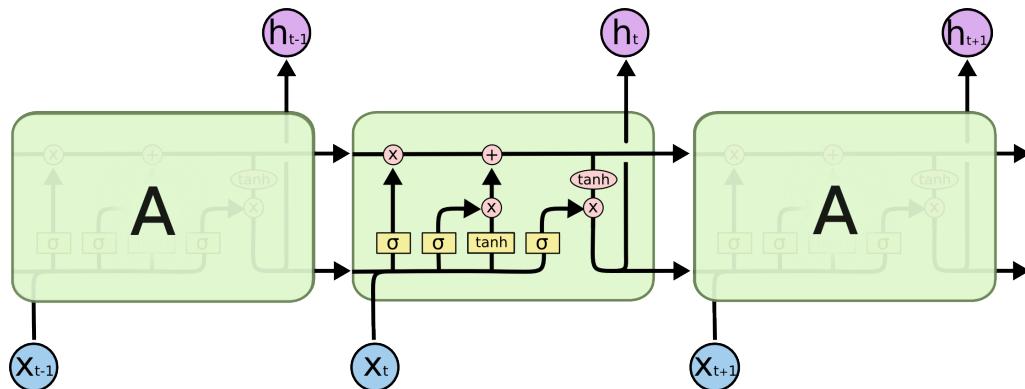


Figura 3.29: Configuración interna de la red LSTM [42]

Este tipo de redes se ha utilizado sobre todo en aplicaciones de NLP, como puede ser la clasificación de texto [14], traducción [10] o generación automática de descripciones con imágenes [11].

Sin embargo, en este trabajo se va a aplicar a la detección de daños y localización de impactos. Para ello, se ha llevado a cabo un cambio de concepto de lo que es un campo de deformaciones, pasando a entenderlo como una descripción de la estructura que hay que clasificar dentro de una determinada clase. Igual que se clasificaría un texto como comentario positivo o negativo.

Las deformaciones se tratan a nivel de red de la misma forma que si fueran palabras. La red LSTM va acumulando en memoria el estado de la estructura a medida que se introducen deformaciones hasta que interrelaciona todas estas deformaciones por muy separadas que estén en la estructura y, finalmente, saca como resultado la localización y tamaño del daño.

#### • Arquitectura usada con los sensores FBG

Para el clasificador de daños y nivel de carga se ha utilizado una arquitectura prácticamente igual, solo diferenciándose en el número de neuronas de la última capa ya que éste coincide con el número de clases a clasificar.

El funcionamiento de estas redes se puede separar en dos zonas. La primera es la etapa de memoria, esta consiste en introducir de forma secuencial el campo de

deformaciones y realimentar la celda *LSTM* con el output de la deformación anterior. En la segunda etapa, se ajusta el tamaño del vector de salida de la *LSTM* hasta el número de clases con una capa *Fully Connected* y se aplica la función de activación *Log Softmax*.

En la Figura 3.30 se puede ver el diagrama de bloques de lo descrito anteriormente.

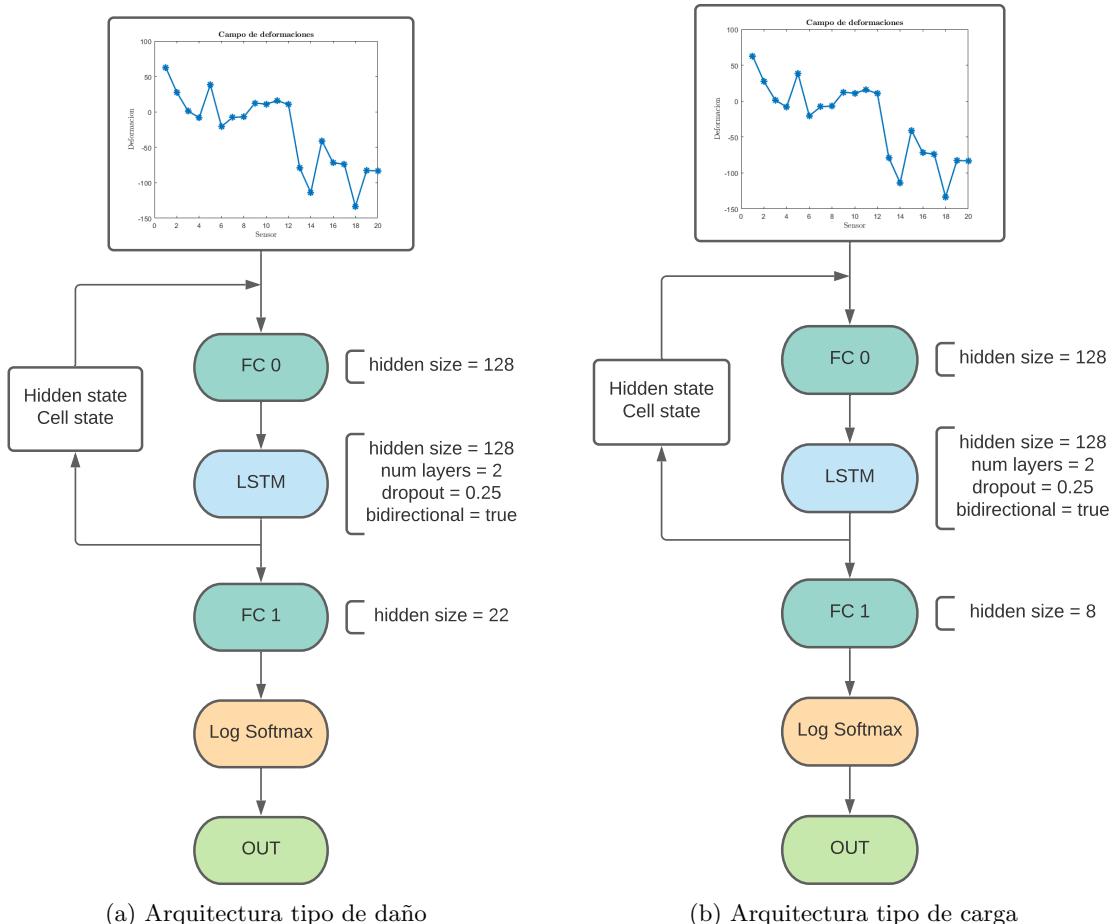
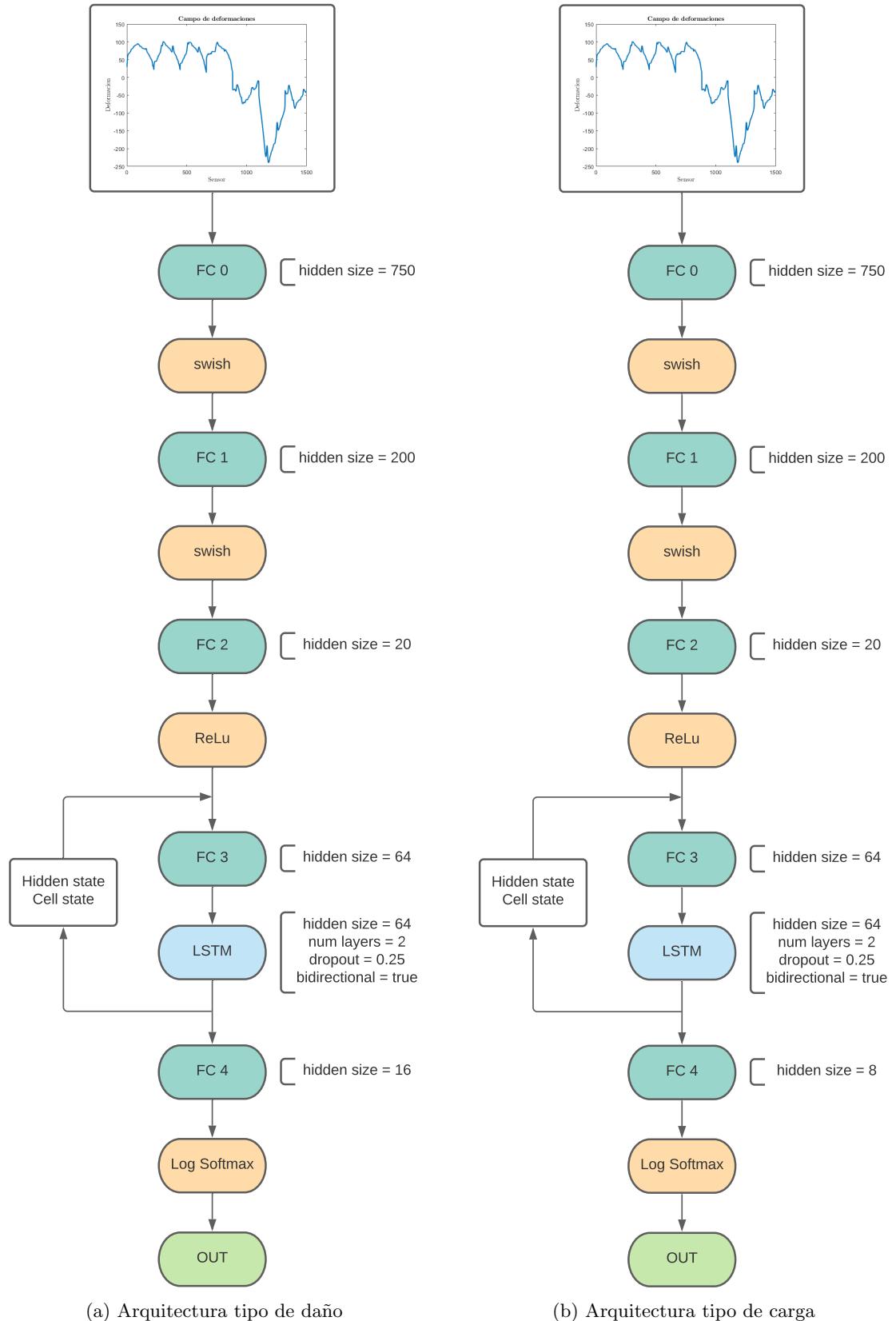


Figura 3.30: Arquitecturas utilizadas en las redes de los sensores FBG

#### • Arquitectura usada con los sensores OBR

Por otro lado está la arquitectura usada para los sensores OBR. Sigue la misma filosofía que la de los sensores FBG, pero se incluyen varias capas FC seguidas de sus funciones de activación para comprimir el campo de deformaciones desde 1499 puntos a los 20 que se usaban en el caso anterior. A partir de la FC 3, el proceso es el mismo que el anterior.

En la Figura 3.31 se puede ver el diagrama de bloques de las arquitecturas para los sensores OBR.



(a) Arquitectura tipo de daño

(b) Arquitectura tipo de carga

Figura 3.31: Arquitecturas utilizadas en las redes de los sensores OBR

### 3.2.3. Resultados de la clasificación

La forma de representar el funcionamiento de una NN en la tarea de clasificación elegida es mediante el uso de Matrices de Confusión (CM). Estas matrices sirven para comprobar la precisión de la red y tener una referencia visual de cuales son las clases que son clasificadas de forma errónea.

En estas matrices, cada columna pertenece a una única clase y la suma de todas las celdas da el número total de muestras de esa clase. Si la red clasifica una muestra correctamente, se sumará uno en la diagonal principal de la matriz, si realiza una predicción errónea se sumará uno a la fila correspondiente a la clase que ha dicho la red, pero dentro de la columna correspondiente a la clase verdadera.

Por lo tanto, cuanto mayor sea el valor de las celdas de la diagonal principal de la CM, mejor realizará la red la tarea de clasificación.

- Sensores FBG

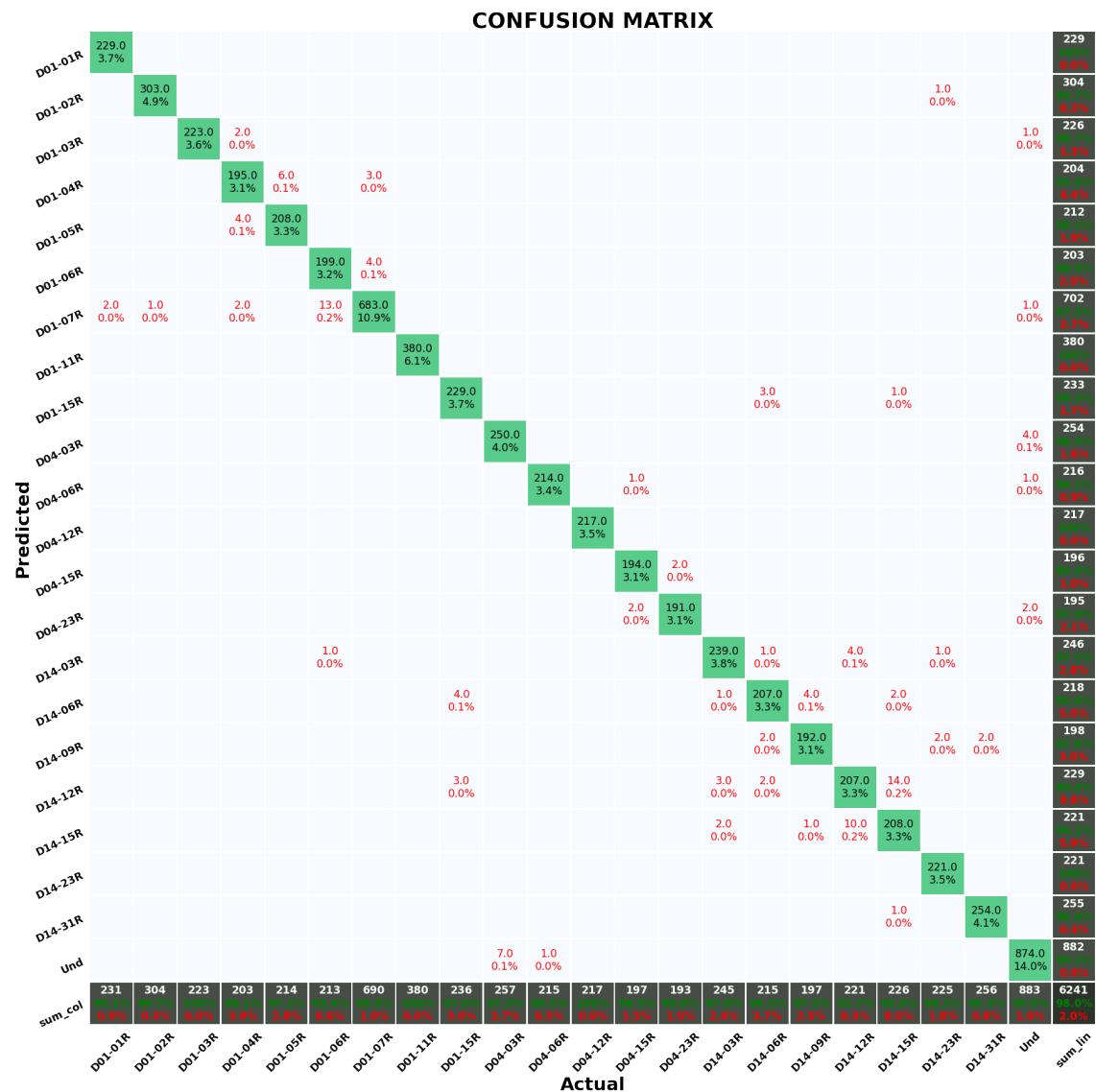


Figura 3.32: Matriz de confusión: Tamaños de daño FBG

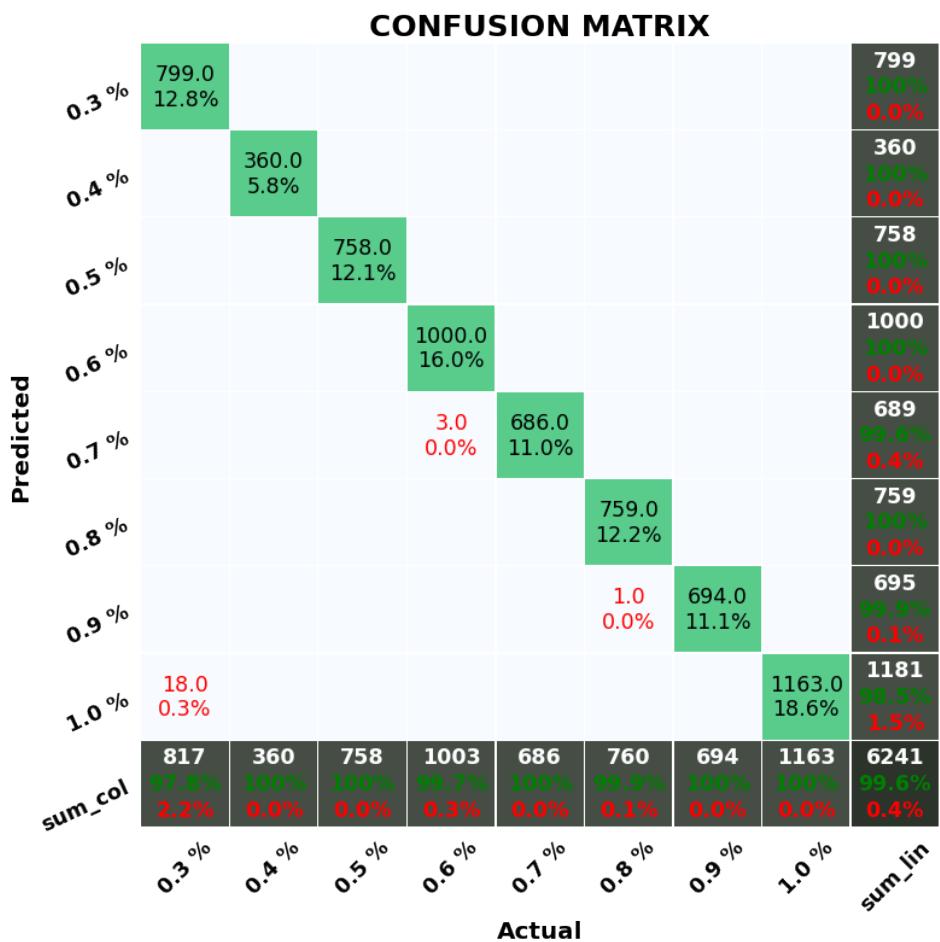


Figura 3.33: Matriz de confusión: Nivel de carga FBG

- Sensores OBR

### CONFUSION MATRIX

		Actual																	
		D01-R03	D01-R07	D01-R11	D01-R15	D02-R03	D02-R09	D02-R11	D04-R06	D04-R15	D04-R23	D04-R31	D14-R06	D14-R15	D14-R23	D14-R31	Und	sum_col	sum_in
Predicted	D01-R03	90.0 5.0%	2.0 0.1%															92 97.5% 2.2%	
	D01-R07	2.0 0.1%	88.0 4.9%															92 95.7% 4.3%	
Predicted	D01-R11		97.0 5.4%															97 100% 0.0%	
	D01-R15		110.0 6.2%															110 100% 0.0%	
Predicted	D02-R03		96.0 5.4%															96 100% 0.0%	
	D02-R09		84.0 4.7%															89 94.4% 5.6%	
Predicted	D02-R11		79.0 4.4%															79 100% 0.0%	
	D04-R06	12.0 0.7%																97 87.6% 12.4%	
Predicted	D04-R15																	106 98.1% 1.9%	
	D04-R23																	109 100% 0.0%	
Predicted	D04-R31																	108 100% 0.0%	
	D14-R06																	97 100% 0.0%	
Predicted	D14-R15																	98 100% 0.0%	
	D14-R23																	104 100% 0.0%	
Predicted	D14-R31																	120 100% 0.0%	
	Und																	291 100% 0.0%	
		104 86.5% 13.5%	90 97.8% 2.2%	97 100% 0.0%	110 100% 0.0%	96 100% 0.0%	84 100% 0.0%	84 94.0% 6.0%	87 97.7% 2.3%	104 100% 0.0%	109 100% 0.9%	109 99.1% 0.9%	98 99.0% 1.0%	98 100% 0.0%	104 100% 0.0%	120 100% 0.0%	291 100% 0.0%	1785 98.0% 1.4%	

Figura 3.34: Matriz de confusión: Tamaños de daño OBR



Figura 3.35: Matriz de confusión: Nivel de carga OBR

Viendo las CM y las Figuras del t-SNE se puede comprobar que las clases que confunden las redes son las que están más próximas en el mapa generado por t-SNE.

