Universidad de Oviedo
Universidá d'Uviéu
University of Oviedo

PROGRAMA DE DOCTORADO EN
ENERGÍA Y CONTROL DE PROCESOS

Escuela Politécnica de Ingeniería de Gijón
Dpto. de Ingeniería Eléctrica, Electrónica, de Computadores y Sistemas

TESIS DOCTORAL

Aplicación de técnicas de aprendizaje profundo *(deep learning)* al análisis y mejora de la eficiencia en sistemas de ingeniería

Ana González Muñiz
Ignacio Díaz Blanco (Director)
Noviembre 2022

Universidad de Oviedo

RESUMEN DEL CONTENIDO DE TESIS DOCTORAL

1.- Título de la Tesis	
Español/Otro Idioma: APLICACIÓN DE TÉCNICAS DE APRENDIZAJE PROFUNDO (*DEEP LEARNING*) AL ANÁLISIS Y MEJORA DE LA EFICIENCIA EN SISTEMAS DE INGENIERÍA	Inglés: APPLICATION OF DEEP LEARNING TECHNIQUES TO THE ANALYSIS AND IMPROVEMENT OF EFFICIENCY IN ENGINEERING SYSTEMS

2.- Autor	
Nombre: ANA GONZÁLEZ MUÑIZ	
Programa de Doctorado: ENERGÍA Y CONTROL DE PROCESOS	
Órgano responsable: CENTRO INTERNACIONAL DE POSTGRADO	

RESUMEN (en español)

A lo largo de la última década, los algoritmos de aprendizaje profundo o *deep learning* se han convertido en un motor de innovación y transformación, con impacto en una amplia variedad de sectores y aplicaciones. Las grandes empresas tecnológicas de nuestro tiempo recurren ya a estos algoritmos como herramientas indispensables en el desarrollo de sus productos e incluso cualquier tarea cotidiana —como el uso del traductor, el reconocimiento facial o de voz en nuestros *smart phones*— lleva implícito hoy en día el uso de tecnología *deep learning*.

Este éxito de los modelos profundos reside en su arquitectura jerárquica, que les confiere la habilidad de transformar grandes volúmenes de datos en información de valor para el usuario. En detalle, las técnicas *deep learning* se han mostrado especialmente útiles en aquellas aplicaciones donde se requiere el manejo de datos de alta dimensionalidad, como ocurre en los ámbitos del procesamiento de imagen o del procesamiento del lenguaje —donde las muestras de trabajo pueden estar constituidas por millones de píxeles o cientos de miles de palabras, respectivamente. Históricamente, los avances más notables del aprendizaje profundo han tenido lugar en estos dos ámbitos, donde las técnicas *deep learning* han obtenido excelentes resultados, superando a otros enfoques del estado del arte.

Dado su gran rendimiento, estas técnicas no dejan de crecer en popularidad y se enfrentan a retos cada vez más ambiciosos. El aprendizaje profundo constituye, por tanto, un campo de investigación en constante evolución, con resultados sorprendentes en una amplia variedad de aplicaciones y que ha revolucionado por completo el procesamiento de datos en ámbitos como el tratamiento de imagen y vídeo, o el análisis del lenguaje. Sin embargo, estas técnicas podrían tener un gran impacto en muchos otros sectores, aún poco explorados. Uno de estos ámbitos es el de los sistemas de ingeniería que —con la madurez de las comunicaciones, los sistemas de almacenamiento y la riqueza en sensores— comienza a gozar de una alta disponibilidad de datos para una amplia variedad de problemas complejos, que podrían verse beneficiados por los avances que las técnicas *deep learning* ya han protagonizado en otros contextos. En consecuencia, se presenta en esta tesis un estudio del estado del arte actual y de las potenciales contribuciones del aprendizaje profundo en el ámbito de los sistemas de ingeniería.

En particular, se ha estudiado la contribución de estas técnicas al análisis y mejora de la eficiencia de los sistemas, para lo cual se ha explorado el uso de distintos tipos de arquitecturas profundas sobre diferentes problemas y contextos de ingeniería. Más en detalle, se ha explorado el rendimiento de arquitecturas profundas bien conocidas (redes *feedforward*, redes convolucionales y *deep autoencoders*) en un amplio rango de aplicaciones, como son la clasificación, la detección de anomalías, la generación de indicadores de salud o la visualización de mapas 2D de los procesos, que faciliten la exploración de los datos al usuario

en busca de nuevo conocimiento acerca de los sistemas.

En primer lugar, se ha explorado el potencial de las arquitecturas profundas en la clasificación del estado de funcionamiento de los procesos. Para ello, se han analizado dos tipos de arquitecturas profundas (redes *feedforward* y redes convolucionales) y se ha valorado su rendimiento en la detección de fallos en motores. En segundo lugar, se ha investigado la contribución de las arquitecturas profundas en el ámbito de la detección de anomalías. En detalle, se ha estudiado el uso de *deep autoencoders*, en combinación con técnicas de análisis de residuos, para la detección de comportamientos anómalos en diferentes contextos de ingeniería, como la operación de motores, sistemas hidráulicos o sistemas de monitorización del movimiento humano. En tercer lugar, se ha explorado el uso de estas arquitecturas para la generación de indicadores de salud de los procesos. Para ello, se han analizado las representaciones de los datos disponibles en las capas ocultas de los *deep autoencoders* y su potencial uso como indicadores de salud de los procesos. En particular, se ha evaluado su rendimiento como indicadores de degradación de las máquinas. En último lugar, se ha investigado el potencial de las arquitecturas profundas para la generación de visualizaciones interpretables de los sistemas. En detalle, se ha explorado la generación de proyecciones de baja dimensión de los datos (en particular, de dos dimensiones o 2D) mediante el uso de *deep autoencoders*. Estas proyecciones han sido integradas en herramientas de visualización interactiva para crear mapas 2D interpretables y de fácil exploración por parte del usuario, cuya contribución ha sido evaluada en el análisis del consumo energético de una gran instalación.

En todos los casos, los enfoques profundos propuestos han permitido abordar los problemas objetivo con éxito, mostrándose altamente competitivos en comparación con otras técnicas de la literatura y contribuyendo, también, a la mejora de la comprensión de los procesos bajo estudio. En consecuencia, los resultados expuestos en esta tesis proporcionan una evidencia del potencial de las arquitecturas profundas como valiosas herramientas para la monitorización de la condición de los sistemas.

Adicionalmente, se ha explorado la aplicabilidad de estos enfoques en otros ámbitos con problemáticas similares, como es el caso de la biomedicina, donde, al igual que en los sistemas de ingeniería, la monitorización de la condición de los procesos también es crítica. En los próximos años, se espera que el aprendizaje profundo tenga un gran impacto tanto en el ámbito biomédico como en el de los sistemas de ingeniería, con lo que la transferencia de conocimiento entre ambos campos podría representar una interesante línea de trabajo futuro. En particular, se ha trasladado con éxito el último de los enfoques propuestos —consistente en el uso de *deep autoencoders* para la generación de mapas 2D de los procesos— al ámbito del análisis y mejora de la comprensión de los mecanismos vinculados con la propagación del cáncer.

RESUMEN (en Inglés)

Over the last decade, deep learning algorithms have become a driver of innovation and transformation, impacting a wide variety of sectors and applications. Some of the most relevant technological companies of our time are already using these algorithms as indispensable tools in the development of their products, and even any everyday task —such as the use of translators, or facial and voice recognition on our smart phones— now involves the use of deep learning technology.

The success of deep models lies in their hierarchical architecture, which gives them the ability to transform large volumes of data into useful information for the user. In particular, deep learning techniques have proven especially useful in those applications where the handling of high-dimensional data is required, such as in the fields of image or language processing — where the working samples may consist of millions of pixels or hundreds of thousands of words, respectively. Historically, the most notable advances in deep learning have been in these two areas, where deep learning techniques have achieved excellent results, outperforming other state-of-the-art approaches.

Given their high performance, these techniques are becoming increasingly popular and facing

ambitious challenges. Deep learning is therefore a constantly evolving field of research, which has achieved impressive results in a wide variety of applications and has completely revolutionised data analysis in areas such as image, video or language processing. However, these techniques may have a major impact in many other, as yet underexplored, areas. One of these areas could be that of engineering systems, which —with the maturity of communications, storage systems and the wealth of sensors— is beginning to exhibit a high availability of data for a wide variety of complex problems, which could benefit from the progress that deep learning techniques have already achieved in other contexts. Accordingly, this thesis presents a survey of the current state of the art and potential contributions of deep learning in the field of engineering systems.

In particular, the contribution of these techniques to the analysis and improvement of system efficiency has been studied by exploring the use of different types of deep architectures on different problems and engineering contexts. In more detail, the performance of well-known deep architectures (feedforward networks, convolutional networks and deep autoencoders) has been explored on a wide range of problems, such as classification, anomaly detection, generation of health indicators or visualisation of 2D process maps which may facilitate the exploration of the data in search of new knowledge about the systems.

Firstly, we have explored the potential of deep architectures in the classification of the operating condition of the machines. To this end, two types of deep architectures (feedforward and convolutional neural networks) have been analysed and their performance has been assessed in the context of machine fault detection. Secondly, we have investigated the contribution of deep architectures in the field of anomaly detection. In detail, we have studied the use of deep autoencoders, in combination with residual analysis techniques, for the detection of anomalous behaviour in different engineering contexts (a rotating machine, a hydraulic system and a body motion system). Thirdly, we have explored the use of these architectures for the generation of process health indicators. To this end, we have analysed the data representations available in the hidden layers of deep autoencoders and their potential use as process health indicators. In particular, their performance as indicators of machine degradation has been evaluated. Finally, we have investigated the potential of deep architectures for the generation of interpretable visualisations of the systems under study. In detail, the generation of low-dimensional projections of data (in particular two-dimensional or 2D) using deep autoencoders has been explored. These projections have been integrated into interactive visualisation tools to create 2D maps that are interpretable and easy to explore by the user, and whose contribution has been evaluated in the analysis of the energy consumption of a large facility.

In all cases, the proposed deep approaches have successfully addressed the target problems, proving to be highly competitive compared to other techniques in the literature and also contributing to the improvement of the understanding of the processes under study. Consequently, the results presented in this thesis provide evidence of the potential of deep architectures as valuable tools for system condition monitoring.

In addition, the applicability of these approaches has been explored in the biomedical domain, where, as in engineering systems, process condition monitoring is also a critical issue. In the coming years, deep learning is expected to have a major impact on both the biomedical and the engineering systems fields, so that knowledge transfer between these two domains could represent an interesting line of future work. In particular, the last of the proposed approaches —the use of deep autoencoders for the generation of 2D process maps— has been successfully transferred to the field of cell motility analysis, contributing to the better understanding of the mechanisms involved in the progression of cancer.

SR. PRESIDENTE DE LA COMISIÓN ACADÉMICA DEL PROGRAMA DE DOCTORADO EN _____

Universidad de Oviedo
Universidá d'Uviéu
University of Oviedo

PROGRAMA DE DOCTORADO EN
ENERGÍA Y CONTROL DE PROCESOS

Escuela Politécnica de Ingeniería de Gijón
Dpto. de Ingeniería Eléctrica, Electrónica, de Computadores y Sistemas

TESIS DOCTORAL

Aplicación de técnicas de aprendizaje profundo *(deep learning)* al análisis y mejora de la eficiencia en sistemas de ingeniería

Ana González Muñiz
Ignacio Díaz Blanco (Director)
Noviembre 2022

Agradecimientos

Esta tesis doctoral no habría sido posible sin la participación y colaboración de las siguientes personas e instituciones, a quienes deseo expresar mi más sincero agradecimiento.

Gracias,

- A mi director de tesis, el profesor Ignacio Díaz Blanco, por su apoyo y motivación durante estos años, por descubrirme el increíble mundo del *deep learning* y darme la oportunidad de iniciar mi carrera como investigadora.
- A todos los miembros de mi grupo de investigación, el GSDPI, que me han acompañado durante esta tesis, compartiendo su sabiduría y ayudándome en incontables ocasiones.
- Al Departamento de Ingeniería Eléctrica, Electrónica, de Computadores y Sistemas de la Universidad de Oviedo, del cual he formado parte estos años y que me ha brindado la oportunidad de iniciar mi carrera docente.
- Al grupo de investigación liderado por la Dra. María Dolores Chiara, perteneciente al área de investigación de Cáncer de Cabeza y Cuello del ISPA, por acogerme durante un periodo de tres meses y mostrarme el mundo de la investigación desde otro punto de vista, dándome la oportunidad de convivir en un ambiente multidisciplinar y tender puentes entre dos ámbitos —habitualmente tan alejados— como el de la industria y el de la biomedicina. Gracias, también, por los conjuntos de datos proporcionados para el desarrollo de esta tesis, fruto de la colaboración en el marco del proyecto PID2020-115401GB-I00.
- Al Hospital Universitario de León y al grupo SUPPRESS de la Universidad de León, que han proporcionado conjuntos de datos empleados en los experimentos de esta tesis, en el marco del proyecto DPI2015-69891-C2-1/2-R (coordinado con el grupo de investigación GSDPI) y, a su vez, fruto de la colaboración actual entre ambos grupos en los proyectos PID2020-115401GB-I00 (GSDPI) y PID2020-117890RB-I00 (SUPPRESS).
- Al Ministerio de Economía y Competitividad (MINECO) y a los Fondos FEDER de la Unión Europea (proyecto DPI2015-69891-C2-1/2-R), así como a la Agencia Estatal de Investigación (proyecto PID2020-115401GB-I00), por la financiación de los dos Planes Nacionales en los que he participado —como miembro del equipo de trabajo— durante el desarrollo de esta tesis.

- Al Gobierno del Principado de Asturias, que a través de la ayuda predoctoral Severo Ochoa (BP17-180) me ha proporcionado el apoyo económico necesario para la realización de la presente tesis.
- A la Universidad de Oviedo y a la Escuela Politécnica de Ingeniería de Gijón, por ser un lugar de encuentro y aprendizaje continuo en el que he tenido la suerte de crecer, personal y profesionalmente, durante esta gran etapa de mi vida.

Resumen

A lo largo de la última década, los algoritmos de aprendizaje profundo o *deep learning* se han convertido en un motor de innovación y transformación, con impacto en una amplia variedad de sectores y aplicaciones. Las grandes empresas tecnológicas de nuestro tiempo recurren ya a estos algoritmos como herramientas indispensables en el desarrollo de sus productos e incluso cualquier tarea cotidiana —como el uso del traductor, el reconocimiento facial o de voz en nuestros *smart phones*— lleva implícito hoy en día el uso de tecnología *deep learning*.

Este éxito de los modelos profundos reside en su arquitectura jerárquica, que les confiere la habilidad de transformar grandes volúmenes de datos en información de valor para el usuario. En detalle, las técnicas *deep learning* se han mostrado especialmente útiles en aquellas aplicaciones donde se requiere el manejo de datos de alta dimensionalidad, como ocurre en los ámbitos del procesamiento de imagen o del procesamiento del lenguaje —donde las muestras de trabajo pueden estar constituidas por millones de píxeles o cientos de miles de palabras, respectivamente. Históricamente, los avances más notables del aprendizaje profundo han tenido lugar en estos dos ámbitos, donde las técnicas *deep learning* han obtenido excelentes resultados, superando a otros enfoques del estado del arte.

Dado su gran rendimiento, estas técnicas no dejan de crecer en popularidad y se enfrentan a retos cada vez más ambiciosos. El aprendizaje profundo constituye, por tanto, un campo de investigación en constante evolución, con resultados sorprendentes en una amplia variedad de aplicaciones y que ha revolucionado por completo el procesamiento de datos en ámbitos como el tratamiento de imagen y vídeo, o el análisis del lenguaje. Sin embargo, estas técnicas podrían tener un gran impacto en muchos otros sectores, aún poco explorados. Uno de estos ámbitos es el de los sistemas de ingeniería que —con la madurez de las comunicaciones, los sistemas de almacenamiento y la riqueza en sensores— comienza a gozar de una alta disponibilidad de datos para una amplia variedad de problemas complejos, que podrían verse beneficiados por los avances que las técnicas *deep learning* ya han protagonizado en otros contextos. En consecuencia, se presenta en esta tesis un estudio del estado del arte actual y de las potenciales contribuciones del aprendizaje profundo en el ámbito de los sistemas de ingeniería.

En particular, se ha estudiado la contribución de estas técnicas al análisis y mejora de la eficiencia de los sistemas, para lo cual se ha explorado el uso de distintos tipos de arquitecturas profundas sobre diferentes problemas y contextos de ingeniería. Más en detalle, se ha explorado el rendimiento de arquitecturas profundas bien conocidas (redes *feedforward*, redes convolucionales y *deep autoencoders*) en un amplio rango de aplicaciones, como son la clasificación, la detección de ano-

malías, la generación de indicadores de salud o la visualización de mapas 2D de los procesos, que faciliten la exploración de los datos al usuario en busca de nuevo conocimiento acerca de los sistemas.

En primer lugar, se ha explorado el potencial de las arquitecturas profundas en la clasificación del estado de funcionamiento de los procesos. Para ello, se han analizado dos tipos de arquitecturas profundas (redes *feedforward* y redes convolucionales) y se ha valorado su rendimiento en la detección de fallos en motores. En segundo lugar, se ha investigado la contribución de las arquitecturas profundas en el ámbito de la detección de anomalías. En detalle, se ha estudiado el uso de *deep autoencoders*, en combinación con técnicas de análisis de residuos, para la detección de comportamientos anómalos en diferentes contextos de ingeniería, como la operación de motores, sistemas hidráulicos o sistemas de monitorización del movimiento humano. En tercer lugar, se ha explorado el uso de estas arquitecturas para la generación de indicadores de salud de los procesos. Para ello, se han analizado las representaciones de los datos disponibles en las capas ocultas de los *deep autoencoders* y su potencial uso como indicadores de salud de los procesos. En particular, se ha evaluado su rendimiento como indicadores de degradación de las máquinas. En último lugar, se ha investigado el potencial de las arquitecturas profundas para la generación de visualizaciones interpretables de los sistemas. En detalle, se ha explorado la generación de proyecciones de baja dimensión de los datos (en particular, de dos dimensiones o 2D) mediante el uso de *deep autoencoders*. Estas proyecciones han sido integradas en herramientas de visualización interactiva para crear mapas 2D interpretables y de fácil exploración por parte del usuario, cuya contribución ha sido evaluada en el análisis del consumo energético de una gran instalación.

En todos los casos, los enfoques profundos propuestos han permitido abordar los problemas objetivo con éxito, mostrándose altamente competitivos en comparación con otras técnicas de la literatura y contribuyendo, también, a la mejora de la comprensión de los procesos bajo estudio. En consecuencia, los resultados expuestos en esta tesis proporcionan una evidencia del potencial de las arquitecturas profundas como valiosas herramientas para la monitorización de la condición de los sistemas.

Adicionalmente, se ha explorado la aplicabilidad de estos enfoques en otros ámbitos con problemáticas similares, como es el caso de la biomedicina, donde, al igual que en los sistemas de ingeniería, la monitorización de la condición de los procesos también es crítica. En los próximos años, se espera que el aprendizaje profundo tenga un gran impacto tanto en el ámbito biomédico como en el de los sistemas de ingeniería, con lo que la transferencia de conocimiento entre ambos campos podría representar una interesante línea de trabajo futuro. En particular, se ha trasladado con éxito el último de los enfoques propuestos —consistente en el uso de *deep autoencoders* para la generación de mapas 2D de los procesos— al ámbito del análisis y mejora de la comprensión de los mecanismos vinculados con la propagación del cáncer.

Índice

Lista de figuras

Lista de tablas

CAPÍTULO 1

Introducción

En este primer capítulo se presenta la temática de la tesis, comenzando con una breve introducción a las técnicas de aprendizaje profundo y su potencial en el ámbito de la ingeniería. A continuación, se define el propósito de esta investigación y se presenta la formulación del problema, terminando con una descripción de la estructura del documento.

1.1. Introducción

A lo largo de la última década, los algoritmos de aprendizaje profundo o *deep learning* se han convertido en un motor de innovación y transformación, con impacto en una amplia variedad de sectores y aplicaciones. Las grandes empresas tecnológicas de nuestro tiempo —Google, Amazon, Facebook, Apple, Microsoft— recurren ya a estos algoritmos como herramientas indispensables en el desarrollo de sus productos. Incluso cualquier tarea cotidiana —como el uso del traductor, el reconocimiento facial o de voz en nuestros *smart phones*— lleva implícito hoy en día el uso de tecnología *deep learning*.

Estos modelos de aprendizaje profundo se han establecido como una de las ramas más destacadas de la inteligencia artificial, aportando resultados muy superiores a los obtenidos hasta el momento con otras técnicas de aprendizaje automático *(machine learning)*. No obstante, cabe destacar que la irrupción de estas técnicas es reciente —sus inicios se sitúan habitualmente en 2006, de la mano de Geoffrey Hinton [1]— y el éxito cosechado en este breve periodo de tiempo es una medida de su gran potencial.

Una muestra de ello es el AlphaGo de DeepMind[1]. En 2016, AlphaGo se proclamó como el primer programa capaz de vencer a un jugador profesional de Go. Poco después, en 2017, una versión mejorada de AlphaGo [2] derrotaría al número uno del mundo, el chino Ke Jie. Esta victoria supuso un hito en la historia de la

[1] DeepMind es una *start-up* inglesa fundada por Demis Hassabis —junto con Shane Legg y Mustafa Suleyman— en 2010 (adquirida más tarde por Google, en 2014) dedicada al desarrollo de aplicaciones basadas en aprendizaje profundo.

inteligencia artificial —y del aprendizaje profundo, en particular— pues por aquel entonces se creía que aún serían necesarios muchos más años de investigación antes de que un ordenador fuese capaz de competir al nivel de un jugador de élite. El Go es un juego muy complejo: el gran número de casillas y jugadas posibles en cada turno hace que el número de posibles combinaciones de movimientos sea increíblemente grande [3]. Esto convierte al Go en un juego enormemente desafiante, tanto para los humanos como para las máquinas. Tanto es así que, a finales de los años noventa, las máquinas ya triunfaban en juegos de mesa como el ajedrez —DeepBlue vencía al campeón del mundo Garry Kaspárov en 1997 [4]— y, sin embargo, los programas de Go estaban aún lejos de competir contra humanos.

Cabe destacar también que, en el caso del ajedrez, fueron necesarios treinta años para que las máquinas evolucionasen de nivel *amateur* a finales de los años sesenta, a campeón del mundo en 1997. Mientras, esta evolución se culminó en una década para el caso del Go, a pesar de su mayor complejidad, gracias a la aparición de las técnicas de aprendizaje profundo. Este progreso sin precedentes en la competición de juegos de mesa es una muestra del impacto que las técnicas *deep learning* podrían tener en una amplia variedad de ámbitos, y cuyo uso se está extendiendo ya con éxito en aplicaciones de visión por computador, traducción inteligente, reconocimiento de voz o de imagen [5, 6].

Detrás de este éxito se encuentran operaciones matemáticas simples. Podríamos decir que, en esencia, un modelo *deep learning* no es más que un gran número de sumas y productos, en combinación con algunas transformaciones no lineales. Pero estas operaciones simples se organizan en capas, dotando a los modelos de una estructura profunda y jerárquica, que les permite identificar y extraer patrones de los datos desde un nivel de abstracción más bajo (capas inferiores) a uno más alto (capas superiores). De esta forma, los modelos profundos son capaces de transformar vastas cantidades de datos en información útil para el usuario. Además, estas técnicas son pioneras en su escalabilidad respecto a la cantidad de datos de entrada: cuanto mayor es el volumen de datos que manejan, mejor es su rendimiento (Figura 1.1), lo cual permite explotar al máximo los beneficios de otros conceptos de vanguardia como son el Big Data o la Industria 4.0.

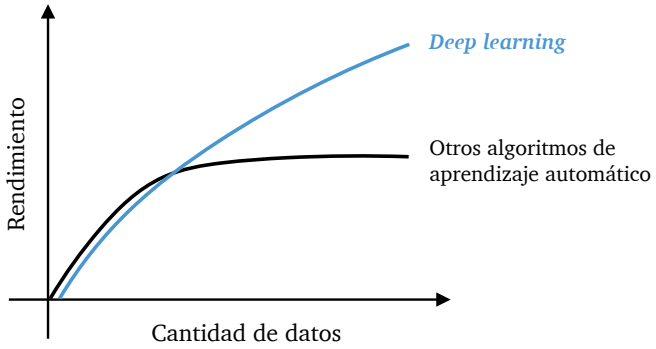


Figura 1.1: Rendimiento de los algoritmos de aprendizaje automático en función del volumen de datos disponible. Figura adaptada de [7].

Un factor clave en la competitividad de las técnicas *deep learning* es su estructura jerárquica, que les confiere la habilidad de aprender representaciones complejas de los datos, también conocida como *representation learning* [8]. Los modelos

profundos establecen una jerarquía de características —donde cada característica resulta de la composición de otras más simples— que les permite llegar a extraer de forma automática, a partir de los datos crudos, características abstractas y de alto nivel con las que abordar eficientemente el problema objetivo.

Este es un aspecto crucial en cualquier problema de análisis de datos, donde los resultados obtenidos dependen en gran medida de la calidad de las características empleadas en el análisis. Tradicionalmente, los algoritmos de aprendizaje automático han recurrido al conocimiento humano experto para la selección y diseño de características, tratándose de una tarea crítica, que habitualmente requiere gran cantidad de tiempo y esfuerzo (se estima que puede llegar a consumir el 80 % del presupuesto de un proyecto de *machine learning* [9]). En cambio, las técnicas *deep learning* desempeñan esta tarea de forma automática, siendo capaces de extraer características relevantes únicamente a partir de los datos de trabajo, sin necesidad de experiencia previa o conocimiento de dominio del problema, lo que representa una gran ventaja respecto a los enfoques tradicionales de *machine learning*. Además, en contextos con alta disponibilidad de datos, los modelos profundos han demostrado alcanzar mejores resultados que los modelos tradicionales basados en características diseñadas manualmente. No obstante, el verdadero potencial de las técnicas *deep learning* no reside en la competición con los enfoques basados en conocimiento experto, sino en la posibilidad de reforzar este conocimiento, así como de abordar aquellos problemas en los que la extracción manual de características pueda ser limitada o incluso imposible.

Esta particular habilidad de los modelos profundos para extraer características relevantes de los datos reside en su estructura de capas, especialmente diseñada para abordar problemas complejos mediante el aprendizaje de una jerarquía —conveniente— de características. Este diseño está inspirado en la idea de *conexionismo*, acorde a la cual, comportamientos complejos pueden ser aproximados por la interacción entre un gran número de unidades de procesamiento simples. En las redes profundas, cada capa aprende una transformación simple de las características extraídas por la capa anterior, de manera que, a través de esta secuencia de pequeñas transformaciones, la arquitectura en su conjunto es capaz de modelar complejas transformaciones de los datos de entrada. En este contexto, la profundidad de los modelos es un aspecto clave, pues cuantas más capas tenga la arquitectura, mayor será su capacidad para capturar la estructura subyacente en los datos. Muchos problemas en la literatura requieren el modelado de comportamientos complejos, en los que las variables implicadas se relacionan entre sí de acuerdo a complejas transformaciones no lineales de los datos, y es precisamente en estos casos donde las técnicas *deep learning* han demostrado un enorme potencial, superando en rendimiento a otras técnicas de aprendizaje automático [10].

En detalle, las técnicas *deep learning* se han mostrado especialmente útiles en aquellas aplicaciones donde se requiere el manejo de datos de alta dimensionalidad, como ocurre en los ámbitos del procesamiento de imagen o del procesamiento del lenguaje —donde las muestras de trabajo pueden estar constituidas por millones de píxeles o cientos de miles de palabras, respectivamente. Históricamente, los avances más notables del aprendizaje profundo han tenido lugar en estos dos ámbitos, donde las técnicas *deep learning* han conseguido una precisión

comparable —en ocasiones, mejor— a la del ser humano [11], con aplicaciones en clasificación de imágenes, reconocimiento de voz, transcripción de escritura manuscrita, etc. Estas aplicaciones han tenido, además, una gran implantación. Empresas como Google, Apple, Amazon o Facebook utilizan enfoques de aprendizaje profundo en algunas de sus herramientas más populares —como el Asistente de Google, Siri o Alexa (reconocimiento de voz); Deep Face o Google Photos (reconocimiento facial); Google Translate (traducción automática de textos); Facebook Deep Text (comprensión y análisis de texto), etc. Dado su excelente rendimiento, estas técnicas no dejan de crecer en popularidad y se enfrentan a retos cada vez más ambiciosos. Entre sus éxitos más recientes se encuentran aplicaciones para la eliminación del ruido en imágenes [12], retoque fotográfico [13], coloreado de fotografías [14], transferencia de estilos pictóricos [15], descripción textual de vídeos [16], generación de vídeos manipulados o *deep fakes* [17], composición de pistas musicales [18], etc.

En definitiva, el aprendizaje profundo constituye un campo de investigación en constante evolución, con resultados sorprendentes en una amplia variedad de aplicaciones y que ha revolucionado por completo el procesamiento de datos en ámbitos como el tratamiento de imagen y vídeo, o el análisis del lenguaje. Sin embargo, estas técnicas podrían tener un gran impacto en muchos otros sectores, aún poco explorados. Uno de estos ámbitos es el de los sistemas de ingeniería que —con la madurez de las comunicaciones, los sistemas de almacenamiento y la riqueza en sensores— comienza a gozar de una alta disponibilidad de datos para una amplia variedad de problemas complejos, que podrían verse beneficiados por los avances que las técnicas *deep learning* ya han protagonizado en otros contextos. En consecuencia, se presenta en esta tesis un estudio del estado del arte actual y de las potenciales contribuciones del aprendizaje profundo en el ámbito de los sistemas de ingeniería.

1.2. Propósito de la investigación

La llegada de la Industria 4.0, también conocida como la *cuarta revolución industrial*, ha impulsado una transformación digital en procesos e instalaciones, que pretende explotar el potencial de las nuevas tecnologías en el ámbito industrial [19]. Liderada por el Internet de las Cosas *(Internet of Things, IoT)*, esta transformación propone un modelo de industria *conectada*, constituida por una red de dispositivos —maquinaria y sensores— conectados tanto entre sí, como a Internet. Este modelo de Industria 4.0 genera una cantidad masiva de datos, que pueden ser transformados en información útil para las empresas mediante herramientas de análisis de datos. En este contexto, destacan los algoritmos de aprendizaje automático, cuyo uso ha tenido un impacto directo en la mejora de la eficiencia de los sistemas productivos, con beneficios como la reducción de costes, reducción de tiempos de ejecución y toma de decisiones, mejoras en la seguridad de las instalaciones o mejoras en la calidad de los productos [20]. No obstante, estos beneficios podrían verse superados con el uso de nuevas técnicas, como los modelos de aprendizaje profundo.

Las técnicas *deep learning* han demostrado un gran potencial en el manejo de

datos complejos y el éxito que ya han tenido en otros campos —como el reconocimiento facial y de voz, o el procesamiento natural del lenguaje— podría ser extrapolable al ámbito de los sistemas de ingeniería. Cabe destacar que, a pesar de la diferencia de contexto, las problemáticas pueden ser similares en muchas ocasiones. A modo de ejemplo, las técnicas de reconocimiento facial podrían ser útiles para el reconocimiento de patrones en un amplio rango de aplicaciones, desde la detección de defectos superficiales en piezas hasta el análisis de espectrogramas; mientras, los enfoques empleados en aplicaciones de reconocimiento de voz podrían ser útiles tanto en problemas de análisis de vibraciones en motores, como en el análisis de estructuras. A estas potenciales conexiones, se suma la disponibilidad de numerosos *frameworks* y librerías de acceso gratuito, que facilitan enormemente la implementación de los modelos profundos. Entre las muchas librerías disponibles, destacan especialmente Tensorflow[2], utilizada por los laboratorios de Google y liberada a la comunidad a finales de 2015, y Keras[3], que aporta una capa de abstracción que simplifica aún más el diseño y entrenamiento de los modelos.

Con el acceso a librerías gratuitas para la implementación de las arquitecturas profundas y la disponibilidad de grandes conjuntos de datos para su entrenamiento, se abre la oportunidad de explorar el potencial de las técnicas *deep learning* en el ámbito de la ingeniería, donde se espera que desempeñen un papel significativo durante los próximos años [21]. En particular, a lo largo de esta tesis se dedicará especial atención a la monitorización de la salud de los sistemas, en busca de mejoras en la comprensión y la eficiencia de los mismos. Esta línea de investigación —enmarcada en el campo del *System Health Management (SHM)*— engloba aplicaciones como la detección, diagnóstico y pronóstico de fallos [22], que son críticas para el buen funcionamiento de procesos e instalaciones, permitiendo garantizar la seguridad en los sistemas o la planificación adecuada de operaciones de mantenimiento.

En este contexto, la capacidad de los modelos profundos para extraer características de forma automática a partir de los datos de entrada —también conocida como *feature learning*— cobra gran interés [23]. Tradicionalmente, el ámbito de la ingeniería ha estado fuertemente ligado a la ingeniería de características o *feature engineering*, donde las características son diseñadas manualmente por un experto. Estas características han obtenido buenos resultados en la literatura, pero su diseño requiere experiencia, así como conocimiento de dominio del problema, y son difícilmente extrapolables a otros ámbitos de aplicación. En su lugar, las redes profundas tienen la habilidad de abordar la tarea objetivo —clasificación, predicción, etc.— extrayendo por sí mismas las características más convenientes para dicha tarea a partir de los datos crudos de trabajo, lo que las convierte en poderosas herramientas de análisis de datos. Entre las arquitecturas presentes en la literatura, las redes convolucionales *(Convolutional Neural Networks, CNNs)* [24] han demostrado un talento especial para el aprendizaje de características, gracias a la incorporación de capas de convolución en su arquitectura, compuestas por bancos de filtros cuyos coeficientes son ajustados durante el entrenamiento de la red.

Estos filtros recorren las muestras de entrada en busca de patrones relevantes

[2] https://www.tensorflow.org
[3] https://keras.io

en los datos y, de forma análoga al funcionamiento de los campos receptivos de las neuronas, cada filtro se especializa en la detección de un patrón específico en los datos: aprovechando la estructura jerárquica de las redes profundas, los filtros de las capas inferiores detectan patrones simples, aportando una caracterización a nivel local de las muestras; mientras que los filtros de las capas superiores aprenden patrones más complejos, resultado de la composición de los patrones aprendidos por las capas anteriores y que proporcionan una descripción más global y abstracta de las muestras, así como conveniente para la resolución del problema objetivo. Este enfoque supone una gran ventaja comparativa respecto al diseño tradicional de características, pues permite prescindir del conocimiento experto, ahorrando con ello una gran cantidad de recursos. Se trata de un enfoque de gran utilidad, especialmente en aquellos casos en que no se dispone de conocimiento a priori del problema a tratar.

Sin embargo, en muchos problemas de ingeniería, la investigación en el diseño de características ha sido extensa y es fácilmente accesible. En tales casos, cabe preguntarse si las características aprendidas por los modelos profundos conducen a mejores resultados que las extraídas manualmente y, por tanto, resulta recomendable prescindir del conocimiento experto, aún disponiendo de él. Otro aspecto a explorar en esta tesis será el análisis de las características aprendidas, que podría revelar información de interés acerca de los sistemas, ayudando a mejorar la comprensión de los mismos y que, adicionalmente, aportaría luz al funcionamiento interno de los modelos, contribuyendo con ello a reforzar la confianza del usuario en los resultados obtenidos. Cabe destacar que las arquitecturas profundas son conocidas, junto con otros enfoques de aprendizaje automático, como modelos de *caja negra*, dado que el usuario desconoce las decisiones y transformaciones intermedias de los datos que el modelo ejecuta antes de llegar a la solución final del problema. En aplicaciones de gran sensibilidad —como podría ser el sistema de navegación de un vehículo autónomo— esta falta de interpretabilidad supone un gran inconveniente, que nuevas líneas de investigación fuera del alcance de esta tesis, como es el caso del *interpretable machine learning* [25], tratan de solventar.

Otra familia de arquitecturas con un gran potencial es la de los *deep autoencoders* o autoencoders profundos [26]. Estos modelos son entrenados para reproducir a su salida la misma información que reciben a la entrada, incorporando ciertas restricciones que impiden que se produzca una copia directa de los datos de entrada. La restricción más común consiste en incluir en la arquitectura una capa intermedia de baja dimensión, que actúe como cuello de botella, de forma que el modelo se vea forzado a aprender una representación compacta y con significado, que capture la estructura de los datos y preserve la información relevante en los mismos. Dada la baja dimensionalidad de esta capa intermedia —conocida como *espacio latente*— los *deep autoencoders* han sido ampliamente empleados en aplicaciones de reducción de la dimensión [27]. También han surgido variantes, como los *variational autoencoders (VAEs)* que, al aprender una distribución de probabilidad del espacio latente, han transformado estas arquitecturas en modelos generativos, con un gran éxito en la generación de datos sintéticos [28]. No obstante, cabe mencionar el potencial del VAE en otras aplicaciones, como la detección de anomalías [29] o el análisis semántico de datos [30], que podrían tener un impacto directo en el ámbito de la ingeniería.

El VAE tiene la capacidad de aprender la distribución de probabilidad de los datos de entrenamiento, con lo que podría llegar a modelar fielmente el funcionamiento de un proceso, tan solo siendo entrenado con datos de operación del mismo. En el ámbito de la ingeniería, esta arquitectura podría ser utilizada para modelar el comportamiento normal de los sistemas, convirtiéndose en una valiosa herramienta para la monitorización de la salud en procesos e instalaciones, cuyo potencial será explorado a lo largo de esta tesis. Uno de los aspectos a considerar será la interpretabilidad de los residuos del VAE, que podrían ser portadores de valiosa información acerca del estado de los sistemas, siendo útiles, por ejemplo, en la detección de comportamientos anómalos, donde los *deep autoencoders* ya han obtenido resultados prometedores [31]. De igual manera, se explorará la semántica del espacio latente del VAE. En trabajos como [32, 33] se han encontrado direcciones en los espacios latentes, llamadas *vectores de atributos*, que capturan conceptos abstractos con significado acerca de los datos —como direcciones de *sonrisa* o *sexo*, en el caso de aplicaciones de reconocimiento facial— y que, en problemas de ingeniería, podrían revelar direcciones de fallo en los sistemas. En consecuencia, otra línea de interés será el estudio del potencial de estos espacios latentes en la generación de indicadores de salud *(Health Indicators, HIs)* de los procesos. Dadas las buenas propiedades de los espacios latentes para capturar la información relevante en los datos y descartar aquella redundante o superflua, estos podrían proporcionar HIs más precisos que los construidos en el espacio original de los datos, como ya apuntan algunos trabajos en la literatura [34].

En último lugar, cabe destacar las aplicaciones de los *deep autoencoders* como técnicas de reducción de la dimensión, que serán también objeto de estudio de esta tesis. Cuando el espacio latente tiene dos dimensiones (2D), este se convierte en un mapa visual del sistema bajo estudio, en el que cada estado del sistema tiene asociada una región diferente del espacio [35]. Estos mapas permiten monitorizar visualmente el comportamiento de los procesos, proporcionando información al usuario de una forma intuitiva. Por tanto, la integración de los espacios latentes en herramientas de visualización de datos, que faciliten su exploración interactiva, podría dar lugar a poderosas herramientas de analítica visual, con potenciales contribuciones en la monitorización de la salud de los sistemas así como en la mejora de la comprensión de los mismos. Estos enfoques podrían ser además fácilmente extrapolables a otros ámbitos con problemáticas similares, como el de la biomedicina, donde la monitorización de la condición de los procesos también es crítica y la visualización de mapas de estado de los procesos podría resultar de gran utilidad. En los próximos años, se espera que el aprendizaje profundo tenga un gran impacto tanto en el ámbito biomédico como en el de los sistemas de ingeniería, con lo que esta transferencia de conocimientos entre ambos campos representa otra interesante línea de estudio [36].

La disponibilidad (o, en su defecto, la ausencia) de datos etiquetados será otro aspecto a considerar en esta investigación, pues se trata de un factor clave que condicionará la estrategia a seguir en el abordaje del problema objetivo. En este contexto, se distinguen habitualmente dos tipos de escenarios o estrategias de aprendizaje diferentes: aprendizaje supervisado y aprendizaje no supervisado. Los modelos de aprendizaje supervisado —utilizados típicamente en problemas de clasificación y predicción— requieren la disponibilidad de datos etiquetados (cada

muestra de trabajo debe tener asociada una etiqueta o valor objetivo); de esta manera, el aprendizaje del modelo consistirá en un proceso iterativo en el que las etiquetas o salidas esperadas para cada muestra serán comparadas con las salidas del modelo y los pesos del mismo serán ajustados hasta que la comparativa arroje resultados satisfactorios. En cambio, los modelos de aprendizaje no supervisado —utilizados en problemas de agrupación o *clustering* o, también, en detección de anomalías— no requieren muestras etiquetadas; el modelo será ajustado de forma iterativa hasta arrojar una agrupación satisfactoria de los datos, de forma que las muestras de cada grupo sean similares entre sí y, al mismo tiempo, distintas a las de otros grupos. Ambos escenarios son comunes en problemas de ingeniería y serán explorados a lo largo de esta tesis.

En definitiva, gracias a la madurez de la tecnología, los sistemas de ingeniería gozan de una alta disponibilidad de datos y requieren de técnicas capaces de extraer —a partir de estos datos— la máxima información útil de los procesos. Con esta información, sería posible mejorar la productividad y eficiencia de los procesos, reducir costes de producción o aumentar la seguridad en las instalaciones. En este contexto, destaca el potencial de las técnicas *deep learning*, que están llamadas a transformar el análisis de datos en el ámbito de la ingeniería, al igual que han hecho ya en otros campos de aplicación. Ante esta situación, se propone explorar el rendimiento de arquitecturas profundas bien conocidas —como redes convolucionales o *deep autoencoders*— en un amplio rango de problemas de ingeniería —clasificación, detección de anomalías, generación de indicadores de salud, etc.— y evaluar con ello sus posibles contribuciones en este campo.

1.3. Formulación del problema

El problema central planteado en esta tesis consiste en investigar las posibilidades de aplicación de las técnicas *deep learning* en el análisis y mejora de la eficiencia en sistemas de ingeniería. Para abordarlo, se propone explorar el uso de distintos tipos de redes profundas sobre diferentes problemas y contextos de ingeniería, y valorar la calidad de los resultados obtenidos. Este objetivo global se divide en otros más concretos, expuestos a continuación.

- **Objetivo 1: Clasificación.**

  El primer objetivo consiste en la exploración del uso de arquitecturas profundas para la clasificación del estado de funcionamiento de los procesos. Para abordar este objetivo, se estudiarán dos tipos de arquitecturas profundas —redes *feedforward* o prealimentadas, que son las arquitecturas estándar en la literatura, y redes convolucionales— y se valorará su rendimiento en la detección de fallos en motores.

- **Objetivo 2: Detección de anomalías.**

  El segundo objetivo consiste en la exploración del uso de arquitecturas profundas para la detección de anomalías en procesos de ingeniería. Para abordar este objetivo, se estudiará un enfoque de redundancia analítica, que consistirá en el uso de *deep autoencoders* en combinación con técnicas de análisis

de residuos, para la detección de comportamientos anómalos en diferentes contextos de ingeniería, como la operación de motores, sistemas hidráulicos o sistemas de monitorización del movimiento humano.

- **Objetivo 3: Generación de indicadores de salud.**

  El tercer objetivo consiste en la exploración del uso de arquitecturas profundas para la generación de indicadores de salud de los procesos. Para abordar este objetivo, se estudiarán las representaciones de los datos disponibles en las capas ocultas de los *deep autoencoders* y su potencial uso como indicadores de salud de los procesos. En detalle, se analizará su rendimiento como indicadores de degradación de las máquinas.

- **Objetivo 4: Visualización de mapas de estados de los procesos.**

  El cuarto objetivo consiste en la exploración del uso de arquitecturas profundas para la generación de visualizaciones interpretables de los procesos. Para abordar este objetivo, se estudiará la generación de proyecciones de baja dimensión de los datos —en concreto, de dos dimensiones (2D)— mediante el uso de *deep autoencoders*. Estas proyecciones, en combinación con técnicas de analítica visual, permitirán crear mapas visuales e interpretables de los procesos, cuya contribución será evaluada en el análisis del consumo energético de grandes instalaciones.

- **Objetivo 5: Conexiones con otros ámbitos.**

  El quinto objetivo consiste en el estudio de la aplicabilidad de estos enfoques en otros ámbitos con problemáticas similares, como es el caso de la biomedicina, donde la monitorización de la condición de los procesos también es crítica. Para abordar este objetivo, se estudiarán las conexiones existentes entre el ámbito industrial y el biomédico, tratando de explorar las posibles contribuciones de nuestros enfoques en el contexto de los procesos biomédicos. En detalle, se estudiará la generación de mapas de estados para el análisis y mejora de la comprensión de los mecanismos vinculados con la propagación del cáncer.

- **Objetivo 6: Extracción automática de características.**

  El último objetivo consiste en la exploración de los descriptores de proceso extraídos de forma automática por las arquitecturas profundas, en busca de nuevo conocimiento acerca de los procesos (así como información que permita contrastar el conocimiento disponible a priori de los mismos). Este objetivo será abordado en conjunto con los anteriores, visualizando parámetros internos de las arquitecturas generadas, como los filtros de las arquitecturas convolucionales (Objetivo 1) o los residuos de los *deep autoencoders* (Objetivo 2). Además, la extracción automática de características es intrínseca tanto a la generación de indicadores de salud de los procesos (Objetivo 3) como a la generación de mapas de estados de los mismos (Objetivos 4 y 5), pues ambos casos son ejemplos de características extraídas de forma automática por los modelos a partir de datos del proceso.

1.4. Estructura del documento

Este documento está estructurado en varios capítulos, a lo largo de los cuales abordaremos los objetivos expuestos en la sección anterior. En detalle, en el **Capítulo 2** se presenta el contexto histórico de las técnicas *deep learning* y se exponen los cuatro tipos de arquitecturas profundas empleadas a lo largo de esta tesis (redes *feedforward*, redes convolucionales, *deep autoencoders* y *variational autoencoders*). A continuación, en los **Capítulos 3 al 6** se expone el uso de estas arquitecturas para abordar los Objetivos 1 al 5, mientras que el Objetivo 6 será transversal a todos ellos (Figura 1.2). Finalmente, en el **Capítulo 7** expondremos las conclusiones de esta investigación y posibles líneas de trabajo futuro.

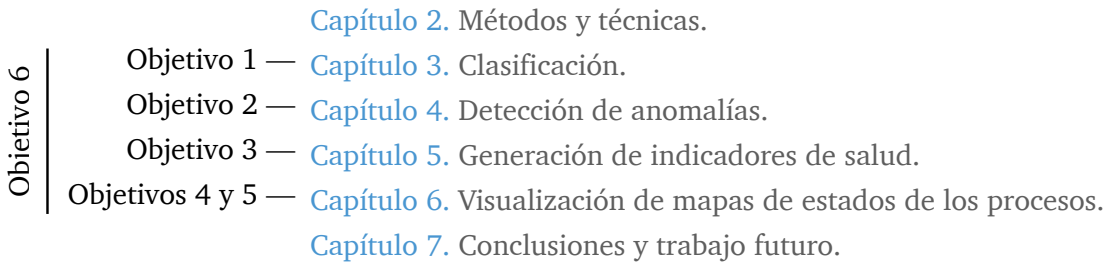


Figura 1.2: Correspondencia entre objetivos y capítulos de esta tesis.

CAPÍTULO 2

Métodos y técnicas

En este capítulo se presentan las técnicas de aprendizaje profundo, comenzando con una revisión de su evolución, desde sus inicios hasta nuestros días. A continuación, se describen los fundamentos de cuatro arquitecturas profundas ampliamente empleadas en la literatura —redes *feedforward*, redes convolucionales, *deep autoencoders* y *variational autoencoders*— que serán las herramientas utilizadas a lo largo de esta investigación.

2.1. Contexto histórico

Las técnicas de aprendizaje profundo constituyen una de las ramas más exitosas del aprendizaje automático que, a su vez, forma parte de una disciplina mucho más amplia, conocida como inteligencia artificial (Figura 2.1).

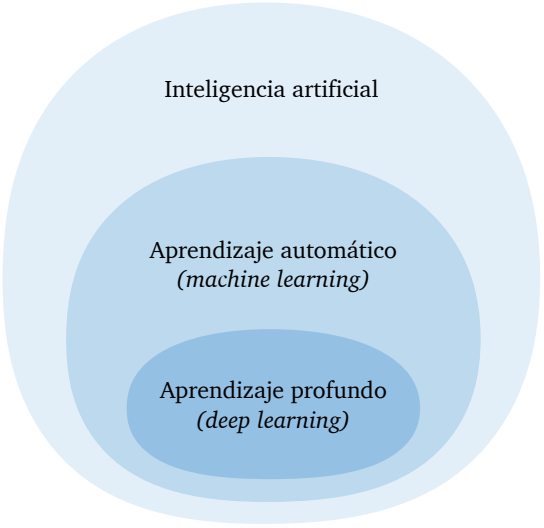


Figura 2.1: Jerarquía e interacción entre inteligencia artificial, aprendizaje automático y aprendizaje profundo.

El nacimiento de la inteligencia artificial se sitúa en el verano de 1956, cuando algunos de los profesionales más brillantes de la época —Marvin Minsky, Claude Shannon, Nathaniel Rochester, John McCarthy, etc.— se reunieron en el Dartmouth College para estudiar el incipiente mundo de los ordenadores y su potencial capacidad para exhibir un comportamiento inteligente. A lo largo de varias semanas, debatieron sobre cuestiones relacionadas con el procesamiento del lenguaje natural, el aprendizaje a partir del ejemplo, la arbitrariedad y la creatividad, o la toma de decisiones. Como resultado, este curso de verano sentó las bases de un nuevo campo de investigación, la inteligencia ar-

tificial, que desde su nacimiento hasta la actualidad ha tratado de replicar —en las máquinas— habilidades típicamente humanas, como son el uso del lenguaje, el aprendizaje o el razonamiento creativo.

Aunque en sus inicios la inteligencia artificial se perfilaba como una disciplina compleja y abstracta, habitualmente ligada a grandes centros de investigación, su uso a lo largo de los últimos años se ha popularizado de forma masiva y sus aplicaciones se han extendido a la escena cotidiana, estando presentes en nuestros móviles, tabletas u ordenadores personales. Este auge ha venido impulsado por dos factores fundamentales: la elevada potencia de cálculo de los computadores, que ha permitido afrontar el alto coste computacional de las técnicas basadas en inteligencia artificial; y la enorme cantidad de información almacenada digitalmente en la actualidad, que ha hecho surgir la demanda de este tipo de técnicas para el procesamiento y análisis automático de datos.

En este contexto se enmarca una de las ramas más destacadas de la inteligencia artificial, inspirada en la capacidad humana para aprender a partir del ejemplo: el aprendizaje automático o *machine learning*. Las técnicas de *machine learning* proponen un modelado analítico y automático de los datos, que les permite identificar patrones recurrentes en los mismos. Para ello, abordan el análisis como un proceso de aprendizaje: el programador proporciona una serie de reglas de partida que el algoritmo de aprendizaje ha de ir adaptando, o también, creando otras nuevas, con el fin de mejorar la tasa de acierto del modelo generado. En consecuencia, los algoritmos de aprendizaje automático son descritos habitualmente como procesos de búsqueda, particularmente diseñados para elegir la función —de una lista de posibles funciones— que mejor explica las relaciones existentes entre las variables de un conjunto de datos. Cabe destacar que, en problemas sencillos, elegir la función adecuada puede ser una tarea fácil, incluso asequible para un humano sin necesidad de recurrir a una máquina. Sin embargo, a medida que aumenta el número de variables en los datos o el número de potenciales funciones a considerar, el espacio de búsqueda crece y la tarea puede volverse enormemente compleja. Los algoritmos de *machine learning* permiten abordar este tipo de problemas con éxito y uso se ha extendido en aplicaciones tan variadas como el filtrado de *spam*, el reconocimiento facial o los traductores inteligentes, que recurren al aprendizaje automático para, a partir de muestras de ejemplo, aprender a detectar correos no deseados, identificar caras en imágenes o traducir textos a otros idiomas.

Las funciones aprendidas por los algoritmos de *machine learning* pueden adoptar formas muy diferentes, como una simple operación aritmética, secuencias de reglas si-entonces, o representaciones mucho más complejas. Entre estas posibles representaciones se encuentran los modelos de aprendizaje profundo o *deep learning*, que en los últimos años se han convertido en uno de los subcampos más populares del aprendizaje automático. Las técnicas *deep learning* proponen modelar abstracciones de alto nivel de los datos, empleando para ello arquitecturas compuestas por un elevado número de capas de transformaciones que pueden ser tanto lineales como no lineales. Se trata de una idea inspirada en la arquitectura y funcionamiento del cerebro humano y, por ello, estos modelos reciben también el nombre de redes neuronales artificiales (RNAs).

A continuación, se presenta un recorrido por la evolución histórica del *deep*

learning, cuyos inicios se sitúan en 2006, gracias a las aportaciones de Geoffrey Hinton [1]. Sin embargo, el germen de esta disciplina se remonta a los años cuarenta, cuando el aprendizaje profundo gozaba de escasa popularidad y ni siquiera era conocido por este nombre. De hecho, este campo ha sido renombrado varias veces, reflejando la influencia de diferentes perspectivas e investigadores. Comúnmente, se distinguen tres etapas históricas: *cybernetics* (1940-1960), *connectionism* (1980-1990) y *deep learning* (2006-actualidad), expuestas a continuación.

2.1.1. *Cybernetics* (1940-1960)

A principios de los años cuarenta, los investigadores de la época comenzaron a reflexionar acerca del funcionamiento de nuestro cerebro y cómo replicarlo artificialmente. En concreto, trataron de modelar la inteligencia humana, proponiendo para ello modelos matemáticos basados en operaciones lógicas. Surgieron así los primeros sistemas computacionales inspirados en el funcionamiento del cerebro humano, que se hicieron populares bajo el nombre de Redes Neuronales Artificiales (RNAs) y cuyo desarrollo constituye el periodo histórico conocido como cibernética o *cybernetics*.

Fueron Warren McCulloch y Walter Pitts quienes, en 1943, sentaron las bases de esta nueva disciplina, al proponer por primera vez un modelo matemático y lógico de una neurona [37]. Esta neurona artificial consistía en una función aditiva, implementada mediante circuitos eléctricos, que recibía como entrada un conjunto de valores binarios. Si la suma de dichos valores superaba cierto umbral, la salida de la neurona —también binaria— se activaba, quedando inhibida en caso contrario. Estas entradas y salidas podían ser conectadas con las de otras neuronas, construyendo así una red neuronal artificial capaz de resolver operaciones lógicas como la conjunción, la disyunción o la negación (operaciones AND, OR y NOT, respectivamente). La propuesta de neurona artificial de McCulloch y Pitts tuvo un gran impacto en la literatura, a pesar de que no abordaba uno de los rasgos más distintivos del cerebro humano, que es su capacidad para aprender a partir de la experiencia.

Años después, en 1949, el neuropsicólogo Donald O. Hebb presentó sus teorías acerca de los mecanismos de aprendizaje neuronal, que hoy conocemos como *postulado de Hebb* [38]. En este postulado, Hebb asume que la experiencia no es más que información almacenada en las conexiones sinápticas entre neuronas y que el aprendizaje tiene lugar al modificar dichas conexiones. Esta teoría supuso una gran contribución en el ámbito de la cibernética, donde las conexiones entre neuronas comenzaron a ser representadas mediante pesos que, idealmente, serían actualizados iterativamente durante el proceso de aprendizaje.

En línea con estos avances, aparecieron en los años siguientes nuevos modelos computacionales entre los que destaca el *perceptrón* [39], propuesto por Frank Rosenblatt en 1958. Inspirado tanto en la red de McCulloch y Pitts, como en el trabajo de Donald O. Hebb, Rosenblatt propuso un modelo de red neuronal que incorporaba su propia regla de aprendizaje. El modelo fue utilizado para llevar a cabo reconocimiento de imágenes y consistía en un clasificador binario capaz de discriminar entre dos clases linealmente separables. Este modelo tuvo una primera

implementación en *software* para el IBM 704, aunque más tarde fue también implementado en *hardware*. El resultado fue la popular máquina Mark I Perceptron, en la que los pesos de la red estaban codificados en potenciómetros y los pesos eran actualizados durante el proceso de aprendizaje mediante motores eléctricos. Rosenblatt también demostró que, siendo la función a representar linealmente separable, el perceptrón siempre convergería a la solución del problema.

Al mismo tiempo que se desarrollaba el perceptrón, Bernard Widrow y Marcian Hoff trabajaban en una propuesta muy similar. En 1960, estos autores presentaron la red neuronal ADALINE *(ADAptative LInear NEuron)* junto con la regla de aprendizaje LMS *(Least Mean Square)* [40]. A diferencia del perceptrón, la red ADALINE no utilizaba una función umbral, sino que la salida de la red era directamente la suma ponderada de sus entradas. Gracias a esta modificación, ADALINE era capaz de resolver problemas de regresión, mientras que el uso del perceptrón estaba restringido a problemas de clasificación. En cuanto a la regla de aprendizaje, esta proponía minimizar el error cuadrático medio de la red, utilizando para ello el método de descenso del gradiente, que sigue siendo el algoritmo predominante en el entrenamiento de los modelos profundos actuales. Adicionalmente, Widrow y Hoff desarrollaron una extensión de su red, llamada MADALINE (resultado de combinar múltiples redes ADALINE), que fue utilizada como filtro adaptativo para la eliminación de ecos en las líneas telefónicas.

Con el éxito de Rosenblatt, Widrow, Hoff y otros, quedó demostrada la capacidad de las RNAs para aprender a distinguir, de forma automática, patrones en los datos. Esto generó gran expectación y las redes neuronales se vieron envueltas en una época dorada, que terminó en 1969. Ese año, Marvin Minsky y Seymour Papert publicaron su escepticismo en un libro [41] que para muchos significó el fin de las RNAs. En él expusieron un análisis detallado de las limitaciones que presentaban los modelos utilizados hasta el momento. Destacaron, por ejemplo, la incapacidad del perceptrón para aprender la sencilla función lógica XOR, que era inabordable al tratarse de un problema no lineal. En definitiva, pusieron en evidencia a estas redes, retratándolas como meros juguetes matemáticos, sin aplicabilidad práctica real.

Conscientes de estas limitaciones, autores como Rosenblatt o Widrow, llegaron a proponer redes más complejas, que serían capaces de resolver problemas no lineales. En concreto, se trataba de modelos constituidos por varias capas de neuronas que, aunque serían el germen de los modelos profundos actuales, no llegaron a implementarse, pues en aquel momento las reglas de aprendizaje estaban diseñadas para manejar una sola capa de neuronas. Incluso, Minsky y Papert trataron este enfoque multicapa en su libro, tildándolo de estéril. Ante esta perspectiva y dadas las limitaciones computacionales de la época, muchos investigadores abandonaron el estudio de las RNAs, que estuvo suspendido durante una década, a lo largo del periodo conocido como *invierno de la Inteligencia Artificial*.

No obstante, cabe destacar que la primera red profunda de la historia fue propuesta durante este *invierno*. Fue en 1971, cuando Alexey Ivakhnenko presentó el algoritmo de aprendizaje GMDH *(Group Method for Data Handling)* y lo utilizó para entrenar una red de ocho capas [42]. Sin embargo, el libro de Minsky y Papert tuvo tal impacto en la comunidad científica, que este éxito pasó desaper-

cibido y quedó prácticamente oculto con el paso de los años. Tanto es así que el método GMDH apenas es conocido en la literatura. En su lugar, el método de retropropagación o *backpropagation* —cuya aparición marca el inicio de la siguiente etapa histórica— se ha convertido en el algoritmo de entrenamiento estándar de los modelos profundos.

2.1.2. *Connectionism* (1980-1990)

A comienzos de los años ochenta el campo de las RNAs se vio activado de nuevo. Las críticas sufridas en las décadas anteriores fueron tachadas de severas y nuevos trabajos hicieron resurgir la ilusión en el potencial de las arquitecturas neuronales. Se inició así la era del conexionismo o *connectionism*, en la que se demostró que unidades computacionales simples (neuronas) eran capaces de reproducir comportamientos inteligentes cuando estas eran conectadas en red.

Entre los trabajos que reactivaron el interés en las RNAs, se encuentran las aportaciones de John Hopfield, que en 1982 propuso una red recurrente para ser empleada como memoria asociativa [43]. Esta red era capaz de memorizar un cierto número de patrones y reproducirlos ante nuevas entradas, permitiendo así completar o corregir patrones corruptos. Otro desarrollo clave en el resurgir de las RNAs fue el algoritmo de retropropagación o *backpropagation*, que hizo posible el entrenamiento de redes con múltiples capas y que, con ello, abriría paso a la resolución de problemas no lineales. La primera descripción de este algoritmo fue presentada en la tesis doctoral de Paul Werbos en 1974 [44]. No obstante, fueron Rumelhart, Hinton y Williams quienes hicieron popular este enfoque en 1986, al publicar un libro en el que hacían una revisión de la investigación en redes neuronales, dedicando todo un capítulo a la descripción del algoritmo de *backpropagation* [45].

Este algoritmo, que sigue siendo el enfoque dominante en el entrenamiento de las redes profundas, propone abordar el proceso de aprendizaje en dos pasos: en el primer paso, se presentan las muestras de entrenamiento ante la red y se calcula el error cometido por la misma (comparando la salida de la red con la salida esperada para cada muestra de entrenamiento); en el segundo paso, el error cometido es propagado hacia atrás, desde la capa de salida hacia las capas intermedias, asignando a cada neurona una fracción de la señal de error y describiendo así su contribución relativa al error total de la red; en último lugar, estos términos de error son empleados para la actualización de los pesos del modelo. En detalle, cada peso será actualizado de forma proporcional, y en dirección opuesta, al gradiente de la función de error con respecto a dicho peso. Dado que el cómputo de este gradiente es una tarea compleja, el algoritmo de *backpropagation* lo aproxima como una combinación entre las activaciones y los términos de error de las neuronas. Este procedimiento de actualización de los pesos (expuesto en detalle en la Sección 2.3) es repetido de forma iterativa hasta que el error de la red alcanza un rango aceptable y el aprendizaje se considera, por tanto, completado. Cabe destacar también que la contribución de cada neurona al error total de la red se estima mediante el cálculo de derivadas parciales en cada punto de la red y esta operación solo es posible si las funciones de activación de las neuronas son diferenciables. Ante esta circunstancia, las funciones de tipo umbral empleadas hasta

el momento comenzaron a ser sustituidas por funciones diferenciables, como la sigmoide o la tangente hiperbólica.

El algoritmo de *backpropagation* también sufría ciertas limitaciones, como el problema de desvanecimiento del gradiente o *vanishing gradient problem*, que fue identificado por Sepp Hochreiter en 1991 [46]. Hochreiter apuntó que el mecanismo de actualización de los pesos —el cual depende no solo de la derivada en cada punto de la red, sino también del producto de las derivadas de las capas posteriores— puede ocasionar que los gradientes menores que uno, al pasar por múltiples capas, se hagan cada vez más pequeños. En esos casos, los pesos del modelo apenas se modifican y el aprendizaje puede volverse extremadamente lento. Este fenómeno afecta especialmente a las primeras capas de la arquitectura, que son vitales para el éxito de la red, pues son las encargadas de extraer las características de los datos de entrada que alimentarán a las capas posteriores. Esta limitación, y otras descubiertas más adelante, son objeto de estudio aún hoy, cuando los investigadores trabajan en proponer nuevas funciones de activación y variantes del mecanismo de aprendizaje que ayuden a mitigar estos efectos.

A pesar de sus limitaciones, el algoritmo de *backpropagation* representa la contribución más relevante de la era del conexionismo. Su éxito transformó el futuro de las RNAs y abrió la puerta a un gran número de investigaciones que continúan hasta nuestros días. Este nuevo enfoque facilitó la aparición de redes multicapa, como los perceptrones multicapa *(Multi-Layer Perceptron, MLP)*, que siguen siendo una de las arquitecturas más empleadas en la actualidad. Al dotar a las redes de varias capas, estas se convirtieron en estructuras complejas, capaces de resolver problemas no lineales. Se trata de la materialización del principio de conexionismo, que da nombre a esta etapa histórica y, según el cual, la inteligencia emerge de la interacción entre múltiples unidades computacionales simples. Como consecuencia, las RNAs estructuran la información de entrada de manera distribuida y jerárquica en capas de diferente nivel de abstracción [47], de manera que cada capa extrae características basadas en aquellas extraídas por las capas anteriores. Esta organización jerárquica se ha convertido en un rasgo característico de las arquitecturas profundas que, a través de representaciones intermedias de los datos, consiguen llegar a aprender complejas relaciones existentes entre las señales de entrada y salida de la red.

Otra gran aportación de esta década fue el *neocognitrón* [48], propuesto por Kunihiko Fukushima en 1982. Dos décadas antes, Hubel y Wiesel estudiaron el funcionamiento de la corteza visual y descubrieron diferentes tipos de células —simples y complejas— que, conectadas entre sí de forma jerárquica, procesaban las imágenes desde un nivel de abstracción más bajo a otro más alto [49]. Inspirado en este trabajo, Fukushima propuso una arquitectura de red multinivel, particularmente diseñada para el reconocimiento de caracteres en imágenes y que sería la precursora de las redes convolucionales. Estaba constituida por capas convolucionales y de submuestreo, que imitaban la jerarquía de conexiones de las células del sistema nervioso visual.

Poco después, en 1989, Yann LeCun presentó la primera red convolucional *(Convolutional Neural Network, CNN)* [50]. Esta arquitectura heredó gran parte del diseño del neocognitrón, pero LeCun aportó también el desarrollo necesario

para entrenar este tipo de redes utilizando el mecanismo de *backpropagation*. Entre sus trabajos se encuentra la famosa LeNet-5 [51], una arquitectura convolucional de siete capas, empleada en algunos bancos de Estados Unidos para reconocer de manera automática los dígitos manuscritos en los cheques bancarios. Desde entonces, las CNNs han demostrado su éxito en una amplia variedad de aplicaciones, especialmente en el ámbito del procesamiento de imagen, y se han convertido en una de las arquitecturas más populares del aprendizaje profundo.

Otra arquitectura ampliamente reconocida en el campo de las RNAs es la de las redes recurrentes *(Recurrent Neural Networks, RNNs)*, que fueron concebidas para el análisis de datos secuenciales en problemas de procesamiento del lenguaje. Llegado este punto, cabe destacar que el resto de redes presentadas hasta el momento son conocidas como *feedforward*, en las que la información fluye *hacia delante*, de forma que las neuronas de cada capa están conectadas con las neuronas de la capa posterior. Sin embargo, las redes recurrentes implementan conexiones arbitrarias entre todas sus neuronas, lo que permite incorporar el concepto de temporalidad y dotar así a las redes de memoria. Entre los primeros trabajos propuestos en la literatura, se encuentra en 1990 la red recurrente de Elman [52], que era capaz de predecir la palabra final en expresiones simples de dos y tres palabras. Para ello, la red fue entrenada con un pequeño conjunto de datos, que contenía expresiones sencillas y un léxico aproximado de veinte palabras; una vez finalizado el entrenamiento, la red era capaz de generar expresiones plausibles. Aunque se tratase de un caso de estudio sencillo, esta contribución tuvo una gran repercusión, pues sirvió para demostrar la potencial capacidad de las RNNs para aprender por sí mismas las reglas de la gramática y de representación del lenguaje, sin necesidad de ninguna enseñanza o codificación explícita previa.

A pesar de que las RNNs arrojaban un buen rendimiento en el manejo de datos secuenciales, su memoria se veía especialmente afectada por los problemas de desvanecimiento del gradiente. Ante esta situación, Hochreiter propuso en 1997 las redes de memoria a largo y corto plazo *(Long Short-Term Memory, LSTM)* [53]. Gracias a una compleja arquitectura, las redes LSTM eran capaces de aprender dependencias a largo plazo —interacciones entre elementos de una secuencia separados por dos o más posiciones— y con ello se convirtieron en la herramienta predominante en aplicaciones de procesamiento del lenguaje natural como son, por ejemplo, los sistemas de traducción automática.

Tras años de contribuciones, el avance de las RNAs se vio pausado de nuevo a finales de los años noventa. Aunque ya habían aparecido los ingredientes que propiciarían la revolución de las arquitecturas profundas —algoritmo de *backpropagation*, CNNs, RNNs—, estas arquitecturas estaban aún lejos de materializarse, pues los problemas de desvanecimiento del gradiente dificultaban la implementación de modelos con un elevado número de capas. Ante estas limitaciones, la enorme expectación generada en torno a las RNAs se tornó, al igual que había ocurrido en los años sesenta, en decepción. Al mismo tiempo, influyó el éxito de otros enfoques, como las Máquinas de Vectores de Soporte *(Support Vector Machines, SVMs)* [54] que arrojaban resultados similares a los de las redes neuronales y eran mucho más fáciles de entrenar. La unión de todos estos factores dio inicio a una etapa de decadencia, que terminaría pocos años después con la aparición del aprendizaje profundo.

2.1.3. *Deep learning* (2006-actualidad)

Tras la era del conexionismo, la investigación en RNAs no fue abandonada por completo. Una muestra de ello es la histórica derrota del campeón del mundo de ajedrez Garry Kaspárov ante DeepBlue en 1997 [4]. También destacan iniciativas como el programa NCAP *(Neural Computation & Adaptive Perception)*, puesto en marcha por el CIFAR *(Canadian Institute For Advanced Research)* en 2004 y que pretendía mantener vivo el estudio de las RNAs. Cabe mencionar que este programa reunió a los investigadores más notables de la etapa histórica que estaba por comenzar: Geoffrey Hinton, Yoshua Bengio y Yann LeCun, cuyas contribuciones al ámbito del *deep learning* les harían merecedores en 2019 del Premio ACM AM Turing y, más recientemente, junto a Demis Hassabis, del Premio Princesa de Asturias de Investigación Científica y Técnica 2022.

A mediados de la década de los 2000, coincidiendo con la creación del programa NCAP, el interés por las RNAs creció de nuevo. Diferentes avances demostraron que el problema de desvanecimiento del gradiente podría ser superado y empezó a popularizarse el término de aprendizaje profundo, haciendo así énfasis en que las nuevas RNAs eran mucho más *profundas* (tenían un mayor número de capas) que las empleadas en épocas anteriores. En concreto, el inicio de la era del *deep learning* se sitúa en 2006, cuando Geoffrey Hinton propuso el algoritmo de entrenamiento *greedy layer-wise* [1]. Este enfoque consistía en un entrenamiento por capas, en el que cada capa de la red era entrenada por separado. Una vez entrenadas todas las capas, la arquitectura obtenida era sometida a un nuevo entrenamiento, en este caso global, para una sintonización fina de los pesos de la red, con la que se daba por completado el proceso de aprendizaje [55]. Hinton demostró que este pre-entrenamiento atenuaba las limitaciones del mecanismo de *backpropagation* y que, gracias a él, sería posible entrenar modelos con un elevado número de capas, lo que atraería la atención de la comunidad científica de nuevo hacia las RNAs.