### 3.3. Aplicación de DL a la detección de cargas térmicas

En una estructura aeronáutica, no solo los daños provocan cambios en el campo de deformaciones, la temperatura también es un factor importante a tener en cuenta. Una aeronave puede estar sometida 50°C en tierra y a -50°C durante un vuelo. Este amplio rango de temperaturas genera cargas y deformaciones de un valor equivalente a las aerodinámicas.

Cuando la temperatura no es uniforme puede generar concentración de tensiones equivalentes a las que provoca un daño localizado, por lo que su estudio es apropiado en SHM.

Ya que este trabajo consiste en una primera aproximación al uso de DL para la detección de cargas térmicas, solo se va a estudiar el caso de varias temperaturas uniformemente distribuidas. Las temperaturas que se han elegido son -10°C, 0°C, 10°C, 20°C, 30°C, 40°C siendo la temperatura de referencia la de curado del material, 180°C.

#### 3.3.1. Estructura y modelo de elementos finitos

Muchas de las estructuras aeronáuticas se pueden considerar vigas. Dependiendo de las cargas que soporten durante un vuelo, estas vigas estarán constituidas por unos componentes u otros.

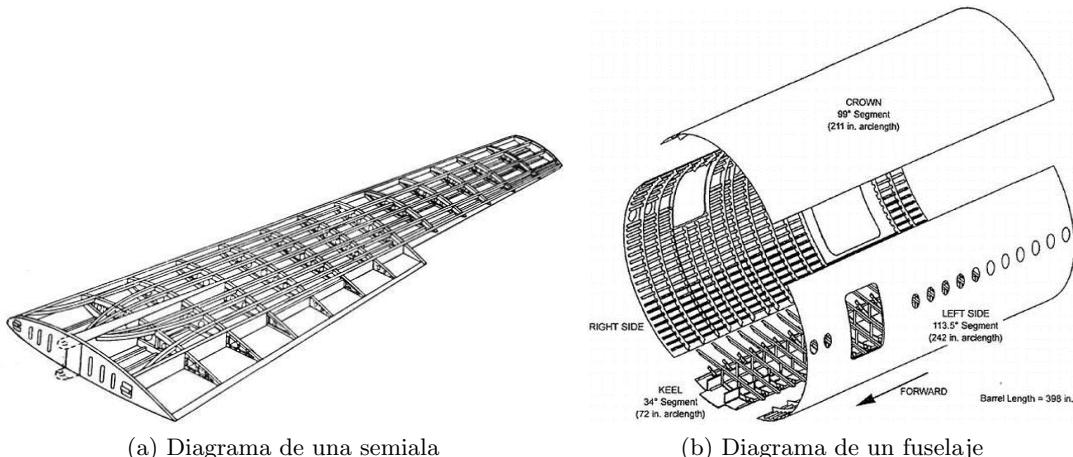


Figura 3.36: Diagramas de estructuras aeronáuticas

En la Figura 3.36 se puede comparar una semiala con un fuselaje. Las semialas están sometidas a flexión y torsión, por ello están formadas por larguerillos y costillas, sin embargo, el fuselaje no trabaja a torsión, por lo que no tiene costillas.

Este estudio se ha centrado en la influencia de diferentes temperaturas sobre una viga fabricada en material compuesto a la que se le han realizado dos tipos de daños adicionales. La viga elegida tiene forma de omega y 4 costillas equiespaciadas, igual a la que se ve en la Figura 3.37. Las dimensiones de la viga son 10000 mm de longitud, una anchura de 850 mm, altura de 500 mm y los laterales forman un ángulo de 60° con la tapa superior.

Para obtener las deformaciones necesarias con las que se alimentará la red se ha creado un modelo de elementos finitos (FEM) utilizando *Nastran-Patran*. Las partes que forman el conjunto de la viga son las siguientes:



Figura 3.37: Imagen de la viga que se va a estudiar

- Tapa superior e inferior fabricada con laminado de 10mm
- Lateral izquiero y derecho fabricada con laminado de 4mm
- Costillas fabricadas con laminado de 4mm

Las características de los laminados se pueden ver en la Tabla 3.2

Espesor laminado	Esperor capa	Número de capas	Orientación
10	0.2	50	$[(\pm 45, 90, 0, 90), 0_{15}]_S$
4	0.2	20	$[(\pm 45, 90, 0_7]_S$

Tabla 3.2: Laminados

Como último paso hay que definir las condiciones de contorno. En la Figura 3.38 se muestra el modelo final una vez creadas las superficies, los elementos y haber aplicado las condiciones de contorno.

Las condiciones de contorno son las siguientes:

- Carga vertical sobre distribuida sobre la parte superior de las costillas centrales
- Empotramiento de los nodos correspondientes a la parte inferior de los laterales.

### 3.3.2. Definición de daños y red sensorial

En esta sección no solo se va a buscar clasificar la temperatura de una estructura, sino que también se le van a añadir dos tipos de daños frecuentes en estructuras de material compuesto.

Estos daños van a ser los siguientes:

- Despegue parcial de la tapa superior y uno de los laterales. El estudio de este daño va a consistir en la localización de varios despegues de un 10 % de la longitud total,

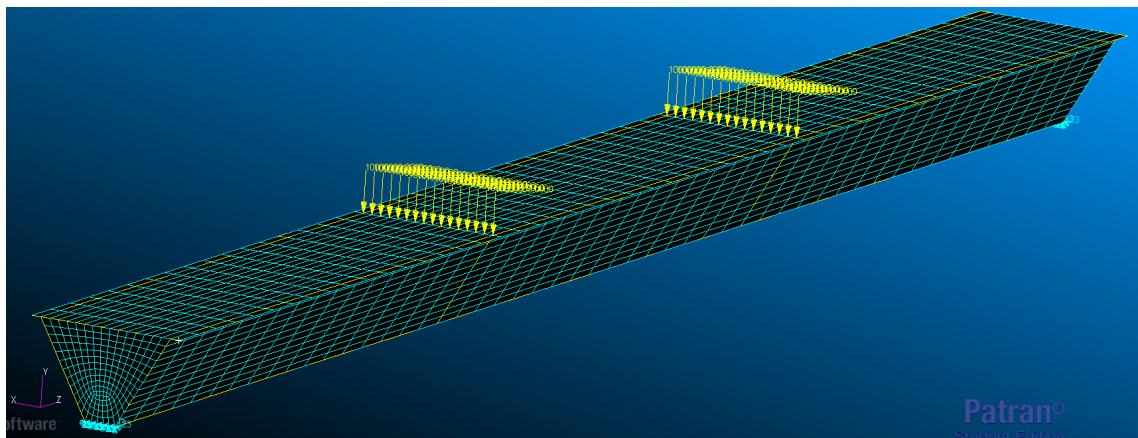


Figura 3.38: Viga con condiciones de contorno aplicadas

desde una costilla lateral hasta la mitad de la longitud total (cinco casos de daño en total). A este daño se le ha denominado *D01*

- Pérdida de rigidez asociada a delaminación. Se va a simular las consecuencias de pérdida de rigidez asociada a un impacto en la zona central-inferior de la viga (unión entre lateral y tapa inferior). El estudio de este daño va a consistir en disminuir la rigidez de un elemento inicial a la mitad y después ir propagando el daño de forma progresiva a los elementos que lo rodean. Se propagará progresivamente el daño hasta tener un total de cinco casos. A este daño se le ha denominado *D02*.

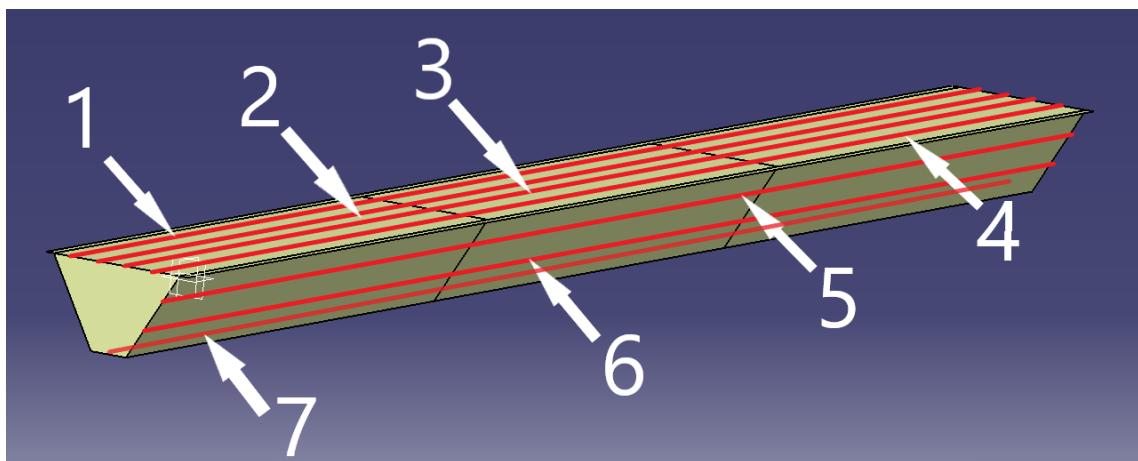


Figura 3.39: EJEMPLOS DE LOS DAÑOS

Para obtener las deformaciones, se van a utilizar varias filas de elementos orientados en la dirección longitudinal. Con esto se consigue simular un sensor OBR similar al que se usó en el proyecto INESASSE. Se pueden ver los elementos que se han seleccionada para hacer de sensor OBR en la Figura 3.40.

### 3.3.3. Pre-procesado de señales

Hay que tener claro cuales van a ser las causas de que la estructura se deforme. Inicialmente, si la estructura se encuentra a la temperatura de curado del material, 180°C,

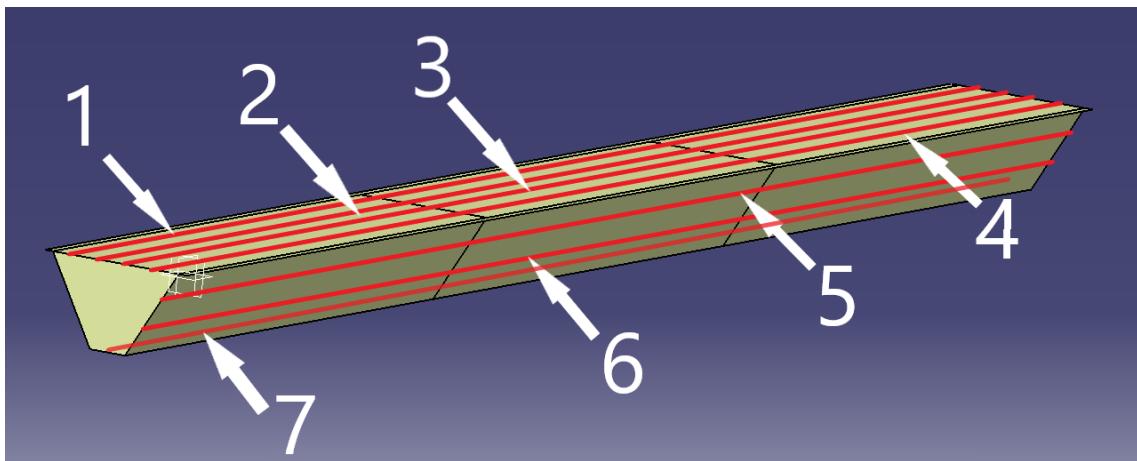


Figura 3.40: Tiras de sensores de los que se van a extraer las deformaciones

y no hay ninguna carga aplicada, el campo de deformaciones será nulo. Conforme la temperatura comience a bajar, el material se contraerá y se generarán deformaciones en la estructura. Por otro lado, si se mantiene la temperatura constante y se aplica la carga vertical, el campo de deformaciones también cambia.

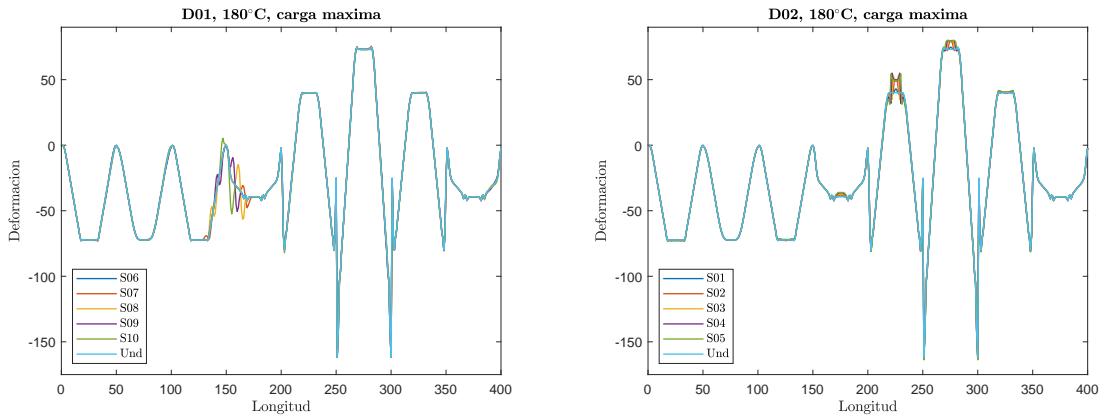
Las deformaciones que provoca la temperatura son equivalentes a las que se producen por una carga distribuida sobre la estructura. Esto quiere decir que las deformaciones producidas por la temperatura y la carga vertical son independientes y aditivas. Las deformaciones de la estructura tiene aplicada una carga de 10 kN a una temperatura de 20°C es igual a la suma de: las deformaciones de la estructura a 180°C (no hay deformaciones térmicas) y 10 kN de carga más la estructura a 20°C y sin carga aplicada.

Esto simplifica el número de simulaciones necesarias, que serán las siguientes:

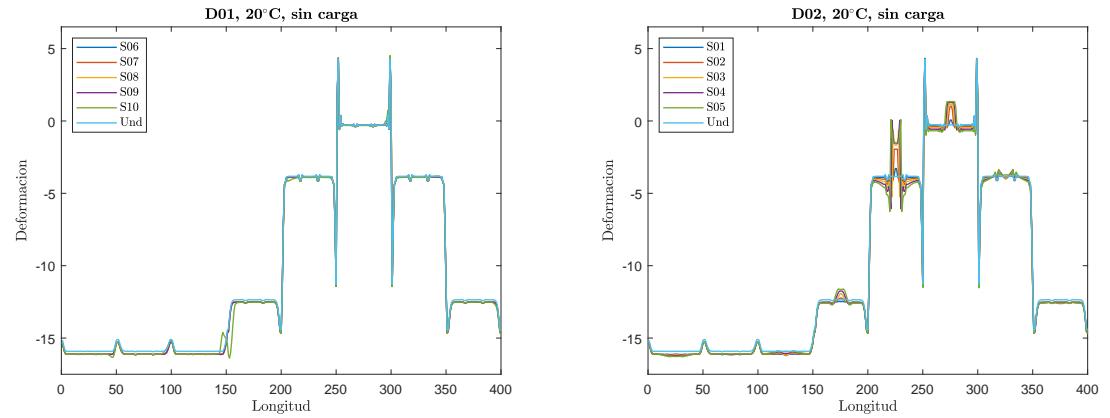
- Medida del campo de deformaciones para cada daño a 180°C con la carga máxima aplicada.
- Medida del campo de deformaciones para cada daño sin carga aplicada y a temperatura diferente de la de referencia.

Como las deformaciones tienen una relación lineal con la carga y la temperatura, se pueden multiplicar por un factor de escala y combinarse para conseguir todos los casos necesarios.

A continuación se va a representar en la Figura 3.41 los campos de deformaciones producidos por la temperatura y carga de forma independiente. A su vez, se representará en la Figura 3.42 la combinación de carga más temperatura para un daño en concreto.

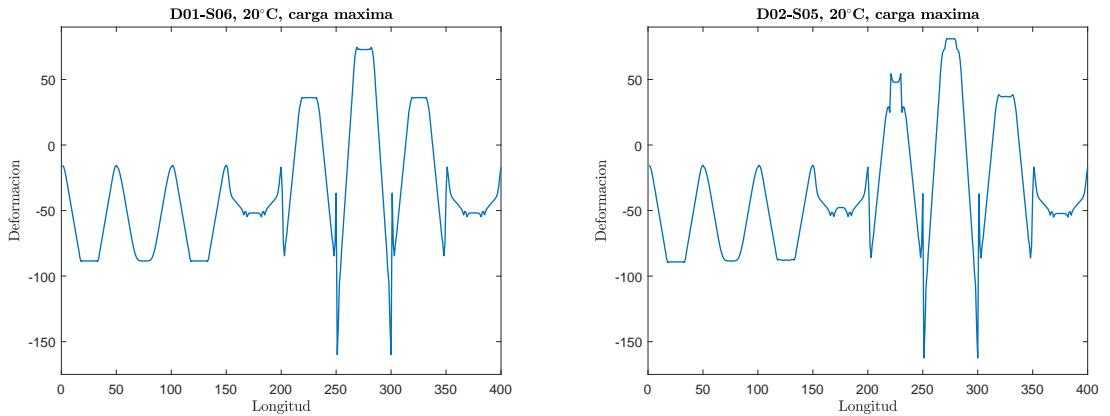


(a) Deformaciones producidas por el daño 1 con una carga vertical a la temperatura de referencia  
 (b) Deformaciones producidas por el daño 2 con una carga vertical a la temperatura de referencia



(c) Deformaciones producidas por el daño 1 a 20°C sin carga vertical  
 (d) Deformaciones producidas por el daño 1 a 20°C sin carga vertical

Figura 3.41: Campos de deformaciones producidos por la carga vertical y temperatura



(a) Deformaciones producidas por la combinación de temperatura y carga ee el daño 2  
 (b) Deformaciones producidas por la combinación de temperatura y carga ee el daño 2

Figura 3.42: Campos de deformaciones producidos por la carga vertical y temperatura

A estos campos de deformaciones se les va a añadir un ruido Gaussiano para simular el producido por los sensores de medida. Se ha seguido el mismo proceso utilizado en *Preprocesado de señales con sensores OBR*. Se puede ver el resultado final en el detalle de la Figura 3.43.

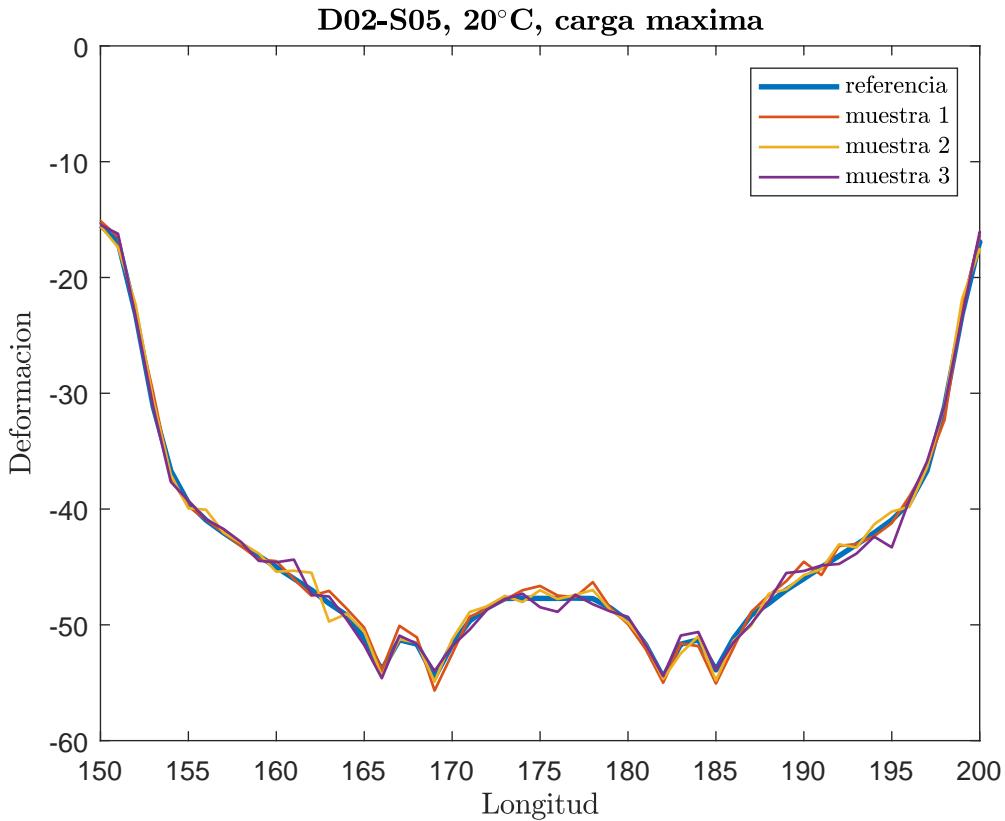


Figura 3.43: Detalle de la linea de sensor 4

### 3.3.4. Visualización del campo de deformaciones

Al igual que se hizo con los datos del proyecto INESASSE, también es conveniente visualizar cómo de parecidos son las deformaciones de cada carga, temperatura y tipo de daño. Con esto se entenderá mejor la clasificación de la red neuronal y se podrán extraer conclusiones posteriormente.

En este caso, los datos tienen dimensión 400 y el algoritmo que se va a utilizar continúa siendo **t-SNE**.

En las Figuras 3.46 y 3.45 se obtienen las mismas conclusiones que con los datos del proyecto INESASSE, se agrupan las cargas y daños en grupos claramente diferenciados.

En la Figura 3.47 se puede comprobar que las deformaciones provocadas por la temperatura se agrupan en grupos pequeños dentro de cada grupo grande de las cargas. A su vez, dentro de los grupos de cada temperatura incorpora cada clase de daño.

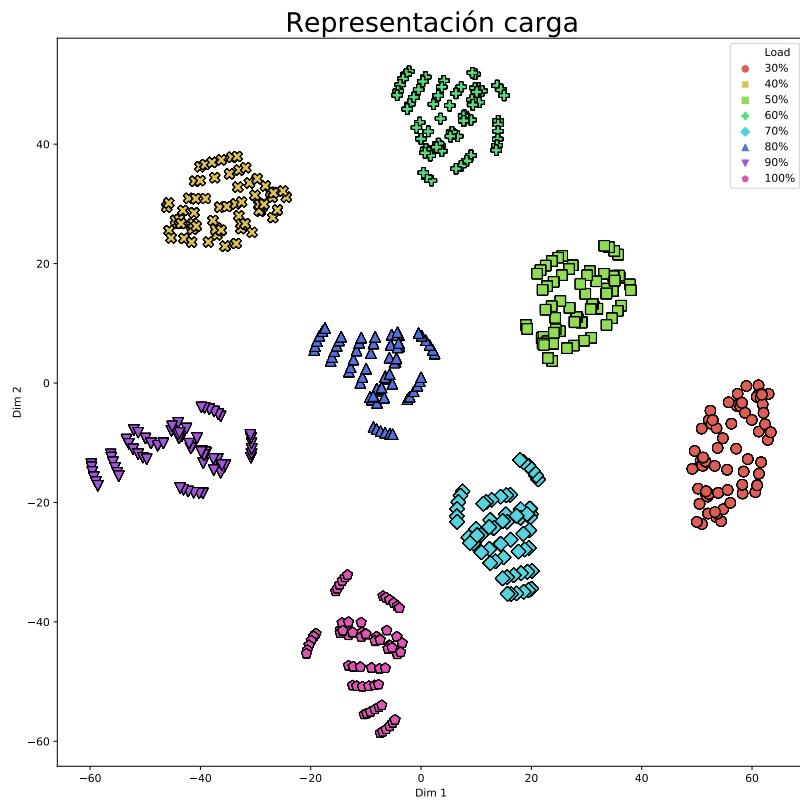


Figura 3.44: Visualización de los diferentes daños

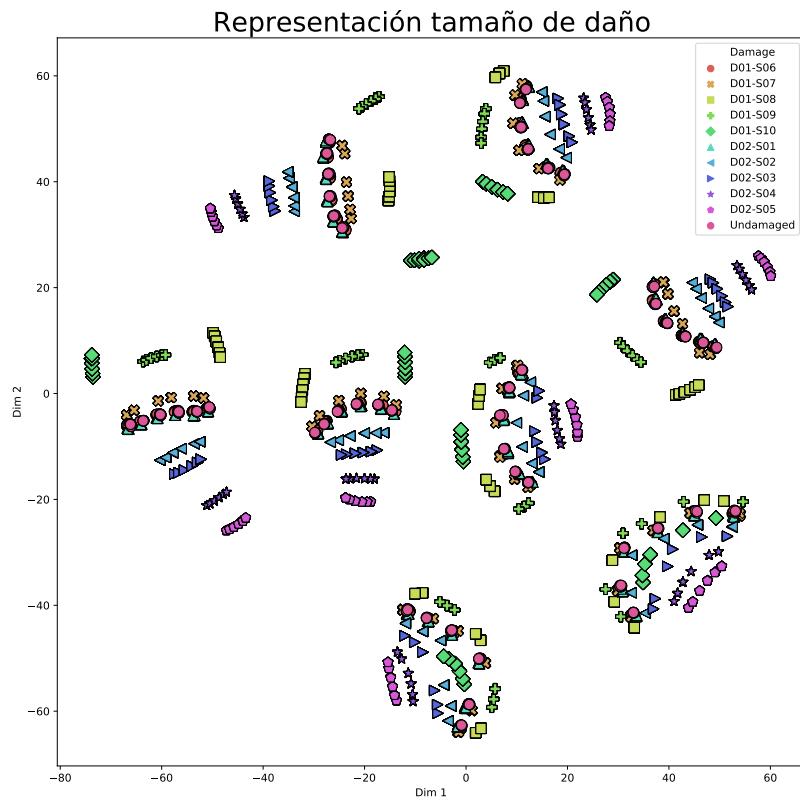


Figura 3.45: Visualización de los diferentes tamaños de daño

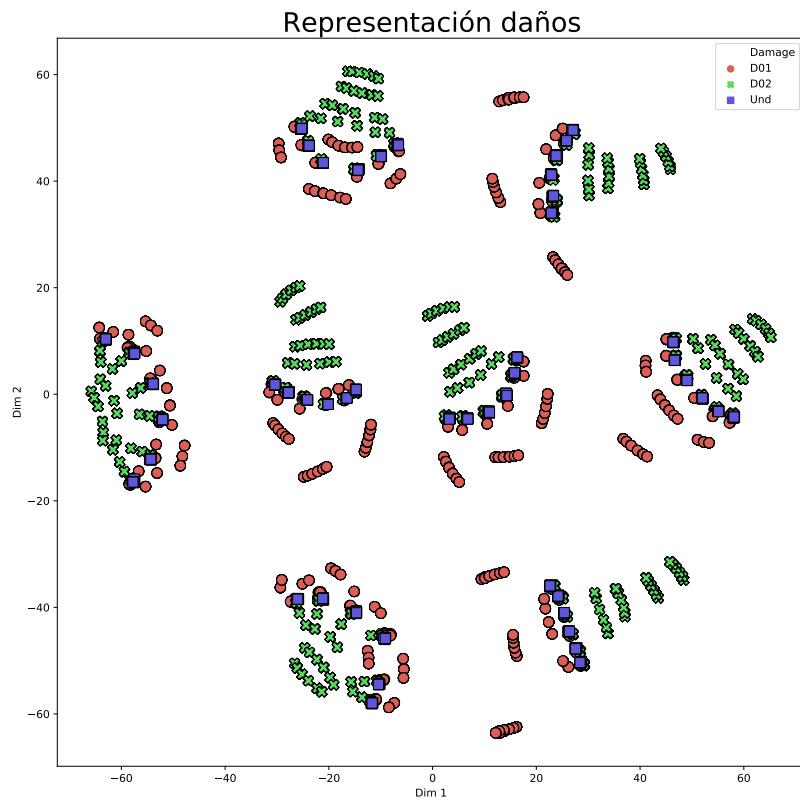


Figura 3.46: Visualización de las diferentes cagas

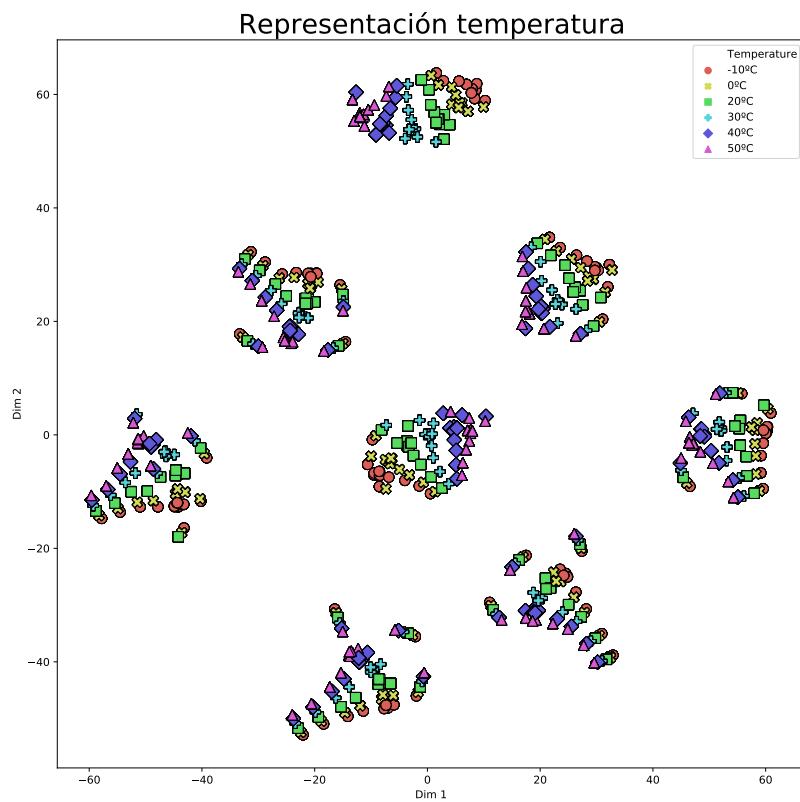
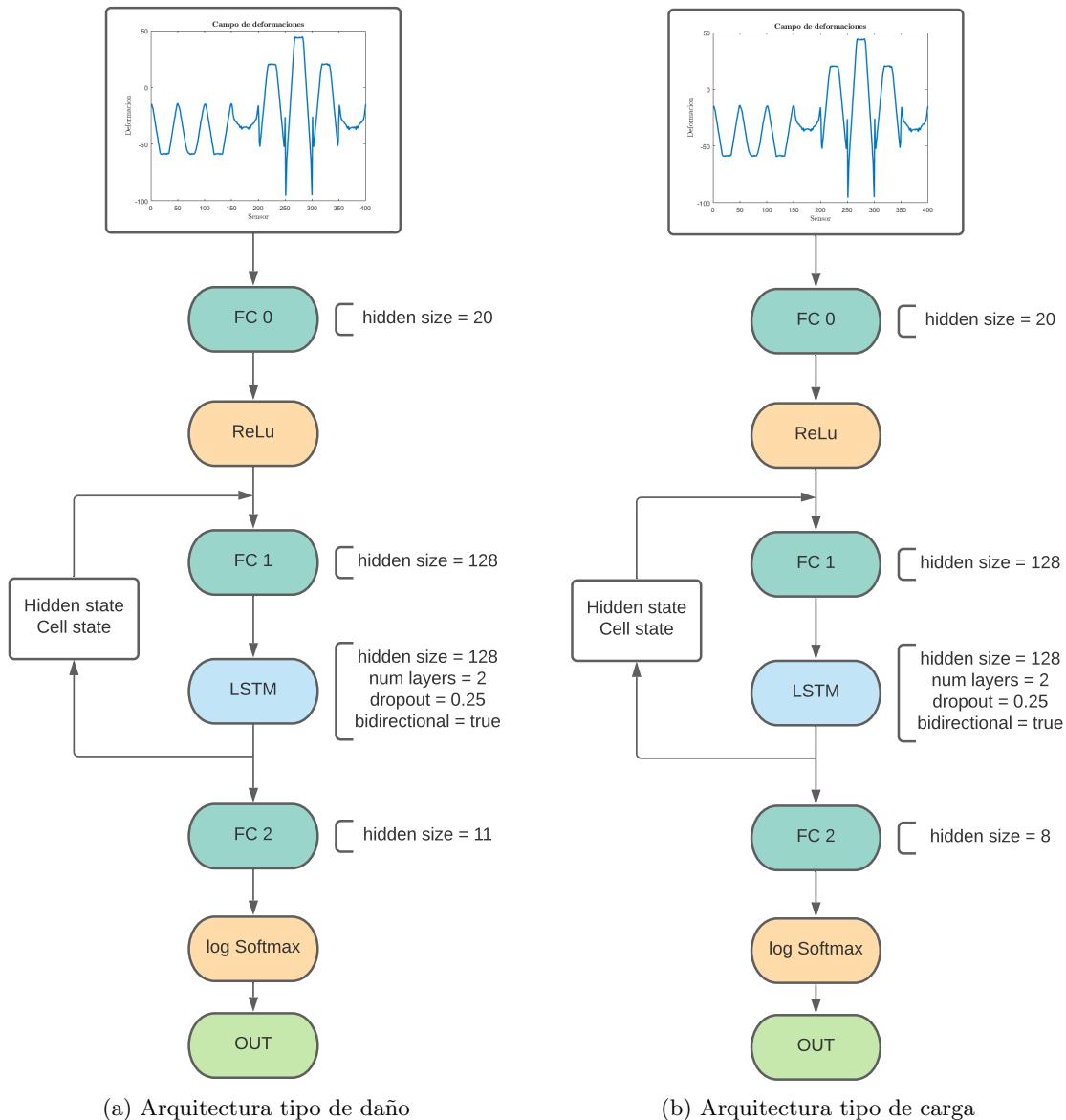


Figura 3.47: Visualización de las diferentes temperaturas

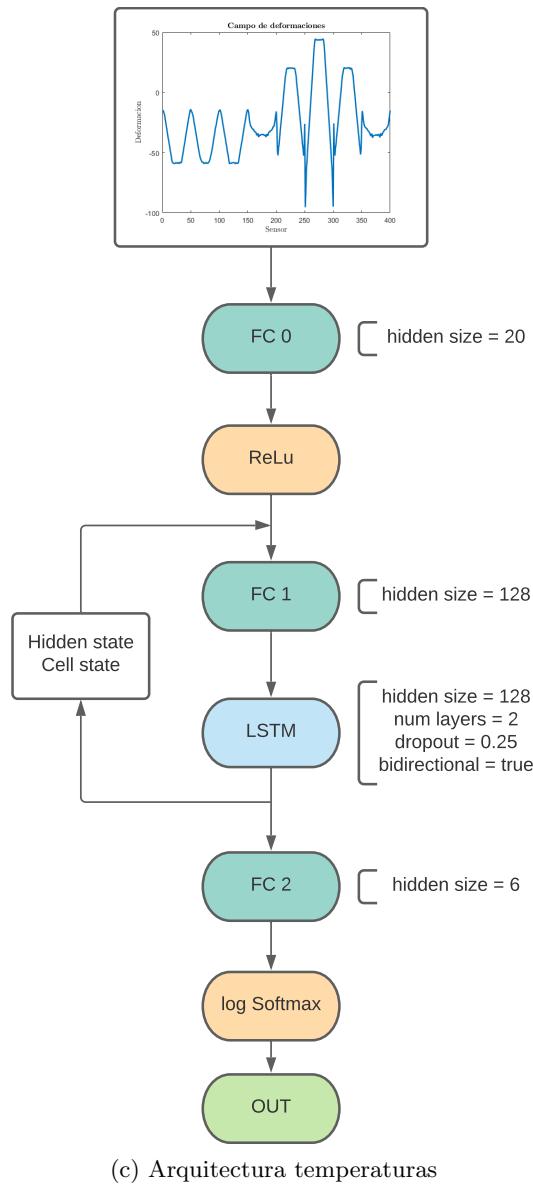
### 3.3.5. Arquitectura de la red

En la Figura 3.48 se pueden ver las diferentes arquitecturas usadas para las deformaciones extraídas de las simulaciones de la viga. El planteamiento ha sido igual que los sensores OBR, pero con una única capa FC para comprimir el campo de deformaciones hasta los 20 elegidos para alimentar al bucle.