Con el paso del tiempo, el algoritmo *greedy layer-wise* fue reemplazado por otras estrategias más eficientes, pero su filosofía siguió vigente. Con su propuesta, Hinton demostró que una inicialización conveniente de los pesos facilitaba el proceso de entrenamiento y, desde entonces, la inicialización de los pesos ha sido considerada un factor clave para el éxito de las redes neuronales. Aunque los principios que rigen la relación entre los valores iniciales de los pesos y la convergencia del entrenamiento son aún desconocidos en la actualidad, existen en la literatura variadas estrategias de inicialización cuyo éxito ha sido demostrado empíricamente. Entre ellas, destaca el esquema de inicialización presentado por Glorot y Bengio en 2010 [56], que trata de equilibrar la varianza de los gradientes de las diferentes capas de la red, de manera que el entrenamiento sea lo más uniforme posible a lo largo de las capas. Para ello, los autores proponen inicializar los pesos de forma aleatoria utilizando una distribución gaussiana, con media cero y varianza dependiente del número de entradas y salidas de cada neurona.

En esta época, también se constató que una elección oportuna de las funciones de activación sería otro factor clave para el éxito del entrenamiento. Sin embargo, las funciones empleadas hasta el momento —sigmoide y tangente hiperbólica— tenían ciertas particularidades que hacían empeorar los problemas de desvaneci-

miento del gradiente: las derivadas de estas funciones eran pequeñas (acotadas en el rango $[0, 0.25]$ para el caso de la sigmoide y en el rango $[0, 1]$ para la tangente hiperbólica), siendo prácticamente nulas en los extremos (donde ambas funciones son casi planas). Ante esta situación, Glorot propuso en 2011 la función de activación lineal rectificada *(Rectified Linear Unit, ReLU)* [57], que tiene una derivada de valor 1 ante entradas positivas, lo cual permite que los gradientes fluyan con facilidad durante el proceso de retropropagación del error. Dadas sus buenas propiedades, la función ReLU mejoró notablemente el entrenamiento de los modelos profundos y se ha convertido en la elección estándar en la configuración de las redes profundas. No obstante, también presenta desventajas: su derivada es cero ante entradas negativas, lo que puede dar lugar a neuronas inactivas durante el entrenamiento. Para solventar este problema, conocido como *dying ReLU*, han surgido variantes que proponen modificaciones en la parte negativa de la función, de manera que el gradiente sea distinto de cero (Leaky ReLU [58], PReLU [59], ELU [60], etc.). A pesar del buen rendimiento de estas variantes, la función ReLU ha tenido una gran implantación en la literatura y continúa siendo la opción más extendida [61].

Tanto los métodos de inicialización de los pesos como la aparición de nuevas funciones de activación contribuyeron al crecimiento de las redes profundas, pero el auge del *deep learning* no surgió tan solo de estas mejoras en los algoritmos, sino también de los avances en otras disciplinas. Por un lado, las Unidades de Procesamiento Gráfico *(Graphics Processing Units, GPUs)*, que aparecieron a finales de la década de los 2000, incrementaron la capacidad de cómputo de los ordenadores y, con ello, permitieron acelerar los procesos de entrenamiento, facilitando enormemente el desarrollo de las RNAs. Por otro lado, la disponibilidad de cantidades masivas de datos —gracias al crecimiento de Internet, la proliferación de dispositivos inteligentes o el desarrollo del Internet de las cosas *(Internet of Things, IoT)*— permitió explotar al máximo el potencial de los modelos profundos, cuyo rendimiento crece al aumentar el volumen de datos de entrenamiento. Además, estos grandes conjuntos de datos demandan el uso de modelos cada vez más complejos, que sean capaces de extraer la información contenida en los mismos; la combinación de un gran número de datos y modelos más complejos genera la necesidad de mejoras en la potencia de cálculo de las máquinas; y se genera así un ciclo virtuoso que llega hasta nuestros días y que explica el crecimiento exponencial que ha vivido el aprendizaje profundo en la última década.

Con la madurez de la tecnología y la riqueza de datos, las redes profundas empezaron a demostrar resultados muy superiores a los obtenidos hasta el momento con otras técnicas de aprendizaje automático y pronto se convirtieron en una de las ramas más destacadas de la inteligencia artificial. A lo largo de los últimos años, estos modelos han mostrado un rendimiento sorprendente y sus resultados se han convertido en la medida de lo que una *máquina inteligente* puede llegar a hacer. Destaca, por ejemplo, la arquitectura Alexnet [62], que en 2012 ganó la competición *ImageNet Large-Scale Visual Recognition Challenge (ILSVRC)*. Esta red CNN fue entrenada con 1.2 millones de imágenes para resolver un problema de clasificación de 1000 clases, consiguiendo mejores resultados —con un amplio margen— que cualquier otra técnica de clasificación. Desde entonces, redes CNN aún más competitivas han seguido liderando la competición, como ZFNet,

GoogLeNet, VGGNet o ResNet [63]. En consecuencia, el aprendizaje profundo se ha consolidado como una de las herramientas más precisas en reconocimiento de objetos y está presente en una amplia variedad de aplicaciones, como sistemas de ayuda a la conducción, equipos de rescate, sistemas de vigilancia o robots autónomos [24]. En la actualidad, el uso de las redes convolucionales ha llegado incluso al ámbito artístico, por ejemplo, en proyectos de transferencia de estilo como [64], donde las CNNs son utilizadas para trasladar el estilo pictórico de una obra al contenido de cualquier imagen de entrada.

De igual manera, destaca el rendimiento de las redes RNN, con excelentes resultados en aplicaciones de procesamiento del lenguaje natural *(Natural Language Processing, NLP)*. Muestra de ello son los modelos *seq2seq* [65], que fueron introducidos en 2014 y se han convertido en la arquitectura estándar en sistemas de traducción inteligente. Al mismo tiempo, los modelos *word2vec* [66] demostraron ser capaces de captar la semántica de las palabras y han sido empleados en sistemas de recomendación, clasificación de documentos o análisis de sentimientos. Más recientemente, han aparecido en la literatura modelos híbridos, que combinan la arquitectura de las redes CNN y RNN, en busca de una mayor versatilidad y eficiencia, con aplicaciones en escenarios diversos que van desde la descripción automática de imágenes [67] hasta la clasificación de llantos de bebés [68].

En esta época, también se extendió el uso de otras arquitecturas relevantes, como los autoencoders profundos *(deep autoencoders)*. Estos modelos son entrenados para reproducir a su salida la misma información que reciben a la entrada, e incluyen ciertas restricciones de diseño que impiden que se produzca una simple copia de los datos de entrada. Una restricción habitual consiste en incluir una capa intermedia en la red, que tenga un menor número de neuronas que las capas de entrada y salida. Gracias a este cuello de botella, el autoencoder es forzado a aprender una representación intermedia o *latente* de los datos, que capturará las características más relevantes de la entrada y descartará información redundante o superflua, lo que convierte a los autoencoders en modelos especialmente útiles en problemas de compresión de datos [26]. Con el tiempo, han surgido variantes de esta arquitectura, como los *denoising autoencoders (DAEs)*, que son capaces de eliminar el ruido de las señales de entrada [69]; o los *variational autoencoders (VAEs)* que, al aprender una distribución de probabilidad del espacio latente, se comportan como modelos generativos y han sido utilizados para generar imágenes realistas de rostros humanos [28], crear fragmentos musicales [70] o diseñar nuevos compuestos químicos [71]. Además del VAE, existen otras arquitecturas generativas en la literatura, entre las que destacan las redes GAN *(Generative Adversarial Networks)* [72], actualmente muy populares por su habilidad para crear *deep fakes* o vídeos manipulados [73, 74].

En definitiva, las técnicas de aprendizaje profundo han experimentado un avance sin precedentes en los últimos años, que explica la riqueza de arquitecturas presentes en la literatura (CNNs, RNNs, LSTMs, VAEs, GANs, etc.). Aunque cada una de estas arquitecturas está especializada en la resolución de un tipo de tarea en concreto, la creciente popularidad de los modelos híbridos ha provocado que muchas de ellas hayan trascendido sus fronteras y hoy podamos encontrar, por ejemplo, redes RNN en aplicaciones de procesamiento de imagen [75] o redes CNN en enfoques de procesamiento del lenguaje natural [76]. Tanto el diseño de

nuevas arquitecturas, como la búsqueda de nuevos usos para las ya existentes, son objeto de estudio en la actualidad. Incluso, en ocasiones son las propias técnicas *deep learning* quienes se abren paso hacia nuevos nichos de aplicación. Muestra de ello es el caso de los *deep fakes* que, dado su realismo, han llegado a ocasionar problemas de fraude y los modelos profundos son ahora utilizados no solo para crear *deep fakes*, sino también para detectarlos [77].

Entre las líneas de trabajo actuales, destaca el estudio de nuevas funciones de activación. Como se ha expuesto a lo largo de esta sección, la evolución del *deep learning* ha ido ligada a la de estas funciones, que son elementos críticos en los modelos, pues en ellas reside su capacidad para aprender transformaciones no lineales de los datos. Las primeras RNAs utilizaban funciones de tipo umbral, que después serían sustituidas por sigmoides y tangentes hiperbólicas, y estas a su vez serían reemplazadas por la función ReLU. En la actualidad, se investiga en nuevas funciones de activación, con mejores propiedades para la retropropagación del error que la función ReLU [78]. Otra línea de trabajo en auge es la del aumento de datos o *data augmentation*. Dado que el rendimiento de los modelos profundos crece al aumentar el volumen de datos de entrenamiento, este enfoque propone aumentar de forma artificial el conjunto de datos disponible, creando nuevas muestras mediante transformaciones de los datos originales. Para ello, se recurre habitualmente a los modelos generativos, dada su capacidad para crear nuevas versiones plausibles de los datos [79].

Nos encontramos, por tanto, ante un campo de investigación en constante evolución, con continuos avances en una amplia variedad de áreas (arquitecturas, algoritmos de entrenamiento, disponibilidad de datos, etc.). Con ello, en esta tercera etapa histórica, el éxito del *deep learning* no solo se ha consolidado, sino que sigue creciendo a un ritmo exponencial y nuevos ámbitos, como el de los sistemas de ingeniería, podrían verse beneficiados por su enorme potencial.

2.2. Arquitectura de una red profunda

Inspiradas en el funcionamiento del cerebro humano, las arquitecturas profundas son también conocidas como Redes Neuronales Artificiales (RNAs). Se trata de modelos computacionales constituidos por un elevado número de unidades de procesamiento simple o *neuronas*, de cuya interacción emerge la capacidad de las arquitecturas profundas para modelar complejas transformaciones de los datos de entrada. En la Figura 2.2 se muestra un ejemplo de arquitectura profunda, que consta de cinco capas de neuronas: capa de entrada, capa de salida y tres capas ocultas o intermedias. Estas capas son las que dotan de profundidad a las arquitecturas, que pueden llegar a contar con decenas o incluso cientos de capas ocultas, aunque se requiere de tan solo un mínimo de dos capas ocultas para que un modelo sea considerado profundo. Dichas capas están constituidas por neuronas —representadas con círculos— las cuales reciben un conjunto de valores numéricos de entrada, que transforman en un único valor de salida. En consecuencia, las variables de salida del modelo serán las salidas de las neuronas de su última capa. En cambio, las unidades de la capa de entrada —representadas con cuadrados— se limitan a recoger las variables de entrada al modelo y no implementan ninguna

transformación de las mismas.

También se observa en la figura —representado con flechas— el flujo de información a lo largo de la red. Este flujo tiene un único sentido, apuntando siempre hacia la capa de salida, y por ello estas arquitecturas son conocidas como redes *feedforward* o prealimentadas. Cada una de estas conexiones —cada flecha— tiene asociado un peso, cuyo valor será ajustado durante el proceso de aprendizaje o entrenamiento de la red, que consistirá en la búsqueda de un conjunto de pesos óptimo en el contexto del problema objetivo. En detalle, la arquitectura será entrenada para minimizar el error cometido por el modelo[1], empleando para ello el algoritmo de descenso del gradiente [80] en combinación con el mecanismo de retropropagación del error o *backpropagation* [45]. No obstante, antes de proceder al entrenamiento del modelo es necesario definir su arquitectura, que vendrá determinada por la naturaleza de sus capas —densas, convolucionales, de submuestreo, etc.—, cuyas particularidades serán descritas a lo largo de esta sección.

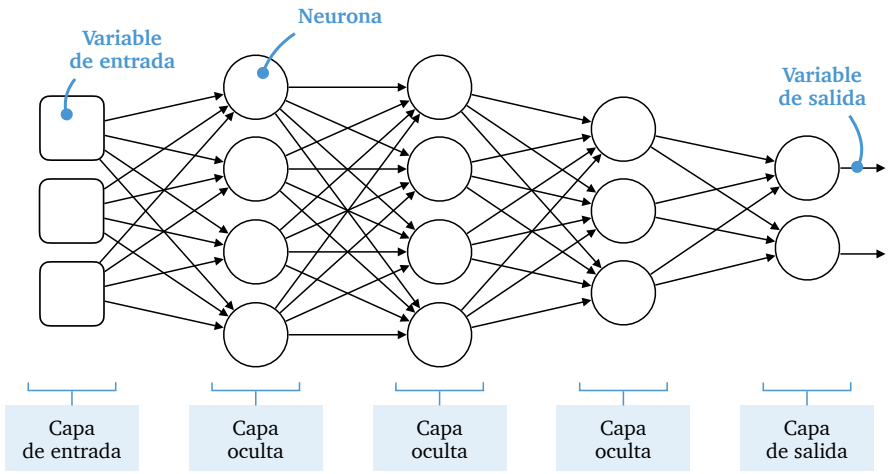


Figura 2.2: Ejemplo de arquitectura de una red profunda prealimentada o *feedforward*.

El procesamiento de información que tiene lugar en las arquitecturas profundas está basado en pequeñas transformaciones de los datos, ejecutadas en sus neuronas. Las neuronas implementan una sencilla operación de dos pasos: en primer lugar, se calcula una suma ponderada de las entradas a la neurona —ponderada según los pesos de sus conexiones— y, en segundo lugar, una función de activación mapea el resultado de dicha suma en un valor de salida o activación de la neurona. Estas funciones de activación son un elemento clave en las redes profundas, pues es en ellas —en combinación con la arquitectura jerárquica de las redes— donde reside la capacidad de los modelos profundos para capturar complejas relaciones no lineales de los datos. A lo largo de la historia se han empleado funciones de activación de tipo umbral, sigmoides, tangentes hiperbólicas, etc., entre las que

[1]Habitualmente, el error del modelo se define como la diferencia entre la salida del modelo y su salida esperada para cada muestra de entrenamiento. No obstante, esta función de error será dependiente del problema objetivo y puede presentar diferentes formas. Un ejemplo de ello es la función de coste del *variational autoencoder*, expuesta en la Sección 2.2.3, en la que no solo se tiene en cuenta la salida deseada del modelo, sino también la representación interna de los datos aprendida en una de sus capas intermedias, conocida como *espacio latente* del autoencoder.

destaca la función ReLU [57], que es la más popular en la actualidad gracias a que atenúa los problemas de desvanecimiento del gradiente durante el entrenamiento, facilitando así la convergencia del mismo. La elección de la función de activación reside en el humano que configura la arquitectura, pues se trata de un hiperparámetro más de las redes profundas, al igual que su número de capas o de neuronas por capa. Habitualmente, se eligen funciones populares como la ReLU o se ejecuta un abanico de experimentos con diferentes funciones de activación para elegir la que demuestre un mejor rendimiento. Cabe añadir que, aunque cada neurona podría utilizar una función diferente, es frecuente que las neuronas de una misma capa empleen el mismo tipo de función de activación.

En este contexto, tendremos que la salida de la capa l-ésima de la red será un vector $\mathbf{x}_l$, descrito en la Ecuación (2.1), donde: σ es la función de activación, $\mathbf{x}_{l-1}$ es el vector de entrada a la capa l, $\mathbf{W}_l$ es la matriz que contiene todos los pesos de las conexiones entre la capa l y la capa $l-1$, y $\mathbf{b}_l$ es un vector de bias que habitualmente acompaña a la suma ponderada de los pesos, para aportar aún mayor flexibilidad en el modelado de los datos.

$$\mathbf{x}_l = \sigma(\mathbf{W}_l \mathbf{x}_{l-1} + \mathbf{b}_l) \tag{2.1}$$

Como se observa en esta ecuación, las operaciones que tienen lugar en las redes profundas pueden ser expresadas en términos de productos matriciales, lo cual tiene importantes implicaciones computacionales y ha sido un factor clave en el éxito del *deep learning*. Cabe recordar que una red profunda puede constar de millones de neuronas, cuyos pesos son ajustados de forma iterativa, empleando tantas iteraciones como sea necesario hasta la convergencia del modelo. Esta elevada carga computacional es asumible gracias al uso de *hardware* específico, como las Unidades de Procesamiento Gráfico *(Graphics Processing Units, GPUs)*, que son altamente eficientes en la ejecución de operaciones matriciales y han sido indispensables en el desarrollo del *deep learning*.

En cuanto al diseño de la arquitectura de red, existe una amplia variedad de topologías presentes en la literatura. En la Figura 2.2 se presenta una arquitectura básica, constituida por capas densas o *fully connected*, que son aquellas en las que cada neurona está conectada con todas las neuronas de la capa anterior, e implementan la transformación de los datos descrita en la Ecuación (2.1). Pero existen alternativas, como las capas convolucionales, en las que cada neurona recibe conexiones entrantes de tan solo algunas de las neuronas de la capa previa, lo cual favorece que cada neurona se especialice en una región concreta de la capa anterior y ha demostrado ser un enfoque exitoso para el reconocimiento de patrones en los datos. Otra configuración posible es la de los *deep autoencoders*, que tienen el mismo número de neuronas en sus capas de entrada y salida, e incluyen una capa oculta de menor dimensión a modo de cuello de botella. Ambas configuraciones, redes convolucionales y *deep autoencoders*, han sido empleadas en esta investigación y su arquitectura será detallada a continuación. Dichas arquitecturas destacan por su amplia aceptación y éxito en la literatura, así como por sus potenciales aplicaciones en problemas de ingeniería. No obstante, otras configuraciones podrían tener también un gran impacto en este ámbito, como las redes recurrentes, que, al incluir conexiones de realimentación en la arquitectura,

han obtenido excelentes resultados en el análisis de secuencias y constituyen una interesante línea de trabajo futuro.

2.2.1. Redes convolucionales

Las redes convolucionales *(Convolutional Neural Networks, CNNs)* [50] se han convertido en uno de los principales exponentes del *deep learning*, destacando por su especial habilidad para extraer características de forma automática a partir de los datos de entrada, lo que se conoce como aprendizaje de características o *feature learning*. En la Figura 2.3 se presenta un ejemplo de arquitectura CNN, que consta de tres tipos de capas: convolucionales, de submuestreo y densas. En primer lugar, se encuentran las capas convolucionales, encargadas de detectar conjunciones locales de características en la representación de los datos proporcionada por la capa anterior. A continuación, las capas de submuestreo tratan de unificar características semánticamente similares, reduciendo con ello la dimensionalidad de los datos y la complejidad computacional del modelo, proporcionando también invarianza ante la presencia de pequeñas distorsiones en los datos. En último lugar, las capas densas conectan todas las neuronas entre sí, en busca de una representación final con un significado global de las muestras.

Con esta combinación de capas se crea una jerarquía composicional, en la que cada característica resulta de la composición de otras más simples [10]. En el contexto del procesamiento de imagen, combinaciones locales de bordes darán lugar a contornos, los contornos conformarán motivos y la agrupación de motivos permitirá representar objetos. De esta manera, las redes CNN implementan un razonamiento de alto nivel que les permite llevar a cabo la tarea objetivo (clasificación de imágenes en la Figura 2.3). No obstante, la capacidad de estas arquitecturas para detectar patrones en los datos se extiende más allá del procesamiento de imagen. Las redes CNN también se han mostrado competitivas en el análisis de señales unidimensionales, destacando en campos como el de reconocimiento de voz [81]. El ámbito de los sistemas de ingeniería, con señales y problemáticas similares, constituye otro interesante nicho de aplicación [82]. Por ello, en el Capítulo 3, se explorará el potencial de las redes CNN en el análisis de señales de vibración y corriente, para la detección de fallos en motores.

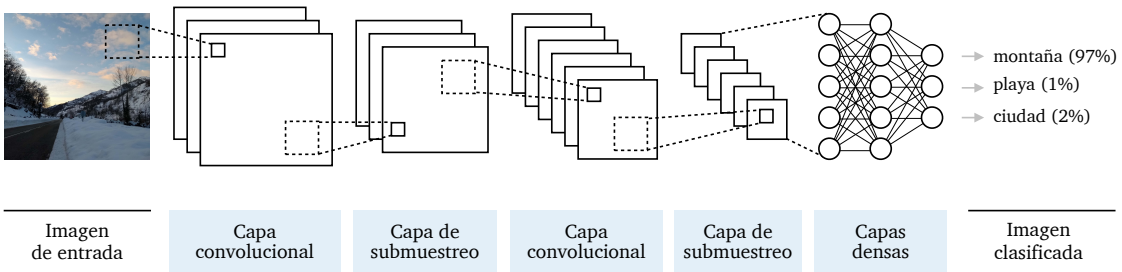


Figura 2.3: Ejemplo de arquitectura CNN para reconocimiento de imagen (cada plano es un mapa de características).

A diferencia de las capas densas, las neuronas de las capas convolucionales y de submuestreo no están totalmente conectadas, lo que les permite especializarse en regiones concretas de la capa previa. De esta manera, las capas convolucionales

y de submuestreo dividen y modelan los datos de entrada en partes más pequeñas, preservando la coherencia espacial de los datos y reduciendo drásticamente el número de operaciones a ejecutar, así como el número de pesos a ajustar durante el entrenamiento [83]. En las capas convolucionales, los pesos se organizan en conjuntos de filtros que convolucionan sobre la señal de entrada —señal, habitualmente, de tipo multicanal (por ejemplo, imágenes con varios canales de color)— dando lugar a un conjunto de vectores de salida llamados mapas de características o *feature maps*. En la Ecuación (2.2) se observa que la salida de la l-ésima capa convolucional consta de m mapas de características, tantos como filtros de convolución. En detalle, el mapa de características $\mathbf{x}_l^{(m)}$ resultará de la convolución ($*$) de cada canal c de la entrada ($\mathbf{x}_{l-1}^{(c)}$) con su filtro m correspondiente ($\mathbf{W}_l^{(c,m)}$), siendo $\mathbf{b}_l^{(m)}$ el vector de bias.

$$\mathbf{x}_l^{(m)} = \sigma\left(\sum_{c=1}^{C} \mathbf{W}_l^{(c,m)} * \mathbf{x}_{l-1}^{(c)} + \mathbf{b}_l^{(m)}\right) \tag{2.2}$$

A continuación de la capa de convolución, se encuentra generalmente una capa de submuestreo. En esta capa, el vector de entrada es recorrido con una máscara, que divide dicho vector en partes más pequeñas y devuelve un valor agregado para cada una de ellas. La función o máscara de *max pooling*, que devuelve el valor máximo de los datos que recibe como entrada, es la más popular en las capas de submuestreo [84].

Con esta sucesión de capas convolucionales y de submuestreo, las redes CNN dividen y modelan la información de entrada que, a continuación, es procesada en las capas más profundas de la red —habitualmente, capas densas— para abordar con ello el problema objetivo. En este contexto, cabe mencionar la presencia de la función de activación (σ) en las transformaciones entre todas las capas del modelo. En la capa de salida de la red, la elección de la función de activación está condicionada al tipo de problema objetivo: en problemas de clasificación, las funciones sigmoide (clasificación binaria) y *softmax* (clasificación multiclase) son las más comunes; mientras, en problemas de predicción, destaca la función de activación lineal. En el resto de capas, como se ha mencionado anteriormente, la función ReLU es la elección más extendida.

2.2.2. *Deep autoencoders*

Los *deep autoencoders*, o autoencoders profundos, son redes prealimentadas particularmente entrenadas para reproducir a su salida la misma información que reciben de entrada. Como se observa en la Figura 2.4, la arquitectura consiste en: un *encoder* f_{enc}, que proporciona una representación latente $\mathbf{z}$ de los datos de entrada $\mathbf{x}$; y un *decoder* f_{dec}, que reconstruye los datos de entrada a partir de su representación latente $\mathbf{z}$, devolviendo una estimación de la entrada $\hat{\mathbf{x}}$. Entre el encoder y el decoder, se encuentra un *cuello de botella* —típicamente, una o varias capas de menor dimensión que la entrada— que impide que el modelo aprenda la función identidad como solución al problema, reproduciendo una simple copia de los datos de entrada. En su lugar, la restricción impuesta por el cuello de botella

hace que el modelo se vea forzado a aprender una representación compacta de los datos, la cual capture la estructura subyacente en los mismos, preservando la información relevante y descartando aquella que sea redundante o superflua para su reconstrucción.

Durante el proceso de aprendizaje, la arquitectura es entrenada para minimizar la diferencia entre $\mathbf{x}$ y $\hat{\mathbf{x}}$. De manera que el objetivo del *deep autoencoder* consiste en encontrar la solución al problema de optimización (2.3), donde se trata de minimizar la función de coste $\mathcal{L}$, siendo $\|.\|$ habitualmente la norma L2.

$$\min_{f_{\text{enc}}, f_{\text{dec}}} \mathcal{L}(\mathbf{x}, \hat{\mathbf{x}}) = \min_{f_{\text{enc}}, f_{\text{dec}}} \|\mathbf{x} - f_{\text{dec}}(f_{\text{enc}}(\mathbf{x}))\| \tag{2.3}$$

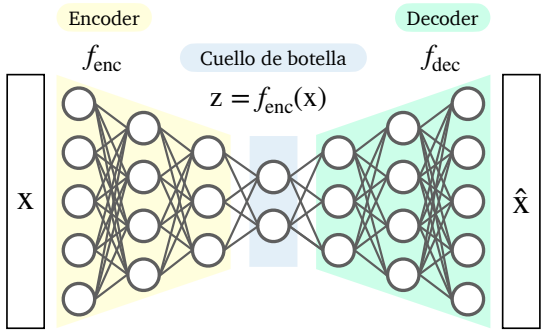


Figura 2.4: Ejemplo de arquitectura de un *deep autoencoder*.

Aunque los primeros autoencoders aparecieron hace décadas [85, 86], estas arquitecturas han seguido evolucionando a lo largo de los años, especialmente con la llegada del aprendizaje profundo. Los autoencoders profundos, al igual que el resto de técnicas *deep learning*, gozan de una arquitectura composicional que les otorga la habilidad de encontrar representaciones con significado acerca de los datos —habilidad conocida como *representation learning* [8]— y que los convierte en excelentes extractores de características. En consecuencia, gracias a su naturaleza profunda, los *deep autoencoders* reducen la dimensionalidad de los datos de entrada de una manera jerárquica, que les permite conseguir reconstrucciones de alta calidad de los datos [87, 88, 89].

Sin embargo, la calidad de la reconstrucción empeora cuando las muestras de entrada no son consistentes con los datos de entrenamiento. Gracias a ello, el error de reconstrucción de los autoencoders ha sido empleado en la literatura para medir la desviación, respecto a los datos de entrenamiento, de cualquier nueva muestra entrante. Además, en aquellos contextos en que los datos de entrenamiento son representativos de un comportamiento normal o esperable del sistema, el error de reconstrucción se convierte en un valioso indicador de anomalía, de manera que aquellas muestras entrantes con errores de reconstrucción elevados son consideradas anómalas [31, 90]. Esta aplicación de los *deep autoencoders* representa una interesante línea de estudio, abordada en el Capítulo 4, donde se explorará el potencial del error de reconstrucción o *residuo* de los autoencoders en la detección y diagnóstico de anomalías en sistema de ingeniería.

La exploración del cuello de botella o espacio latente de los *deep autoencoders*

también podría revelar información de interés para la monitorización de la salud de los sistemas. Estos espacios —al capturar la información relevante en los datos y, potencialmente, las principales fuentes de variación en los mismos— constituyen un interesante nicho para la construcción de indicadores de salud de los procesos, como ya apuntan algunos trabajos en la literatura [34]. En consecuencia, en el Capítulo 5 se explorará el potencial de los espacios latentes para la generación de indicadores que, en particular, permitan monitorizar el nivel de degradación de las máquinas.

Adicionalmente, en el Capítulo 6, se estudiará la integración de estos espacios latentes en herramientas de visualización interactiva, que permitan una exploración sencilla de los mismos para una monitorización visual e intuitiva de los procesos. En detalle, se utilizará este enfoque para el análisis del consumo energético en grandes edificios. Complementariamente, se trasladará también esta idea al ámbito biomédico, como herramienta de análisis para el estudio de los movimientos celulares en procesos de cáncer, estableciendo así potenciales conexiones entre el ámbito de la biomedicina y el de la ingeniería, en ambos de los cuales se espera que el aprendizaje profundo tenga un gran impacto durante los próximos años.

Finalmente cabe destacar que, gracias a sus buenas propiedades, los *deep autoencoders* han sido ampliamente utilizados en la literatura, facilitando el estudio y diseño de nuevas variantes, entre las que destaca el autoencoder variacional, que también ha sido objeto de estudio de esta tesis.

2.2.3. *Deep variational autoencoders*

Los autoencoders variacionales o *variational autoencoders* (VAEs) [91] heredan la estructura de los *deep autoencoders*, imponiendo restricciones adicionales en el cuello de botella, con las que transforman la arquitectura determinista del autoencoder en un modelo probabilístico. En detalle, el VAE es entrenado para aprender una distribución de probabilidad del espacio latente, lo que le convierte además en un valioso modelo generativo. En los últimos años, esta arquitectura ha ido creciendo en popularidad y ha mostrado resultados prometedores en aplicaciones de generación de imagen [28], creación de pistas musicales [70] o diseño de compuestos moleculares [30].

Gracias a su naturaleza probabilística, el VAE ha demostrado una especial habilidad para capturar la estructura subyacente en los datos, que le ha llevado al éxito en tareas generativas, pero que podría suponer también una gran contribución en el modelado del comportamiento normal de los procesos, con potenciales aplicaciones en la detección de anomalías o en la generación de indicadores de salud de los procesos [22]. Además, estas arquitecturas proporcionan espacios latentes semánticamente relevantes [92], cuya exploración podría contribuir también a la monitorización y mejora de la comprensión de los sistemas. En consecuencia, a lo largo de esta tesis se explorará el potencial de los *deep autoencoders* en diferentes problemas y contextos de ingeniería, poniendo especial interés en su extensión variacional (Capítulos 4 y 5).

En la Figura 2.5 se observa que el VAE consta de un encoder $q_\phi(\mathbf{z}|\mathbf{x})$, que repre-

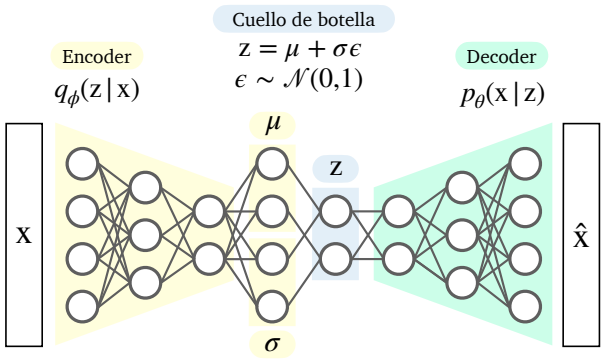


Figura 2.5: Ejemplo de arquitectura de un *deep variational autoencoder*.

senta una aproximación de la distribución a posteriori de los datos, y un decoder $p_\theta(\mathbf{x}|\mathbf{z})$, que representa la verosimilitud de $\mathbf{x}$ dada una variable latente $\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})$. Acorde a este esquema, el encoder actúa como una red de inferencia variacional que mapea los datos de entrada en una distribución a posteriori (aproximada) en el espacio latente del autoencoder. A continuación, el decoder opera como una red generativa, que mapea las coordenadas latentes de vuelta hacia el espacio original de los datos. Adicionalmente, se asume que los datos de entrada $\mathbf{x}$ se ajustan a una distribución gaussiana unitaria, cuya media $\boldsymbol{\mu}$ y desviación típica $\boldsymbol{\sigma}$ serán variables latentes del modelo. En consecuencia, el proceso de entrenamiento de la red consiste en la optimización simultánea de dos funciones de coste (2.4): una función asociada al error de reconstrucción del autoencoder $\mathcal{L}$, al igual que en los *deep autoencoders* convencionales; y la divergencia de Kullback-Leibler D_{KL}, entre la distribución latente aprendida y una distribución a priori de los datos de tipo gaussiana unitaria. Como resultado, el VAE se convierte en un caso especial de *deep autoencoder*, que al incorporar una regularización adicional —proporcionada por el término D_{KL}— adquiere las propiedades de un modelo generativo.

$$\mathcal{L}_{VAE}(\theta, \phi; \mathbf{x}) = \mathcal{L}(\mathbf{x}, \hat{\mathbf{x}}) + D_{\mathrm{KL}}(q_\phi(\mathbf{z}|\mathbf{x})||p_\theta(\mathbf{z})) \tag{2.4}$$

En cuanto a la implementación de estas arquitecturas, cabe destacar que la función de activación ReLU es la más popular en la literatura, tanto en el caso de los *deep autoencoders* como en su extensión variacional. No obstante, en la capa de salida, así como en el espacio latente del modelo, es habitual encontrar otras variantes, como la función de activación lineal o la sigmoide [93, 94].

2.3. Entrenamiento de una red profunda

Las redes profundas son entrenadas para aprender una función, o mapeo de los datos, que relacione de forma satisfactoria las entradas y salidas del modelo. En detalle, la función aprendida por la red estará determinada por los pesos de su arquitectura. Por tanto, el proceso de aprendizaje o entrenamiento de las redes profundas consiste en la búsqueda de un conjunto de pesos óptimo que, para el conjunto de datos de trabajo, permita completar con éxito la tarea objetivo

(clasificación, predicción, reconstrucción de datos, etc.). De forma simplificada, el entrenamiento comenzará con una inicialización aleatoria de los pesos, que serán actualizados de forma iterativa hasta que el error cometido por la red alcance un rango aceptable. Para la optimización de los pesos que tiene lugar a lo largo de estas iteraciones, se recurre tradicionalmente al algoritmo de descenso del gradiente [80], en combinación con el método de retropropagación del error o *backpropagation* [45], ambos descritos a continuación.

2.3.1. Descenso del gradiente

El algoritmo de descenso del gradiente propone actualizar los pesos del modelo mediante la minimización de su función de coste, la cual mide el error cometido por el modelo sobre el conjunto de datos de trabajo. Esta función de coste es elegida por el usuario y está fuertemente vinculada al tipo de problema objetivo: en problemas de predicción destaca la función del error cuadrático medio *(Mean Squared Error, MSE)*; mientras que en problemas de clasificación, son frecuentes las funciones de entropía cruzada o *cross entropy* (clasificación multiclase) y entropía cruzada binaria o *binary cross entropy* (clasificación binaria).

La minimización analítica de esta función de coste representa un problema enormemente complejo, especialmente en el caso de las arquitecturas profundas, que constan de un elevado número de capas y miles de pesos a optimizar. La solución propuesta por el algoritmo de descenso del gradiente consiste en desplazarse de forma iterativa por el espacio de búsqueda —espacio de los parámetros entrenables o pesos del modelo— en la dirección negativa del gradiente de la función de coste. Este procedimiento se refleja en la regla de aprendizaje del algoritmo (2.5), donde cada peso de la red $w_{j,k}$ —que es aquel que conecta las neuronas j y k— se actualiza en cada iteración i en base a su valor en la iteración anterior y al gradiente de la función de coste —derivada parcial de la función de coste $\mathcal{L}$ respecto a dicho peso— ponderado por un factor α, conocido como tasa de aprendizaje.

$$w_{j,k}^{i} = w_{j,k}^{i-1} - \alpha \frac{\partial \mathcal{L}}{\partial w_{j,k}^{i-1}} \tag{2.5}$$

Como resultado, los pesos son actualizados de forma proporcional al gradiente y en la dirección opuesta al mismo. Adicionalmente, la tasa de aprendizaje o *learning rate* pondera el tamaño de los desplazamientos realizados en el espacio de búsqueda y, con ello, permite ajustar el ritmo de aprendizaje de la red. Se trata de un parámetro crítico, pues tanto tasas demasiado pequeñas, como demasiado elevadas, pueden dificultar la convergencia del modelo. En la actualidad, es frecuente recurrir a algoritmos adaptativos —como AdaGrad [95], RMSProp [96] o ADAM [97]— que reducen la tasa de aprendizaje de forma progresiva a lo largo del entrenamiento, acelerando así la convergencia en las primeras iteraciones y facilitando una sintonización fina de los pesos en las últimas.

Otro aspecto a considerar es el número de muestras empleadas durante el entrenamiento. Cabe recordar que, en cada iteración, el modelo de red será aplicado

a dichas muestras y los pesos serán actualizados en función del error cometido, siguiendo la regla de aprendizaje (2.5). Por ello, cuando se trabaja con conjuntos de datos de gran tamaño, el procesamiento de todas las muestras en cada iteración del entrenamiento puede suponer una elevada carga computacional. Ante esta situación, es habitual tomar tan solo una porción del total de las muestras —que será aleatoria y, por tanto, diferente, en cada iteración— en lo que se conoce como entrenamiento *mini-batch*, donde el tamaño de esta porción o *batch* será un hiperparámetro a definir por el usuario.

2.3.2. *Backpropagation*

El algoritmo de descenso del gradiente define la regla de aprendizaje del modelo en términos del gradiente de la función de coste (2.5). Esta regla requiere el cálculo de la derivada parcial de la función de coste respecto a cada uno de los parámetros de la red, lo cual, en modelos profundos con un elevado número de capas ocultas y de neuronas por capa, es una tarea enormemente compleja. Por ello, la implementación de las arquitecturas profundas fue inasumible en la práctica hasta la aparición, en los años ochenta, del algoritmo de retropropagación del error o *backpropagation* [45].

Este algoritmo propone un mecanismo eficiente para el cálculo del gradiente, que consiste en propagar hacia atrás el error cometido por el modelo, desde la capa de salida hacia las capas intermedias, asignando a cada neurona una fracción de la señal de error y describiendo así su contribución relativa al error total de la red. En detalle, el término de error δ asociado a cada neurona k, se obtiene a partir del producto de dos términos (2.6): el gradiente del error cometido por la red $\mathcal{L}$ respecto a la salida o activación de la neurona a_k, y el gradiente de la función de activación a_k respecto a la suma ponderada de los pesos de entrada que recibe la neurona z_k.

$$\delta_k = \frac{\partial \mathcal{L}}{\partial a_k} \cdot \frac{\partial a_k}{\partial z_k} \tag{2.6}$$

El primer término de esta expresión requiere una descomposición adicional. En el caso de las neuronas pertenecientes a la última capa del modelo (2.7), el gradiente del error consistirá en la diferencia entre la salida esperada de la neurona t_k y la salida real obtenida a_k. En cambio, en las neuronas de las capas ocultas (2.8), este gradiente será la suma de los errores de las N neuronas de la capa posterior, ponderados por los pesos de sus conexiones con la neurona k. A la luz de estas expresiones, podemos comprobar que, salvo en la capa de salida, el error de cada neurona está expresado en términos del error de sus neuronas posteriores. Esto explica el nombre del algoritmo de *backpropagation*, donde el error es retropropagado hacia atrás, comenzando por la última capa del modelo.

$$\frac{\partial \mathcal{L}}{\partial a_k} = t_k - a_k \tag{2.7}$$

$$\frac{\partial \mathcal{L}}{\partial a_k} = \sum_{i=1}^{N} w_{k,i} \cdot \delta_i \tag{2.8}$$

Una vez calculado el término de error de todas las neuronas, se procede finalmente a la obtención del gradiente de la función de coste respecto a cada peso de la red (2.9). Usando la regla de la cadena, se obtiene que la derivada parcial del error respecto al peso $w_{j,k}$ —que conecta la neurona j con la neurona k— se define como el producto entre: la activación de la neurona j y el error asociado a la neurona k.

$$\frac{\partial \mathcal{L}}{\partial w_{j,k}} = a_j \cdot \delta_k \tag{2.9}$$

En cada iteración del algoritmo de descenso del gradiente, los pesos del modelo serán actualizados de acuerdo a la regla de aprendizaje expuesta en (2.5), donde el gradiente de la función de coste respecto a cada peso será calculado a través del mecanismo de *backpropagation* (2.9). Esta combinación de algoritmos es la que hace posible el entrenamiento de las arquitecturas profundas, que, en esta investigación, hemos abordado haciendo uso de la librería de programación Keras —especializada en aprendizaje profundo— que ya incluye la implementación de los mismos [98].

2.3.3. Sobreajuste de los datos

El proceso de entrenamiento de las redes profundas no siempre es completado con éxito. Idealmente, una vez finalizado, las arquitecturas habrán *aprendido* a resolver el problema objetivo y, por tanto, obtendrán un buen rendimiento no solo ante los datos de entrenamiento, sino también ante nuevas muestras entrantes no vistas durante el mismo. Sin embargo, las redes profundas tienden a sufrir problemas de *overfitting* o sobreajuste, que les hacen perder su capacidad de generalización.

Esta limitación es una consecuencia de la poderosa arquitectura de los modelos profundos, que, en ocasiones, puede llevarles a aprender funciones demasiado complejas de los datos. Para evitar este problema, se han propuesto en la literatura diferentes mecanismos de regularización que ayudan a limitar la capacidad de aprendizaje de los modelos. Entre las opciones más frecuentes se encuentran la de reducir el número de iteraciones del entrenamiento o la de simplificar la arquitectura de red —disminuyendo, por ejemplo, el número de capas o el número de neuronas por capa. Otros mecanismos consisten en incluir restricciones en los pesos de las neuronas, entre los que destacan: la técnica *weight decay*, que propone incluir penalizaciones para los pesos elevados, en función de sus valores al cuadrado (penalización L2) o absolutos (penalización L1); o la técnica *max-norm*, que propone restringir a un valor máximo la norma del vector de pesos.

Sin embargo, estos mecanismos pueden ser insuficientes para prevenir el sobreajuste del modelo. En tales casos, se recurre a técnicas más elaboradas, como

el mecanismo de *dropout* [99] o la técnica de normalización del *batch* o *batch normalization* [100]. El mecanismo de *dropout* propone transformar el entrenamiento de una red compleja en el entrenamiento de varias redes más simples, donde cada red simple resulta de ignorar parte de las neuronas de la red original, que son omitidas aleatoriamente durante el proceso de entrenamiento. Mientras, el mecanismo de *batch normalization* propone acelerar el entrenamiento de la red mediante la reducción de la covarianza interna de los datos. Se trata de una técnica de optimización que, como efecto secundario, introduce ruido en la red, lo que contribuye a la regularización de la misma.

En la práctica, la elección del mecanismo de regularización dependerá de cada problema y será necesaria la ejecución de diferentes experimentos para seleccionar la técnica —o combinación de técnicas— óptima. Como se expone en los siguientes capítulos, a lo largo de esta investigación el problema de sobreajuste se ha solventado mediante la limitación del número de iteraciones del entrenamiento y la simplificación de la arquitectura de los modelos.

CAPÍTULO 3

Clasificación

En este capítulo se aborda el uso de arquitecturas profundas para la clasificación del estado de funcionamiento de los procesos. En detalle, se presenta el uso de una red convolucional para la detección de fallos en motores, incluyendo un análisis de las características aprendidas por la red durante el entrenamiento. El capítulo comienza con una revisión del estado del arte, para detallar a continuación los experimentos realizados y los resultados obtenidos, terminando con un apartado de conclusiones.

El contenido de este capítulo ha sido publicado en la revista *Heliyon*, bajo el título *«DCNN for condition monitoring and fault detection in rotating machines and its contribution to the understanding of machine nature»* [101].

3.1. Antecedentes

En el ámbito de los sistemas de ingeniería, las aplicaciones de detección y diagnóstico de fallos resultan de gran importancia, pues permiten garantizar la seguridad en la operación de las máquinas, así como optimizar su funcionamiento, en busca de una mayor productividad y eficiencia, y con beneficios como reducciones de costes o un mayor tiempo de vida útil para las máquinas [102]. En este contexto, se ha dedicado gran esfuerzo al estudio de componentes críticos, como los rodamientos, cuyo fallo es una de las causas de avería más comunes en las máquinas rotativas [102, 103].

Estos fallos en los rodamientos son frecuentes y difíciles de detectar, pero, detectados a tiempo, se encuentran entre los fallos menos costosos de reparar [104]. En este contexto, se recurre habitualmente a algoritmos de aprendizaje automático que, a partir de un conjunto de características representativas de la máquina, son capaces de detectar fallos en su funcionamiento de forma eficiente y automática, permitiendo planificar así operaciones de mantenimiento en el sistema. Dichas características son extraídas a partir de datos crudos de operación —típicamente, corrientes y vibraciones, que portan gran cantidad de información acerca de la condición de la máquina— y es un experto quien decide cuál es el conjunto de

características más apropiado a extraer [105], en base a su experiencia y conocimiento previo de la máquina. En consecuencia, se dice que estos enfoques están basados en características *diseñadas manualmente*, lo que también se conoce como *ingeniería de características*.

El enventanado de datos es una técnica ampliamente utilizada en esta fase de extracción de características [106], que consiste en dividir los datos en ventanas —con o sin solapamiento— y calcular una característica para cada ventana. En este contexto, descriptores como el valor eficaz o RMS [107], la curtosis [108] o el factor de cresta [108] han demostrado ser características eficientes para la detección de fallos en los rodamientos. La expresión de estas características se muestra en las Ecuaciones (3.1, 3.2, 3.3), donde x representa una ventana o vector de N elementos, con media μ y desviación típica σ.

$$RMS = \sqrt{\frac{1}{N}\sum_{i}^{N} {x_i}^2} \tag{3.1}$$

$$Curtosis = \frac{\sum_{i}^{N}(x_i - \mu)^4}{N\sigma^4} \tag{3.2}$$

$$Factor\ de\ cresta = \frac{max(|x_i|)}{RMS} \tag{3.3}$$

La monitorización de características frecuenciales de la máquina también ha proporcionado buenos resultados en la literatura. Los rodamientos inducen vibraciones inherentes al sistema, que manifestará patrones de vibración diferentes según opere en condiciones de funcionamiento normal o, en su defecto, en condiciones de fallo (debidas, por ejemplo, a la presencia de defectos en la pista de rodadura del rodamiento, defectos en sus rodillos, deterioro de la jaula, desequilibrios, desalineaciones, etc.). Por ello, las técnicas de análisis de vibraciones son herramientas frecuentes en la monitorización de la condición de las máquinas [109, 110]. Habitualmente, estos enfoques proponen un análisis de las vibraciones que tiene lugar en el dominio de la frecuencia [105, 111, 112] y que requiere de conocimiento previo acerca de las frecuencias de fallo del sistema. En detalle, dichos enfoques proponen monitorizar la amplitud de las vibraciones de la máquina a las frecuencias de fallo, para detectar así posibles defectos o anomalías en su funcionamiento. Cuando tanto la geometría del rodamiento, como la velocidad de su eje, son conocidos (Figura 3.1), estas frecuencias pueden ser calculadas de acuerdo a las siguientes ecuaciones [113]:

$$BPFI = \frac{n}{2} \cdot N \cdot \left[1 + \frac{d}{D}\right] \tag{3.4}$$

$$BPFO = \frac{n}{2} \cdot N \cdot \left[1 - \frac{d}{D}\right] \tag{3.5}$$

$$BPFI = 0.6 \cdot N \cdot n \tag{3.6}$$

$$BPFO = 0.4 \cdot N \cdot n \tag{3.7}$$

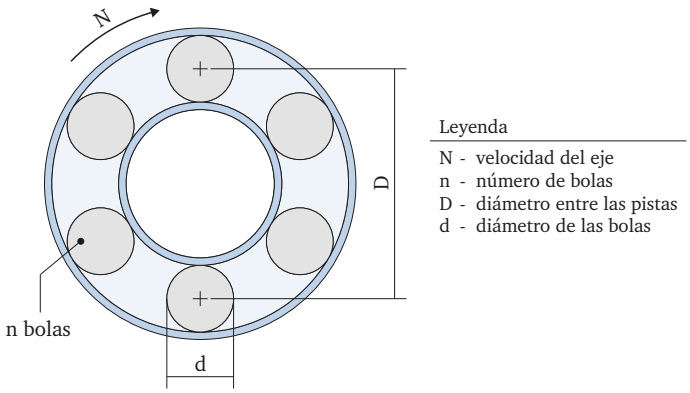


Figura 3.1: Geometría de un rodamiento.

En (3.4) y (3.5) se presentan las frecuencias de paso de las bolas en la pista interior *(Ball Pass Frequency Inner, BPFI)* y exterior *(Ball Pass Frequency Outer, BPFO)* del rodamiento, que están relacionadas con la presencia de defectos en dichas pistas. Estas ecuaciones requieren cierto conocimiento previo acerca de la máquina, como su velocidad N, el diámetro entre las pistas D, el diámetro de las bolas d, o el número de bolas del rodamiento n. Alternativamente, cuando los parámetros D y d son desconocidos, dichas expresiones son reemplazadas por las ecuaciones experimentales (3.6) y (3.7) [103].

Todas estas características —descritas en las Ecuaciones (3.1) a (3.7)— son populares en la literatura, pero existe una amplia variedad de alternativas. Otros descriptores de interés podrían ser la frecuencia de rotación de la jaula, la frecuencia de paso de las bolas, la asimetría de los datos o *skewness*, el factor de forma, el factor de holgura, etc. En definitiva, se requiere de conocimiento humano experto, para decidir cuál es el conjunto de características más apropiado para el problema objetivo. A continuación, los algoritmos de aprendizaje automático son alimentados con estas características y, en base a ellas, monitorizan la condición de las máquinas para detectar fallos en su funcionamiento. Algoritmos como las máquinas de vectores de soporte *(Support Vector Machines, SVMs)*, los árboles de decisión *(Decision Trees, DTs)* o los perceptrones multicapa *(Multi-Layer Perceptrons, MLPs)* han demostrado buenos resultados en este ámbito de detección de fallos [114, 115, 116].

Por tanto, la combinación de métodos de ingeniería de características con algoritmos de aprendizaje automático ha dado lugar a útiles herramientas de monitorización, capaces de detectar con precisión diferentes tipos de fallo en las máquinas. No obstante, estos enfoques también presentan importantes inconvenientes. Por ejemplo, para el cálculo de las frecuencias de fallo, se asumen ciertas simplificaciones —como el movimiento de rodadura en los rodamientos, que en realidad están sometidos a una combinación de rodadura y deslizamiento— que hacen que

dichas frecuencias diseñadas manualmente puedan diferir ligeramente de las frecuencias reales del sistema, afectando a la precisión en la detección [117]. Otra fuente de error es la presencia simultánea de diferentes tipos de fallos, así como la interferencia de fuentes adicionales de vibración, que dificultan enormemente la caracterización del fallo [105]. En último lugar, ciertos defectos, como los derivados de problemas de lubricación, no tienen una naturaleza cíclica y son difíciles de detectar a través de un análisis en frecuencia, con lo que requieren de descriptores específicos para su monitorización [118].

Estos inconvenientes reflejan las debilidades de los métodos basados en características diseñadas manualmente, que son altamente dependientes del problema y cuyo rendimiento estará sujeto a la calidad de las características elegidas para el análisis. Dada su importancia, el diseño y selección de un conjunto óptimo de características ha sido objeto de estudio de numerosas investigaciones y se ha demostrado que, en ciertos contextos [119, 120, 121, 122], es posible extraer manualmente características óptimas para la monitorización del estado de los sistemas. Sin embargo, esta extracción de características supone un gran reto en sistemas complejos [123]. En tales casos, se requiere de conocimiento experto, información previa sobre la máquina y una fuerte base matemática, para llevar a cabo un diseño óptimo de características.

Ante esta situación, existe un interés creciente en el aprendizaje de características o *feature learning* [10]. Este enfoque propone *aprender* características relevantes del sistema a partir de datos de operación del mismo, en lugar de diseñarlas manualmente. Como se muestra en la Figura 3.2, en los enfoques de ingeniería de características es un experto el que extrae las características de interés $\boldsymbol{\phi}$ a partir de los datos crudos de entrada $\mathbf{x}$. Mientras, los enfoques de aprendizaje de características (Figura 3.3) proponen aprender la transformación $t_\theta(\mathbf{x})$ de los datos de entrada $\mathbf{x}$ que produce una representación adecuada $\boldsymbol{\phi}$ para la posterior tarea de clasificación. En ambos casos, las características $\boldsymbol{\phi}$ serán empleadas para entrenar un algoritmo de clasificación $f_\theta(\boldsymbol{\phi})$ encargado de determinar la condición $\mathbf{y}$ de los datos de entrada.

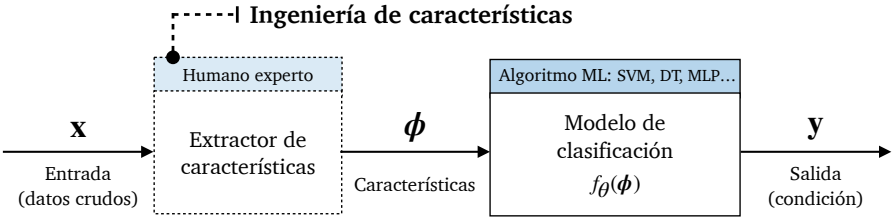


Figura 3.2: Clasificación utilizando ingeniería de características.

El aprendizaje de características ha sido abordado en la literatura por medio de técnicas populares como el Análisis de Componentes Principales *(Principal Components Analysis, PCA)* [124], los métodos de codificación dispersa *(sparse coding)* [125] o las redes causales *(sigmoid belief networks)* [126]. No obstante, estas técnicas han comenzado a ser reemplazadas por modelos profundos, capaces de abordar conjuntamente las tareas de extracción de características y clasificación (como se muestra en la Figura 3.3), y cuya habilidad para encontrar de forma automática representaciones con significado de los datos, conocida como *representation*

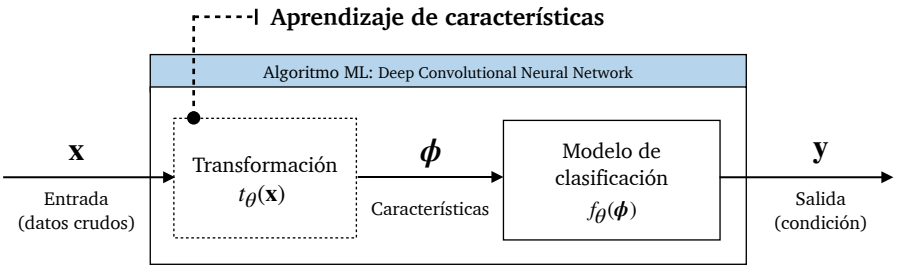


Figura 3.3: Clasificación utilizando aprendizaje de características.

learning [8], les convierte en poderosos extractores de características.

Varios trabajos en el estado del arte han abordado un enfoque intermedio, que consiste en combinar los modelos profundos con una breve etapa previa de extracción de características. Estos enfoques han demostrado ser capaces de simplificar el proceso de diseño de características y, al mismo tiempo, detectar con éxito la aparición de fallos en las máquinas [114, 127, 128, 129, 130, 131]. Sin embargo, el auténtico reto no consistiría en simplificar, sino en prescindir de cualquier diseño manual de características, y es en este campo donde se espera que las redes profundas convolucionales *(Convolutional Neural Networks, CNNs)* [132] —que destacan por encima de otros modelos profundos por sus habilidades para el reconocimiento de patrones en los datos— tengan una gran contribución.

En los últimos años, las redes CNN se han convertido en una de las arquitecturas profundas más populares, demostrando una excelente habilidad para la detección de patrones en imágenes y señales. Estos modelos han sido empleados con éxito en otros ámbitos, como el reconocimiento de voz o el procesamiento de imagen, donde han superado en rendimiento a otras técnicas del estado del arte [62, 133, 134]. Su habilidad para la detección de fallos en máquinas rotativas también ha sido objeto de algunos estudios en la literatura, pero pocos trabajos han explorado las características aprendidas por las arquitecturas CNN o la información que podrían proporcionar acerca de la naturaleza de las máquinas [135, 136].

Ante estas circunstancias, hemos explorado en la presente tesis el potencial de las redes CNN para la monitorización de la condición en máquinas rotativas. El enfoque propuesto, descrito a continuación, ha demostrado reconocer el estado de funcionamiento de la máquina bajo estudio con una exactitud del 98 %, a partir de datos crudos de vibración y corriente en la misma. Además, hemos analizado el filtro de convolución aprendido por el modelo, cuyo estudio ha revelado parámetros característicos de la máquina, como su velocidad de rotación o el número de bolas en sus rodamientos. Adicionalmente, el rendimiento de este modelo ha sido comparado con el de los enfoques tradicionales, que combinan ingeniería de características y métodos clásicos de detección de fallos. En detalle, se han considerado el perceptrón multicapa *(Multi-Layer Perceptron, MLP)* [114], el clasificador de bosque aleatorio *(Random Forest Classifier, RFC)* [137] y el clasificador de vectores de soporte *(Support Vector Classifier, SVC)* [138]. En esta comparativa, la red CNN ha demostrado un rendimiento similar al de los clasificadores tradicionales, con la ventaja de que el modelo profundo no requiere de ningún tipo de conocimiento a priori sobre la máquina para llevar a cabo la clasificación. Finalmente, se ha

trasladado este mismo enfoque a otra máquina rotativa, donde el modelo también ha conseguido resultados satisfactorios, con una exactitud en la clasificación del 92 %.

3.2. Método propuesto

En esta sección se presenta la arquitectura convolucional empleada para abordar la detección y diagnóstico de fallos en máquinas rotativas. También se incluye la descripción del conjunto de datos utilizado en los experimentos.

3.2.1. Conjunto de datos: *dataicann*

La máquina empleada en nuestros experimentos se presenta en la Figura 3.4. Se trata de un motor de inducción de 4 kW con rodamientos 6306-2Z/C3 que gira a 1500 rpm (25 Hz), con una frecuencia de alimentación de 50 Hz. Esta máquina ha sido sometida a siete ensayos de funcionamiento (Tabla 3.1) de cuatro segundos de duración, para cada uno de los cuales se han registrado los valores de tres variables de operación (Tabla 3.2) con una frecuencia de muestreo de 5000 Hz. El conjunto de datos resultante, generado por el grupo de investigación GSDPI[1], recibe el nombre de *dataicann* y se encuentra disponible en [139].

Figura 3.4: Máquina utilizada en los experimentos.

En cuanto al preprocesamiento de los datos, se ha igualado el rango de las tres variables, mediante un escalado min-max [140] de rango $[0, 1]$. A continuación, se ha realizado un enventanado de los datos, utilizando ventanas sin solapamiento de tamaño 800 elementos.

Tabla 3.1: Ensayos *dataicann*.

ID Ensayo	Condición de la máquina
T1	Fallo mecánico (masa excéntrica en polea)
T2	Fallo eléctrico y mecánico combinado
T3	Funcionamiento normal
T4	Fallo eléctrico (resistencia 5 Ω en fase R)
T5	Fallo eléctrico (resistencia 10 Ω en fase R)
T6	Fallo eléctrico (resistencia 15 Ω en fase R)
T7	Fallo eléctrico (resistencia 20 Ω en fase R)

[1]GSDPI: Grupo de Supervisión y Diagnóstico de Procesos Industriales de la Universidad de Oviedo (`http://isa.uniovi.es/GSDPI/`).

Tabla 3.2: Variables disponibles en *dataicann*.

Variable	Descripción
a_x	Aceleración de la vibración en dirección X (horizontal)
a_y	Aceleración de la vibración en dirección Y (vertical)
i_r	Corriente en fase R

3.2.2. Modelo CNN

Como se describe en la Figura 3.5, el modelo CNN propuesto determina la condición de la máquina en cualquier instante k, a partir de un vector de entrada que contiene datos de operación de la misma (valores de vibración y corriente). El vector de entrada $(\mathbf{cnn}_{input})$ tiene un tamaño (N, c), donde N es el número de elementos en una muestra ($N = 800$) y c es el número de canales en los datos ($c = 3$, ya que se dispone de tres variables de operación: a_x, a_y, i_r) (Tabla 3.2). La salida del modelo $(\mathbf{cnn}_{output})$ es un vector $\mathbf{p}_k$ de n elementos, que refleja la probabilidad de pertenencia de la muestra k a cada uno de los n posibles estados de la máquina ($n = 7$, de acuerdo a los siete ensayos disponibles) (Tabla 3.1).

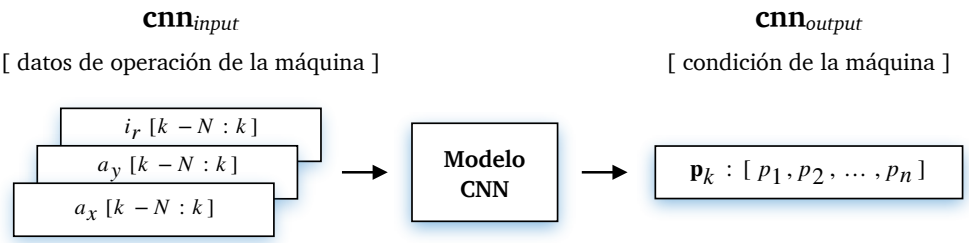


Figura 3.5: Contexto de trabajo del modelo CNN.

Para establecer la correspondencia entre los datos de operación de la máquina y su estado de funcionamiento, hemos utilizado la arquitectura convolucional expuesta en la Figura 3.6. Este modelo consta de una capa de entrada constituida por 3 canales de 800 neuronas y una capa de salida de 7 neuronas. El aprendizaje de características reside en una capa convolucional constituida por un filtro de convolución unidimensional (Conv1D) de tamaño 600 elementos, seguida de una capa de submuestreo (Pool1D) con una máscara de tipo *max pooling* de 4 elementos. Las características aprendidas pasan a continuación por una capa densa de 20 neuronas, previa a la capa final. En cuanto a las funciones de activación, se ha utilizado la función *softmax* en la capa de salida y la función ReLU en el resto de capas.

Esta arquitectura fue entrenada por medio del algoritmo de descenso del gradiente [80] en combinación con el optimizador ADAM [97], durante 100 épocas o iteraciones, utilizando un *mini-batch* de 40 muestras y tomando la función de entropía cruzada categórica como función de coste del modelo. Para ello, el conjunto de datos de trabajo fue dividido aleatoriamente en un subconjunto de entrenamiento (70 % de las muestras) y otro de test (30 % de las muestras): el conjunto de entrenamiento fue empleado en este proceso de aprendizaje, mientras que el conjunto de test fue utilizado para evaluar el rendimiento del modelo.

En cuanto a la elección de los hiperparámetros del modelo (tamaño del filtro

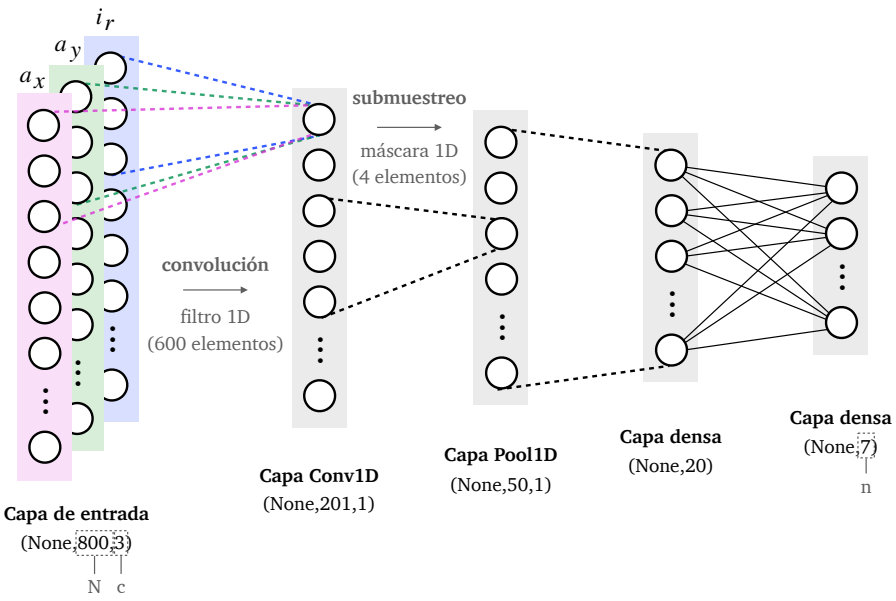


Figura 3.6: Modelo CNN propuesto. La dimensión del *batch* (número de muestras utilizadas en cada iteración del entrenamiento) es un parámetro no prefijado en el modelo y por ello se denota habitualmente como *None*.

de convolución, tamaño de la máscara de submuestreo, número de capas densas, número de neuronas en las capas densas, número de épocas, tamaño del *mini-batch*, etc.), se han ejecutado diferentes abanicos de experimentos y se han elegido aquellos hiperparámetros con los que el modelo ha demostrado mejores resultados de clasificación en términos de exactitud o *accuracy*. Finalmente, cabe destacar que la arquitectura de red ha sido implementada utilizando la combinación de librerías Keras-Tensorflow, con lo que el modelo CNN ha sido descrito en esta sección de acuerdo al estándar de dichas librerías.

3.3. Resultados

El modelo CNN expuesto tiene como objetivo determinar la condición de la máquina bajo estudio, a partir de sus datos de operación. Para ello, el modelo aprende a extraer características relevantes de los datos, que a continuación utiliza para clasificar el estado de funcionamiento de la máquina. En esta sección, se presentan tanto los resultados de la clasificación como las características aprendidas por el modelo.

3.3.1. Resultados de la clasificación

En la Figura 3.7 se muestran los resultados de la clasificación en términos de la matriz de confusión. En esta figura podemos observar que el modelo determina la condición de la máquina con una exactitud del 100 % para todas las clases en el conjunto de entrenamiento, al igual que en el conjunto de test, salvo para el ensayo T4, donde la exactitud cae al 90 %. Por tanto, el enfoque propuesto tiene

un acierto del 100 % en los datos de entrenamiento y del 98 % en los datos de test.

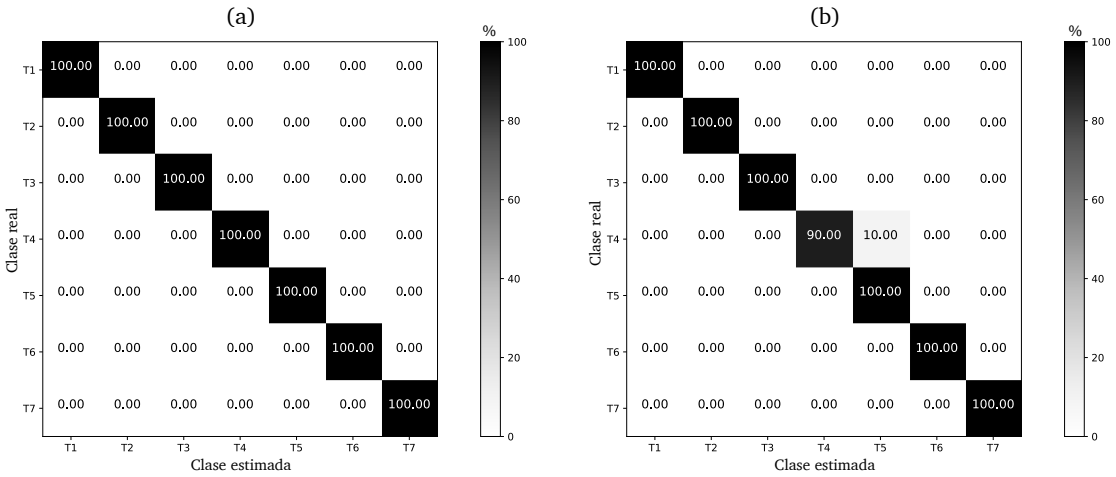


Figura 3.7: Matriz de confusión de los resultados de clasificación del modelo CNN propuesto: (a) resultados en el conjunto de entrenamiento; (b) resultados en el conjunto de test.

Tomando otras particiones aleatorias del conjunto de datos original para la creación de los subconjuntos de entrenamiento y test, se obtienen unas exactitudes promedio del 100 % (σ=0.00) y del 98.43 % (σ=0.88), respectivamente, como se observa en la Tabla 3.3. Por tanto, a la vista de los resultados obtenidos, podemos afirmar que el modelo propuesto es capaz de clasificar satisfactoriamente el estado de funcionamiento de la máquina.

3.3.1.1. Comparativa con métodos basados en ingeniería de características

El rendimiento del modelo ha sido comparado con el de otros clasificadores convencionales, basados en características diseñadas manualmente. Para este propósito, se ha extraído un conjunto de características representativas de la máquina, a partir de sus datos de operación y de acuerdo al conocimiento a priori disponible sobre el sistema. Se trata de las cinco características detalladas a continuación: el valor RMS de a_x y de a_y en la banda de frecuencia 20-30 Hz (esta banda, centrada en 1$\times$ frecuencia de rotación de la máquina, está asociada a los desequilibrios mecánicos); el valor RMS de a_x y de a_y en la banda de frecuencia 95-105 Hz (esta banda, centrada en 2$\times$ frecuencia de alimentación de la máquina, está asociada a los desequilibrios eléctricos); y valor RMS de i_r en la banda de frecuencia 45-55 Hz (esta banda, centrada en 1$\times$ frecuencia de alimentación de la máquina, se relaciona con fallos en la alimentación de la máquina). Los clasificadores empleados en la comparativa reciben un vector de entrada con estas cinco características y devuelven como salida la condición de la máquina (Figura 3.8).

En detalle, se han considerado los siguientes tipos de clasificadores: (1) Perceptrón multicapa o MLP, con 4 capas ocultas de 20 neuronas cada una; (2) Clasificador de vectores de soporte o SVC, con *kernel* lineal y un parámetro de penalización del error $C = 1000$; (3) SVC con *kernel* polinomial, de grado 3, penalización $C = 10$ y un coeficiente del *kernel* $\gamma = 10$; (4) SVC con *kernel* RBF *(Radial Basis*

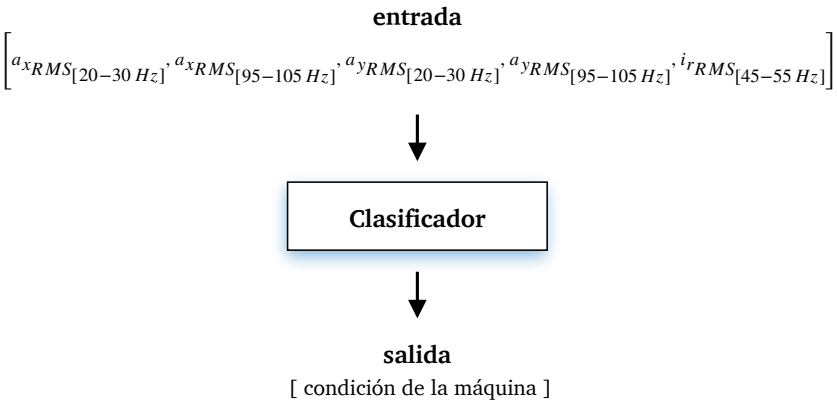


Figura 3.8: Contexto de trabajo de los clasificadores empleados en la comparativa.