(a) Arquitectura tipo de daño

(b) Arquitectura tipo de carga



(c) Arquitectura temperaturas

Figura 3.48: Arquitecturas utilizadas en las redes de los sensores en el FEM

### 3.3.6. Resultados de la clasificación

Al igual que se ha hecho anteriormente, el método elegido para representar el funcionamiento de las redes ha sido las Matrices de Confusión.

Se puede comprobar en las Figuras 3.49 y 3.50 que la red consigue una precisión elevada en la detección de las diferentes cargas a las que está sometida la estructura independientemente del tipo que sean.

Al igual que pasaban en los casos anteriores, en la Figura 3.51 y 3.45 se comprueba que los daños que no es capaz de clasificar bien la red están muy juntos en la representación con t-SNE.

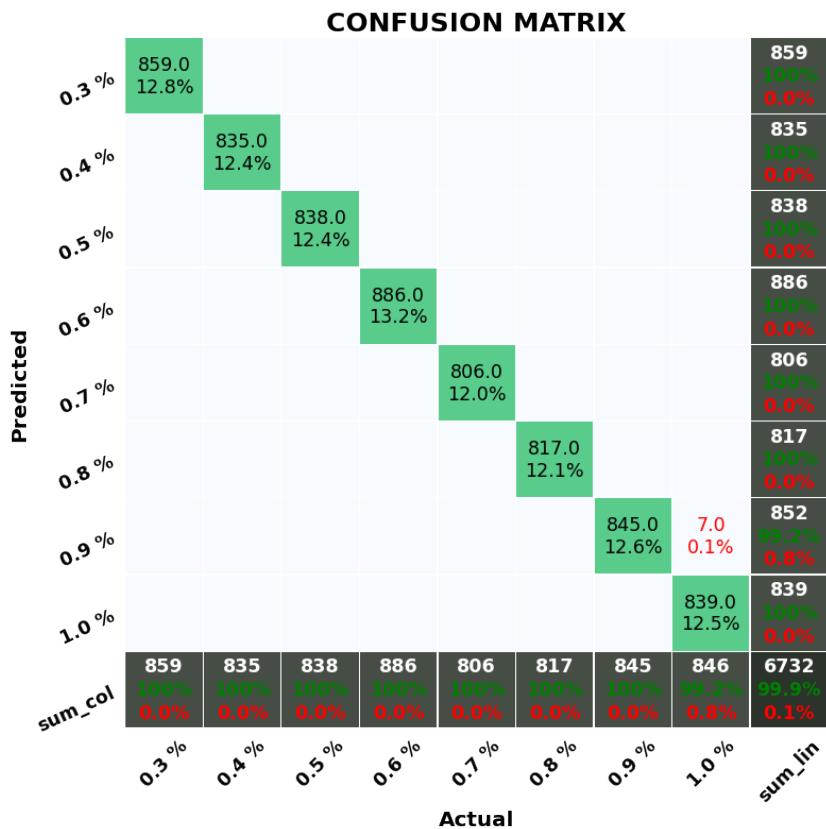


Figura 3.49: Matriz de confusión: nivel de carga

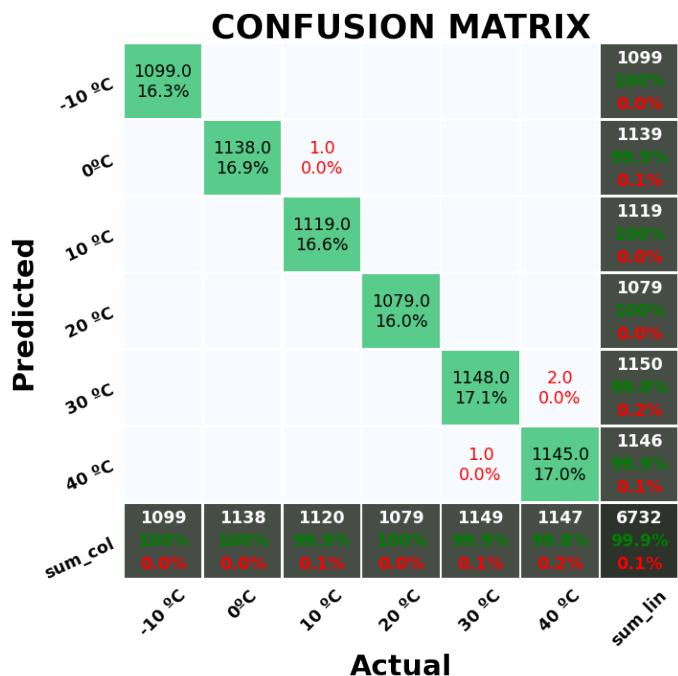


Figura 3.50: Matriz de confusión: temperatura distribuida

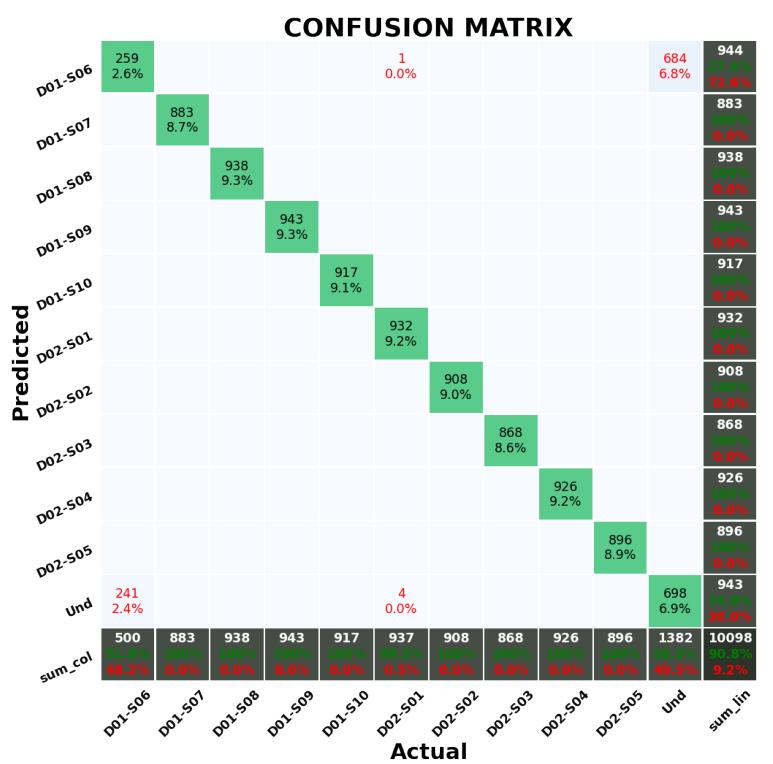


Figura 3.51: Matriz de confusión: tamaño de daño

## Capítulo 4

# Caracterización de impactos con Deep Learning

Como se ha mencionado en la introducción, el objetivo a cumplir de la segunda herramienta es la localización de impactos.

Los impactos se pueden considerar como cargas impulsivas y si se clasifican en función de la velocidad con la que el proyectil impacta, se pueden dividir en dos grupos: impactos de baja velocidad e impactos de alta velocidad.

En un impacto de baja velocidad las ondas de tensión que se generan alcanzan los bordes de la estructura y son reflejadas durante todo el proceso de impacto. Por esta razón, la respuesta de la estructura es global y está controlada por la combinación de geometría y condiciones de contorno. Por el contrario, en los impactos de alta velocidad la respuesta de la estructura es controlada por la propagación de las ondas de tensión en la dirección del espesor del laminado. En este caso, la respuesta frente a este tipo de impacto está gobernada por las propiedades del material en la zona cercana al punto de impacto y es, básicamente, independiente de la geometría y condiciones de contorno.

C. Aguilar basó su estudio, [2], en la utilización de diversos algoritmos de localización de daños e impactos de baja velocidad en estructuras de material compuesto.

En varios de los algoritmos utilizados, el cálculo de la velocidad de propagación de las ondas generadas por el impacto en el material era esencial. Por ejemplo, en la localización de impactos y daños por triangulación se utilizan los tiempos de llegada de las ondas a los distintos PZTs integrados en la estructura junto a las velocidades de propagación.

Los materiales compuestos tienen propiedades anisótropas, por lo que la velocidad de propagación de las ondas mecánicas no es igual en todas las direcciones. Esto hizo que en C. Aguilar se viera obligado a calcular el campo de velocidades de propagación sobre la estructura estudiada para realizar la triangulación.

Sin embargo, cuando la geometría de la estructura es compleja, el campo de velocidades ya no se puede calcular de forma sencilla y este tipo de algoritmos pasan a ser inviables. Por este motivo, cuando C. Aguilar estudió geometrías más complejas optó por utilizar algoritmos de Deep Learning que solo necesitan las medidas de los PZTs.

Este capítulo se va a centrar en la creación de una herramienta de Deep Learning para la localización de impactos en estructuras complejas de material compuesto. En la Figura 4.1 se puede ver un ejemplo de este tipo de estructuras.

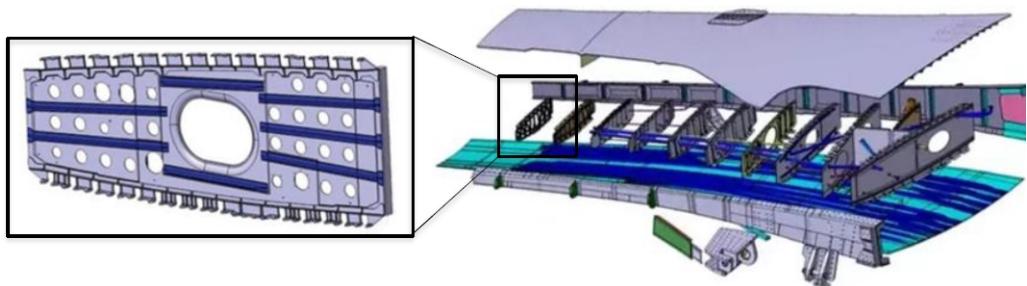


Figura 4.1: Costillas en una semiala del Airbus A380

El desarrollo de esta herramienta se ha basado en el trabajo y datos obtenidos por C. Aguilar en [2].

La estructura que se va a utilizar en este estudio es un cuarto de costilla del Airbus A380. En la Figura 4.2 se puede apreciar la complejidad de la estructura con múltiples vaciados y rigidizadores. También se ven los piezoelectrómicos encargados de recoger las ondas de Lamb provocadas por los impactos.

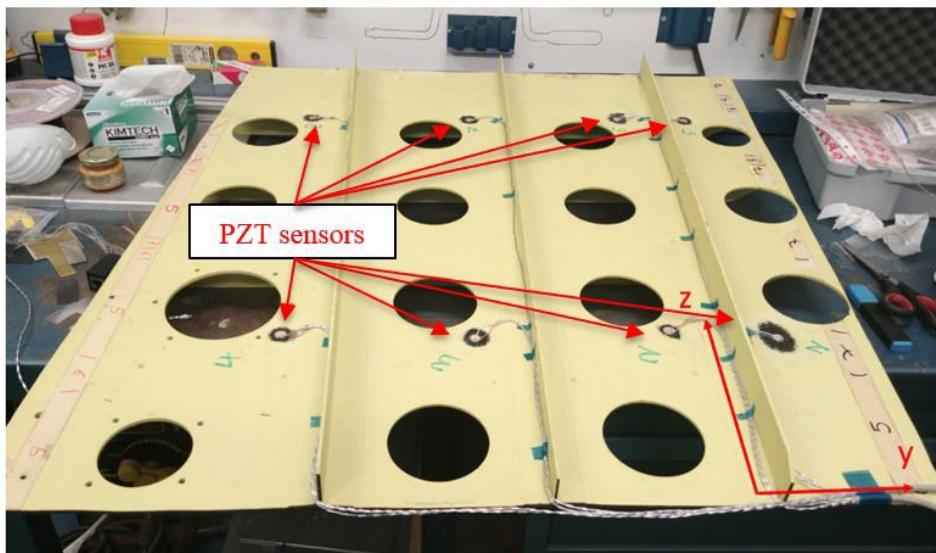


Figura 4.2: Estructura en estudio instrumentada

Como se ha visto anteriormente, las NN clasificadoras dan un resultado discreto, en forma de clases. Para que la red pueda clasificar, se ha discretizado la superficie de la costilla en diferentes celdas, cuyos índices serán considerados como clases. Esta discretización se puede ver en la Figura 4.3 junto con la posición de los PZT encargados de recoger los impactos.

Al igual que para las redes de detección de daño, es necesario un DS con una cantidad de muestras del orden de  $10^3$  por clase para obtener unos resultados fiables. Para esta clasificación se va a utilizar el mismo DS que C. Aguilar generó de forma manual y, posteriormente, utilizó para alimentar a su red. Sin embargo, este DS es demasiado escaso (20 muestras por clase) por lo que no se podrían utilizar resultados fiables.

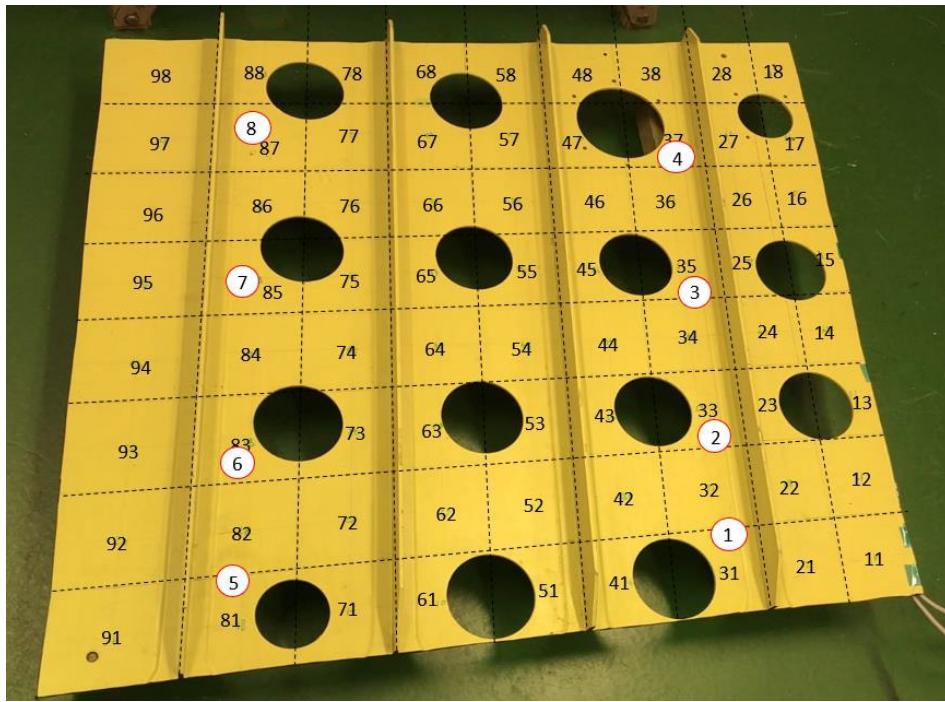


Figura 4.3: Discretización de la superficie de la costilla

La forma ideal de entrenar a una NN es mediante datos que recojan la mayor cantidad de variabilidad dentro de una clase. En el caso de la costilla se traduce en una gran cantidad de impactos a distintas alturas, distintas masas de objeto impactador y repartidos por toda la celda. De este modo se conseguirá que la NN aprenda de una forma completa la dinámica de los datos de entrada y realizará una clasificación de una forma completa y eficiente.

Sin embargo, el realizar una cantidad de impactos tan elevada de forma manual no es viable y su automatización pasa a ser esencial.

Como en este trabajo se parte de un DS reducido para poder aumentar el número de muestras se va a utilizar un tipo de NN llamado GAN (Generative Adversarial Network). Esta red se caracteriza por su capacidad de aprender las relaciones físicas entre las variables (impacto recogido por cada piezoelectrónico) y después ser capaces de generar nuevas muestras sintéticas muy similares a las reales.

Una vez se hayan generado los impactos sintéticos, se realizará una visualización con t-SNE para comprobar su similitud con los reales. Si los impactos sintéticos están agrupados juntos los reales, se utilizarán para alimentar a una nueva red que tendrá como objetivo clasificar un impacto en las celdas de la costilla.

## 4.1. Generative Adversarial Networks

Muchos algoritmos de Machine Learning tienen como entrada datos complejos (una imagen, por ejemplo) y generan una salida simple (la clase avión). Sin embargo, el objetivo de un modelo generativo es completamente opuesto. A partir de una entrada simple, como puede ser un vector de números aleatorios, generar como salida una imagen realista de un avión.

Las Generative Adversarial Networks (GANs) son un tipo de modelo generativo muy

efectivo, creado en el 2014 por Ian J. Goodfellow [4], que ha suscitado un gran interés en la comunidad de ML.

Un concepto importante de las GANs es que utiliza la aleatoriedad como instrumento creador. Esto hace que cuando introduzcas una entrada aleatoria nunca genere una imagen repetida, ya que hay muy pocas probabilidades de generar dos grupos de datos aleatorios iguales.

Pero igual de importante es que pensar en términos de probabilidades también nos ayuda a traducir el problema de la generación de imágenes en un marco matemático natural. Obviamente no se quieren elegir imágenes uniformemente al azar, ya que eso sólo produciría ruido. En su lugar, queremos que el sistema aprenda qué imágenes tienen probabilidades de ser caras, y cuáles no. Matemáticamente, esto implica modelar una distribución de probabilidad en las imágenes, es decir, una función que diga qué imágenes son probables de ser caras y cuáles no. Este tipo de problema se puede entender como modelar una función en un espacio de altas dimensiones y es exactamente el tipo de cosas para las que están hechas las redes neuronales.

La gran idea que define a una GAN es establecer este problema de modelización como una especie de concurso. De aquí viene la parte „adversaria” del nombre. La idea clave es construir no una, sino dos redes enfrentadas: un generador y un discriminador. El generador trata de crear salidas sintéticas aleatorias (imágenes de rostros), mientras que el discriminador trata de diferenciarlas de las salidas reales (una base de datos de famosos). El objetivo es que a medida que las dos redes se enfrenten, ambas mejoren cada vez más, con el resultado final de una red generadora que produzca salidas realistas.

En resumen: Las redes generadoras adversarias son redes neuronales que aprenden a elegir muestras de una distribución especial (la parte “generativa” del nombre), y lo hacen estableciendo una competencia (por lo tanto „adversaria”).