Function), con $C = 10$ y $\gamma = 10$; (5) Clasificador de bosque aleatorio o RFC, con una profundida máxima de nivel 6. El rendimiento del modelo CNN, junto con el de estos clasificadores, se presenta en la Tabla 3.3, en términos de las siguientes métricas:

- Exactitud o *accuracy* (ratio de predicciones correctas, sobre el total de predicciones), precisión (ratio de predicciones positivas correctas, sobre el total de predicciones positivas), *recall* (ratio de predicciones positivas correctas, sobre el total de muestras positivas) y *f1-score* (promedio ponderado de precisión y *recall*). Estas métricas de error pueden ser obtenidas a partir de la matriz de confusión, utilizando las expresiones (3.8) a (3.11), donde: VP es la cantidad de verdaderos positivos en la clasificación, VN es la cantidad de verdaderos negativos, FP es la cantidad de falsos positivos y FN es la cantidad de falsos negativos.

$$exactitud = \frac{VP + VN}{VP + VN + FP + FN} \tag{3.8}$$

$$precisión = \frac{VP}{VP + FP} \tag{3.9}$$

$$recall = \frac{VP}{VP + FN} \tag{3.10}$$

$$f1 - score = 2 \cdot \frac{precisión \cdot recall}{precisión + recall} \tag{3.11}$$

- Número de parámetros del modelo, que sirve como indicador de la complejidad computacional del modelo.
- En último lugar, el modelo es evaluado en términos de su eficiencia computacional. Para ello, se han registrado dos tiempos: el tiempo que el modelo requiere para completar el entrenamiento y el tiempo que invierte en clasificar una nueva muestra entrante.

Como se observa en la Tabla 3.3, el modelo CNN ha obtenido excelentes resultados en todas las métricas derivadas de la matriz de confusión: exactitud del 98.43 %, precisión del 98.40 %, *recall* del 98.40 % y *f1-score* del 98.40 %. No obstante, estos resultados son semejantes a los obtenidos por parte de los clasificadores convencionales. También se observa que el modelo CNN es el enfoque más complejo —consta de un elevado número de parámetros a optimizar durante el entrenamiento— y el más demandante desde el punto de vista computacional —requiere mayor tiempo que el resto de clasificadores tanto para completar el entrenamiento como para clasificar una nueva muestra entrante.

En vista de estos resultados, podría parecer que el modelo profundo no presenta grandes ventajas frente a los enfoques tradicionales. Sin embargo, cabe recordar que el modelo CNN ha demostrado ser tan competitivo en la clasificación del estado de funcionamiento del sistema como dichos enfoques tradicionales, pero trabajando en un contexto menos favorable, sin ningún tipo de información a priori sobre la máquina y aprendiendo por sí mismo las características más relevantes para llevar a cabo la clasificación. Además, el consumo de recursos computacionales, aunque superior al del resto de clasificadores, no es elevado y permitiría una monitorización en tiempo real de la máquina.

Tabla 3.3: Rendimiento del modelo CNN en comparación con otros clasificadores convencionales, utilizando la técnica de validación cruzada con cinco ejecuciones (se muestran la media y desviación típica de todas las ejecuciones).

	Exactitud[1]	Precisión[1]	Recall[1]	f1-score[1]	Número de parámetros[2]	Rendimiento computacional	
						Entrenamiento (s)	Test (ms)
Modelo CNN propuesto	98.43 % (σ=0.88)	98.40 % (σ=0.89)	98.40 % (σ=0.89)	98.40 % (σ=0.89)	2968.00 (σ=0.00)	16.38 (σ=1.08)	4.32 (σ=0.38)
MLP	100.00 % (σ=0.00)	100.00 % (σ=0.00)	100.00 % (σ=0.00)	100.00 % (σ=0.00)	1527.00 (σ=0.00)	4.08 (σ=0.31)	2.77 (σ=2.19)
SVC (lineal)	100.00 (σ=0.00)	100.00 (σ=0.00)	100.00 (σ=0.00)	100.00 (σ=0.00)	114.00 (σ=3.74)	$2.84\text{x}10^{-3}$ ($\sigma=1.70\text{x}10^{-3}$)	0.36 (σ=0.16)
SVC (polinomial)	100.00 % (σ=0.00)	100.00 % (σ=0.00)	100.00 % (σ=0.00)	100.00 % (σ=0.00)	320.00 (σ=19.75)	$2.09\text{x}10^{-3}$ ($\sigma=6.38\text{x}10^{-4}$)	0.40 (σ=0.33)
SVC (RBF)	99.22 % (σ=1.57)	99.40 % (σ=1.20)	99.20 % (σ=1.60)	99.20 % (σ=1.60)	332.00 (σ=23.37)	$1.95\text{x}10^{-3}$ ($\sigma=3.47\text{x}10^{-4}$)	0.28 ($\sigma=7.56\text{x}10^{-2}$)
RFC	100.00 % (σ=0.00)	100.00 % (σ=0.00)	100.00 % (σ=0.00)	100.00 % (σ=0.00)	10.00 (σ=0.00)	$1.50\text{x}10^{-2}$ ($\sigma=1.31\text{x}10^{-3}$)	1.92 (σ=0.63)

[1] Métricas de clasificación sobre el conjunto de test. En el conjunto de entrenamiento, los clasificadores han conseguido un rendimiento del 100.00 % (σ=0.00) en todas las métricas.
[2] En el modelo CNN y en el MLP, hace referencia al número de parámetros del modelo. En los modelos SVC, hace referencia a su número de coeficientes. En el RFC, se trata del número de árboles en el bosque.

3.3.1.2. Resultados sobre otro conjunto de datos

Los clasificadores basados en ingeniería de características son altamente dependientes del problema objetivo y, por tanto, son también difícilmente extrapolables a otros contextos. En cambio, el modelo CNN podría ser empleado en la monitorización de la condición de nuevas máquinas, tan solo siendo entrenado con datos de operación de las mismas, gracias a que tiene la capacidad de extraer por sí mismo las características de interés necesarias para llevar a cabo la clasificación.

Con el propósito de explorar el potencial del modelo CNN en este contexto, se ha evaluado su rendimiento sobre otro conjunto de datos [141]. El *bearing fault dataset* contiene datos de la aceleración radial de una máquina rotativa (a_r), operando ante diferentes condiciones y escenarios de trabajo: tres escenarios de operación en condiciones normales y misma carga; diez escenarios de fallo en la pista exterior del rodamiento ante diferentes cargas; y siete escenarios de fallo en la pista interior del rodamiento ante diferentes cargas (Tabla 3.4). Estos escenarios han sido seleccionados —de forma aleatoria— para construir los conjuntos de entrenamiento y test. En detalle, se ha utilizado un escenario de funcionamiento normal, tres de fallo en la pista exterior y dos de fallo en la pista interior, para crear el conjunto de test; el resto de escenarios han sido empleados para constituir el conjunto de entrenamiento.

Tabla 3.4: Contenido del conjunto de datos *bearing fault dataset*.

Condición de la máquina	Escenario (Carga en lbs, número de muestras)
Operación normal (N)	N1(270,292968), N2(270,292968), N3(270,292968)
Fallo en la pista exterior (O)	O1(25,146484), O2(50,146484), O3(100,146484), O4(150,146484), O5(200,146484), O6(250,146484), O7(270,292968), O8(270,292968), O9(270,292968), 10(300,146484)
Fallo en la pista interior (I)	I1(0,146484), I2(50,146484), I3(100,146484), I4(150,146484), I5(200,146484), I6(250,146484), I7(300,146484)

Este conjunto de datos ha sido normalizado y enventanado, siguiendo el mismo procedimiento descrito en la Sección 3.2.1. A continuación, se ha entrenado un modelo CNN con estos datos. El modelo empleado tiene los mismos hiperparámetros (tamaño del filtro de convolución, tamaño de la máscara de submuestreo, número de capas densas, número de neuronas en las capas densas, número de épocas, tamaño del *mini-batch*, etc.) que el modelo presentado en la Sección 3.2.2. Tan solo ha sido necesario modificar su arquitectura en las capas de entrada y salida, para ajustar el número de canales de entrada ($c = 1$ en lugar de $c = 3$) y el número de neuronas de salida ($n = 3$ en lugar de $n = 7$) (Figura 3.9). Los resultados de la clasificación se muestran en la Tabla 3.5 (b).

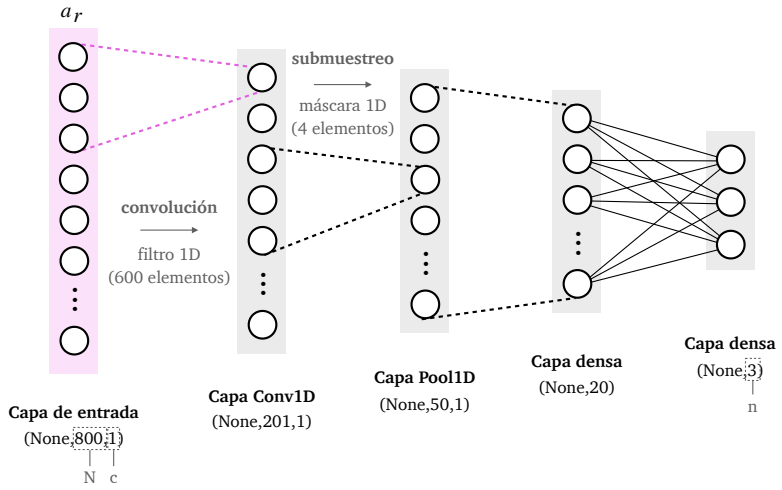


Figura 3.9: Modelo CNN adaptado a otro conjunto de datos.

La Tabla 3.5 compara los resultados de la clasificación para los dos conjuntos de datos empleados en los experimentos: (a) el conjunto *dataicann*, presentado en la Sección 3.2.1 y en base al cual ha sido diseñado el modelo de red; (b) el conjunto *bearing fault dataset*, presentado en esta sección. Como se aprecia en la tabla, el sistema demuestra buenos resultados en la clasificación de este nuevo conjunto de datos (b), con una exactitud del 91.81 % en el conjunto de test, a pesar de haber sido entrenado con los mismos hiperparámetros elegidos para el conjunto (a). Estos resultados son una evidencia del enorme potencial de los enfoques *deep learning* en el ámbito de los sistemas de ingeniería, que en este caso nos han permitido monitorizar con éxito la condición de dos máquinas diferentes —utilizando la misma arquitectura base— y sin necesidad de ningún tipo de conocimiento previo sobre las mismas.

Tabla 3.5: Resultados del modelo CNN para los dos conjuntos de datos, utilizando la técnica de validación cruzada con cinco ejecuciones (se muestran la media y desviación típica de todas las ejecuciones).

	Exactitud[1]	Precisión[1]	Recall[1]	f1-score[1]
(a) Dataicann dataset	98.43 % (σ=0.88)	98.40 % (σ=0.89)	98.40 % (σ=0.89)	98.40 % (σ=0.89)
(b) Bearing fault dataset	91.81 % (σ=5.88)	93.60 % (σ=4.22)	91.60 % (σ=5.77)	91.20 % (σ=6.69)

[1] Métricas de clasificación sobre el conjunto de test. En el conjunto de entrenamiento, el modelo CNN ha conseguido un rendimiento del 100.00 % (σ=0.00) en todas las métricas.

3.3.2. Características aprendidas por el modelo

La capa de convolución del modelo es optimizada durante el proceso de entrenamiento, para aprender a extraer características relevantes de los datos de entrada. Esta capa implementa, por tanto, una transformación de los datos que será de vital importancia para el éxito de la red y cuyo estudio hemos abordado en esta sección.

En la Figura 3.10 se muestra el vector de salida de la capa de convolución, para cada muestra de los datos, donde se aprecia que las muestras de un mismo ensayo presentan características similares entre sí. Por tanto, la capa de convolución aprende una representación de los datos de entrada que parece ser útil para la posterior clasificación del estado de la máquina. Cabe destacar también que los estados más confusos parecen ser T4/T5/T6, lo cual es consistente con los resultados de las matrices de confusión (Figura 3.7), donde el modelo confundía las muestras asociadas a los ensayos T4/T5.

Esta capa de convolución consta de un filtro convolucional 1D de tres canales, responsable de transformar los datos de entrada en vectores de características. Este filtro aprende a resaltar aquellos armónicos que portan información útil para la tarea de clasificación, con lo que su respuesta en frecuencia podría revelar valiosa información acerca de la máquina. Una vez finalizado el entrenamiento, el filtro queda definido por tres vectores de pesos que, para su visualización y correspondiente análisis, hemos transferido al dominio de la frecuencia. Para ello, hemos

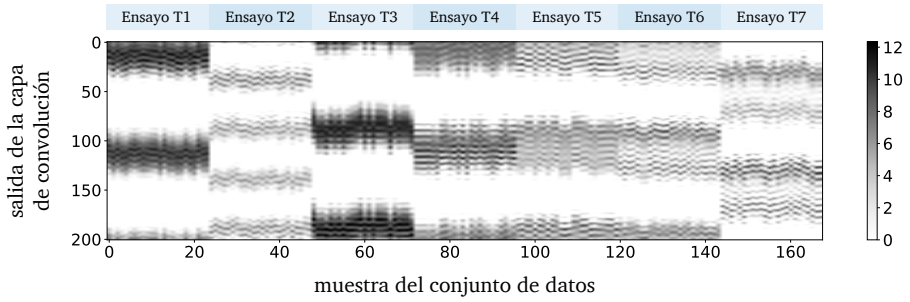


Figura 3.10: Salida de la capa de convolución para cada muestra del conjunto de datos *dataicann*.

utilizado la FFT *(Fast Fourier Transform)* (3.12), donde $\mathbf{x}_n$ representa el vector de pesos y N_f es el tamaño del filtro ($N_f = 600$). Siguiendo este enfoque, se presenta en la Figura 3.11 la respuesta en frecuencia del filtro aprendido por el modelo CNN, donde cada vector de pesos está asociado a un canal de entrada diferente: a_x (aceleración horizontal), a_y (aceleración vertical) e i_r (corriente en fase R).

$$\mathbf{x}_k = \sum_{n=0}^{N_f-1} \mathbf{x}_n e^{-i2\pi kn/N_f} \qquad k = 0, \ldots, N_f - 1. \tag{3.12}$$

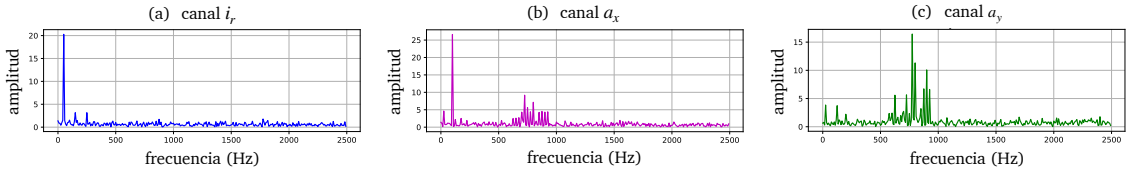


Figura 3.11: Respuesta en frecuencia del filtro de convolución aprendido por el modelo CNN propuesto: (a) canal i_r, (b) canal a_x, (c) canal a_y.

Dado que el filtro de convolución extrae las frecuencias relevantes para la tarea de clasificación, sus vectores de pesos (Figura 3.11) contienen gran cantidad de información acerca de la máquina bajo estudio y su análisis permite deducir parámetros constructivos y operativos de la máquina, como se detalla a continuación:

- En el canal i_r (Figura 3.12) podemos identificar la frecuencia de alimentación de la máquina (f_{PS}) como frecuencia fundamental del filtro. Esto indica que el filtro es capaz de detectar frecuencias representativas de la máquina y, coherentemente, identificarlas como características relevantes para la discriminación de su estado de funcionamiento. Adicionalmente, se observan armónicos de esta misma frecuencia a $3f_{PS}$ y $5f_{PS}$.

- En el canal a_x (Figura 3.13) está de nuevo presente la frecuencia de alimentación $(2f_{PS})$, junto con otras frecuencias relevantes, como la frecuencia de paso de las bolas por la pista exterior del rodamiento $(BPFO)$ o la velocidad de rotación de la máquina (f_R). Estas frecuencias aparecen juntas en la zona sombreada de la figura, donde existe una modulación de alta frecuencia

con bandas laterales f_R, siendo $BPFO$ la distancia entre las dos frecuencias portadoras.

- En el canal a_y (Figura 3.14) podemos observar las frecuencias ya mencionadas ($4f_{PS}$,f_R), así como la frecuencia de paso de las bolas por la pista interior del rodamiento ($BPFI$), que está presente tanto a baja como a alta frecuencia.

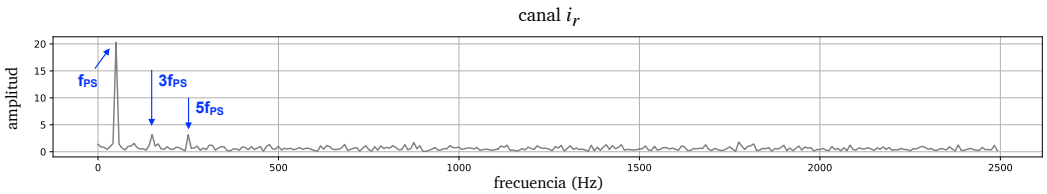


Figura 3.12: Análisis del canal i_r del filtro de convolución.

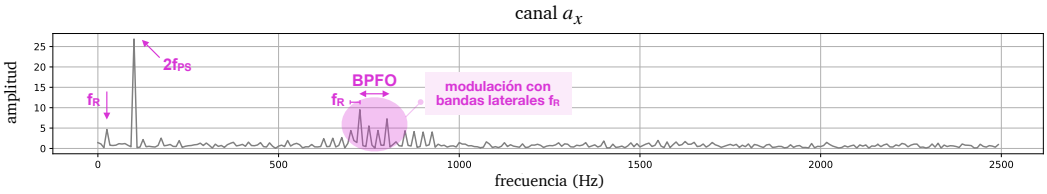


Figura 3.13: Análisis del canal a_x del filtro de convolución.

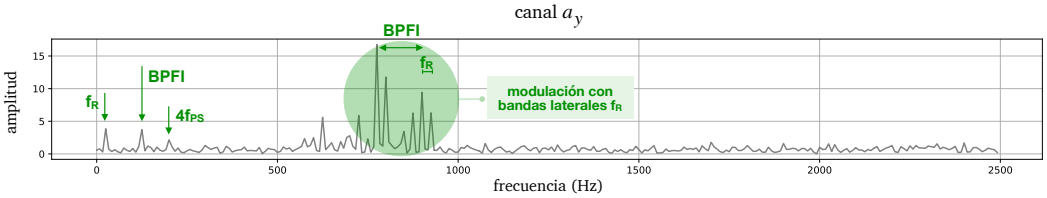


Figura 3.14: Análisis del canal a_y del filtro de convolución.

Tras este análisis frecuencial del filtro de convolución, podemos afirmar que el modelo propuesto ha sido capaz de identificar las frecuencias características del sistema, cuyos valores pueden ser extraídos de las figuras previas ($f_{PS} = 50\,Hz$, $f_R = 25\,Hz$, $BPFI = 125\,Hz$ y $BPFO = 75\,Hz$). A partir de estos parámetros operativos, es posible deducir también parámetros constructivos de la máquina, como el número de bolas en los rodamientos (n) o el ratio entre los diámetros de la bola y la pista (d/D). Empleando las frecuencias extraídas de los filtros y las expresiones (3.4-3.7), se ha obtenido un número de ocho bolas en los rodamientos ($n = 8$) y un ratio entre diámetros de 0.25 ($d/D = 0.25$), ambos de los cuales se corresponden con la realidad, como se muestra en la Figura 3.15.

Como resultado, el modelo CNN no solo ha permitido monitorizar la máquina sin necesidad de conocimiento previo sobre la misma, sino que el estudio de las características aprendidas por su filtro de convolución ha proporcionado información a priori desconocida, como frecuencias características o parámetros constructivos de la máquina. Por tanto, la introspección en los modelos profundos podría proporcionar información útil para la mejora de la compresión de los sistemas bajo

estudio, facilitando un diagnóstico no invasivo de los mismos. Además, esta introspección aporta luz a las transformaciones de los datos que tienen lugar en las capas internas de la red, ayudando con ello a aumentar la confianza del usuario en los resultados obtenidos.

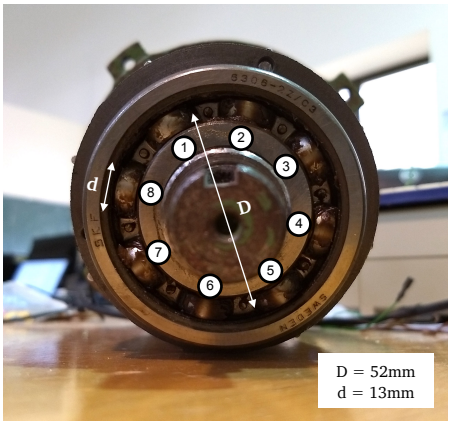


Figura 3.15: Geometría de los rodamientos de la máquina.

3.4. Conclusiones

A lo largo de este capítulo se ha explorado el potencial de las redes convolucionales (CNNs) para la detección de fallos en máquinas rotativas. Para ello, se ha propuesto un modelo CNN capaz de clasificar el estado de funcionamiento de la máquina a partir de datos crudos de operación —vibraciones y corrientes— de la misma. Dicho enfoque integra en una única arquitectura las dos tareas de los enfoques tradicionales de detección de fallos: extracción de características y clasificación de las características extraídas. Esto se debe a que el modelo CNN tiene la habilidad de aprender por sí mismo una representación óptima de los datos de entrada para, en base a ella, llevar a cabo la clasificación. Por tanto, al tener la capacidad para aprender —extraer automáticamente— características de los datos, se elimina la necesidad de recurrir a métodos de ingeniería de características, que requieren experiencia y conocimiento de dominio del sistema para monitorizar su estado de funcionamiento.

El modelo CNN propuesto ha sido utilizado para clasificar el estado de una máquina con siete posibles estados de operación y se ha comparado su rendimiento con el de otros clasificadores basados en características extraídas manualmente. Los resultados obtenidos indican que el modelo CNN es capaz de determinar la condición de la máquina con un acierto del 98 %, mostrando un rendimiento semejante al de los enfoques tradicionales. Adicionalmente, el análisis de las características aprendidas por el modelo ha revelado parámetros operativos y constructivos de la máquina, como su velocidad de rotación o el número de bolas en los rodamientos. En último lugar, este enfoque ha sido empleado en la monitorización de otra máquina rotativa, donde también ha obtenido buenos resultados de clasificación.

En conclusión, el modelo profundo nos ha permitido monitorizar con éxito la

condición del sistema, sin necesidad de experiencia o conocimiento de dominio del problema y proporcionando, además, información sobre la máquina. Este enfoque también se ha mostrado competitivo en la monitorización de una máquina diferente, demostrando que el éxito del modelo podría ser fácilmente extrapolable a nuevos contextos de trabajo. Por tanto, estos resultados son una evidencia del potencial de las arquitecturas profundas como valiosas herramientas para la detección y diagnóstico de fallos en sistemas de ingeniería.

CAPÍTULO 4

Detección de anomalías

En este capítulo se aborda el uso de arquitecturas profundas para la detección de anomalías en procesos de ingeniería. En detalle, se presenta el uso de un *variational autoencoder*, en combinación con un enfoque de análisis de residuos, para la detección de comportamientos anómalos en diferentes contextos de ingeniería (se analiza el caso de un motor, un sistema hidráulico y un sistema de monitorización del movimiento humano). El capítulo comienza con una revisión del estado del arte, para detallar a continuación los experimentos realizados y los resultados obtenidos, terminando con un apartado de conclusiones.

El contenido de este capítulo ha sido publicado en la revista *Computers and Electrical Engineering*, bajo el título *«Two-step residual-error based approach for anomaly detection in engineering systems using variational autoencoders»* [142].

4.1. Antecedentes

La detección de anomalías consiste en encontrar aquellos patrones en los datos que no se ajustan al comportamiento esperado del proceso. Dichos patrones, comúnmente conocidos como anomalías o *outliers*, representan desviaciones del comportamiento normal, con lo que su detección no solo resulta de gran valor, sino que es crítica en una amplia variedad de aplicaciones, como ocurre en problemas de detección de intrusos [143], detección de fraudes [144] o detección de anomalías médicas [145]. En el ámbito de los sistemas de ingeniería, la detección de anomalías resulta también de gran importancia, pues permite optimizar el funcionamiento de los procesos, así como garantizar la seguridad en la operación de los mismos [102].

Dada su relevancia, podemos encontrar numerosos enfoques de detección de anomalías en la literatura, los cuales son, generalmente, altamente dependientes del tipo de información disponible en el conjunto de datos de trabajo [146]. En los sistemas de ingeniería, los procesos bajo estudio operan la mayor parte del tiempo en condiciones normales de funcionamiento, con lo que los registros o muestras disponibles acerca de un proceso se corresponden en su mayoría con di-

cho comportamiento normal. Por ello, en el escenario habitual, los *datasets* están compuestos por un gran número de muestras de comportamiento normal, junto con algunas muestras de comportamiento anómalo, que, a menudo, ni siquiera son representativas de todos los posibles modos de anomalía del sistema. En este contexto, los enfoques de clasificación multiclase —como podría ser el modelo CNN expuesto en el Capítulo 3— no resultan adecuados para la detección de anomalías, puesto que no se dispone de muestras representativas de cada uno de los posibles estados de funcionamiento del sistema [147].

En su lugar, un enfoque habitual en la literatura consiste en construir un modelo del comportamiento normal del proceso (en base a la gran cantidad de muestras normales disponibles), de manera que la detección de anomalías en nuevas muestras entrantes dependa de cuánto se desvíen estas de dicho modelo de normalidad [147]. Por tanto, el objetivo de este enfoque es construir un modelo analítico del proceso bajo estudio, de manera que su comparación con el proceso real permita sacar a la luz posibles comportamientos anómalos en el sistema. Se trata, en definitiva, de una comparativa en términos de residuos, donde las muestras con residuos elevados son consideradas más susceptibles de ser anómalas.

El modelado del comportamiento normal del proceso ha sido abordado tradicionalmente mediante modelos diseñados en base a la experiencia y conocimiento de dominio del sistema a modelar [148, 149, 150]. En contraste, existe un interés creciente en el uso de métodos basados en datos, especialmente con la proliferación en los últimos años de las técnicas *deep learning* [151]. En el contexto de los sistemas de ingeniería, podemos encontrar varios trabajos en la literatura que proponen el uso de *deep autoencoders* para la detección de anomalías [31, 152, 153], donde muestras con errores de reconstrucción —residuos— elevados son clasificadas como muestras anómalas. Aunque los detalles de la decisión de anomalía varían en cada propuesta, dos enfoques comunes para la clasificación de los residuos son: (1) el enfoque basado en umbral, que consiste en clasificar los residuos por comparación con un *umbral de anomalía* (error de reconstrucción por encima del cual una muestra es considerada anómala) previamente definido [31, 154, 155, 156]; y (2) el enfoque basado en clasificador, que propone entrenar una arquitectura adicional para la clasificación automática de los residuos [90, 157, 158]. Un ejemplo de (1) se presenta en [154], donde el error de reconstrucción máximo sobre los datos de entrenamiento es elegido como umbral de anomalía; otra propuesta se encuentra en [156], donde los autores toman un percentil del error de reconstrucción como umbral de anomalía. En [157] se observa un ejemplo de (2), donde una máquina de vectores de soporte *(Support Vector Machine, SVM)* de una sola clase, o *one-class*, es entrenada para clasificar las muestras entrantes en base a sus residuos.

A pesar del gran interés en los enfoques *deep learning*, nuevas técnicas han emergido recientemente cuyo potencial ha sido aún poco explorado. Este es el caso del autoencoder variacional *(variational autoencoder, VAE)* [91], que se ha posicionado como un valioso algoritmo de aprendizaje no supervisado, demostrando resultados prometedores en tareas generativas y con aplicaciones en una amplia variedad de ámbitos, como pueden ser el procesamiento de audio, texto o imagen [28, 30, 159, 160]. En contraste con los autoencoders convencionales, el VAE impone restricciones en la distribución del espacio latente y, con ello, es capaz

de aprender la función de densidad de probabilidad de los datos de entrenamiento. Por tanto, siendo entrenado con muestras representativas del comportamiento normal o saludable del proceso, el VAE puede llegar a aprender la distribución *saludable* de los datos, convirtiéndose así en una poderosa herramienta de detección de anomalías [29, 161, 162].

En este contexto, los residuos del VAE se convierten en poderosos indicadores de anomalía, pues capturan cualquier desviación del comportamiento normal y portan, por tanto, valiosa información acerca del estado de salud del proceso. En consecuencia, estos residuos han empezado a ser empleados en enfoques de detección de anomalías [163, 164] que, en línea con los trabajos basados en *deep autoencoders*, proponen clasificar los residuos por comparación con un umbral de anomalía [165] o mediante el uso de clasificadores de una sola clase [166]. La aplicabilidad de estas ideas en el ámbito de la ingeniería ha comenzado a ser explorada recientemente [167, 168] y, dado su potencial, se espera que el VAE desempeñe un papel clave en el futuro de los algoritmos de monitorización de la salud [22].

Llegado este punto, cabe recordar que las técnicas basadas en análisis de residuos han demostrado en la literatura ser capaces de abordar con éxito, y en una amplia variedad de ámbitos, la crítica tarea de detectar anomalías en los datos. Para ello, se requiere de residuos con gran significado acerca del proceso y es precisamente en este contexto donde los residuos del VAE podrían tener un gran impacto. Como se apunta en la literatura [169], cuanto mejor aproxime el modelo los datos de entrenamiento, más significativos serán sus residuos. En este contexto, la habilidad del VAE para obtener estimaciones precisas de la distribución de los datos de entrenamiento se traduce, entre otros beneficios, en valiosos residuos, de gran interés en aplicaciones de detección y diagnóstico de anomalías. A lo largo de esta investigación, hemos tratado de explotar este potencial mediante un novedoso algoritmo de clasificación que, a partir de los residuos del VAE, ha permitido identificar no solo la naturaleza normal/anómala de las muestras, sino también la de cada una de sus componentes, obteniendo así una visión más completa del estado de salud de los procesos.

El enfoque propuesto, descrito a continuación, consiste en un VAE —entrenado con muestras de comportamiento normal del proceso— y un algoritmo de clasificación —encargado de clasificar las muestras entrantes en base a su error de reconstrucción. Este algoritmo, llamado *two-step classifier*, clasifica cada muestra en dos pasos: (1) clasificación por componentes o *component-wise* (clasificación de cada elemento —componente— en la muestra) y (2) clasificación global (clasificación de la muestra). Esta propuesta ha sido evaluada en diferentes contextos de ingeniería (se ha analizado el caso de un motor, un sistema hidráulico y un sistema de monitorización del movimiento humano) y su rendimiento ha sido comparado con el de otros enfoques del estado del arte, tanto en términos de modelado (autoencoder variacional vs. *deep autoencoder*) como de clasificación (*two-step classifier* vs. clasificadores convencionales). Los resultados de la investigación han demostrado la capacidad del modelo para detectar con éxito la presencia de anomalías en los tres contextos de trabajo, obteniendo un mejor rendimiento que los enfoques del estado del arte. Adicionalmente, se ha realizado un análisis visual de los resultados, que ha permitido extraer información de interés sobre el sistema a

partir de la clasificación *component-wise* de sus muestras, aportando también luz a la decisión de anomalía del algoritmo y contribuyendo con ello a una mayor interpretabilidad de los resultados obtenidos.

4.2. Método propuesto

En esta sección se presenta el método de detección de anomalías empleado a lo largo de la investigación, que consta de dos elementos: (1) un modelo de normalidad del proceso, construido a partir de muestras normales, de manera que los residuos de las nuevas muestras entrantes sirvan como una medida de su desviación respecto al comportamiento normal esperado; (2) un algoritmo de clasificación, encargado de clasificar las muestras entrantes a partir de sus residuos. Como se ilustra en la Figura 4.1, el modelo recibe una muestra $\mathbf{x}$ y devuelve su reconstrucción $\hat{\mathbf{x}}$; a continuación, el clasificador analiza los residuos de la muestra $|\mathbf{x} - \hat{\mathbf{x}}|$ y devuelve la clasificación de la misma, junto con la clasificación de sus componentes (tal como se indica en la figura). Ambos elementos —el modelo del proceso y el algoritmo de clasificación— serán presentados a continuación, junto con la descripción de los conjuntos de datos utilizados en los experimentos.

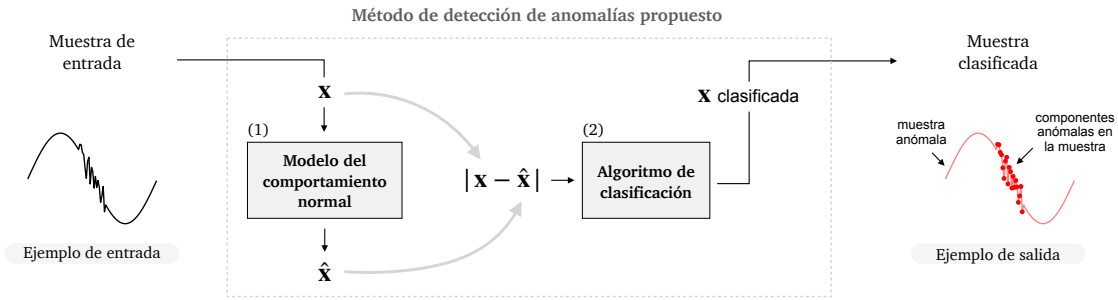


Figura 4.1: Método propuesto para la detección de anomalías.

4.2.1. Conjuntos de datos

El método propuesto ha sido evaluado sobre tres conjuntos de datos diferentes, elegidos para explorar el potencial de nuestro enfoque en un amplio rango de contextos de ingeniería, que cubren desde la monitorización de la salud en máquinas e instalaciones, hasta el análisis de la conducta humana (se ha contemplado el caso de un motor, un sistema hidráulico y un sistema de monitorización del movimiento humano, respectivamente). Estos conjuntos de datos, junto con su correspondiente preprocesamiento, serán descritos en los siguientes apartados.

Los estados seleccionados como representativos del comportamiento normal/anómalo del sistema también serán indicados a continuación. Esta selección de estados es diferente para cada contexto de ingeniería y ha sido tomada de manera conveniente, eligiendo como estados normales/anómalos aquellos que hemos considerado más apropiados según las particularidades de cada *dataset*. En las tablas 3.1, 4.2 y 4.6 se describen los estados de funcionamiento disponibles para cada

conjunto de datos y en las tablas 4.1, 4.4 y 4.7 se indica la selección de estados normales/anómalos considerada para nuestros experimentos.

4.2.1.1. Máquina rotativa: *dataicann*

Este conjunto de datos ha sido descrito previamente en la Sección 3.2.1, donde se presentaba la máquina de estudio (Figura 3.4), junto con los ensayos realizados (Tabla 3.1) y las variables medidas en cada uno de ellos (Tabla 3.2). En cuanto al preprocesamiento de los datos, se ha igualado el rango de las tres variables de operación, mediante un escalado min-max [140] de rango $[0, 1]$. A continuación, se ha realizado un enventanado de los datos, utilizando ventanas de tamaño 100 elementos y con un solapamiento del 50 %. Finalmente, las tres ventanas resultantes (una por variable) han sido concatenadas, dando lugar a las muestras de trabajo, de 300 elementos de tamaño y con información temporal de las tres variables bajo estudio. En cuanto a la selección de estados, los ensayos de naturaleza mecánica (T1, T2) han sido tomados como anómalos y el resto —el ensayo normal (T3), así como los ensayos de naturaleza eléctrica (T4, T5, T6, T7)— han sido considerados representativos del comportamiento normal de la máquina (Tabla 4.1).

Tabla 4.1: Subconjuntos de comportamiento normal y anómalo de la máquina rotativa (el tamaño del subconjunto está expresado como: número de muestras $\times$ número de elementos en las muestras).

	Comportamiento normal (T3, T4, T5, T6, T7)	Comportamiento anómalo (T1, T2)
Tamaño:	1990×300	796×300

4.2.1.2. Sistema hidráulico

En segundo lugar, se ha considerado un *dataset* que contiene datos de funcionamiento de un banco de pruebas hidráulico [170], constituido por un circuito primario de trabajo y otro secundario de refrigeración-filtración. Este sistema ejecuta ciclos de carga constante, ante diferentes condiciones de refrigeración (Tabla 4.2) y dispone de 17 variables de proceso (Tabla 4.3). El conjunto de datos ha sido normalizado mediante un escalado min-max [140] de rango $[0, 1]$ y se ha tomado el fallo total del circuito de refrigeración (T3) como representativo del comportamiento anómalo del sistema (Tabla 4.4).

Tabla 4.2: Ensayos en el sistema hidráulico.

ID Ensayo	Condición del circuito de refrigeración
T1	Eficiencia máxima
T2	Eficiencia reducida
T3	Fallo total

Tabla 4.3: Variables en el sistema hidráulico.

Variable	Descripción
$PS_1, PS_2, PS_3, PS_4, PS_5, PS_6$	Presión
EPS_1	Potencia del motor
FS_1, FS_2	Volumen de flujo
TS_1, TS_2, TS_3, TS_4	Temperatura
VS_1	Vibración
CE	Eficiencia de la refrigeración
CP	Potencia de la refrigeración
SE	Factor de eficiencia

Tabla 4.4: Subconjuntos de comportamiento normal y anómalo del sistema hidráulico (el tamaño del subconjunto está expresado como: número de muestras $\times$ número de elementos en las muestras).

	Comportamiento normal (T1, T2)	Comportamiento anómalo (T3)
Tamaño:	88380 $\times$ 17	43920 $\times$ 17

4.2.1.3. Sistema de monitorización del movimiento humano

En último lugar, se ha considerado el *dataset* MHEALTH [171], que contiene registros de 23 variables de proceso asociadas al movimiento y señales vitales (Tabla 4.5) de 10 voluntarios, mientras ejecutan 12 tipos de actividades físicas diferentes (Tabla 4.6). Cabe destacar que este *dataset* contiene muestras sin etiquetar, asociadas a tiempos muertos entre actividades físicas, que han sido descartadas para nuestros experimentos. El conjunto de datos resultante consta de 12 ensayos, donde cada ensayo (T1, T2,...,Tn) contiene las muestras registradas de los 10 voluntarios mientras ejecutan la actividad n. Estos datos han sido normalizados mediante un escalado min-max [140] de rango $[0, 1]$ y se han tomado las actividades de baja intensidad física (T1, T2, T3, T4) como representativas del comportamiento normal del sistema (Tabla 4.7).

Tabla 4.5: Variables en el sistema de monitorización del movimiento humano.

Variable	Descripción
a_{cx}, a_{cy}, a_{cz}	Aceleración en el pecho (ejes: x, y, z)
ecg_1, ecg_2	Señal del electrocardiograma (derivaciones: 1, 2)
a_{ax}, a_{ay}, a_{az}	Aceleración en el tobillo (ejes: x, y, z)
g_{ax}, g_{ay}, g_{az}	Velocidad angular en el tobillo (ejes: x, y, z)
m_{ax}, m_{ay}, m_{az}	Campo magnético en el tobillo (ejes: x, y, z)
$a_{lax}, a_{lay}, a_{laz}$	Aceleración en el antebrazo (ejes: x, y, z)
$g_{lax}, g_{lay}, g_{laz}$	Velocidad angular en el antebrazo (ejes: x, y, z)
$m_{lax}, m_{lay}, m_{laz}$	Campo magnético en el antebrazo (ejes: x, y, z)

Tabla 4.6: Ensayos en el sistema de monitorización del movimiento humano.

ID Ensayo	Actividad física
T1	De pie
T2	Sentado
T3	Tumbado
T4	Caminando
T5	Subiendo escaleras
T6	Flexiones de cintura
T7	Elevación de los brazos
T8	Flexiones de rodillas
T9	Ciclismo
T10	Trotar
T11	Correr
T12	Saltos adelante y atrás

Tabla 4.7: Subconjuntos de comportamiento normal y anómalo del sistema de monitorización del movimiento humano (el tamaño del subconjunto está expresado como: número de muestras $\times$ número de elementos en las muestras).

	Comportamiento normal (T1, T2, T3, T4)	Comportamiento anómalo (T5, T6, T7, T8, T9, T10, T11, T12)
Tamaño:	122880×23	220315×23

4.2.2. Modelo del comportamiento normal

Como se indicaba en la Figura 4.1, nuestro enfoque de detección de anomalías requiere de un modelo encargado de reconstruir los datos de entrada, para obtener así los residuos con los que a continuación será alimentado el algoritmo de clasificación. Para este propósito, hemos utilizado un *variational autoencoder*, entrenado con muestras representativas del comportamiento normal del proceso, de manera que los residuos de las nuevas muestras entrantes sirvan como medida de su desviación respecto al comportamiento normal esperado.

En detalle, fueron entrenados tres VAEs (uno por *dataset*) utilizando para ello el algoritmo de descenso del gradiente [80] en combinación con el optimizador RMSProp [96]. El número de épocas y el tamaño del *mini-batch* se indican en la Tabla 4.8, junto con la arquitectura de cada modelo (número de capas y número de neuronas por capa). En cuanto a las funciones de activación, se ha utilizado la función ReLU en todas las capas, excepto en la capa de salida y en el cuello de botella del modelo, donde se han empleado la función sigmoide y la función lineal, respectivamente.

A fin de estudiar el potencial del VAE, se ha comparado su rendimiento con el de un autoencoder convencional y, para ello, fueron entrenados tres *deep autoencoders* (uno por *dataset*) utilizando el algoritmo de descenso del gradiente [80] en combinación con el optimizador ADAM [96]. El número de épocas, el tamaño del *mini-batch* y la arquitectura de cada modelo se indican en la Tabla 4.9. En cuanto a las funciones de activación, se ha utilizado la función ReLU en todas las capas, excepto en la capa de salida y en el cuello de botella del modelo, donde se ha empleado la función de activación lineal.

Tabla 4.8: Arquitectura del VAE para cada *dataset*.

	Número de épocas	Tamaño *mini-batch*	Número de capas	Número de neuronas en las capas		
				Encoder	Cuello de botella	Decoder
Máquina rotativa	800	400	9	(300,60,30)	(2,2,2)	(30,60,300)
Sistema hidráulico	600	400	11	(17,20,40,20)	(2,2,2)	(20,40,20,17)
Movimiento humano	800	800	11	(23,40,40,40)	(2,2,2)	(40,40,40,23)

Tabla 4.9: Arquitectura del *deep autoencoder* para cada *dataset*.

	Número de épocas	Tamaño *mini-batch*	Número de capas	Número de neuronas en las capas		
				Encoder	Cuello de botella	Decoder
Máquina rotativa	800	100	7	(300,60,30)	(2)	(30,60,300)
Sistema hidráulico	200	300	9	(17,20,20,20)	(2)	(20,20,20,17)
Movimiento humano	400	600	9	(23,40,40,40)	(2)	(40,40,40,23)

En cuanto al proceso de entrenamiento de estas arquitecturas, el 70 % de las muestras normales han sido tomadas aleatoriamente para componer los conjuntos de entrenamiento. El 30 % restante de las muestras normales, junto con las muestras representativas del comportamiento anómalo, constituyen los conjuntos de test, que han sido empleados para evaluar el rendimiento del método. Finalmente, cabe destacar que para la elección de los hiperparámetros de cada modelo (número de capas, número de neuronas en las capas, número de épocas, tamaño del *mini-batch*, etc.), se han ejecutado diferentes abanicos de experimentos y se han elegido aquellos hiperparámetros con los que el modelo ha demostrado un menor error de reconstrucción.

4.2.3. Algoritmo de clasificación: *two-step classifier*

Al tratarse de una tarea crítica en una amplia variedad de ámbitos, la detección de anomalías ha sido objeto de numerosos estudios en la literatura, dando lugar a variadas propuestas de detección (basadas en análisis estadísticos, métodos de *clustering*, técnicas de clasificación, etc.) [146]. Un enfoque común consiste en abordar la detección de anomalías como un problema de clasificación de una sola clase, en el que se pretenden diferenciar las muestras anómalas (clase positiva, representada como 1) de las muestras normales (clase negativa, representada como 0). Para llevar a cabo esta clasificación, hemos propuesto un algoritmo de clasificación basado en umbral, que recibe como entrada los residuos de la muestra a clasificar y devuelve como salida la clasificación de la muestra. Más en detalle (Figura 4.2), la salida del clasificador para una muestra entrante $\mathbf{x} \in \mathbb{R}^n$ consiste en: un vector $\hat{\mathbf{y}} \in \mathbb{R}^n$, que contiene la clasificación por componentes o *component-wise* de la muestra; y un escalar $\hat{y}_{\text{global}}$, que representa la clasificación global de la muestra. Los dos pasos requeridos para conseguir esta clasificación son descritos a continuación.

1. **Clasificación *component-wise*.**
   En este primer paso se clasifica cada componente $\{x_j, j = 1, 2, \ldots, n\}$ de la muestra $\mathbf{x}$: si el residuo de la componente x_j excede su correspondiente

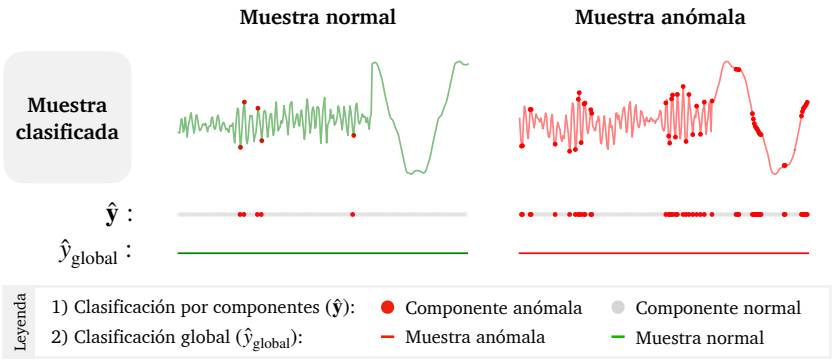


Figura 4.2: Resultados de la clasificación de dos muestras (normal y anómala) del conjunto de datos de la máquina rotativa.

umbral de anomalía th_j, la componente es considerada anómala; en caso contrario, la componente es clasificada como normal.

$$\hat{y}_j = \begin{cases} 1 & \text{if } |x_j - \hat{x}_j| > th_j \\ 0 & \text{otherwise} \end{cases} \tag{4.1}$$

La definición del umbral de anomalía th_j se hace en base a los residuos del conjunto de entrenamiento $\{\mathbf{X}_{\text{train}} = \left(x_{\text{train}_{ij}}\right) \in \mathbb{R}^{m \times n}\}$, que está constituido por muestras normales. Dado que se espera que los residuos de las muestras anómalas sean mayores que aquellos de las muestras normales, el umbral de cada componente th_j se define como el percentil 95 del error de reconstrucción en los datos de entrenamiento para la componente j:

$$th_j = 95\text{th}(|x_{\text{train}_{*j}} - \hat{x}_{\text{train}_{*j}}|) \tag{4.2}$$

De esta manera, la decisión de anomalía depende tan solo del comportamiento normal del proceso, del que a menudo hay grandes cantidades de datos disponibles, y no se requiere de ningún tipo de información previa acerca de los posibles modos de anomalía del sistema. Como umbral de anomalía se ha elegido el percentil 95 del error, que es un enfoque común [172, 173] y proporciona una medida representativa de los datos de entrenamiento, siendo más robusto ante la presencia de ruido en los datos que otras elecciones en la literatura (como, por ejemplo, el valor máximo del error de reconstrucción [154]). Cabe destacar que este percentil 95 puede ser sustituido por un percentil diferente, dependiendo del equilibrio deseado en la clasificación entre falsos positivos y falsos negativos (un umbral inferior incrementará el número de falsos positivos, mientras que un umbral superior incrementará el número de falsos negativos). En la Sección 4.3.1 analizaremos el impacto del percentil elegido en el rendimiento de la propuesta.

2. **Clasificación global.**
   En el segundo paso de la clasificación, asumimos que la naturaleza de una muestra depende del número de componentes normales/anómalas presentes

en ella[1]. Por tanto, la muestra $\mathbf{x}$ es clasificada en términos de su clasificación por componentes: si el número de componentes anómalas en la muestra excede el umbral de anomalía th_{global}, la muestra es considerada anómala; en caso contrario, la muestra es clasificada como normal (4.3).

$$\hat{y}_{\text{global}} = \begin{cases} 1 & \text{if } \sum_{j=1}^{n} \hat{y}_j > th_{\text{global}} \\ 0 & \text{otherwise} \end{cases} \tag{4.3}$$

El umbral global th_{global} representa el percentil 95 del número de componentes anómalas por muestra en el conjunto de entrenamiento:

$$th_{\text{global}} = 95\text{th}\left(\sum_{j=1}^{n} \hat{y}_{\text{train}_{*j}}\right) \tag{4.4}$$

Cabe destacar que los *datasets* utilizados en los experimentos incluyen la clasificación global de las muestras, pero no su clasificación por componentes, que es desconocida. Por este motivo, en (4.4) se opera en base a la clasificación estimada ($\hat{y}_{\text{train}}$), obtenida de acuerdo a la ecuación (4.1).

Estos dos pasos dan lugar al algoritmo de clasificación propuesto, el cual nos permite detectar muestras anómalas, pero también nos informa de la presencia de componentes anómalas en las muestras, proporcionando así un diagnóstico detallado acerca del estado de salud del sistema bajo estudio. Una visión general de este algoritmo se presenta en la Figura 4.3.

A fin de explorar el potencial del clasificador propuesto, su rendimiento ha sido comparado con el de otros enfoques populares del estado del arte: un clasificador basado en umbral [156, 168] y un clasificador de una sola clase [157, 166]. En línea con el primer enfoque, hemos utilizado un clasificador de umbral, que clasifica las muestras entrantes como anómalas si la norma L2 de sus residuos excede el umbral de anomalía (por motivos de comparación con el algoritmo *two-step*, se ha tomado como umbral de anomalía el percentil 95 de la norma L2 de los residuos en el conjunto de entrenamiento). Con respecto al segundo enfoque, hemos empleado un clasificador SVM de una clase con un *kernel* no lineal de tipo RBF, entrenado con los residuos del conjunto de entrenamiento y diferentes configuraciones para cada *dataset* (máquina rotativa: $\nu = 0.01, \gamma = 0.1$; sistema hidráulico: $\nu = 0.01, \gamma = 0.1$; movimiento humano: $\nu = 0.03, \gamma = 0.1$). Cabe destacar que las configuraciones elegidas son aquellas que obtuvieron mejores resultados en la clasificación (habiendo considerado posibles valores de ν y γ en el rango 0.001–10). En la siguiente sección, se presentan los resultados obtenidos por estos clasificadores, en comparación con el clasificador propuesto. Adicionalmente, se analizará

[1]Esta asunción ha permitido detectar anomalías con éxito en tres conjuntos de trabajo diferentes, como se refleja en la Sección 4.3. Sin embargo cabe señalar que, bajo esta consideración, muestras con anomalías de carácter impulsional serían rara vez clasificadas como anómalas, al manifestarse las anomalías en tan solo una o unas pocas componentes de la muestra. En tales casos, podrían contemplarse variaciones del método propuesto, en las que la clasificación global de las muestras no solo estuviese basada en su número de componentes anómalas, sino también en la magnitud del error cometido en las mismas.

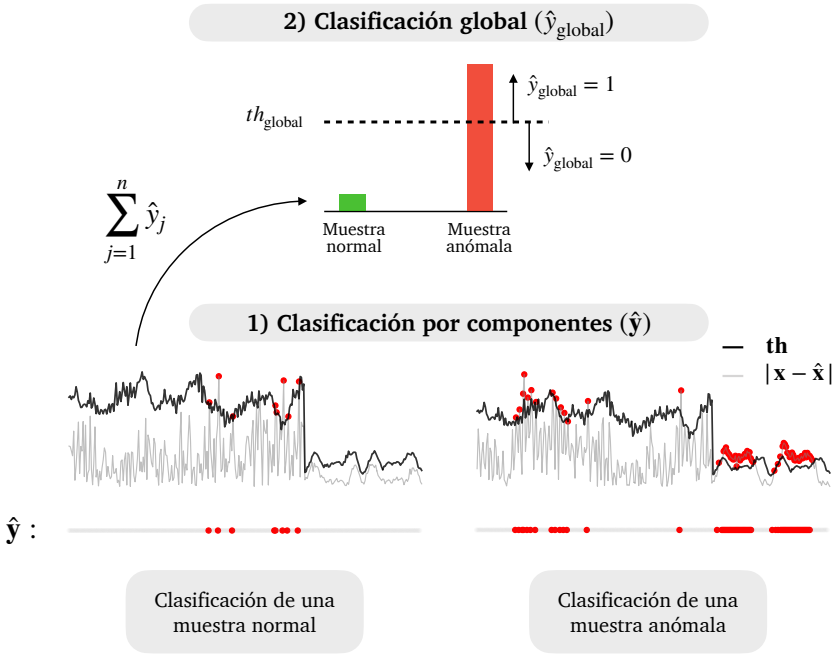


Figura 4.3: Clasificación de los residuos de dos muestras (normal y anómala) del conjunto de datos de la máquina rotativa.

la contribución de la clasificación a la mejora de la compresión del proceso bajo estudio.

4.3. Resultados

El método de detección de anomalías propuesto —constituido por un autoencoder variacional y un algoritmo para la clasificación de sus residuos— es capaz de determinar la naturaleza normal/anómala de las muestras entrantes y también la de sus componentes. A continuación, se presentan los resultados de la clasificación de las muestras, en tres contextos de ingeniería diferentes y en comparación con otros enfoques del estado del arte. Adicionalmente, se incluye un análisis visual de la clasificación por componentes de las muestras, que ha revelado valiosa información acerca del estado de salud de los procesos.

4.3.1. Resultados de la clasificación de las muestras

En la Tabla 4.10 se muestran los resultados de clasificación obtenidos en el conjunto de test de cada sistema de ingeniería, comparando el rendimiento de nuestra propuesta (combinación de *variational autoencoder* y clasificador *two-step*) con el de otros enfoques del estado del arte, tanto en lo relativo a técnicas de reconstrucción (*deep autoencoder*) como a clasificadores (clasificador de umbral, *one-class* SVM).

De acuerdo a esta tabla, los residuos del VAE conducen a mejores resultados de clasificación que los residuos del *deep autoencoder*, lo cual se aprecia en los re-

sultados de los tres conjuntos de datos. Con respecto al algoritmo de clasificación, el clasificador *two-step* muestra mejor un rendimiento que los clasificadores convencionales en todos los contextos, a excepción del sistema hidráulico, donde los enfoques convencionales se muestran levemente superiores cuando son alimentados con los residuos del VAE.

Tabla 4.10: Resultados de la clasificación global de las muestras en términos de *f1-score* (%) para los tres conjuntos de test, utilizando la técnica de validación cruzada con cinco ejecuciones (se muestran la media y desviación típica de todas las ejecuciones).

Modelo de normalidad	Algoritmo de clasificación		Dataset: Máquina rotativa	Dataset: Movimiento humano	Dataset: Sistema hidráulico
Variational autoencoder	Clasificador two-step		**97.75 (σ=0.23)**	**95.85 (σ=0.34)**	97.92 (σ=0.44)
	Otros enfoques	Clasificador de umbral	93.83 (σ=2.02)	95.59 (σ=0.37)	98.42 (σ=0.08)
		One-class SVM	94.33 (σ=2.33)	94.38 (σ=0.25)	**99.68 (σ=0.02)**
Deep autoencoder	Clasificador two-step		93.26 (σ=2.70)	93.74 (σ=1.10)	96.45 (σ=1.56)
	Otros enfoques	Clasificador de umbral	92.28 (σ=2.89)	92.91 (σ=1.00)	95.57 (σ=2.78)
		One-class SVM	92.46 (σ=2.65)	90.69 (σ=0.49)	91.61 (σ=5.44)

Adicionalmente, hemos visualizado la influencia del umbral de anomalía en los resultados del método propuesto. Como se mencionaba previamente, se ha elegido como umbral de anomalía el percentil 95 del error de reconstrucción obtenido sobre el conjunto de entrenamiento. Por tanto, este enfoque depende tan solo del comportamiento normal del proceso y no precisa de ninguna información acerca de anomalías previas en el sistema para llevar a cabo la clasificación. Como se observa en la Figura 4.4, este umbral no es el óptimo en ninguno de los tres conjuntos de datos y, aún así, el clasificador *two-step* ha demostrado un mejor rendimiento que los enfoques tradicionales. En este contexto, cabe destacar que los resultados del clasificador propuesto podrían ser mejorados, en caso de sustituir el percentil 95 por un percentil óptimo, acorde a cada *dataset*. Sin embargo, para determinar este umbral óptimo, se requiere de muestras normales y anómalas del sistema, con lo que el método propuesto ya no dependería tan solo del comportamiento normal del sistema, sino que requeriría también cierta información acerca de sus anomalías.

En definitiva, los resultados obtenidos demuestran el potencial de los residuos del VAE en la monitorización de la salud de los procesos, así como la contribución del clasificador *two-step* a un mejor rendimiento en la clasificación de los residuos que los enfoques convencionales.

4.3.2. Contribuciones de la clasificación *component-wise* a la mejora de la comprensión de los procesos

El método propuesto determina no solo la naturaleza normal/anómala de las muestras, sino también la de sus componentes, proporcionando así valiosa información acerca del estado de salud del proceso bajo estudio. En consecuencia,

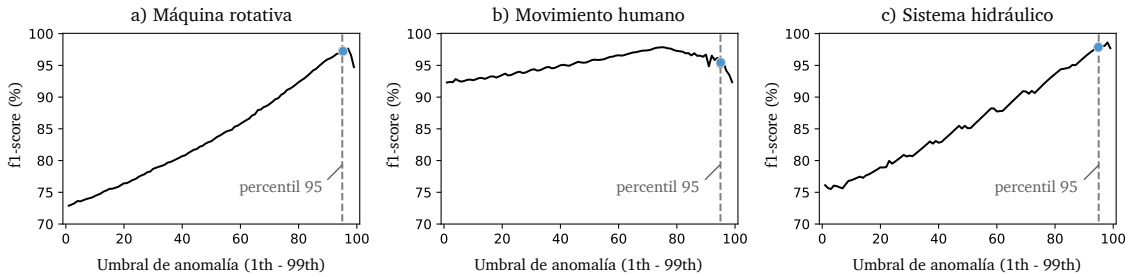


Figura 4.4: Resultados de clasificación del método propuesto (VAE y clasificador *two-step*) en términos de *f1-score* (%) para los tres conjuntos de test ante diferentes umbrales de anomalía (se ha utilizado la técnica de validación cruzada con cinco ejecuciones; se muestran la media y desviación típica de todas las ejecuciones).

exploraremos a lo largo de esta sección la contribución de dicha clasificación a la mejora de la comprensión de los sistemas, analizando para ello la clasificación *component-wise* de las muestras en cada contexto de ingeniería.

4.3.2.1. Movimiento humano: clases subyacentes en los datos

En el caso del *dataset* que contiene datos de monitorización del movimiento humano, hemos explorado los resultados de la clasificación de las muestras a través de una técnica de visualización de datos. En detalle, se ha empleado una técnica de reducción de la dimensión (*t-stochastic neighbour embedding*, t-SNE [174]) para generar una representación 2D del conjunto de datos (Figura 4.5.a) y de su correspondiente clasificación por componentes (Figura 4.5.b), donde se aprecia fácilmente la aparición de clases subyacentes en los datos. Cabe destacar que, a fin de simplificar la visualización de los resultados y favorecer la interpretabilidad, las 343195 muestras del *dataset* y sus correspondientes clasificaciones fueron promediadas por sujeto y actividad, para después ser proyectadas, dando lugar a 120 puntos que representan los resultados promedio para las 12 actividades y los 10 sujetos de estudio.

Los residuos del VAE —al eliminar la variabilidad asociada al comportamiento normal del proceso y, con ello, realzar los modos de anomalía presentes en las muestras— portan información precisa sobre el estado del sistema, que se vuelve interpretable gracias a su clasificación *component-wise*. En este caso, como se observa en la Figura 4.5.b, la clasificación por componentes de las muestras ha sacado a la luz las actividades realizadas por los voluntarios, aparentemente agrupadas de acuerdo a su nivel de intensidad. Mientras, es difícil extraer cualquier tipo de información de la representación original de los datos (Figura 4.5.a).

Por tanto, el método propuesto no solo nos informa del comportamiento normal/anómalo de los sistemas, sino que además proporciona intuición acerca de las clases —en este contexto, actividades físicas— subyacentes tras dichos comportamientos.

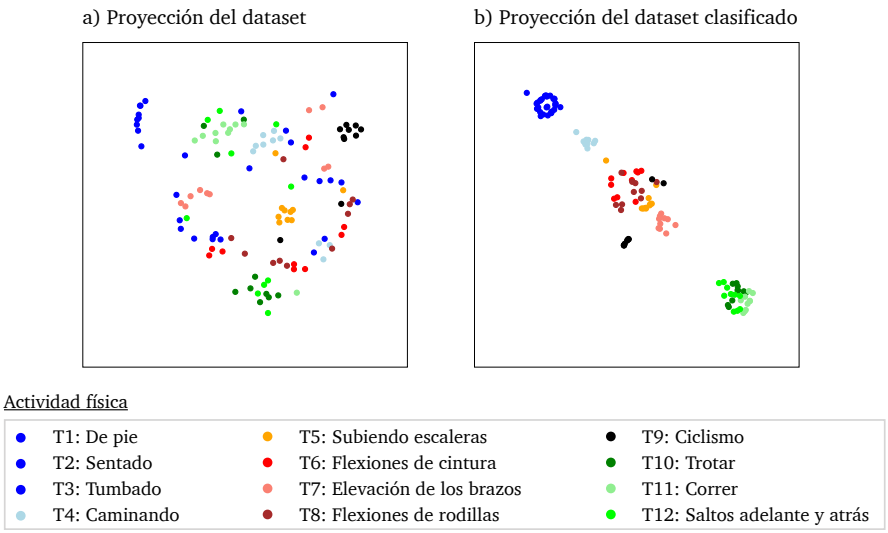


Figura 4.5: Reducción de la dimensión del conjunto de datos de movimiento humano (a) y de su clasificación por componentes (b), ambas reducciones obtenidas empleando un t-SNE con perplejidad 30.

4.3.2.2. Máquina rotativa: componentes anómalas en las muestras

En ocasiones, no solo es relevante la naturaleza de las muestras, sino también la de sus elementos. En tales casos, la clasificación por componentes resulta de gran valor, como se ilustra en la Figura 4.6. En esta figura podemos observar un registro de 100 ms de tiempo, representativo del comportamiento normal de la máquina (Figura 4.6.a) y que ha sido contaminado (Figura 4.6.b) con datos corruptos en las variables i_r y a_y: hemos añadido ruido blanco a i_r y simulado datos faltantes, fallo del sensor, etc. en a_y. La clasificación *component-wise* de los datos contaminados se muestra en la Figura 4.6.c.

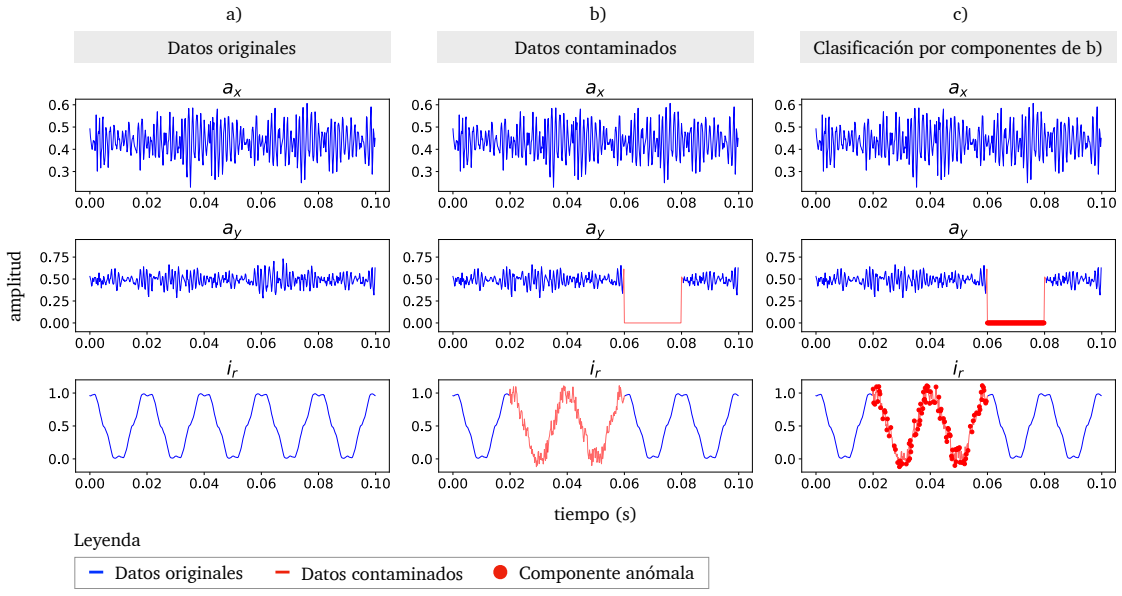


Figura 4.6: Clasificación por componentes de muestras contaminadas en la máquina rotativa.

Los resultados de la clasificación (Figura 4.6.c) revelan la presencia de componentes anómalas en las muestras, que se corresponden con los datos previamente contaminados. Por tanto, esta figura ilustra el valor de la información disponible en la clasificación por componentes, que en este caso nos ha permitido detectar con éxito en qué variables e instantes de tiempo la máquina ha sido desviada de su comportamiento normal. En vista de estos resultados, podemos afirmar que el método propuesto proporciona, gracias a la clasificación *component-wise*, una imagen explicativa del estado de salud de los procesos.

4.3.2.3. Sistema hidráulico: causas del comportamiento anómalo

La clasificación *component-wise* de las muestras puede resultar también de ayuda en la identificación de las causas de anomalía del sistema. Para explorar esta posibilidad, hemos analizado los resultados de la clasificación en todo el conjunto de muestras anómalas del sistema hidráulico.

En la Figura 4.7 se presenta el promedio de la clasificación por componentes de las muestras anómalas, donde observamos la contribución de cada componente al comportamiento anómalo del sistema, siendo las más relevantes: FS_2, $TS_1 - TS_4$, VS_1, CE y CP (volumen de flujo, temperaturas, vibración, eficiencia de la refrigeración y potencia de la refrigeración, respectivamente).

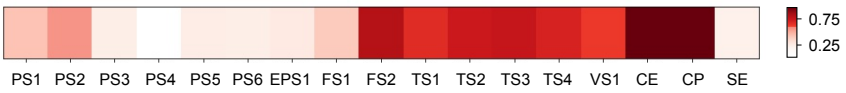


Figura 4.7: Promedio de la clasificación por componentes de todas las muestras anómalas del sistema hidráulico —las componentes están clasificadas como normales (valor 0) o anómalas (valor 1), con lo que su promedio para todas las muestras se encuentra acotado en el rango $[0, 1]$.

Estas componentes están relacionadas con la temperatura en el sistema hidráulico, o bien presentes en el circuito de refrigeración (Figura 4.8), lo que hace intuir un posible fallo del sistema de refrigeración. Como se mencionaba en la Sección 4.2.1.2, el comportamiento anómalo del sistema se debe al fallo total del circuito de refrigeración, con lo que la información proporcionada por la clasificación es consistente con la naturaleza del proceso. Por tanto, el método propuesto ha demostrado ser útil de nuevo en la mejora de la comprensión de los procesos, revelando en este caso las componentes involucradas en el comportamiento anómalo del sistema hidráulico y aportando cierta intuición acerca de las posibles causas de dicha anomalía.

4.4. Conclusiones

A lo largo de este capítulo se ha explorado el potencial de las arquitecturas profundas para la detección de anomalías en sistemas de ingeniería. Para ello, hemos propuesto un enfoque de análisis de residuos que consiste en: (1) un autoencoder

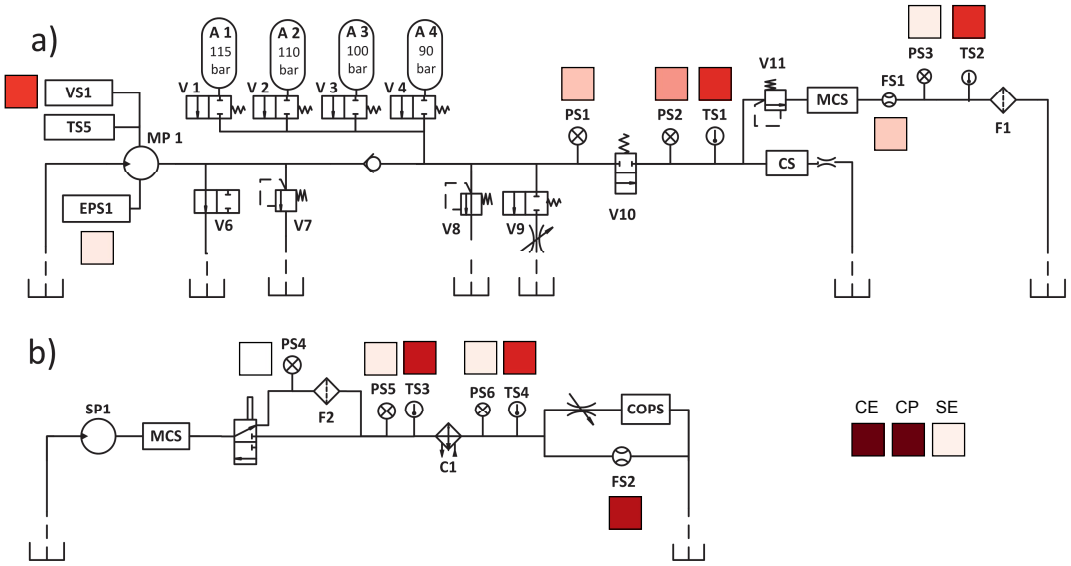


Figura 4.8: Esquema del sistema hidráulico [170] —constituido por un circuito primario de trabajo (a) y otro secundario de refrigeración-filtración (b)— incluyendo la contribución de cada componente al comportamiento anómalo del sistema (fallo total del circuito de refrigeración).

variacional (VAE), previamente entrenado para reconstruir muestras del comportamiento normal del proceso, de manera que los residuos de las nuevas muestras entrantes sirvan como una medida de su desviación respecto al comportamiento normal esperado; (2) un algoritmo de clasificación en dos pasos o *two-step*, que clasifica las muestras entrantes a partir de sus residuos, determinando tanto su naturaleza normal/anómala, como la de sus componentes.

Este enfoque ha sido evaluado en tres contextos de ingeniería diferentes (una máquina rotativa, un sistema hidráulico y un sistema de monitorización del movimiento humano) y su rendimiento ha sido comparado con el de otros enfoques de la literatura, tanto en términos de técnicas de reconstrucción (autoencoder variacional vs. *deep autoencoder*) como de clasificadores (clasificador *two-step* vs. clasificadores convencionales). Los resultados de la investigación han demostrado la capacidad de nuestra propuesta para detectar anomalías con éxito en los tres contextos y con un rendimiento superior al de los enfoques convencionales. Esto sugiere que la habilidad del VAE para aprender la función de densidad de probabilidad (PDF) de los datos normales, le confiere un rendimiento superior al obtenido por el *deep autoencoder*. Los *deep autoencoders* tienen la capacidad de modelar la geometría de los datos en el espacio de entrada, pero no su densidad, lo que les hace precisos en la reconstrucción de las muestras normales, pero también de algunas muestras anómalas; los VAEs, en cambio, restringen la reconstrucción de los datos al soporte de la PDF aprendida, lo que les convierte en herramientas más eficientes y con un gran potencial en contextos de detección de anomalías. Adicionalmente, se ha presentado un análisis visual de la clasificación por componentes de las muestras, que ha ilustrado la contribución del clasificador *two-step* a la interpretabilidad de la decisión de anomalía y a la mejora de la comprensión de los procesos bajo estudio.