Esta idea de dos redes enfrentadas se puede ver en la Figura 4.4

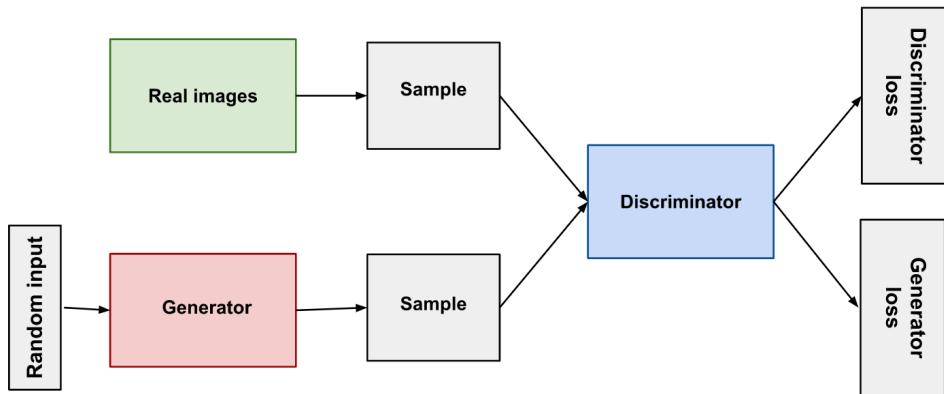


Figura 4.4: Representación simplificada de la arquitectura de una red GAN

Descripción extraída de [28]

En este estudio no se tiene como objetivo crear imágenes realistas, sino impactos que sean similares a los que se recogieron en [2]. Para entender mejor la forma de los datos que se van a utilizar se han representado varias muestras en la Figura 4.5. Cada muestra consta de las medidas de tensión (voltios) recogidas por los ocho piezoeléctricos en una ventana de tiempo de 0.0125 segundos. Este tipo de datos se puede clasificar como series

de datos temporales multivariable.

Si se presta atención a las subfiguras 4.5a y 4.5b puede apreciarse que, aun siendo un impacto en la misma celda, las mediciones son muy dispares. Recuperando la idea de que una parte esencial en las GANs es que se generan los nuevos impactos a partir de datos aleatorios, el que haya variación dentro de impactos en una misma celda facilita a la red el aprendizaje y, por lo tanto, el generar impactos más similares a los reales.

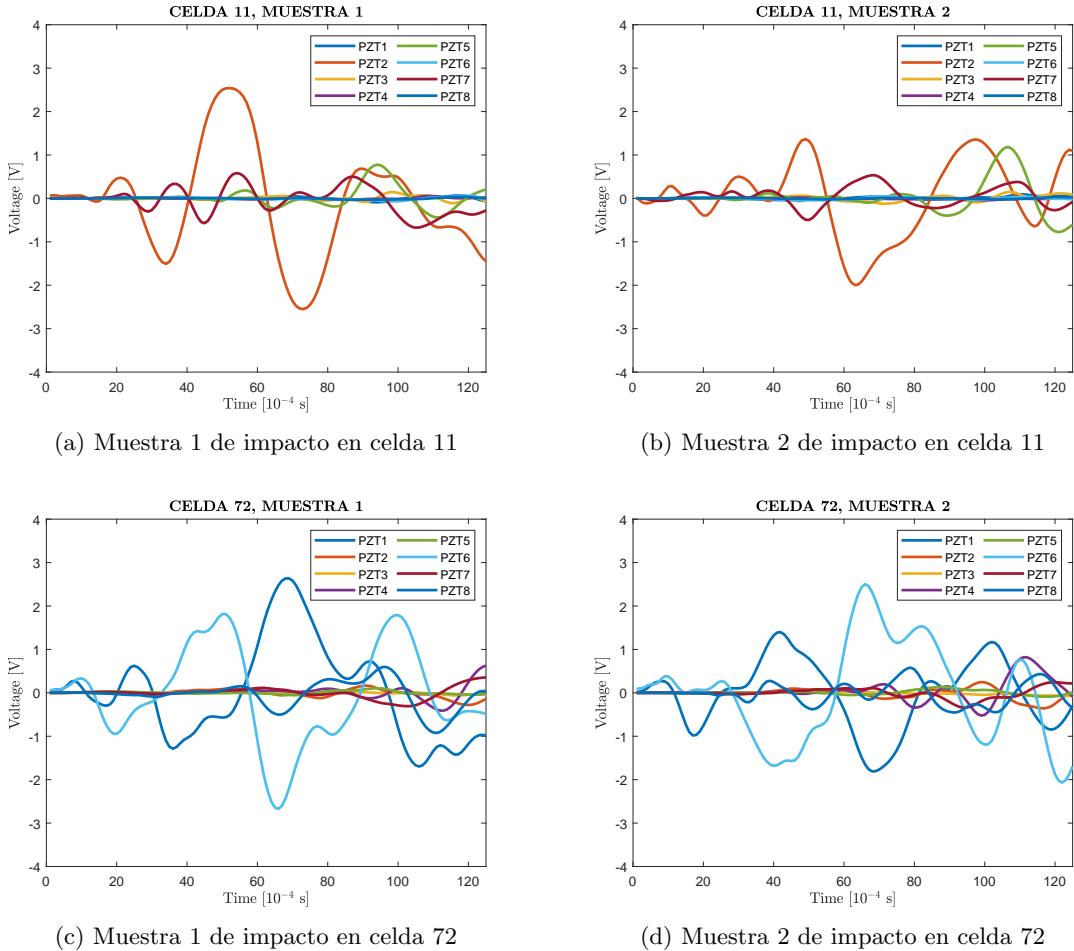


Figura 4.5: Diferentes ejemplos de impactos

#### 4.1.1. Modelo generativo de secuencias temporales

¿Qué es un buen modelo generativo para datos de series temporales? El entorno temporal plantea un desafío único para el modelado generativo. Un modelo no sólo tiene la tarea de capturar las distribuciones de los rasgos dentro de cada punto temporal, sino que también debe capturar la dinámica potencialmente compleja de esas variables a lo largo del tiempo. Específicamente, en el modelado de datos secuenciales multivariable  $x_{1:T} = (x_1; \dots; x_T)$ , también deseamos capturar con precisión la distribución condicional  $p(x_t|x_{1:t})$  de las transiciones temporales.

Desde el 2014, gran parte del trabajo se ha centrado en mejorar la dinámica temporal de los modelos autorregresivos para la predicción de secuencias. Éstos abordan principalmente el problema de los errores de composición durante el muestreo de múltiples etapas,

introduciendo diversas modificaciones en el momento de su formación para reflejar con mayor precisión las condiciones del tiempo de validación. Los modelos autorregresivos factorizan explícitamente la distribución de las secuencias en un producto de los condicionales  $\Pi_t p(x_t|x_{1:t_1})$ . Sin embargo, aunque es útil en el contexto de la predicción, este enfoque es fundamentalmente determinista y no es verdaderamente generativo en el sentido de que se pueden tomar muestras aleatorias de nuevas secuencias a partir de ellas sin condicionamientos externos. Por otra parte, otra línea de trabajo se ha centrado en la aplicación directa del marco de las GANs a los datos secuenciales, principalmente mediante la instanciación de redes recurrentes para las funciones de generador y discriminador. Aunque es sencillo, el objetivo adversario busca modelar  $p(x_{1:t_1})$  directamente, sin apalancar el previo autorregresivo. Es importante, simplemente sumar la función de costes (loss) estándar de la GAN sobre las secuencias de vectores puede no ser suficiente para asegurar que la dinámica de la red capte eficientemente las dependencias escalonadas presentes en los datos de entrenamiento.

Las Redes Generativas Adversas de Series Temporales (TimeGAN) proponen un novedoso mecanismo para unir ambos hilos de la investigación, dando lugar a un modelo generativo explícitamente entrenado para preservar la dinámica temporal. TimeGAN crea, un marco natural para generar datos de series temporales realistas en varios dominios.

#### 4.1.2. Arquitectura TimeGAN

Como ya se ha mencionado al inicio de este capítulo, las GANs están constituidas principalmente por un generador y un discriminador. Sin embargo, la arquitectura de TimeGAN varía ligeramente de esta generalización con el objetivo de poder retener la dinámica de los datos temporales.

TimeGAN está formada por cuatro componentes: una función de integración, función de recuperación, generador de secuencia y discriminador de secuencia.

La idea clave es que los componentes de autocodificación (los dos primeros) se entrenan conjuntamente con los componentes adversarios (los dos últimos), de modo que TimeGAN aprende simultáneamente a codificar características, generar representaciones e iterar a través del tiempo. La red de integración proporciona el espacio latente, la red adversaria opera dentro de este espacio y la dinámica latente de los datos reales y sintéticos se sincronizan a través de una pérdida supervisada (función de costes). Describimos cada uno a su vez.

- **Funciones de integración y recuperación**

Las funciones de integración y recuperación proporcionan asignaciones entre la característica y el espacio latente, lo que permite a la red adversaria aprender la dinámica temporal subyacente de los datos a través de representaciones de dimensiones más bajas.

Sean  $H_S$  y  $H_X$  los espacios vectoriales latentes correspondientes a los espacios de las características  $S$  y  $X$ . Entonces la función de integración  $e : S \times \Pi_t X \rightarrow H_S \times \Pi_t H_X$  lleva las características estáticas y temporales a su estado latente  $h_S, h_{1:t} = e(s, x_{1:t})$ .  $e$  se ha implementado mediante una RNN.

$$\mathbf{h}_S = e_S(s), \quad \mathbf{h}_t = e_X(\mathbf{h}_S, \mathbf{h}_{t-1}, \mathbf{x}_t) \quad (4.1)$$

donde  $e_S : S \rightarrow H_S$  es una red integradora para variables estáticas, y  $e_X : H_S \times H_X \times X \rightarrow H_X$  una red recurrente integradora para características temporales. En

la dirección opuesta, la función recuperadora  $r : H_S \times \Pi_t H_X \rightarrow S \times \Pi_t X$  coge las características latentes y las devuelve al espacio de características  $\tilde{s}, \tilde{x}_{1:T} = r(\mathbf{h}_S, \mathbf{h}_{1:T})$ . Aquí se implementa  $r$  usando una red *feedforward* en cada paso

$$\tilde{s} = r_S(\mathbf{h}_s), \quad \tilde{x}_t = r_X(\mathbf{h}_t) \quad (4.2)$$

donde  $r_S : H_S \rightarrow S$  y  $r_X : H_X \rightarrow X$  son redes de recuperación para las integraciones estáticas y temporales.

- **Generador y discriminador de secuencias**

En vez de generar datos sintéticos directamente en el espacio de características, el generador primero crea su salida en el espacio de integración. Sean  $Z_S$  y  $Z_X$  los espacios vectoriales sobre los que sabemos que las distribuciones están definidas y de donde se extraen los vectores aleatorios como entrada para generar  $H_S$  y  $H_X$ . Entonces la función de generación  $g : Z_S \times \Pi_t Z_X \rightarrow H_S \times \Pi_t H_X$  lleva una pareja de vectores aleatorios estáticos y temporales al espacio latente sintético  $\hat{h}_S, \hat{h}_{1:T} = g(\mathbf{z}_S, \mathbf{z}_{1:T})$ . Se implementa  $g$  con una RNN.

$$\hat{\mathbf{h}}_S = g_S(\mathbf{z}_S), \quad \hat{\mathbf{h}}_t = g_X(\hat{\mathbf{h}}_S, \hat{\mathbf{h}}_{t-1}, \mathbf{z}_t) \quad (4.3)$$

donde  $g_S : Z_S \rightarrow H_S$  es una red generadora de características estáticas y  $g_X : H_S \times H_X \times Z_X \rightarrow H_X$  es un generador recurrente para características temporales. Vectores aleatorios  $\mathbf{z}_S$  pueden ser muestrados desde una distribución cualquiera y  $\mathbf{z}_t$  sigue un proceso estocástico. Aquí se usa una distribución y un proceso Winer respectivamente. Finalmente el discriminador también opera en el espacio integrado. La función discriminatoria  $d : H_S \times \Pi_t H_X \rightarrow [0, 1] \times \Pi_t [0, 1]$  recibe las características temporales y estáticas, devolviendo la clasificación  $\tilde{y}_S, \tilde{y}_{1:T} = d(\tilde{\mathbf{h}}_S, \tilde{\mathbf{h}}_{1:T})$ . La notación  $\tilde{\mathbf{h}}$  refiere a datos reales ( $h$ ) como sintéticos ( $\hat{h}$ ) en el espacio vectorial de integración, al igual que  $\tilde{\mathbf{y}}$  indica clasificación real ( $y$ ) como sintética ( $\hat{y}$ ). Aquí se implementa  $d$  a partir de una RNN bidimensional con una capa *feedforward* en la salida

$$\tilde{\mathbf{y}}_S = d_S(\tilde{\mathbf{h}}_S), \quad \tilde{\mathbf{y}}_t = d_X(\tilde{\mathbf{u}}_t, \tilde{\mathbf{u}}_t) \quad (4.4)$$

donde  $\tilde{\mathbf{u}}_t = \vec{c}_X(\tilde{\mathbf{h}}_S, \tilde{\mathbf{h}}_t, \tilde{\mathbf{u}}_{t-1})$  y  $\tilde{\mathbf{u}}_t = \vec{c}_X(\tilde{\mathbf{h}}_S, \tilde{\mathbf{h}}_t, \tilde{\mathbf{u}}_{t+1})$  respectivamente denotan las secuencias hacia adelante y atrás de los *hidden states*,  $\vec{c}_X$  y  $\tilde{c}_X$  son funciones recurrentes y  $d_S, d_X$  son las funciones de clasificación de la última capa.

El diagrama de bloques del funcionamiento de TimeGAN se puede ver en la Figura 4.6

Todo lo comentado sobre TimeGAN en este trabajo ha sido extraído de [30]. Aquí se puede encontrar tanto el artículo que lo describe como el código necesario para la implementación de esta red.

#### 4.1.3. Resultados de la generación de impactos con TimeGAN

Antes de comenzar con el proceso de generación de impactos es necesario definir una serie de parámetros de arquitectura y del proceso de entrenamiento.

- Tipo de capa recurrente: GRU

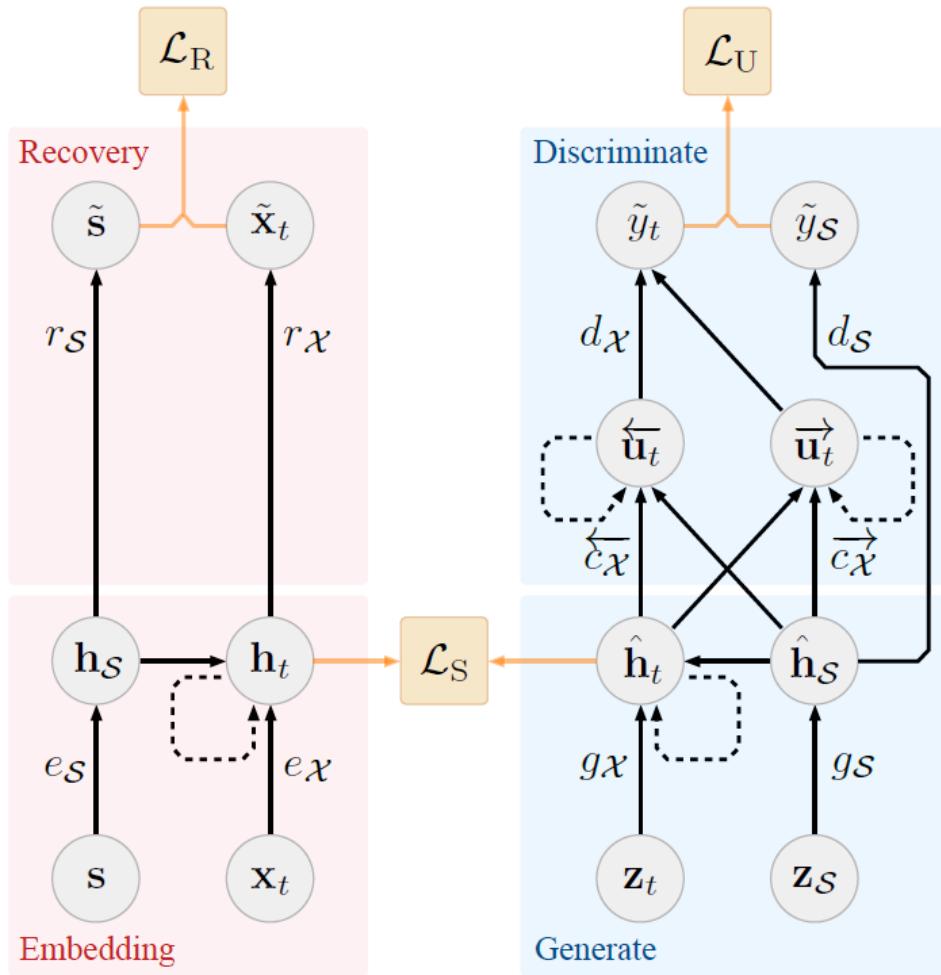


Figura 4.6: Diagrama de bloques del funcionamiento de TimeGAN

- Número de neuronas en las capas ocultas: 64
- Número de épocas en el entrenamiento: 400

Con los parámetros ya definidos se puede comenzar el proceso. Es importante tener en cuenta que se van a generar los impactos de forma individual e independiente. Esto quiere decir que se introducirán a la red todos los impactos reales de, por ejemplo, la celda 11 y una vez que se haya entrenado la TimeGAN se generarán 1500 impactos sintéticos. Después se borrará todo el aprendizaje que la red haya generado y se introducirán los impactos de la siguiente celda y volverá a repetirse el ciclo.

En la Figura 4.7 se puede ver varios impactos sintéticos que coinciden con las celdas elegidas en la Figura 4.5. Si se comparan visualmente se podrían confundir con un impacto real sin ninguna duda.

Pero para tener un nivel alto de confianza en que los nuevos datos que se han generado sintéticamente son ciertamente equivalentes a los impactos reales se ha decidido utilizar el algoritmo t-SNE de visualización.

En el capítulo 3, este algoritmo demostró que los daños similares se agrupaban en grupos o *clusters* independientes, por lo tanto, para poder considerar que los impactos

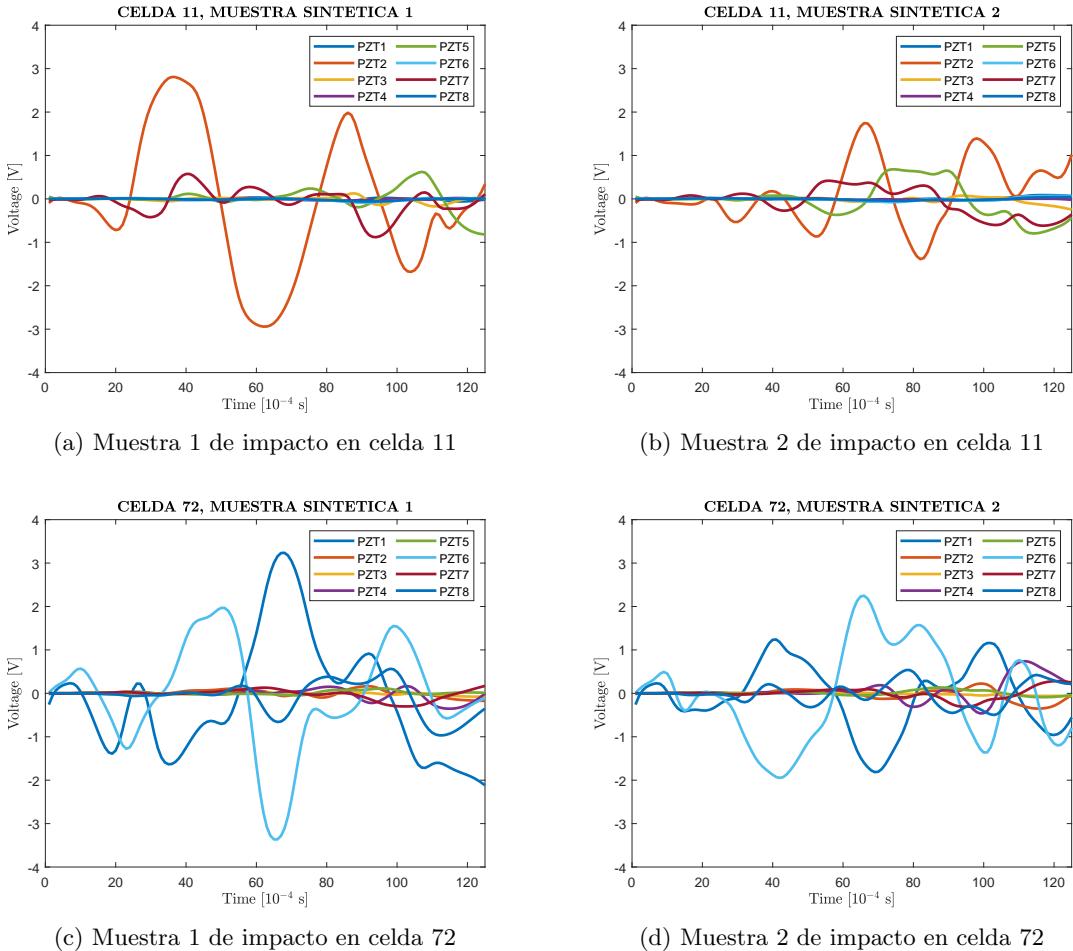


Figura 4.7: Diferentes ejemplos de impactos generados sintéticamente con TimeGAN

reales y generados son similares dentro de una celda y diferentes al resto de celdas lo que se espera al ejecutar el algoritmo es lo siguiente:

- cada celda esté en un cluster independiente
- los impactos reales y sintéticos de una misma celda estén distribuidos de forma uniforme en el cluster correspondiente

De la Figura 4.8 a 4.16 se pueden ver la representación de los impactos en las diferentes celdas (t-SNE). Al haber un total de 72 celdas, se ha decidido no realizar la visualización de todas ellas ya que el objetivo es tener una idea general de la similitud de los impactos reales y sintéticos.

En la Figura 4.8 se puede ver con claridad que los puntos correspondientes a la celda 34 están distribuidos entre dos agrupaciones pequeñas de los puntos 134. Esto quiere decir que los impactos sintéticos son similares a los reales y, a su vez, diferentes al resto de impactos.

Con este resultado se puede considerar que TimeGAN ha generado unos impactos similares a los reales y se pueden utilizar para alimentar a una red clasificadora.