En conclusión, el enfoque profundo propuesto nos ha permitido detectar anomalías con éxito en tres contextos de ingeniería diferentes, proporcionando valiosa información adicional acerca de las muestras de trabajo y sin necesidad de ninguna información de contexto acerca de los procesos o de anomalías previas en los mismos. Además, este enfoque ha demostrado ser capaz de transformar los residuos de los autoencoders en información de valor para el usuario, explotando así la expresividad de dichos residuos, cuya explicabilidad acerca del estado del proceso se ve especialmente favorecida por la reducción de la dimensión aplicada en el cuello de botella del autoencoder. En definitiva, estos resultados son una evidencia del potencial de las arquitecturas profundas como valiosas herramientas para la detección y diagnóstico de anomalías en sistemas de ingeniería.

CAPÍTULO 5

Generación de indicadores de salud

En este capítulo se aborda el uso de arquitecturas profundas para la generación de indicadores de salud de los procesos. En detalle, se presenta un indicador de salud construido en el espacio latente de un *deep autoencoder*, que será empleado como indicador del nivel de degradación de diferentes tipos de máquinas: se analiza el caso de una máquina fresadora y el de un motor de tipo turboventilador o *turbofan*. El capítulo comienza con una revisión del estado del arte, para detallar a continuación los experimentos realizados y los resultados obtenidos, terminando con un apartado de conclusiones.

El contenido de este capítulo ha sido publicado en la revista *Reliability Engineering & System Safety*, bajo el título *«Health indicator for machine condition monitoring built in the latent space of a deep autoencoder»* [175].

5.1. Antecedentes

La monitorización de la condición de las máquinas constituye un campo de investigación en continuo crecimiento y de gran interés en el ámbito de los sistemas de ingeniería [176, 177, 178]. Las máquinas están presentes como equipamiento esencial en una amplia variedad de procesos e instalaciones, con lo que resulta de vital importancia optimizar su funcionamiento, así como garantizar la seguridad durante la operación de las mismas. La monitorización de su condición adquiere, por tanto, gran valor, conllevando mejoras en la productividad y eficiencia de los procesos, y con beneficios como reducciones de costes o un mayor tiempo de vida útil para las máquinas [102].

En el estado del arte, la monitorización de la condición ha sido abordada tradicionalmente por medio de enfoques basados en la experiencia y conocimiento previo de los sistemas [179], así como a través de modelos físicos de los mismos [180]. Sin embargo, con la aparición de los sensores inteligentes y los avances en redes de comunicaciones y sistemas de almacenamiento, existe en la actualidad una cantidad creciente de datos de operación de las máquinas, lo que ha atraído la atención de los investigadores hacia métodos de monitorización de la condición

basados en datos [135].

En el estudio de estos métodos, se ha dedicado especial interés a la construcción de indicadores de salud *(Health Indicators, HIs)* de los procesos [181, 182, 183]. Los HIs reflejan el grado de degradación del sistema bajo estudio, proporcionando así valiosa información acerca del mismo, a menudo empleada como punto de partida en una amplia variedad de aplicaciones, como pueden ser el diagnóstico de fallos, la detección de anomalías o la estimación del tiempo de vida útil *(Remaining Useful Life —RUL— estimation)* de las máquinas. Por ejemplo, en el ámbito de la detección de anomalías (objeto del Capítulo 4 de esta tesis), el estado de un sistema es habitualmente declarado anómalo cuando su HI —o conjunto de HIs— cumple ciertas condiciones, como exceder un determinado umbral de anomalía [152, 184]. Otro ámbito de relevancia en el que se hace uso de estos HIs es el de las tareas de pronóstico: a modo de ejemplo, los modelos de similitud estiman el RUL de las máquinas por comparación de perfiles de degradación construidos sobre uno o varios HIs del sistema [185]; otros enfoques utilizan modelos de degradación, elaborados en base a valores previos de los HIs, para predecir la futura condición de las máquinas [186], etc. El rendimiento de todos estos enfoques depende en gran medida de la calidad de los indicadores de salud empleados en el análisis [187] [188], con lo que la construcción de HIs precisos desempeña un papel clave en la monitorización de la salud de las máquinas.

Para la generación de HIs de los procesos, se ha recurrido tradicionalmente a métodos de diseño manuales, basados en la experiencia y conocimiento previo del sistema, que, en ocasiones, puede ser difícil o incluso imposible de obtener. Cabe destacar también que estos HIs son a menudo diseñados para un proceso específico de degradación y no resultan fácilmente extrapolables a nuevos contextos de trabajo. Por tanto, la construcción de HIs representativos del estado de salud de los procesos supone una tarea compleja, tradicionalmente abordada a través de conocimiento experto y fuertemente dependiente del contexto de trabajo. En contraste, los métodos basados en datos proponen recurrir a técnicas de aprendizaje automático capaces de extraer HIs de forma automática a partir de datos de operación del proceso, lo que también se conoce como aprendizaje de características o *feature learning* [10].

Entre las técnicas de aprendizaje automático, destacan los modelos de aprendizaje profundo o *deep learning*, que en los últimos años se han convertido en una de las ramas más destacadas de la inteligencia artificial [189], con un gran éxito en aplicaciones de reconocimiento de voz y procesamiento de imagen [190, 191], y demostrando también resultados prometedores en la monitorización de la salud de los sistemas [22]. Gracias a su arquitectura de capas, los modelos profundos establecen una jerarquía composicional de características —en la que las características extraídas por cada capa se expresan en términos de las extraídas por la capa anterior— que les confiere la habilidad de encontrar representaciones con significado de los datos, también conocida como *representation learning* [8]. Dado su potencial, los modelos profundos se han establecido como poderosas herramientas de aprendizaje de características y, en consecuencia, han comenzado a ser empleadas en la construcción de indicadores de salud de los procesos [192, 193, 194].

En la literatura se han utilizado diferentes tipos de técnicas *deep learning* para

la construcción de HIs, entre las que destacan las redes convolucionales (CNNs) [195] [196], las redes recurrentes (RNNs) [187] [197], las redes GAN [198] y los *deep autoencoders* (AEs) [199] [200], en torno a los cuales hemos desarrollado nuestra investigación. Los *deep autoencoders* representan un caso particular de arquitectura profunda, entrenada para reproducir a su salida la misma información que recibe de entrada. En detalle, los autoencoders aprenden una proyección de baja dimensión de los datos de entrada, a partir de la cual reconstruyen los datos de salida. Dada su naturaleza profunda, estos modelos tienen la habilidad de reducir la dimensionalidad de los datos de entrada de una manera jerárquica, que les permite conseguir reconstrucciones de alta calidad de los datos [88, 89].

En el ámbito de la monitorización de la salud, los *deep autoencoders* son habitualmente entrenados con datos representativos del comportamiento normal o saludable del proceso, de manera que: el modelo resultante es capaz de reconstruir satisfactoriamente nuevas muestras entrantes, cuando estas son normales; en cambio, su habilidad para reconstruir muestras anómalas es limitada, dado que no se ajustan al comportamiento visto durante el entrenamiento, lo que se traduce en errores de reconstrucción elevados. En consecuencia, el error de reconstrucción se ha convertido en un popular indicador de salud en la literatura [31, 90, 155, 201], donde errores elevados están asociados a desviaciones del sistema respecto a su comportamiento normal esperado.

Por tanto, cuando los autoencoders son entrenados con muestras normales del proceso, el grado de novedad —o desviación respecto al comportamiento normal— de cualquier nueva muestra entrante es habitualmente medido en términos de su error de reconstrucción (Ecuación 5.1), el cual compara la muestra entrante $\mathbf{x}$ con su reconstrucción $\hat{\mathbf{x}}$. En este contexto, se considera que cuanto mayor sea el error residual de una muestra, mayor será su grado de desviación respecto a la normalidad o comportamiento esperado del proceso [147].

$$\varepsilon_{REC}(\mathbf{x}) = \|\mathbf{x} - \hat{\mathbf{x}}\|_2 \tag{5.1}$$

Este enfoque ha resultado exitoso en la literatura [31, 90, 152, 153], como así se demuestra en el Capítulo 4 de esta tesis, donde se ha explorado su potencial para la detección de anomalías en sistemas de ingeniería. No obstante, estudios recientes apuntan que los HIs construidos sobre los residuos del *deep autoencoder* tienen un potencial limitado: estos enfoques miden la calidad de la reconstrucción tan solo en el espacio de entrada del autoencoder, con lo que no explotan una de las más notables fortalezas de los modelos profundos, que es su habilidad para aprender representaciones jerárquicas de los datos [8]. Ante esta situación, algunos autores han propuesto explotar dicha naturaleza jerárquica extendiendo el cálculo del error de reconstrucción a los espacios ocultos del autoencoder, en lo que se conoce como enfoque RaPP *(Reconstruction along Projection Pathway)* [34]. En lugar de comparar $\mathbf{x}$ y $\hat{\mathbf{x}}$ tan solo en el espacio de entrada, el enfoque RaPP propone proyectar tanto la entrada, como su reconstrucción, en los espacios ocultos del autoencoder, para obtener así parejas de valores de activación, que serán a continuación agregadas para cuantificar el grado de novedad de la muestra entrante.

Para expresar en detalle el cómputo del error en el enfoque RaPP, consideraremos que $A = f \circ g$ es el autoencoder entrenado (donde f es el decoder y g es el encoder) y l es el número de capas en g. Entonces, el cómputo parcial de g quedaría definido como: $g_{:i} = g_i \circ \cdots \circ g_1$, para $1 \leq i \leq l$. En concordancia, cuando alimentamos A con $\mathbf{x}$ y $\hat{\mathbf{x}}$, obtenemos parejas $(h_i, \hat{h}_i)$ de sus representaciones ocultas, donde $h_i(\mathbf{x}) = g_{:i}(\mathbf{x})$ y $\hat{h}_i(\mathbf{x}) = g_{:i}(\hat{\mathbf{x}})$. Este procedimiento se ilustra en la Figura 5.1.

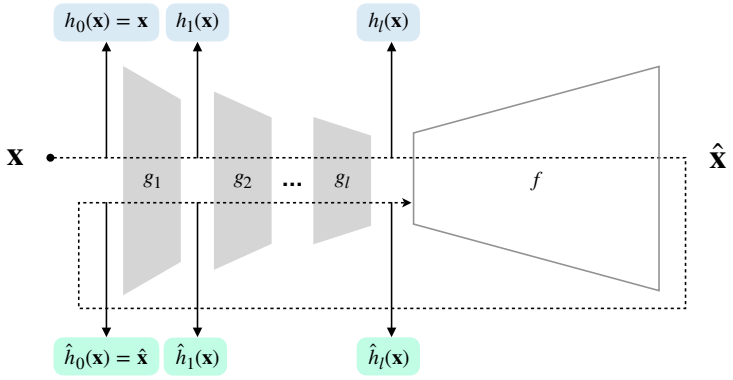


Figura 5.1: Enfoque RaPP (figura adaptada de [34]).

En cuanto a la agregación de las parejas de activaciones, los autores presentan dos métricas, que dan lugar a dos indicadores de salud diferentes derivados del enfoque RaPP: ε_{SAP} y ε_{NAP}. La agregación simple o SAP *(Simple Aggregation along Pathway)* para una muestra $\mathbf{x}$ se define como la norma L2 de las distancias entre las parejas de activaciones:

$$\varepsilon_{SAP}(\mathbf{x}) = \|h(\mathbf{x}) - \hat{h}(\mathbf{x})\|_2 \tag{5.2}$$

donde los vectores $h(\mathbf{x})$ y $\hat{h}(\mathbf{x})$ son el resultado de concatenar los vectores de activación de todas las capas del autoencoder:

$$h(\mathbf{x}) = [h_0(\mathbf{x}), \ldots, h_l(\mathbf{x})] \tag{5.3}$$

$$\hat{h}(\mathbf{x}) = [\hat{h_0}(\mathbf{x}), \ldots, \hat{h_l}(\mathbf{x})] \tag{5.4}$$

En cuanto a la agregación normalizada o NAP *(Normalized Aggregation along Pathway)*, se trata de una extensión de la métrica SAP que propone normalizar las distancias antes de su agregación, para mitigar así las dependencias entre capas ocultas. Para expresar esta agregación, consideraremos: las distancias entre activaciones $d(\mathbf{x}) = h(\mathbf{x}) - \hat{h}(\mathbf{x})$; la matriz $\mathbf{D}$, que se trata de una matriz cuya fila i-ésima corresponde a $d(\mathbf{x}_i)$ para $\mathbf{x}_i \in \mathbf{X}$, siendo $\mathbf{X}$ el conjunto de entrenamiento; y la matriz $\bar{\mathbf{D}}$, que representa la matriz centrada por columnas de $\mathbf{D}$. En este contexto, el error ε_{NAP} para una muestra $\mathbf{x}$ se define como:

$$\varepsilon_{NAP}(\mathbf{x}) = \|(d(\mathbf{x}) - \mu_{\mathbf{X}})^\top \mathbf{V}\mathbf{\Sigma}^{-1}\|_2 \tag{5.5}$$

donde $d(\mathbf{x})$ se expresa como un vector columna, $\mu_{\mathbf{X}}$ es la media por columnas

de $\mathbf{D}$, $\mathbf{\Sigma}$ es una matriz cuadrada diagonal que contiene los valores singulares de $\bar{\mathbf{D}}$ y $\mathbf{V}$ es una matriz que contiene los vectores singulares derechos de $\bar{\mathbf{D}}$. Cabe destacar también que esta métrica ε_{NAP} es equivalente al concepto de distancia de Mahalanobis.

Estos indicadores RaPP han obtenido mejores resultados en la literatura que los indicadores de salud tradicionales construidos en el espacio de entrada de los datos [34, 202], con lo que los espacios ocultos de los *deep autoencoders* han demostrado proporcionar valiosa información acerca del estado de salud de los sistemas, estableciéndose como poderosas herramientas para la construcción de HIs. En línea con estos hallazgos, hemos explorado a lo largo de esta investigación el potencial del error de reconstrucción como indicador de salud de las máquinas, poniendo especial atención en la capa oculta que proporciona la representación más compacta de los datos: el espacio latente del autoencoder.

En detalle, estudiaremos a lo largo del presente capítulo el uso del *error de reconstrucción latente* como indicador de la salud de las máquinas y evaluaremos la calidad de este indicador para ser empleado en tareas de pronóstico. De esta manera, proponemos una variante del enfoque RaPP en la que, en lugar de considerar los residuos en cada capa oculta, restringiremos nuestro análisis al espacio latente del modelo. Los espacios latentes proporcionan representaciones de baja dimensión, compactas y con significado, que capturan la estructura subyacente en los datos y que han demostrado ser capaces de desacoplar con éxito los principales modos de variación presentes en los mismos [92]. En consecuencia, los residuos calculados en el espacio latente del autoencoder podrían revelar el estado de salud de los procesos con mayor precisión que aquellos calculados tanto en el espacio de entrada de los datos (error de reconstrucción convencional), como a lo largo de todos los espacios ocultos del modelo (enfoque RaPP). El error de reconstrucción latente proporciona, por tanto, una representación compacta del error de reconstrucción, que podría tratarse de un HI más preciso y robusto que aquellos propuestos en los enfoques previos. En este contexto proponemos un método, descrito a continuación, que consta de: (1) un *deep autoencoder* entrenado con muestras representativas del comportamiento saludable del sistema; y (2) un indicador de salud, que será el error de reconstrucción calculado en el espacio latente del autoencoder. Esta propuesta ha sido evaluada sobre tres conjuntos de datos diferentes y su rendimiento ha sido comparado con el de otros enfoques del estado del arte, en términos de métricas populares en tareas de pronóstico (monotonicidad o *monotonicity*, tendencialidad o *trendability* y pronosticabilidad o *prognosability*).

Cabe destacar que en esta comparativa se ha contemplado también el uso de diferentes tipos de autoencoders, *deep autoencoder* y *variational autoencoder*, para la construcción del indicador de salud. Los *variational autoencoders* han mostrado en la literatura un rendimiento superior al de los *deep autoencoders* —sus residuos han conseguido mejores resultados en aplicaciones de detección de anomalías [29]— y sus espacios latentes han demostrado proporcionar valiosas representaciones de los datos, de gran utilidad en aplicaciones de detección de fallos [203] [204].

Los resultados de la investigación han demostrado que el indicador de salud propuesto alcanza un mejor rendimiento que el resto de enfoques —error tradicio-

nal y enfoque RaPP— para los dos tipos de autoencoders considerados, aportando así evidencias del potencial de los espacios latentes como herramientas para la monitorización de la condición de las máquinas. Adicionalmente, se incluye un análisis visual de la geometría del espacio latente, que facilita la interpretación de los resultados y arroja luz acerca de las diferencias de rendimiento entre los diferentes HIs considerados en la investigación.

5.2. Método propuesto

En esta sección se presenta el indicador de salud propuesto, el cual —de acuerdo a la filosofía RaPP— ha sido construido a partir de las representaciones intermedias de los datos disponibles en los espacios ocultos de un *deep autoencoder*. Como novedad con respecto al enfoque RaPP original, nuestro indicador ha sido elaborado teniendo tan solo en cuenta la información proporcionada por el espacio latente del autoencoder. La motivación de este enfoque y su formulación matemática se presentan a continuación, junto con la descripción de los conjuntos de datos utilizados en los experimentos.

5.2.1. Conjuntos de datos

El método propuesto ha sido evaluado sobre tres conjuntos de datos diferentes, expuestos en la Tabla 5.1. Los *datasets* FD001 y FD003 forman parte del repositorio C-MAPSS [205], desarrollado por la NASA, que proporciona datos simulados de degradación de motores de tipo *turbofan* ante diferentes condiciones de operación y patrones de fallo. El *dataset* Mill [206], proporcionado por el laboratorio BEST de la Universidad de California en Berkeley, contiene datos de desgaste de una máquina fresadora ante diferentes condiciones de funcionamiento.

Tabla 5.1: Conjuntos de datos empleados en los experimentos.

Dataset	Sensores	Ejecuciones	Ciclos	Tamaño	Conjunto de entrenamiento	Conjunto de test
FD001	21	200	33727	33727×21	23692×21	10035×21
FD003	21	519	41316	41316×21	31289×21	10027×21
Mill	6	16	167	8350×600	3000×600	5350×600

Más en detalle, el *dataset* FD001 contiene registros de 21 sensores a lo largo de 200 ejecuciones (100 ejecuciones de motores operando desde su funcionamiento normal hasta el fallo total y 100 ejecuciones interrumpidas en algún momento previo al fallo), cada una de ellas de diferente duración, dando lugar a un conjunto de datos con 33727 registros o ciclos de operación. Para cada ciclo, se dispone también de cierta información adicional de la máquina (parámetros operacionales de los motores) que no han sido incluidos en nuestros experimentos. El *dataset* FD003 proporciona registros de operación de otro conjunto de motores y está construido de igual manera: consta de 519 ejecuciones (260 ejecuciones de motores operando desde su funcionamiento normal hasta el fallo total y 259 ejecuciones interrumpidas en algún momento previo al fallo), dando lugar a un conjunto de datos de

41316 muestras. En lo que respecta al preprocesamiento de los datos, ambos *datasets* han sido escalados en el rango $[0, 1]$ utilizando un escalado min-max [140].

El *dataset* Mill contiene registros de 6 sensores para 16 ejecuciones de diferente duración —en las que la máquina opera desde el funcionamiento normal hasta el fallo total— dando lugar a un conjunto de datos con 167 ciclos de operación. En este conjunto, cada ciclo de la máquina consta de 6 capturas de datos (una por sensor) de 9000 elementos, que hemos preprocesado de la siguiente manera: los primeros y últimos 2000 elementos —que se corresponden con el encendido y apagado de la fresadora— han sido descartados, obteniendo vectores de tamaño 5000 elementos, que han sido enventanados en 50 ventanas de 100 elementos (sin solapamiento). A continuación, las ventanas de todos los sensores han sido concatenadas para obtener un conjunto final con 8350 muestras de tamaño 600 elementos. También se dispone de información adicional de la máquina (parámetros operacionales y tipo de material de trabajo) que no ha sido incluida en nuestros experimentos. Finalmente, el conjunto de datos ha sido escalado en el rango $[0, 1]$ utilizando un escalado min-max [140].

En cuanto al entrenamiento y evaluación de la propuesta, los *datasets* descritos han sido divididos en dos subconjuntos, entrenamiento y test, como se indica en la Tabla 5.1. En detalle, se han considerado los ciclos iniciales de las máquinas como representativos del comportamiento normal del proceso y, por tanto, han sido empleados para crear los conjuntos de entrenamiento. En particular, para el conjunto Mill hemos considerado como muestras normales aquellas correspondientes a los cuatro primeros ciclos de operación de la máquina; para los conjuntos FD001 y FD003, hemos elegido como muestras normales aquellas con un RUL superior a 80 ciclos (en estos *datasets* el RUL asociado a cada muestra es un dato conocido). Estas muestras normales han sido empleadas para construir un modelo del comportamiento normal del proceso y las muestras restantes han sido utilizadas como datos de test para la evaluación del rendimiento del indicador de salud propuesto.

5.2.2. Modelo del comportamiento normal

El indicador de salud propuesto se construye a partir de los residuos de un *deep autoencoder*, previamente entrenado con muestras representativas del comportamiento normal del proceso, de manera que los residuos de las nuevas muestras entrantes sirvan como medida del grado de desviación o degradación de la máquina respecto a dicho comportamiento normal esperado. Con este propósito, fueron entrenados los autoencoders expuestos en las Tablas 5.2 y 5.3. Como se aprecia en estas tablas, se han empleado dos tipos de autoencoders —*deep autoencoder* y *variational autoencoder*— a fin de evaluar la influencia del tipo de modelo elegido en la calidad del indicador de salud obtenido.

En detalle, fueron entrenados tres *deep autoencoders* (uno por *dataset*) utilizando el algoritmo de descenso del gradiente [80] en combinación con el optimizador ADAM [97]. El número de épocas, el tamaño del *mini-batch* y la arquitectura de cada modelo se indican en la Tabla 5.2. En cuanto a las funciones de activación, se ha utilizado la función ReLU en todas las capas, excepto en la capa de salida y en el cuello de botella del modelo, donde se ha empleado la función de activación

lineal.

De igual manera, fueron entrenados tres *variational autoencoders* (uno por *dataset*) utilizando para ello el algoritmo de descenso del gradiente [80] en combinación con el optimizador RMSProp [96]. El número de épocas y el tamaño del *mini-batch* se indican en la Tabla 5.3, junto con la arquitectura de cada modelo (número de capas y número de neuronas por capa). En cuanto a las funciones de activación, se ha utilizado la función ReLU en todas las capas, excepto en la capa de salida y en el cuello de botella del modelo, donde se han empleado la función sigmoide y la función lineal, respectivamente.

Tabla 5.2: Arquitectura del *deep autoencoder* para cada *dataset*.

	Número de épocas	Tamaño *mini-batch*	Número de capas	Número de neuronas en las capas		
				Encoder	Cuello de botella	Decoder
FD001	200	800	9	(21,10,20,10)	(2)	(10,20,10,21)
FD003	200	800	9	(21,10,20,10)	(2)	(10,20,10,21)
Mill	300	1000	7	(600,200,100)	(10)	(100,200,600)

Tabla 5.3: Arquitectura del *variational autoencoder* para cada *dataset*.

	Número de épocas	Tamaño *mini-batch*	Número de capas	Número de neuronas en las capas		
				Encoder	Cuello de botella	Decoder
FD001	400	300	11	(21,10,20,10)	(2,2,2)	(10,20,10,21)
FD003	400	300	11	(21,10,20,10)	(2,2,2)	(10,20,10,21)
Mill	400	300	9	(600,200,100)	(10,10,10)	(100,200,600)

En lo que respecta a la elección de los hiperparámetros de cada modelo (número de capas, número de neuronas en las capas, número de épocas, tamaño del *mini-batch*, etc.), se han ejecutado diferentes abanicos de experimentos y se han elegido aquellos hiperparámetros con los que el modelo ha demostrado un menor error de reconstrucción. En cuanto al entrenamiento de las arquitecturas, los modelos han sido alimentados con los conjuntos de entrenamiento señalados en la Tabla 5.1.

Finalmente, cabe destacar que los conjuntos de entrenamiento han sido considerados representativos de todo el rango de condiciones de funcionamiento normal o saludable del proceso. Sin embargo, en sistemas de naturaleza cambiante, las condiciones normales del proceso podrían variar con el tiempo, afectando al rendimiento de la propuesta. En tales casos, podría requerirse un reentrenamiento del autoencoder —activado, por ejemplo, por un detector de deriva [207]— para modelar de nuevo el comportamiento normal del proceso.

5.2.3. Error de reconstrucción latente

Con la llegada de la Industria 4.0, los sistemas de ingeniería gozan de una enorme riqueza en sensores, cuyos registros proporcionan una imagen detallada del estado de funcionamiento de los procesos e instalaciones bajo estudio. Esta valiosa información da lugar a conjuntos de datos de alta dimensionalidad, donde

cada muestra de trabajo puede contener decenas, cientos o incluso miles de elementos. No obstante, a pesar de vivir en espacios de alta dimensión, estos datos gozan a menudo de un reducido número de grados de libertad, de manera que tras ellos subyace habitualmente una estructura de dimensión menor [208, 209]. En consecuencia, en muchas ocasiones se asume que estos grandes conjuntos de datos pueden ser explicados en términos de un pequeño conjunto de variables, en lo que se conoce como hipótesis manifold o *manifold hypothesis* [210]. Por tanto, una reducción conveniente de la dimensión de los datos —compacta y con significado de los mismos— como la proporcionada por los espacios latentes de los *deep autoencoders*, podría revelar con éxito las fuentes de variación presentes en los sistemas.

Los *deep autoencoders* son particularmente entrenados para proyectar los datos de entrada en un espacio latente de baja dimensión y, a continuación, reconstruir los datos de salida a partir de esta representación compacta. Gracias a su naturaleza jerárquica, han demostrado una gran habilidad para aprender representaciones con significado de los datos y sus espacios latentes se han establecido como poderosos extractores de características [92, 211]. Estos espacios capturan la estructura subyacente en los datos, proporcionando proyecciones de baja dimensión de los mismos, que son más robustas ante la presencia de ruido o artefactos en los datos, que los espacios de alta dimensión [212, 213].

Ante esta situación, proponemos trasladar el cálculo del error de reconstrucción, del espacio original de los datos a este espacio latente de características. Como resultado, obtendremos una versión compacta del error de reconstrucción tradicional, que hemos llamado *error de reconstrucción latente* y que podría tratarse de un HI más robusto y, por tanto, más conveniente para tareas de pronóstico que dicho error tradicional. Así puede apreciarse en las siguientes figuras, donde ilustramos esta comparativa para dos ejemplos tridimensionales (3D): un conjunto de *clusters* y una espiral (Figuras 5.2 y 5.3, respectivamente). En ambos casos, un *deep autoencoder* fue entrenado para aprender a reconstruir las muestras 3D (considerándolas representativas del comportamiento normal), empleando un espacio latente 2D. A continuación, se creó una malla o *grid*, para la cual fueron calculados tanto el error de reconstrucción tradicional ε_{REC} como el latente ε_{LS}. Estos errores se representan mediante una escala de color que va de blanco (error bajo, comportamiento normal) a rojo (error alto, comportamiento de fallo), reflejando así el grado de desviación del proceso con respecto a su comportamiento normal. En detalle, podemos observar en las subfiguras 5.2.c y 5.3.c que ε_{REC} es sensible a múltiples fuentes de variación en los datos, dando lugar a la aparición de artefactos en el espacio latente. En su lugar, el error ε_{LS} presenta en las subfiguras 5.2.d y 5.3.d un comportamiento más regularizado, siendo tan solo sensible ante desviaciones del funcionamiento normal: el error es bajo en las zonas de alta densidad de datos y aumenta gradualmente hacia las zonas de baja densidad, comportándose como un indicador de salud más consistente.

En último lugar, cabe destacar que esta propuesta es una simplificación intencionada del enfoque RaPP: en lugar de calcular los residuos a lo largo de todos los espacios ocultos del autoencoder, en nuestra propuesta hemos puesto el foco en el espacio latente, que se trata del espacio oculto que proporciona la representación más compacta y parsimoniosa de los datos, resumiendo en un número reducido de

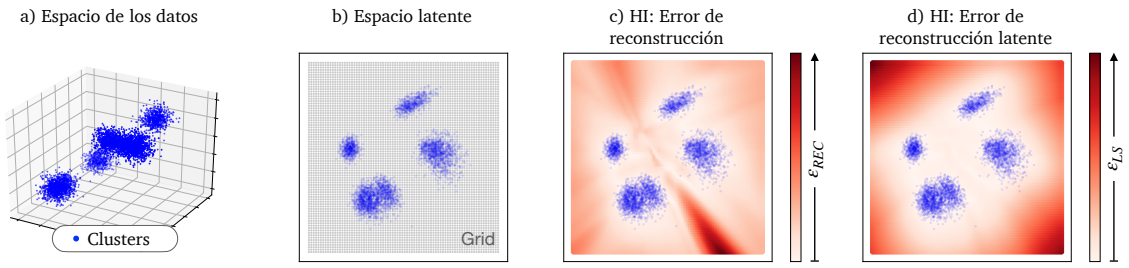


Figura 5.2: Error de reconstrucción latente para un conjunto de *clusters* 3D.

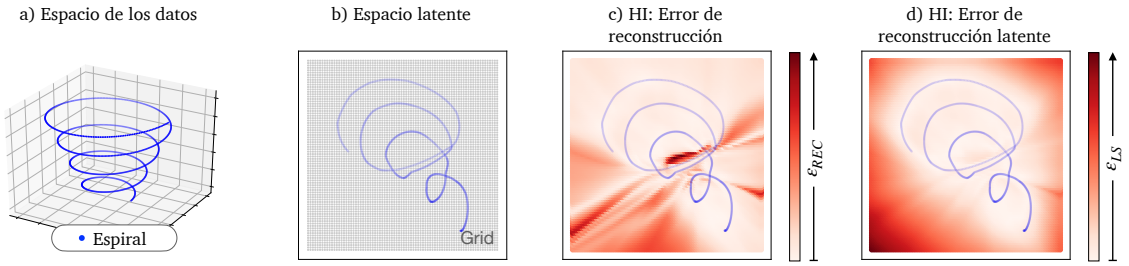


Figura 5.3: Error de reconstrucción latente para una espiral 3D.

factores los principales modos de variación del proceso. En la Sección 5.3, evaluaremos la calidad de nuestra propuesta, en comparación con el enfoque RaPP y con el error de reconstrucción tradicional, como indicadores del grado de degradación de las máquinas.

5.2.3.1. Formulación matemática

La construcción de nuestro indicador de salud comienza por entrenar un autoencoder —constituido por un encoder g y un decoder f— con muestras normales del proceso. A continuación, para evaluar el grado de degradación de cualquier muestra entrante, la muestra $\mathbf{x}$ y su reconstrucción $\hat{\mathbf{x}}$ serán proyectadas en el espacio latente del autoencoder (Figura 5.4), dando lugar a parejas de vectores de activación $[h_l(\mathbf{x}), \hat{h}_l(\mathbf{x})]$ que serán agregadas para determinar el error de reconstrucción latente. Para la agregación, hemos empleado las mismas métricas que el enfoque RaPP (SAP, NAP), pero en nuestro caso limitadas al espacio latente. Como resultado, se obtienen dos indicadores de salud diferentes: $\varepsilon_{SAP_{LS}}$ y $\varepsilon_{NAP_{LS}}$.

En detalle, el error de reconstrucción latente $\varepsilon_{SAP_{LS}}$[1] de una muestra $\mathbf{x}$ quedaría definido como la norma L2 de la diferencia entre su representación latente $h_l(\mathbf{x})$ y la representación latente de su reconstrucción $\hat{h}_l(\mathbf{x})$:

$$\varepsilon_{SAP_{LS}}(\mathbf{x}) = \|h_l(\mathbf{x}) - \hat{h}_l(\mathbf{x})\|_2 \tag{5.6}$$

De igual manera que en el método RaPP, las distancias también pueden ser normalizadas antes de su agregación. Para ello, el error normalizado $\varepsilon_{NAP_{LS}}$ será calculado de acuerdo a la Ecuación (5.5), tomando como distancias $d(\mathbf{x}) = h_l(\mathbf{x}) - \hat{h}_l(\mathbf{x})$.

[1] Para mayor claridad, hasta el momento nos habíamos referido a él como ε_{LS}.

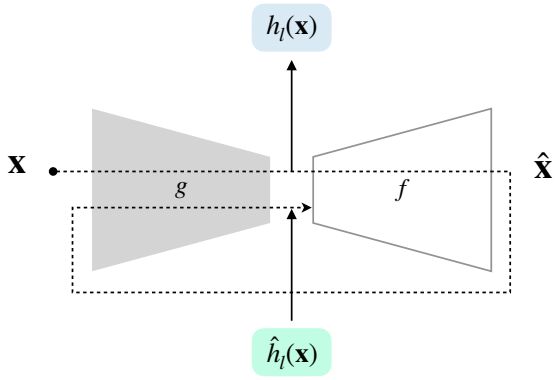


Figura 5.4: Método propuesto: limitamos el enfoque RaPP al espacio latente.

Siguiendo este procedimiento, el enfoque propuesto da lugar a dos indicadores de salud ($\varepsilon_{SAP_{LS}}$, $\varepsilon_{NAP_{LS}}$), cuyo rendimiento será analizado en la siguiente sección.

5.3. Resultados

El indicador de salud propuesto ha sido evaluado sobre tres conjuntos de datos diferentes, considerando dos tipos de autoencoders para su construcción y comparando su rendimiento con el de otros enfoques del estado del arte. A continuación se presenta una evaluación de su rendimiento, en términos de *monotonicity*, *trendability* y *prognosability*. Adicionalmente, se incluye un análisis visual del espacio latente de los autoencoders entrenados, aportando luz al potencial de los residuos latentes para la monitorización de la degradación de las máquinas.

5.3.1. Evaluación del indicador de salud

El método propuesto ha sido evaluado ante tres conjuntos de datos (FD001, FD003, Mill), considerando diferentes arquitecturas para la generación de los residuos (*deep autoencoder*, *variational autoencoder*) y comparando su rendimiento con el de otros enfoques del estado del arte. En detalle, se han considerado tres métodos diferentes para la construcción de HIs: el enfoque tradicional ε_{REC}, en el que los residuos son calculados en el espacio de entrada del autoencoder; el enfoque RaPP, en el que los residuos son calculados en los espacios ocultos del autoencoder (en concordancia con la literatura, se han considerado la agregación simple ε_{SAP} y normalizada ε_{NAP} de los residuos); y nuestro enfoque, que propone calcular los residuos en el espacio latente del autoencoder (de igual manera, presentamos dos posibles indicadores según el tipo de agregación de los residuos, $\varepsilon_{SAP_{LS}}$ y $\varepsilon_{NAP_{LS}}$). Para evaluar el rendimiento de estos indicadores, se han empleado tres métricas populares en la literatura [214]: monotonicidad o *monotonicity* (medida de la tendencia del HI a medida que la máquina evoluciona hacia el fallo), tendencialidad o *trendability* (medida de similitud del HI entre diferentes ejecuciones de la máquina) y pronosticabilidad o *prognosability* (medida de la variabilidad del HI en el momento del fallo en relación con el rango entre sus valores iniciales y finales). Las expresiones de estas métricas se muestran en las Ecuaciones (5.7)

a (5.9), donde $\mathbf{x}_j$ representa el perfil de degradación de la máquina j, M es el número de máquinas consideradas y N_j es el tamaño del vector $\mathbf{x}_j$. Los resultados obtenidos se muestran en la Tabla 5.4, donde se ha incluido también un gráfico de barras para cada columna de la tabla.

$$Monotonicity = \frac{1}{M}\sum_{j=1}^{M}\left|\sum_{k=1}^{N_j-1}\frac{sign(\mathbf{x}_j(k+1)-\mathbf{x}_j(k))}{N_j-1}\right| \tag{5.7}$$

$$Trendability = \min_{j,k}|corr(\mathbf{x}_j,\mathbf{x}_k)|\,, \qquad j,k=1,\ldots,M \tag{5.8}$$

$$Prognosability = exp\left(-\frac{std_j(\mathbf{x}_j(N_j))}{mean_j|\mathbf{x}_j(1)-\mathbf{x}_j(N_j)|}\right), \qquad j=1,\ldots,M \tag{5.9}$$

De acuerdo a la Tabla 5.4, los HIs construidos sobre el error de reconstrucción latente ($\varepsilon_{SAP_{LS}}$, $\varepsilon_{NAP_{LS}}$) consiguen mejores resultados que el resto de indicadores considerados en la comparativa (así se observa en todas las métricas, para los tres *datasets* y en ambos tipos de autoencoders; con excepción de la *prognosability* en el conjunto FD003). Por tanto, nuestro método ha demostrado superar en rendimiento a otros enfoques de la literatura, en diferentes contextos, e independientemente del tipo de autoencoder empleado para la generación de los residuos. También se aprecia en esta tabla que los dos indicadores derivados de nuestra propuesta presentan un buen rendimiento, destacando especialmente el HI basado en la agregación normalizada de los residuos ($\varepsilon_{NAP_{LS}}$).