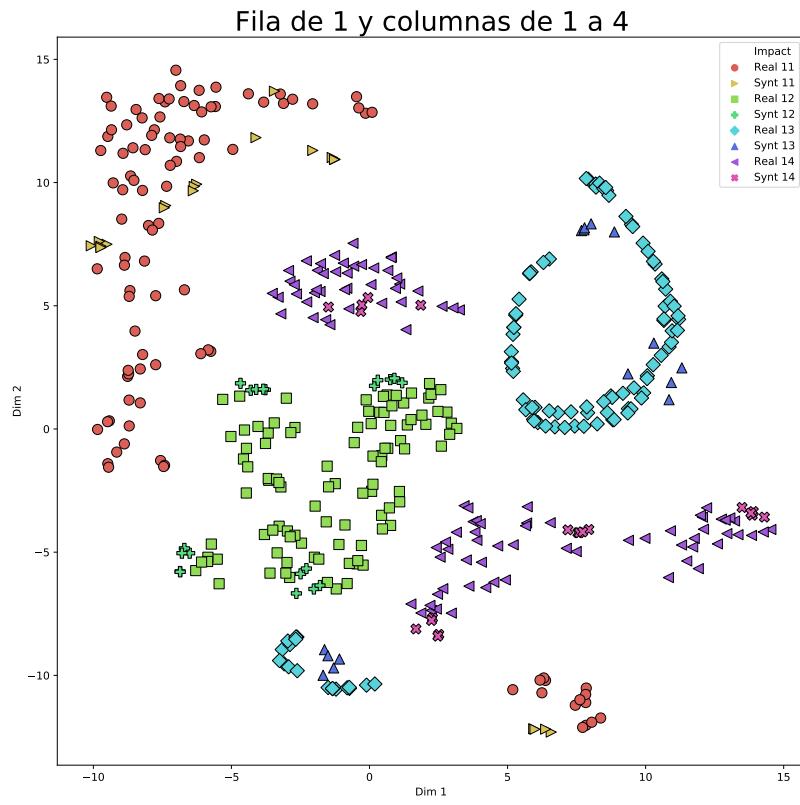


Figura 4.8: t-SNE de las celdas 11, 12, 13, 14

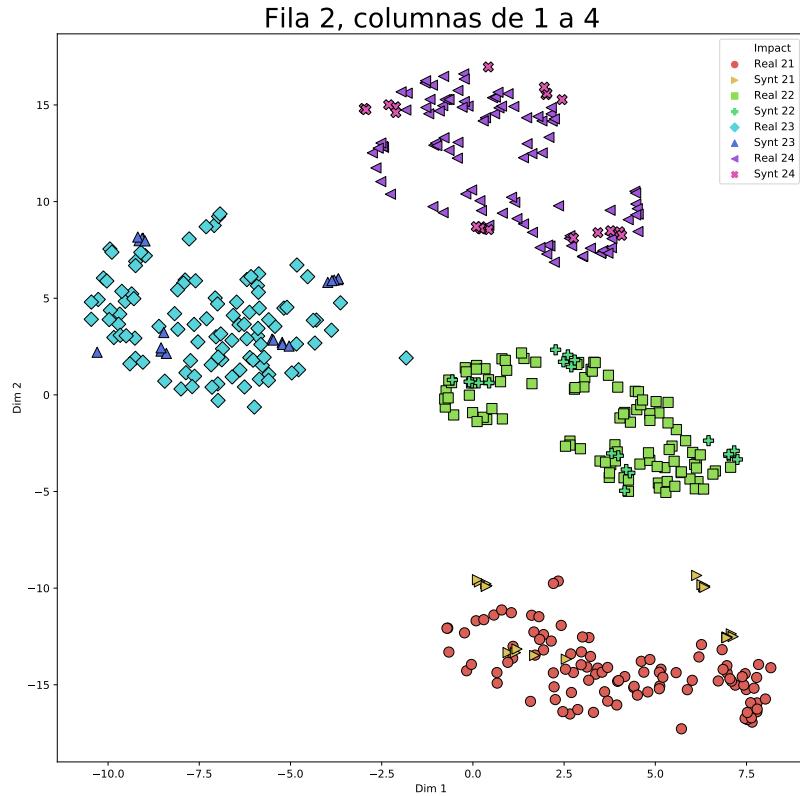


Figura 4.9: t-SNE de las celdas 21, 22, 23, 24

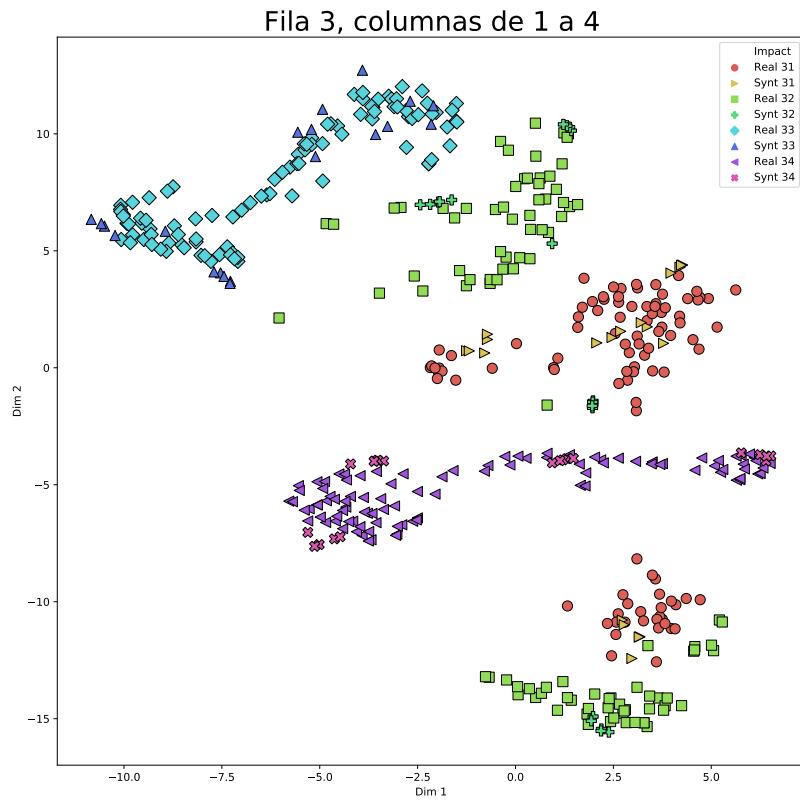


Figura 4.10: t-SNE de las celdas 31, 32, 33, 34

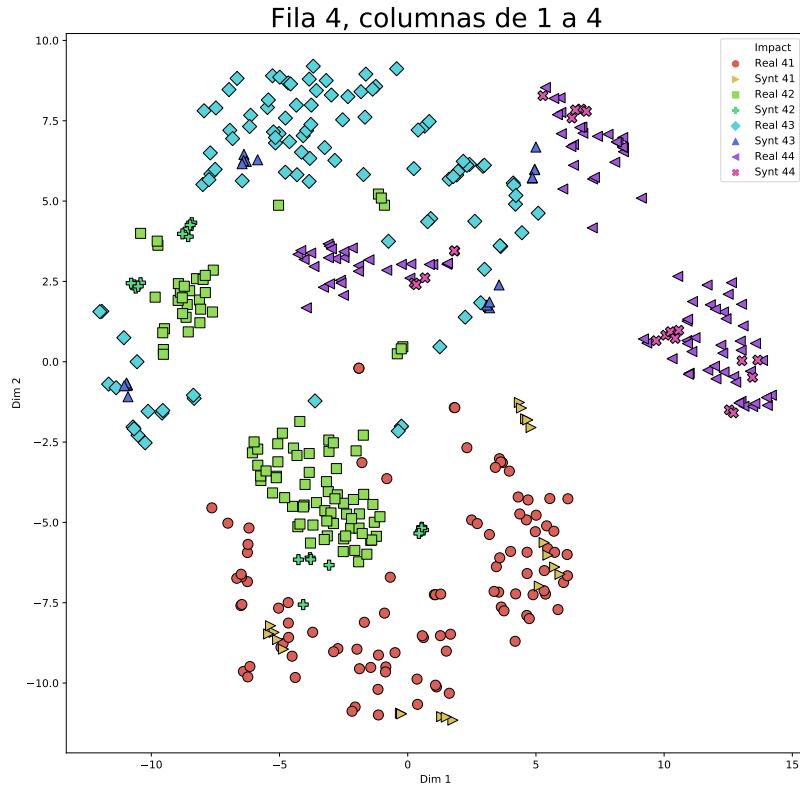


Figura 4.11: t-SNE de las celdas 41, 42, 43, 44

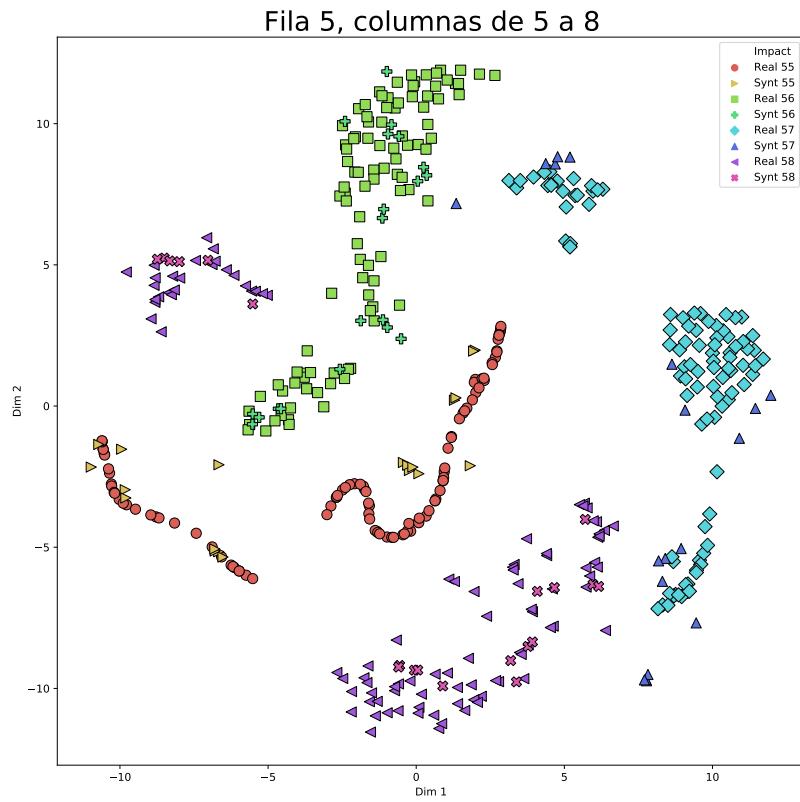


Figura 4.12: t-SNE de las celdas 55, 56, 57, 58

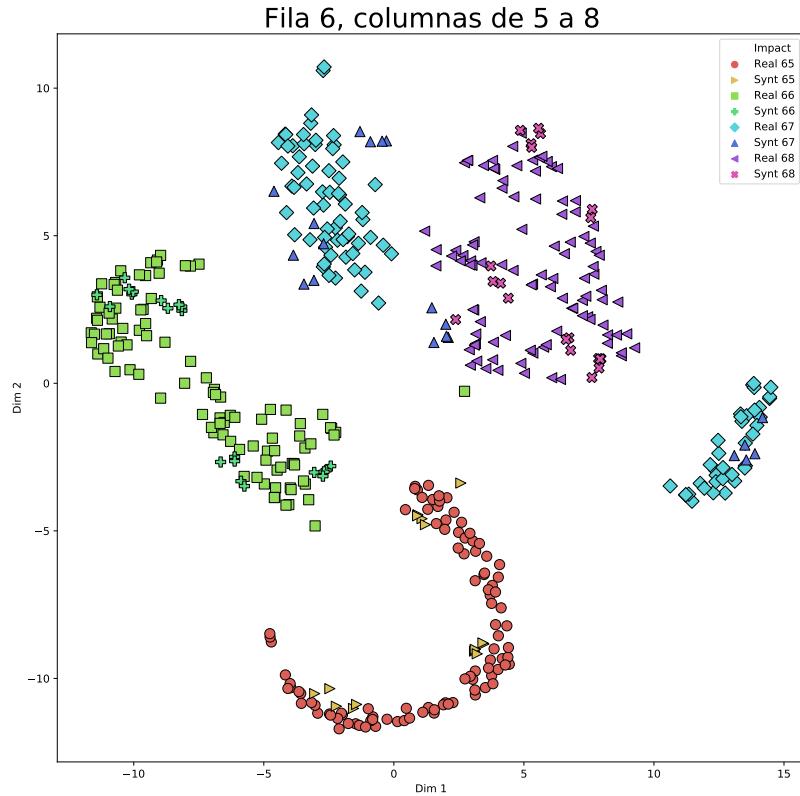


Figura 4.13: t-SNE de las celdas 65, 66, 67, 68

Fila 7, columnas de 5 a 8

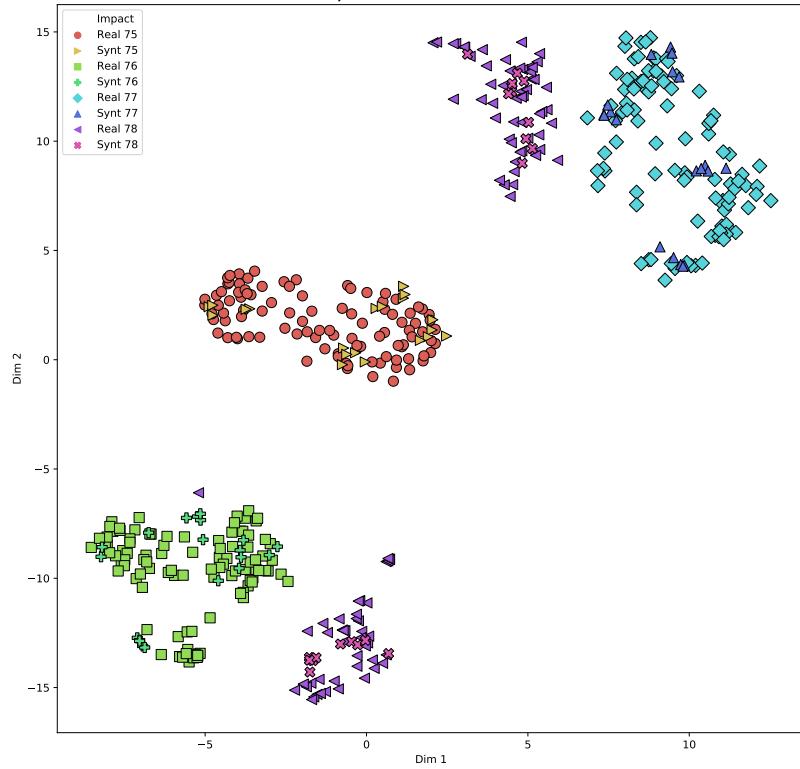


Figura 4.14: t-SNE de las celdas 75, 76, 77, 78

Fila 8, columnas de 5 a 8

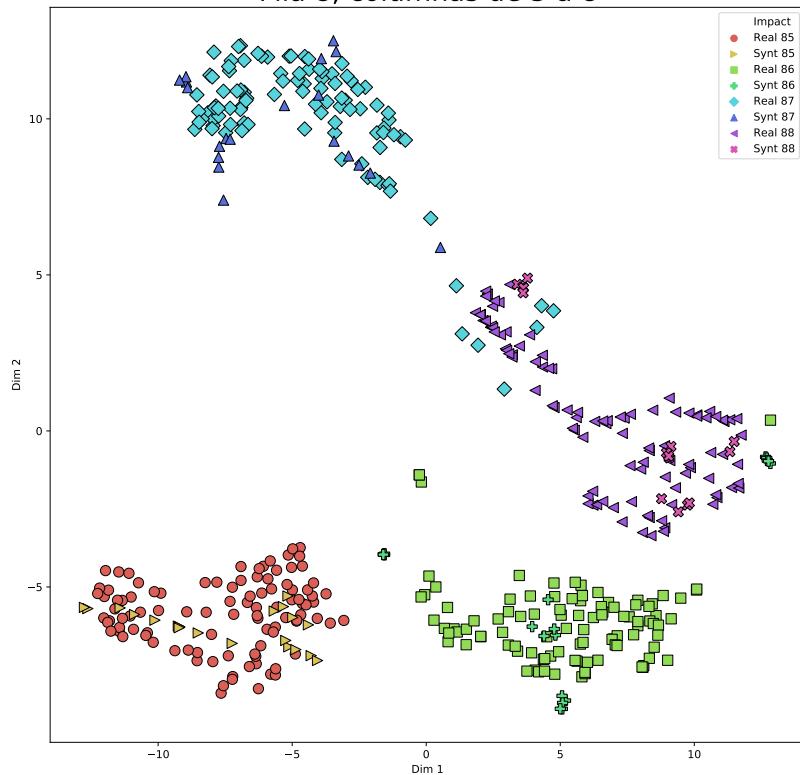


Figura 4.15: t-SNE de las celdas 85, 86, 87, 88

Fila 9, columnas de 5 a 8

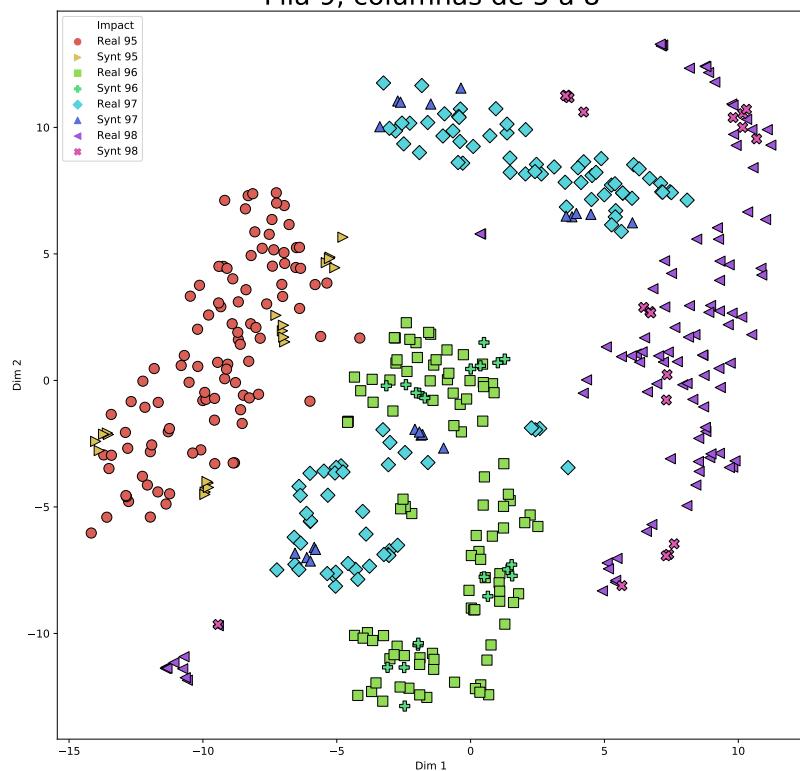


Figura 4.16: t-SNE de las celdas 95, 96, 97, 98

## 4.2. Implementación del localizador de impactos

### 4.2.1. Arquitectura del clasificador

En este caso, se tiene una única arquitectura (Figura 4.17) idéntica visualmente a la usada en el modelo FEM.

Sin embargo, a diferencia de las redes que se alimentaban con el campo de deformaciones, esta red procesa de forma simultánea los 8 impactos, en vez de trabajar con los datos como si fueran vectores, lo hace de forma matricial. En cada ejecución del bucle, a la red LSTM le va a entrar un vector de 8x1 (8 sensores en cada incremento de tiempo).

De esta forma se consigue que la red aprenda las dependencias temporales del impacto de con todos los sensores a la vez.

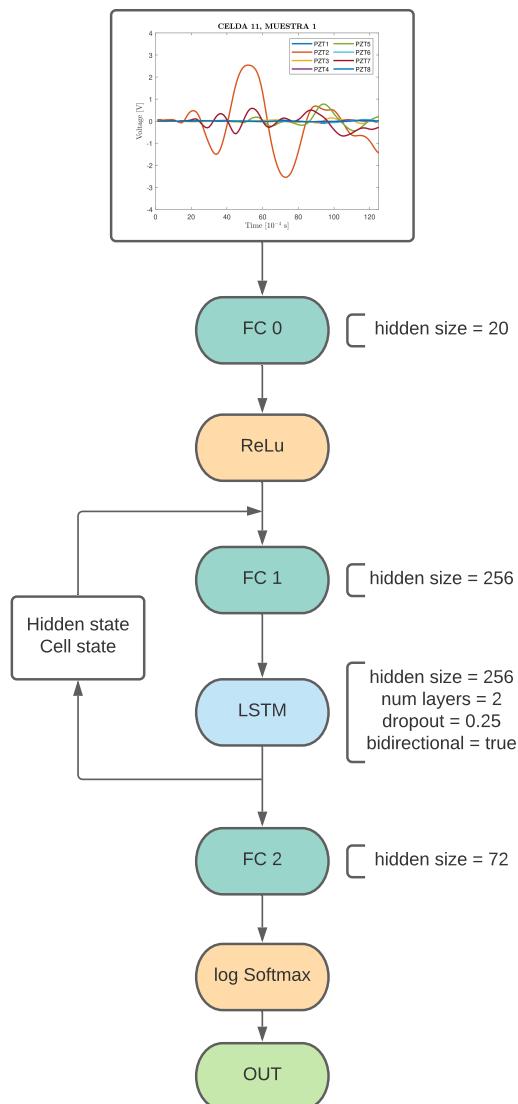


Figura 4.17: Arquitectura utilizada para el localizador de impactos

Falta preparar el modelo de arquitectura para que se vea bien.

#### **4.2.2. Resultados**

Como la cantidad de impactos reales que se tienen es muy reducida, se ha decidido entrenar al clasificador únicamente con los impactos generados sintéticamente. Esto permite comprobar también el funcionamiento de un clasificador cuando los datos de entrenamiento y validación tienen cierta dispersión.

En las Figuras 4.18 y 4.19 se pueden ver los resultados obtenidos con la red entrenada únicamente con los datos generados de forma sintética. Con esta red se ha conseguido una precisión de 99,38 % de precisión.

Por otro lado, en las Figuras 4.20 y 4.21 se ha validado la red entrenada con los impactos sintéticos introduciendo únicamente los impactos reales y ésta red ha conseguido una precisión de 88,29 %.

Se puede comparar ambos resultados de las redes anteriores para resaltar el que las redes funcionan mejor si se validan con el mismo grupo de datos con el que se ha entrenado. Por otra parte, también sirve para validar la generación de nuevas muestras con TimeGAN ya que se consigue una precisión elevada usando para entrenar los sintéticos.

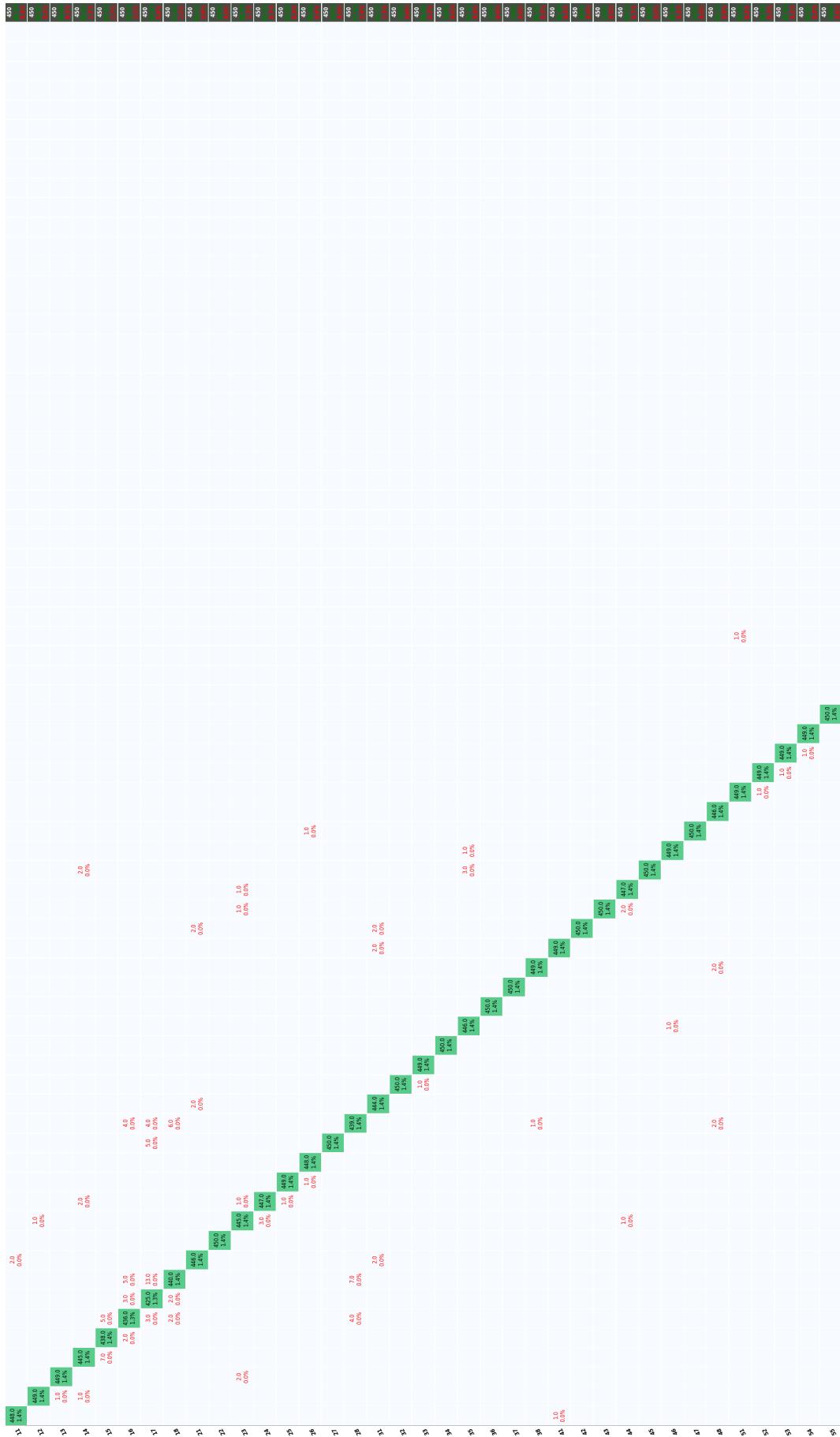


Figura 4.18: Matriz de confusión de impactos sintéticos 1

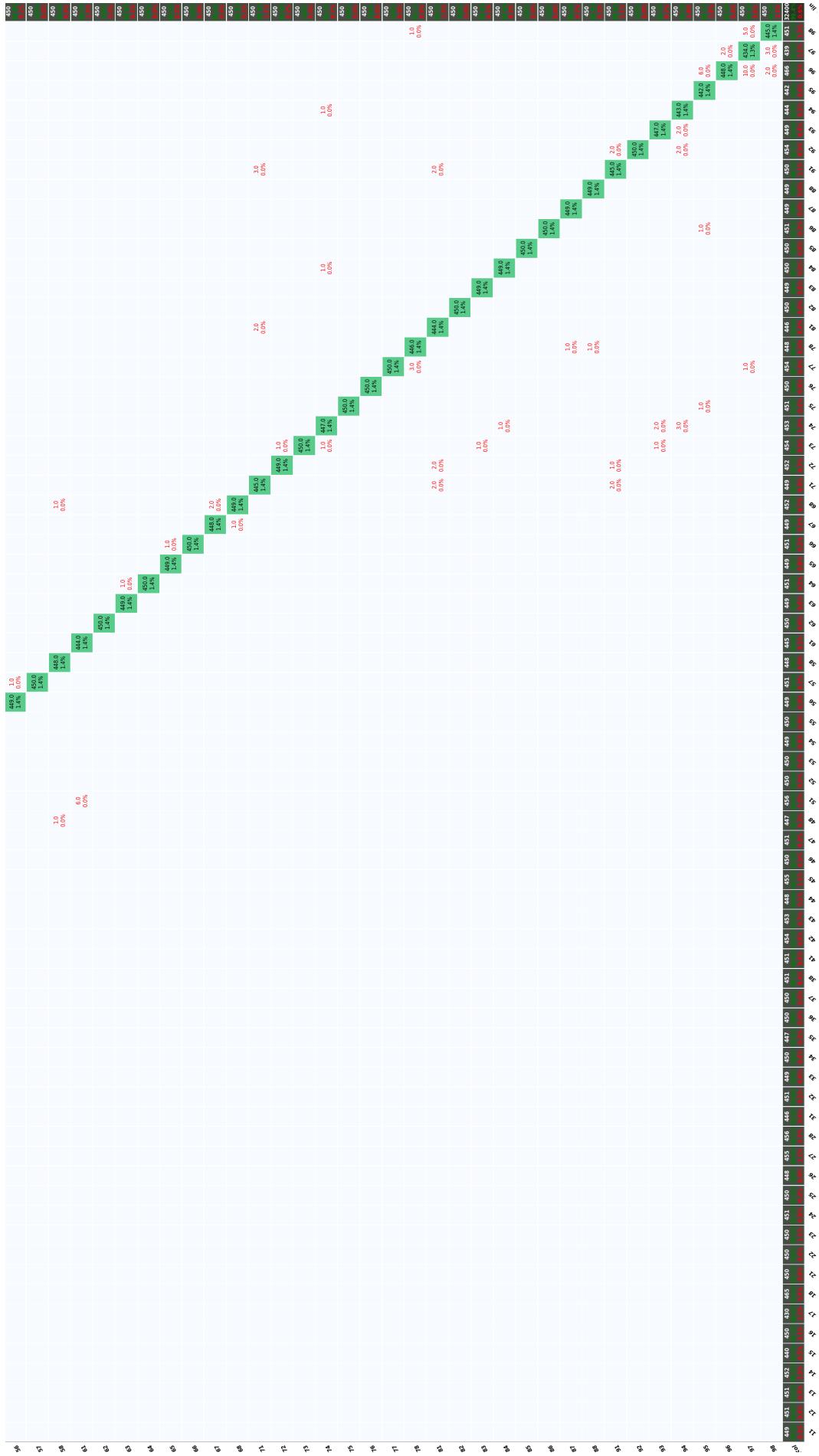


Figura 4.19: Matriz de confusión de impactos sintéticos 2

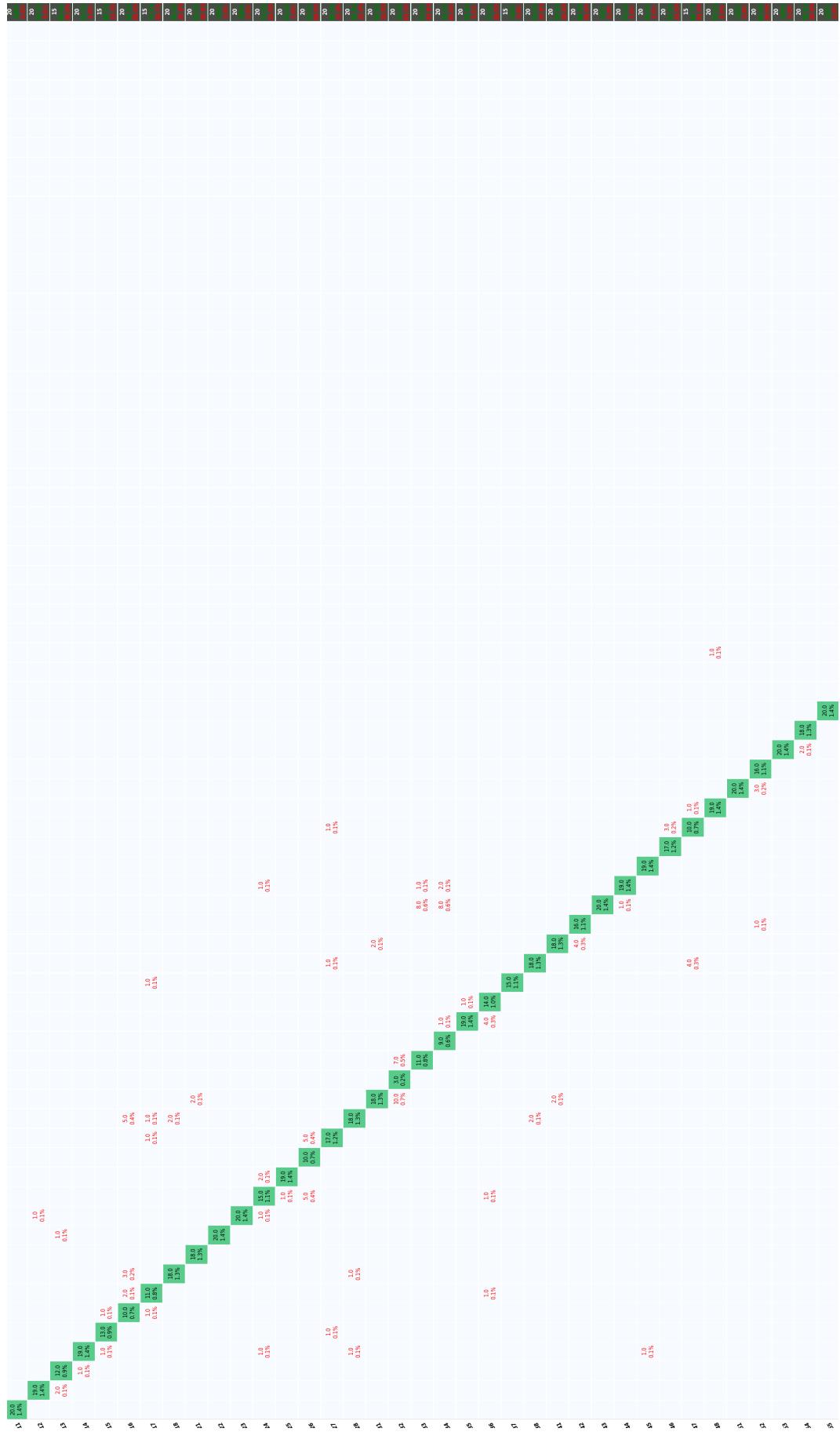


Figura 4.20: Matriz de confusión de impactos reales 1

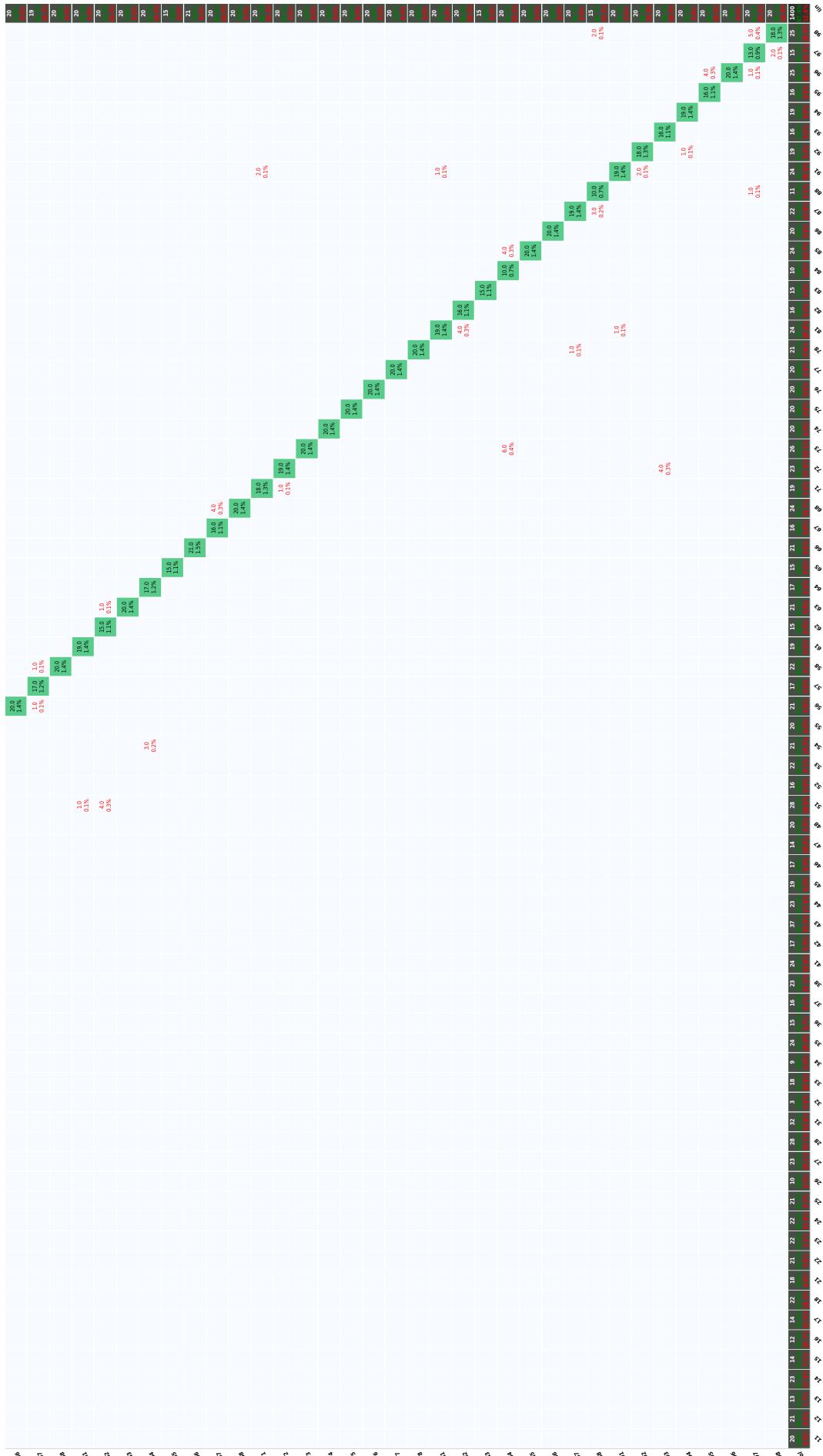


Figura 4.21: Matriz de confusión de impactos sintéticos 2

## Capítulo 5

# Conclusiones y trabajos futuros

Una vez concluido el desarrollo del proyecto se procede a valorar los resultados obtenidos y se introducirán posteriormente las posibles líneas de estudio futuras.

El estudio de la tecnología SHM es una tarea compleja que engloba varios niveles de ingeniería. Como el tiempo del que se ha dispuesto para realizar este trabajo ha sido limitado, se marcó como objetivo principal la localización de daños en diferentes estructuras complejas fabricadas en material compuesto.

### 5.1. Conclusiones

Las primeras conclusiones se sacan de los resultados obtenidos en la sección *Aplicación de DL a la detección de daños mediante sensores de deformaciones* utilizando los datos obtenidos durante los ensayos del proyecto INESASSE.

- La utilización de distribuciones de ruido gaussiano es un método eficaz para aumentar el número de muestras de un DS.
- Utilizando el filtro Savitzky-Golay se consigue eliminar el ruido de los sensores OBR de forma satisfactoria.
- El algoritmo de visualización t-SNE ha mostrado un gran potencial a la hora de encontrar relaciones entre grupos de datos multidimensionales.
- El algoritmo t-SNE es capaz de agrupar los vectores de deformación en grupos de daños y en niveles de carga independiente pero de forma simultánea.
- Variando el parámetro *perplexity* del t-SNE se consigue aumentar o disminuir el nivel de abstracción sobre los datos.
- Los precisión de la clasificación, tanto del nivel de carga, como para tipos de daño han sido muy elevadas. Entre 98 % y 99,6 % con el dataset de validación.
- La clasificación ha sido más precisa con un número de sensores bajo (FBG).

Una vez terminadas las conclusiones extraídas del proyecto INESASSE, se puede proceder a comentar las conclusiones de *Aplicación de DL a la detección de cargas térmicas*.

En este apartado se añade la componente de temperatura, la cual es interesante en el estudio del SHM.

- Los modelos de elementos finitos son una herramienta útil para comprobar de forma rápida y efectiva la viabilidad de un determinado algoritmo basado en el campo de deformaciones.
- La aditividad de los campos de deformaciones provocados por la carga vertical y el provocado por la diferencia de temperatura facilita en gran medida la obtención de las muestras.
- El algoritmo t-SNE es capaz de agrupar los datos en grupos de nivel de carga, temperatura de la estructura y tipo de daño de forma simultánea y claramente diferenciada.
- Los resultados de la red en la clasificación de nivel de carga y temperatura han sido elevados (99,9 %), sin embargo, en el tipo de daño ha bajado a un 90,79 %.
- En la Figura ?? se puede comprobar que las muestras mal clasificadas en la Figura 3.51 están agrupadas muy juntas. Con esto se llega a la conclusión de que estos campos de deformaciones son casi equivalentes para la red, por lo que no es capaz de distinguirlos de forma efectiva.

Una vez concluida las partes correspondientes a las deformaciones, se va a pasar al capítulo *Caracterización de impactos con Deep Learning*

Este capítulo ha sido la parte conceptualmente más compleja, por lo que se han extraído conclusiones interesantes.

- El mayor inconveniente de los algoritmos de DL es que se necesita un dataset con un número elevado de muestras. Esto complica su aplicación en determinados casos.
- Cuando se entrena una NN con un número pequeño de muestras se produce un fenómeno llamado *overfitting* o sobrealmimentación y hace que los resultados no sean fiables.
- Para aumentar el número de muestras el uso de redes GAN es recomendable.
- El uso de NN hace que la física del problema a resolver no sea relevante. La red interioriza estas dependencias y el usuario no necesita tener esos conocimientos.
- La red TimeGAN es capaz de sintetizar de forma efectiva un impacto medido por varios piezoelectrinos.
- La visualización de los impactos reales y sintéticos con el algoritmo t-SNE sirve para comprobar si los impactos sintéticos y reales son similares entre sí y diferentes al resto.
- Al ser agrupados la mayoría de impactos reales dentro de las nubes de impactos sintéticos, se puede considerar que los impactos son similares entre sí.
- Aunque la visualización da unos resultados positivos, el número de muestras reales es demasiado pequeña y no sería recomendable generar impactos con un dataset real tan pequeño.
- El coste en tiempo que conlleva la generación de 72 clases de impacto es muy elevado, del orden de 2500 minutos considerando que no falle nada.

- La clasificación de los impactos es muy precisa cuando se valida con los datos de entrenamiento y baja la precisión cuando se utilizan los reales.
- En la matriz de confusión validada con reales se ve que la red falla mayoritariamente con las celdas próximas.

## 5.2. Trabajos futuros

Tras el esfuerzo realizado durante este Trabajo Fin de Grado se ha abierto la puerta a varias líneas de trabajo futuras para seguir desarrollando la aplicación de Deep Learning y Machine Learning dentro del campo del SHM.

Como cada herramienta que se ha desarrollado se ha centrado en un tipo de datos en concreto, se van a comentar las posibles líneas de desarrollo de forma independiente.

Se ha comprobado que la utilización de redes neuronales para la detección de carga aplicada, temperatura y daños es muy efectiva consiguiendo precisiones superiores a un 90 %, por lo que se propone los siguientes desarrollos:

- Para conseguir que estos sistemas se puedan integrar en una aeronave, se propone utilizar modelos de elementos finitos con los que se puedan obtener los campos de deformaciones producidos por cargas aerodinámicas en distintas configuraciones de vuelo.

Con estos datos, se podrá entrenar a una NN y validar los resultados con un modelo en un túnel de viento.

A su vez, se pueden simular diferentes daños, entrenar un clasificador y validar su funcionamiento en diversas condiciones de vuelo con el modelo en el túnel de viento.

- Saliendo de la aeronáutica, se puede introducir esta herramienta de SHM en el sector eólico. Al tratarse de estructuras aeronáuticas, que se fabrican con materiales compuestos, la interacción de los sensores se puede realizar de una forma sencilla y así se podrá monitorizar de forma eficiente y rápida la salud de las estructuras de los aerogeneradores.

Para terminar, se van a proponer futuros desarrollos relacionados con la herramienta de detección de impactos:

- Fabricación de un impactador automático con el que se puedan realizar una gran cantidad de impactos de forma autónoma y sin necesidad de supervisión humana.

Este impactador también deberá ser capaz de adaptarse a superficies complejas y curvas.

Con esto, se podrá generar un dataset del tamaño necesario para conseguir unos resultados fiables con las redes clasificadoras.

- Validar el funcionamiento de TimeGAN con un dataset de impactos reales más grande.
- Investigar si las NN son capaces de clasificar diferentes impactos por niveles de energía y masa del objeto que ha impactado.

# Bibliografía

- [1] Mohamed Abdel-Basset Abdo. “Structural Health Monitoring: History, Applications and Future”. En: (2014).
- [2] C. Aguilar. “Data analysis for Structural Health Monitoring.” En: (2019).
- [3] Antonio Carlos Ancelotti Juniora et al. “Use of the Mar-Lin Criteria to Determine the Influence of Porosity on the Iosipescu and Short Beam Shear Properties in Carbon Fiber Polymer Matrix Composites”. En: (2009).
- [4] Ian J. Goodfellow et al. “Generative Adversarial Networks”. En: (2019). DOI: \url{https://arxiv.org/abs/1406.2661}.
- [5] Mohsen Azimi et al. “Data-Driven Structural Health Monitoring and Damage Detection through Deep Learning: State-of-the-Art Review”. En: *Sensors* (13 May 2020). DOI: 10.3390/s20102778.
- [6] Syafiqah Nur Azrie Bt Safri et al. “Damage analysis of glass fiber reinforced composites”. En: (2019). DOI: 10.1016/B978-0-08-102290-0.00007-6.
- [7] V. J. Hawyes et al. “Effect of impact damage on the compressive response of composite laminates”. En: (2000).
- [8] A.Khan et al. “Structural vibration-based classification and prediction of delamination in smart composite laminates using deep learning neural network”. En: *Composites Part B: Engineering* 161 (2019), págs. 586-594. DOI: 10.1016/j.compositesb.2018.12.118.
- [9] Abdeljaber et al. “Wireless and real-time structural damage detection: A novel decentralized method for wireless sensor networks”. En: *Journal of Sound and Vibration* 424 (2018), págs. 158-172. DOI: 10.1016/j.jsv.2018.03.008.
- [10] Alexandre Bérard et al. “Listen and Translate: A Proof of Concept for End-to-End Speech-to-Text Translation”. En: *30th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain*. (2016). DOI: \url{arXiv:1612.01744v1}.
- [11] ASubhashini Venugopalan et al. “Improving LSTM-based Video Description with Linguistic Knowledge Mined from Text”. En: (2016). DOI: \url{arXiv:1604.01729v2}.
- [12] Bao et al. “Computer vision and deep learning-based data anomaly detection method for structural health monitoring”. En: *Sage* (2018). DOI: 10.1177/1475921718757405.
- [13] Chul Min Yeum et al. “Automated region-of-interest localization and classification for vision-based visual assessment of civil infrastructure”. En: *Structural Health Monitoring* (2018). DOI: 10.1177/1475921718765419.
- [14] Chunting Zhou et al. “A C-LSTM Neural Network for Text Classification”. En: (2015). DOI: \url{http://arxiv.org/abs/1511.08630v2}.

- [15] Javier Gómez-Escaloni et al. “Strain predictions using Artificial Neural Networks for a full-scale fatigue monitoring system”. En: *Sixth DSTO International Conference on Health Usage Monitoring* ().
- [16] Kim et al. “Crack and Noncrack Classification from Concrete Surface Images Using Machine Learning”. En: *Structural Health Monitoring* 18 (2019), págs. 725-738. DOI: 10.1177/1475921718768747.
- [17] Mia Xu Chen et al. “Gmail Smart Compose: Real-Time Assisted Writing”. En: (2019). DOI: arXiv:1906.00080v1.
- [18] Vincentius Ewald et al. “DeepSHM: A Deep Learning Approach for Structural Health Monitoring Based on Guided Lamb Wave Techniques”. En: (2019). DOI: 10.1117/12.2506794.
- [19] JSarthak V. Doshi et all. “AArtificial Intelligence Chatbot in Android System using Open Source Program-O”. En: *Third Year Student, Department of Computer Engineering, VPKBIET, Baramati, Pune, India* (). DOI: 10.17148/IJARCCE.2017.64151.
- [20] Tom B. Brown et all. “Language Models are Few-Shot Learners”. En: (2020). arXiv: 2005.14165 [cs.CL].
- [21] J. García Alonso. “Monitorización de estructuras aeronáuticas mediante técnicas de inteligencia artificial.” En: (2016).
- [22] Christian Boller. “Structural Health Management of Ageing Aircraft and Other Infrastructure.” En: (2008).
- [23] Jatin Borana. “Applications of Artificial Intelligence Associated Technologies”. En: *Btech IV year, Department of Electrical Engineering, Jodhpur National University* ().
- [24] J.M. W. Brownjohn. “Structural Health Monitoring od civil infraestructure.” En: (2007).
- [25] P.W.R. Beaumont C. Soutis. “Multi-scale Modelling of Composite Material Systems: The Art of Predictive Damage Modelling.” En: *Woodhead Publishing Ltd.* (2005).
- [26] Albert Einstein. “Zur Elektrodynamik bewegter Körper. (German) [On the electrodynamics of moving bodies]”. En: *Annalen der Physik* 322.10 (1905), págs. 891-921. DOI: 10.1002/andp.19053221004.
- [27] J Feldman. “Artificial Intelligence in Cognitive Science”. En: *International Encyclopedia of the Social & Behavioral Science* 2 (2001), págs. 792-796. DOI: 10.1016/B978-008045046-9.00434-4.
- [28] *GAN Lab*. <https://poloclub.github.io/ganlab/>. Accessed: 2020-07-12.
- [29] Victor Giurgiutiu. “Structural health monitoring (SHM) of aerospace composites.” En: *Polymer Composites in the Aerospace Industry*. (2019). DOI: 10.1016/B978-0-08-102679-3.00017-4.
- [30] Daniel Jarrett Jinsung Yoon y Mihaela van der Schaar. “Time-series Generative Adversarial Networks”. En: (2019). DOI: \url{https://github.com/jsyoon0823/TimeGAN}.
- [31] G. Manson y G. Park K. Worden C.R. Farrar. “The fundamental axioms of structural health monitoring.” En: (2007).

- [32] *Learning Paradigms in Machine Learning*. <https://medium.com/datadriveninvestor/learning-paradigms-in-machine-learning-146ebf8b5943>. Accessed: 2020-08-24.
- [33] Van der Maaten end Hinton. “Visualizing Data using t-SNE”. En: *Journal of Machine Learning Research* 9 (2008), págs. 2579-2605.
- [34] Mohammad Jawaid y Mohamed Thariq. “Sustainable Composites for Aerospace Applications”. En: (2018). DOI: 10.1016/B978-0-08-102131-6.00007-4.
- [35] TensorFlow - Playground. En: (). DOI: <https://playground.tensorflow.org/>.
- [36] Verma Rahul. “Structural health monitoring of aerospace composites.” En: (). DOI: 10.1016/B978-0-08-102291-7.00003-4.
- [37] J. García Ramirez. “Análisis de datos para la monitorización de la salud estructural.” En: (2018).
- [38] *Recurrent Neural Network*. <http://karpathy.github.io/2015/05/21/rnn-effectiveness/>. Accessed: 2020-09-02.
- [39] *Redes Neuronales Recurrentes*. <https://torres.ai/redes-neuronales-recurrentes/>. Accessed: 2020-09-02.
- [40] M. Scheerer. “SHM of Composite Structures using Acoustic Emission Methods.” En: *European Ground Testing Instrumentation Aerospace Testing Expo* (2008). DOI: 10.1016/B978-0-08-102679-3.00017-4.
- [41] *The State of the Octoverse: machine learning*. <https://github.blog/2019-01-24-the-state-of-the-octoverse-machine-learning/>. Accessed: 2020-08-25.
- [42] *Understanding LSTM Networks*. <https://colah.github.io/posts/2015-08-Understanding-LSTMs/>. Accessed: 2020-09-02.
- [43] Saul A. Teukolsky William H. Press. “Savitzky-Golay Smoothing Filters”. En: (). DOI: 10.1063/1.4822961.
- [44] Dongho Kang y Young-Jin Cha. “Autonomous UAVs for Structural Health Monitoring Using Deep Learning and an Ultrasonic Beacon System with Geo-Tagging”. En: *Computer-Aided Civil and Infrastructure Engineering* (2018). DOI: 10.1111/mice.12375.