Adicionalmente, hemos considerado cinco descriptores estadísticos habitualmente empleados en la literatura como indicadores de salud de los procesos: *skewness*, curtosis, valor RMS, factor de cresta y varianza [215]. La expresión de estas características se muestra en las Ecuaciones (5.10) a (5.14), donde $\mathbf{x}$ representa una ventana de datos de N elementos, con media μ y desviación típica σ. En nuestros experimentos, estos descriptores fueron extraídos para cada variable disponible en los *datasets*, tomando ventanas de 20 elementos de tamaño en los *datasets* FD001 y FD003, y de 100 elementos en el *dataset* Mill. En la Tabla 5.5 se muestran los mejores resultados obtenidos para cada descriptor, donde se observa que, a pesar de demostrar un buen rendimiento, estos descriptores son menos competitivos que los indicadores extraídos automáticamente por los autoencoders profundos. Por tanto, aunque las características diseñadas manualmente son a menudo eficientes y han demostrado su éxito en variadas aplicaciones, escoger el conjunto oportuno de características a extraer puede convertirse, en ocasiones, en una tarea compleja —requiriendo experiencia, conocimiento previo de la máquina o incluso una fuerte base matemática— que las técnicas de aprendizaje automático pueden abordar satisfactoriamente por sí mismas.

$$Skewness = \frac{\frac{1}{N}\sum_{i=1}^{N}(x_i-\mu)^3}{\left(\sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i-\mu)^2}\right)^3} \tag{5.10}$$

	FD001			FD003			Mill		
HI	Mono	Tren	Prog	Mono	Tren	Prog	Mono	Tren	Prog
$\varepsilon_{NAP_{LS}}$	**0.455**	**3.025×10^{-04}**	**0.991**	**0.413**	**2.980×10^{-04}**	0.904	**0.406**	5.318×10^{-02}	**0.392**
$\varepsilon_{SAP_{LS}}$	0.416	1.867×10^{-04}	**0.991**	0.401	4.529×10^{-05}	0.898	0.357	**2.320×10^{-01}**	0.326
ε_{NAP}	0.156	2.252×10^{-05}	0.979	0.179	3.310×10^{-05}	**1.000**	0.260	7.083×10^{-03}	0.043
ε_{SAP}	0.265	1.358×10^{-04}	0.909	0.187	3.562×10^{-05}	0.790	0.306	1.589×10^{-01}	0.366
ε_{REC}	0.199	1.859×10^{-17}	0.893	0.214	2.629×10^{-18}	0.803	0.302	1.060×10^{-01}	0.344

(a) Resultados de los HIs construidos sobre los residuos del *deep autoencoder*.

	FD001			FD003			Mill		
HI	Mono	Tren	Prog	Mono	Tren	Prog	Mono	Tren	Prog
$\varepsilon_{NAP_{LS}}$	**0.557**	**5.535×10^{-03}**	0.963	**0.561**	1.914×10^{-03}	0.950	0.267	6.358×10^{-03}	**0.384**
$\varepsilon_{SAP_{LS}}$	0.546	2.857×10^{-04}	**0.965**	0.534	**2.048×10^{-03}**	0.926	**0.371**	**9.227×10^{-02}**	0.349
ε_{NAP}	0.409	3.303×10^{-04}	0.964	0.224	1.204×10^{-05}	**1.000**	0.209	1.074×10^{-03}	0.021
ε_{SAP}	0.459	8.019×10^{-04}	0.964	0.494	4.833×10^{-04}	0.938	0.312	4.071×10^{-03}	0.351
ε_{REC}	0.263	7.423×10^{-05}	0.913	0.314	8.646×10^{-19}	0.913	0.333	4.451×10^{-02}	0.305

(b) Resultados de los HIs construidos sobre los residuos del *variational autoencoder*.

Tabla 5.4: El rendimiento de los HIs se expresa en términos de *monotonicity* (mono), *trendability* (tren) y *prognosability* (prog). Los mejores resultados para cada métrica y *dataset* están señalados en negrita.

	FD001			FD003			Mill		
HI	Mono	Tren	Prog	Mono	Tren	Prog	Mono	Tren	Prog
SKEWNESS	0.154	3.278×10^{-20}	0.416	0.196	4.791×10^{-20}	0.362	0.291	4.000×10^{-03}	0.563
KURTOSIS	0.117	9.828×10^{-19}	0.378	0.142	**1.325×10^{-05}**	0.339	0.292	**3.860×10^{-02}**	**0.758**
RMS	**0.777**	5.677×10^{-20}	**0.651**	**0.838**	3.807×10^{-17}	**0.693**	**0.369**	9.000×10^{-04}	0.413
CRESTFACTOR	0.232	**5.799×10^{-05}**	0.358	0.264	4.562×10^{-06}	0.205	0.270	7.600×10^{-03}	0.670
VARIANCE	0.151	3.333×10^{-05}	0.595	0.199	2.790×10^{-18}	0.108	0.338	7.000×10^{-04}	0.656

Tabla 5.5: Resultados de los HIs construidos a partir de estadísticos. El rendimiento de los HIs se expresa en términos de *monotonicity* (mono), *trendability* (tren) y *prognosability* (prog). Los mejores resultados para cada métrica y *dataset* están señalados en negrita; los resultados que superan en rendimiento a los enfoques *deep learning* (Tabla 5.4) aparecen subrayados.

$$Curtosis = \frac{\sum_i^N (x_i - \mu)^4}{N\sigma^4} \tag{5.11}$$

$$RMS = \sqrt{\frac{1}{N}\sum_i^N {x_i}^2} \tag{5.12}$$

$$Factor\ de\ cresta = \frac{max(|x_i|)}{RMS} \tag{5.13}$$

$$Varianza = \frac{1}{N-1}\sum_{i=1}^{N} |x_i - \mu|^2 \tag{5.14}$$

En último lugar, se presenta una comparativa de los perfiles de degradación arrojados por los indicadores de salud, para cuatro ejecuciones diferentes de las máquinas —mientras evolucionan desde su funcionamiento normal hasta el fallo total— en los tres conjuntos de trabajo (Figuras 5.5, 5.6 y 5.7). En estas figuras se observa que los HIs basados en el enfoque RaPP (ε_{SAP}, ε_{NAP}) demuestran un mejor rendimiento, con perfiles de degradación menos ruidosos, que el error de reconstrucción convencional (ε_{REC}). Cabe recordar que el indicador ε_{NAP} ha demostrado excelentes resultados en la literatura en aplicaciones de detección de anomalías, con un rendimiento superior al del indicador ε_{SAP}. Sin embargo, en el contexto de tareas de pronóstico, este indicador presenta una propiedad indeseada, que se manifiesta en un incremento abrupto al final de la vida de la máquina, como puede observarse en las Figuras 5.5.a.3, 5.6.a.3, 5.7.a.3 y 5.7.b.3. Este comportamiento dificulta la detección temprana del fallo, que es un aspecto crítico en las aplicaciones de mantenimiento predictivo y que podría explicar los modestos resultados conseguidos por el indicador ε_{NAP} en la Tabla 5.4. También se aprecia que los HIs derivados de nuestra propuesta ($\varepsilon_{SAP_{LS}}$, $\varepsilon_{NAP_{LS}}$) comparten perfiles de degradación similares, con una particularidad notable: los residuos permanecen bajos para las primeras muestras —cuando la máquina aún no ha comenzado el proceso de desgaste— y van incrementándose gradualmente hasta el fallo de la máquina. Por tanto, estos HIs parecen ser sensibles tan solo ante desviaciones del comportamiento normal —en este contexto, ante degradación de la máquina— dando lugar a perfiles de degradación más robustos y regularizados, y en consecuencia con un mejor rendimiento que los enfoques del estado del arte.

En vista de estos resultados, el error de reconstrucción latente ha demostrado ser un indicador fiable del estado de salud de los sistemas, con valiosas cualidades como indicador para tareas de pronóstico. Entre las aplicaciones que podrían beneficiarse de su potencial, se encuentran los métodos de estimación del RUL, que constituyen otro campo de investigación relevante en el ámbito de los sistemas de ingeniería y cuyo objetivo es estimar el tiempo de vida útil de las máquinas a partir de indicadores de salud de las mismas [185, 216, 217].

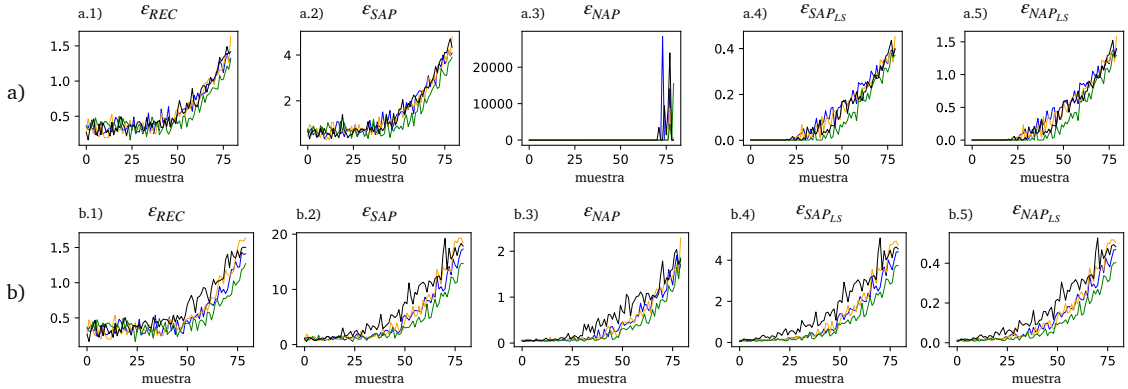


Figura 5.5: HIs construidos sobre los residuos del a) *deep autoencoder* y del b) *variational autoencoder*, para cuatro trayectorias de degradación pertenecientes al conjunto de test del *dataset* FD001.

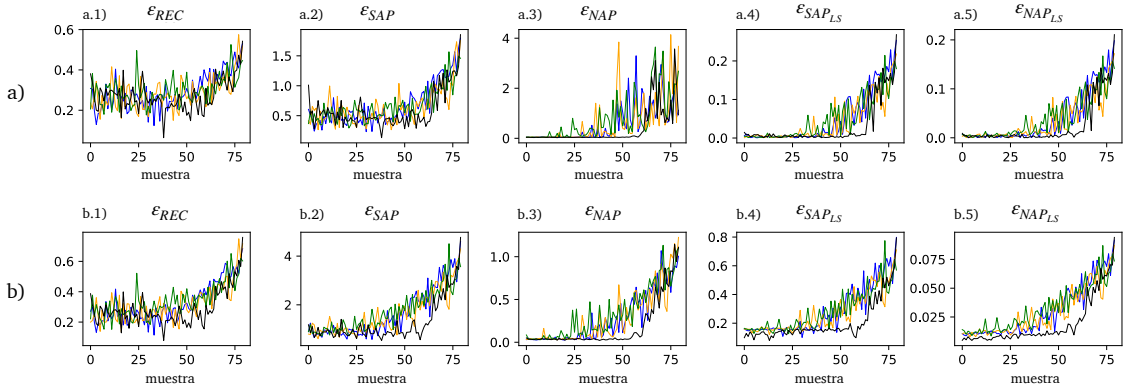


Figura 5.6: HIs construidos sobre los residuos del a) *deep autoencoder* y del b) *variational autoencoder*, para cuatro trayectorias de degradación pertenecientes al conjunto de test del *dataset* FD003.

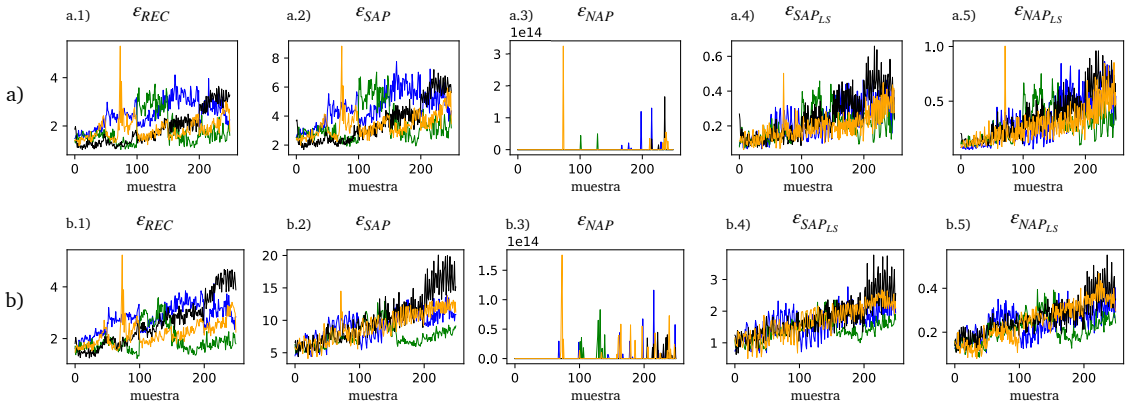


Figura 5.7: HIs construidos sobre los residuos del a) *deep autoencoder* y del b) *variational autoencoder*, para cuatro trayectorias de degradación pertenecientes al conjunto de test del *dataset* Mill.

5.3.2. Interpretación geométrica del indicador de salud

Procediendo de igual manera que en la Sección 5.2.3, es posible generar un *grid* en el espacio latente del autoencoder, para emplearlo como mapa de degradación del proceso bajo estudio. En particular, se analiza a continuación el proceso de degradación de las máquinas en el conjunto de datos FD001, a partir del espacio latente del *deep autoencoder* entrenado.

En la Figura 5.8 se observa el espacio latente del autoencoder, sobre el cual se han dispuesto cinco mapas de degradación, cada uno construido en base a un indicador de salud diferente. También se han proyectado las trayectorias de degradación de cuatro máquinas (las mismas que se presentan en la Figura 5.5.a), incluyendo: un vector que indica la dirección de degradación; y la función de densidad de probabilidad de los datos de entrenamiento (señalada en color azul y estimada mediante KDE —*Kernel Density Estimation*— con *kernel* gaussiano).

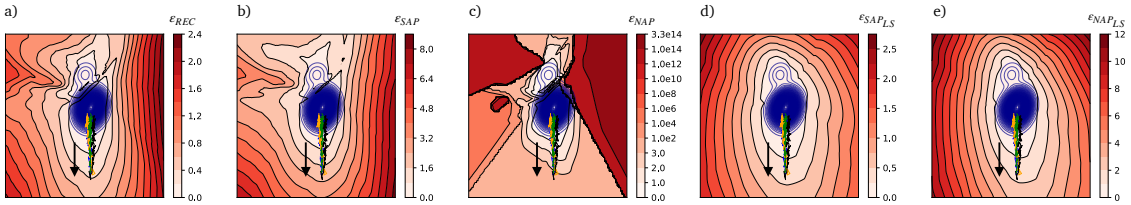


Figura 5.8: Representación de diferentes indicadores de salud sobre el espacio latente del *deep autoencoder* para el conjunto de datos FD001: a) ε_{REC}, b) ε_{SAP}, c) ε_{NAP}, d) $\varepsilon_{SAP_{LS}}$, e) $\varepsilon_{NAP_{LS}}$. Cabe señalar que la proyección de las trayectorias de degradación de las máquinas es la misma en todas las subfiguras —se trata del mismo espacio latente en los cinco casos— pero en cada una de ellas se ha representado un mapa de degradación diferente.

En esta comparativa podemos apreciar que, en los cinco casos, el valor del HI aumenta en la dirección de degradación de la máquina, con lo que todos los indicadores son consistentes con la naturaleza del proceso. Sin embargo, en una vista general, se observa que cada HI proporciona una interpretación geométrica diferente del espacio latente. En detalle, los indicadores derivados de nuestra propuesta (subfiguras d y e) dan lugar a mapas con una geometría más suave y regularizada que los enfoques del estado del arte (subfiguras a, b y c) y, también, mejor alineada con la función de densidad de los datos de entrenamiento (los cuales eran representativos del funcionamiento normal de la máquina). Por tanto, los residuos latentes revelan una geometría más consistente con el proceso de degradación, que explica los buenos resultados obtenidos por los indicadores $\varepsilon_{SAP_{LS}}$ y $\varepsilon_{NAP_{LS}}$ en la Tabla 5.4.

Gracias a esta visualización, es posible apreciar al completo la información proporcionada por los indicadores de salud y cómo esta podría resultar de ayuda en la monitorización de la condición de los sistemas de ingeniería. Como se mostraba en la sección previa (Figura 5.5, Figura 5.6, Figura 5.7), la magnitud del indicador de salud revela el grado de degradación de la máquina bajo estudio, proporcionando así valiosa información acerca de la condición del sistema, que podría ser empleada, por ejemplo, como soporte para la planificación de operaciones de mantenimiento [218] o, también, como información de entrada en tareas de pro-

nóstico como la estimación del RUL [219]. Adicionalmente, el análisis semántico del espacio latente, que en este caso ha revelado la dirección de degradación del proceso (Figura 5.8), podría sacar a la luz diferentes tipos de degradación en las máquinas, permitiendo identificar qué clase de anomalía está experimentando el sistema y planificar así operaciones de mantenimiento en concordancia.

5.4. Conclusiones

A lo largo de este capítulo se ha explorado el potencial de las arquitecturas profundas para la construcción de indicadores de salud de los procesos. En detalle, se ha propuesto emplear el error de reconstrucción latente de los autoencoders profundos como indicador de salud de las máquinas. Este enfoque ha sido evaluado ante tres conjuntos de datos diferentes, considerando dos tipos de autoencoders (*deep autoencoder* y *variational autoencoder*) y comparando su rendimiento con el de otros enfoques del estado del arte: el error de reconstrucción convencional, en el que los residuos son calculados en el espacio de entrada del autoencoder; y el enfoque RaPP —recientemente presentado en la literatura— en el que los residuos son calculados a lo largo de los espacios ocultos del autoencoder. Los resultados de la investigación —expresados en términos de *monotonicity*, *trendability* y *prognosability*— han demostrado la capacidad de nuestra propuesta para superar en rendimiento a sus competidores, en los tres conjuntos de datos considerados, e independientemente del tipo de autoencoder empleado para la generación de los residuos.

En vista de estos resultados, podemos afirmar que el error latente de los *deep autoencoders* es un valioso indicador de salud de los procesos que, en esta investigación, ha demostrado ser más preciso y coherente con respecto al proceso de degradación de las máquinas, que el error calculado tanto en el espacio de entrada como a lo largo de todos los espacios ocultos del autoencoder. Esto sugiere que la capacidad de los espacios latentes para aprender representaciones compactas y con significado de los datos les permite capturar con éxito las principales fuentes de variación presentes en los mismos, como, por ejemplo, la degradación de las máquinas.

En conclusión, el enfoque profundo propuesto ha sido capaz de capturar fielmente el grado de degradación de las máquinas bajo estudio, ante tres conjuntos de datos diferentes, y sin necesidad de ninguna información de contexto acerca de los sistemas. Por tanto, estos resultados son una evidencia del potencial de los espacios latentes como valiosas herramientas para la monitorización de la salud en sistemas de ingeniería.

CAPÍTULO 6

Visualización de mapas de estados de los procesos

En este capítulo se aborda el uso de arquitecturas profundas para la generación de mapas de estados 2D de los procesos. En detalle, se presenta el uso de *deep autoencoders*, en combinación con técnicas de analítica visual, para la creación de mapas visuales e interactivos de los sistemas bajo estudio, que permitan explorar su estado de funcionamiento de forma sencilla e intuitiva. En particular, se presenta un mapa 2D que ha sido utilizado para analizar los patrones de consumo energético de una gran instalación, como es el Hospital de León. Adicionalmente, se ha trasladado este enfoque al ámbito biomédico, a fin de explorar su potencial en este campo, donde, al igual que ocurre en los sistemas de ingeniería, la monitorización de la condición de los procesos también es crítica. En este caso, el mapa 2D generado ha sido empleado como herramienta para el análisis de la motilidad celular en procesos de cáncer. El presente capítulo comienza con una revisión del estado del arte, para detallar a continuación los experimentos realizados y los resultados obtenidos, terminando con un apartado de conclusiones.

6.1. Antecedentes

A lo largo de los últimos años, han tenido lugar numerosos avances en las redes de comunicaciones, así como en los sistemas de adquisición y almacenamiento de datos, que han propiciado la aparición de entornos *inteligentes*. Estos entornos están constituidos por dispositivos conectados, entre sí y a la red, en lo que se conoce como Internet de las cosas *(Internet of Things, IoT)*, y desempeñan un papel clave en la transformación digital de la industria, con lo que se encuentran habitualmente integrados en el concepto de Industria 4.0. En este contexto, los entornos industriales gozan de un elevado número de sensores inteligentes, que capturan el estado de los sistemas en cada instante de tiempo, generando así cantidades masivas de datos.

Esta alta disponibilidad de datos se traduce en conjuntos de datos de gran tamaño, que representan una valiosa fuente de información acerca del estado de sa-

lud de los procesos. No obstante, el gran tamaño de estos *datasets* también conlleva ciertas limitaciones: dado su elevado número de muestras y la alta dimensionalidad de las mismas, la exploración de estos conjuntos —en busca de información útil sobre los sistemas— puede convertirse en una tarea enormemente compleja para el operador humano.

Ante esta situación, se recurre habitualmente a técnicas de análisis de datos capaces de transformar de forma automática los datos brutos de trabajo en información de valor para el usuario. Varios ejemplos han sido presentados en los Capítulos 3, 4 y 5, donde se han propuesto diferentes enfoques de aprendizaje profundo, que han demostrado resolver con éxito problemas de interés en la literatura —como son la clasificación, la detección de anomalías o la generación de indicadores de salud— a partir de datos de operación de los sistemas.

Complementariamente, otra línea de trabajo en la literatura consiste en recurrir a enfoques de visualización de datos, que faciliten al usuario la exploración de estos *datasets* de gran tamaño. El objetivo de dichos enfoques consiste en proporcionar al usuario representaciones convenientes de los conjuntos de trabajo, que le permitan sacar conclusiones de forma visual e intuitiva, a pesar del gran volumen de muestras y/o dimensiones a manejar. En este contexto, cabe destacar que nuestra capacidad de percepción visual nos impide manejar cómodamente más de dos o tres variables (2D, 3D) de forma simultánea en una misma representación. Por tanto, el estudio de las interacciones entre variables en *datasets* de alta dimensionalidad puede convertirse en una tarea compleja y tediosa, que limita nuestra habilidad para extraer conocimiento a partir de los datos. Por ello, visualizaciones de baja dimensión como las basadas en gráficos de dispersión, histogramas, mapas de calor, etc., son populares en el estado del arte y han demostrado facilitar al usuario la exploración de los datos, ayudando a extraer información de valor para la monitorización de la condición de los procesos [220, 221]. De esta manera, las técnicas de visualización de datos se han convertido en valiosas herramientas de análisis, gracias a las cuales es posible mejorar nuestra comprensión acerca de los sistemas bajo estudio, favoreciendo así la generación de nuevo conocimiento y, también, la formulación de nuevas preguntas a resolver sobre los sistemas.

No obstante, a pesar de que estas visualizaciones facilitan la exploración de los conjuntos de datos, cabe puntualizar que un *dataset* de alta dimensión puede llegar a tener decenas, cientos o incluso miles de variables, con lo que la visualización directa de estas variables a través de gráficos de baja dimensión daría lugar a un elevado número de gráficos a analizar por parte del usuario, lo cual representa de nuevo una tarea tediosa para el operador humano y dificulta la obtención de una visión global del proceso bajo estudio. Ante esta situación, se recurre a técnicas de reducción de la dimensión *(Dimensionality Reduction, DR)*, que permiten visualizar los *datasets* al completo mediante su proyección en un espacio de dimensión menor [222, 223].

En este contexto, las técnicas DR tienen como objetivo proporcionar representaciones de baja dimensión de los datos —habitualmente, de dimensión dos (2D) para favorecer su exploración mediante técnicas de visualización— que, a su vez, han de preservar la información de valor presente en el espacio original de trabajo (Figura 6.1). Para ello, las técnicas DR capturan la información relevante en los da-

tos, descartando aquella redundante o superflua, proporcionando como resultado representaciones compactas y con significado de los datos originales. La integración de estas representaciones en enfoques de visualización de datos da lugar a visualizaciones interpretables de los *datasets*, que permiten explorar su estructura de una forma intuitiva, ayudando a mejorar la comprensión de los procesos y facilitando, por ejemplo, la detección visual de anomalías o la identificación de *clusters* en los datos [224, 225]. En consecuencia, estas técnicas tienen un gran potencial en el ámbito de la monitorización de la condición de los procesos y han sido empleadas en variadas aplicaciones, como la detección de intrusiones en el tráfico de red [226], el análisis de los sistemas de climatización o del consumo eléctrico en grandes edificios [227, 228], la clasificación de materiales o el estudio de los mecanismos químicos que explican sus propiedades [229, 230], la detección de fallos en procesos metalúrgicos [231], la monitorización del estado de carga de las baterías en vehículos eléctricos [232], el diagnóstico de fallos en instalaciones hidráulicas [233], etc.

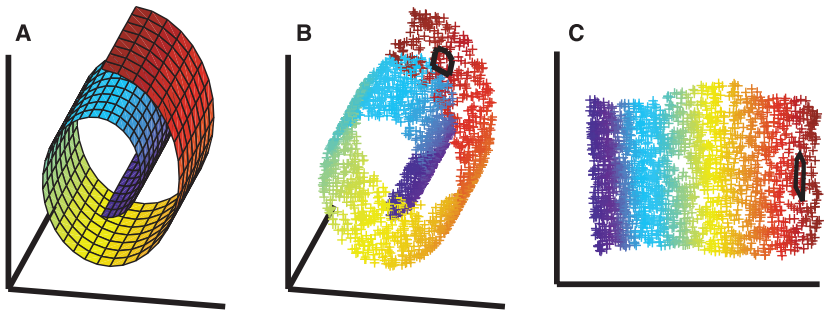


Figura 6.1: Ejemplo de reducción de la dimensión para un conjunto de datos de tipo *swiss roll*: a) Estructura *swiss roll*; b) Conjunto de datos 3D, muestreado a partir de (a); c) Reducción de la dimensión del conjunto de trabajo de 3D a 2D. Figura extraída de [234].

Entre las técnicas DR más populares en la literatura se encuentra el Análisis de Componentes Principales *(Principal Component Analysis, PCA)* [235, 236], que propone transformar linealmente el conjunto de variables originales en un nuevo conjunto de variables denominadas *componentes principales*, que no están correlacionadas entre sí y tienen varianza máxima. Este análisis PCA ha sido ampliamente empleado en aplicaciones de ingeniería [237] pero, a lo largo de la última década, ha sido reemplazado por nuevos enfoques, capaces de encontrar no solo dependencias lineales en los datos, sino también aquellas más complejas de tipo no lineal. Entre estos enfoques, destaca el t-SNE *(t-distributed Stochastic Neighbour Embedding)* [174], que es capaz de identificar *clusters* de muestras semejantes entre sí en el espacio original de los datos y separarlos en un espacio de baja dimensión, también llamado *espacio latente*. Para ello, el algoritmo t-SNE caracteriza la similitud entre muestras como una probabilidad, que después trata de preservar en el espacio latente. Este método ha obtenido un gran éxito en la literatura [238, 239] aunque, recientemente, ha sido superado por la técnica UMAP *(Uniform Manifold Approximation and Projection)* [240], que no solo identifica la presencia de *clusters* en los datos, sino que proporciona también información acerca de la estructura global de los mismos, de manera que es posible inferir similitudes y diferencias entre *clusters* a partir de su proximidad en el espacio latente

[241].

A pesar del éxito de estos métodos DR, con la llegada del *deep learning* ha cobrado relevancia otra línea de estudio, que consiste en aprender representaciones de baja dimensión de los datos a través de autoencoders profundos [242]. Estos modelos han demostrado proporcionar representaciones desenredadas *(disentangled representations)* de los datos [243, 244], con una mejor interpretabilidad y un mayor significado de los mismos que aquellas representaciones obtenidas a través de otros enfoques DR y, también, demostrando una mayor habilidad en el manejo de conjuntos de datos de gran tamaño, con un elevado número de muestras y de dimensiones por muestra [245, 246]. Los autoencoders profundos, al igual que el resto de técnicas *deep learning*, gozan de una arquitectura composicional que les otorga la habilidad de encontrar representaciones con significado acerca de los datos —habilidad conocida como *representation learning* [8]— y que los convierte en excelentes extractores de características. Se trata de modelos de aprendizaje no supervisado, entrenados para reconstruir a su salida la misma información que reciben de entrada y que, adicionalmente, incorporan algún tipo de restricción en su arquitectura, como la presencia de un cuello de botella o *espacio latente*, con una dimensionalidad menor que la de los datos de entrada. De esta manera, los autoencoders se ven forzados a aprender la estructura intrínseca de los datos, lo cual les permite generar reconstrucciones de alta calidad, a pesar de la restricción impuesta en el cuello de botella del modelo [26].

Como resultado, los autoencoders proporcionan en sus espacios latentes versiones comprimidas de los datos, que capturan la estructura subyacente en los mismos, preservando sus patrones o características más relevantes y tratándose, por tanto, de valiosas representaciones de los datos originales [27, 247]. No obstante, cuando la dimensión impuesta en el cuello de botella es demasiado baja —inferior a la dimensionalidad intrínseca de los datos— la calidad de la representación latente empeora y, con ella, empobrece también la reconstrucción de los datos. Esta situación puede suponer una limitación en aplicaciones de visualización, donde se persiguen representaciones de muy baja dimensión —típicamente, 2D— de los datos, lo cual ha dado lugar a otra rama de estudio en la literatura, en la que se combina el uso de *deep autoencoders* junto con otras técnicas DR [248, 249, 250], aprovechando así los beneficios de ambas familias de métodos. Dichos enfoques proponen llevar a cabo una primera reducción de la dimensión de los datos utilizando un *deep autoencoder* para, a continuación, reducir la dimensión de su representación latente utilizando otra técnica DR, como UMAP. Como se expone en las próximas secciones, en nuestra investigación hemos explorado tanto el potencial de los *deep autoencoders*, como el de su combinación con la técnica UMAP, para la generación de mapas 2D —gráficos de dispersión 2D— de los *datasets* de trabajo.

Otro aspecto a considerar en esta tesis será la integración de dichos mapas en herramientas de visualización interactiva, que faciliten al usuario el análisis exploratorio de los datos. La visualización interactiva de datos constituye un interesante campo de estudio en la literatura [251, 252, 253], que propone recurrir a elementos de interacción —*zoom*, navegación, selección múltiple, etc.— para generar interfaces dinámicas que favorezcan una exploración intuitiva de los *datasets* y faciliten al operador humano la identificación de relaciones relevantes

en los datos, así como la generación de nuevas hipótesis. Estas interfaces permiten, además, llevar a cabo operaciones sobre la representación de los datos —por ejemplo, operaciones de ordenación o filtrado— en base a criterios establecidos de forma interactiva por el operador humano (en la Figura 6.2 se ilustra el modelo de visualización de Van Wijk [254], que refleja el flujo de información que tiene lugar en estas interfaces). De esta manera, se obtienen potentes herramientas de análisis, que no solo combinan los beneficios de la interacción, la visualización de datos y el aprendizaje automático, sino que incorporan también conocimiento humano experto, introducido por el propio usuario en el manejo de la herramienta. En nuestra investigación, se explorará el potencial de estos enfoques empleando para ello la conocida librería de visualización interactiva Bokeh[1] de Python.

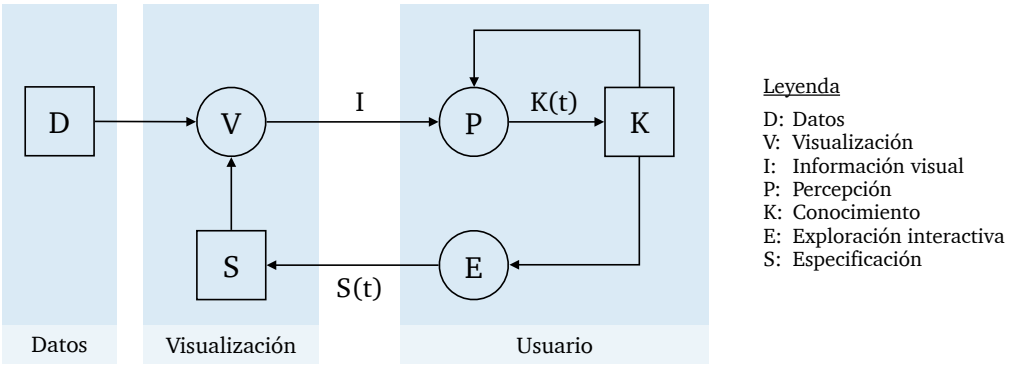


Figura 6.2: Modelo de visualización de Van Wijk que ilustra la relación entre los datos, la visualización y el usuario. El conocimiento del usuario K depende de su conocimiento actual y de la información visual I que le llega a través del sistema de percepción P. Basándose en su conocimiento K, el usuario puede modificar la visualización V mediante su exploración interactiva E, que le permite fijar nuevas especificaciones S para reconfigurar así la vista actual de la visualización V. Los círculos denotan procesos que transforman entradas en salidas, mientras que los cuadrados denotan contenedores de datos. Figura adaptada de [254].

La generación de mapas interactivos representa, por tanto, un interesante nicho de investigación, con potenciales aplicaciones en una amplia variedad de ámbitos. En particular, en esta tesis exploraremos los mapas generados a partir de autoencoders profundos, considerando dos casos de estudio que cubren diferentes ámbitos de aplicación. En primer lugar, se abordará el análisis del consumo energético en grandes edificios, mediante la exploración de un conjunto de datos proporcionado por el grupo SUPPRESS[2] de la Universidad de León, el cual contiene registros de la potencia consumida en una de las alas del Hospital de León[3]. En segundo lugar, se propone abordar el análisis de la motilidad celular, a partir de un conjunto de datos proporcionado por el grupo de Cáncer de Cabeza y Cuello

[1] `https://bokeh.org`

[2] SUPPRESS: Grupo de Supervisión, Control y Automatización de la Universidad de León (`https://suppress.unileon.es`).

[3] El conjunto de datos ha sido obtenido gracias a la colaboración entre el Hospital de León y el grupo SUPPRESS en el marco del proyecto DPI2015-69891-C2-1/2-R coordinado con el grupo de investigación GSDPI de la Universidad de Oviedo, y, a su vez, gracias a la colaboración actual entre ambos grupos en el marco de los proyectos PID2020-115401GB-I00 (GSDPI) y PID2020-117890RB-I00 (SUPPRESS).

del ISPA[4], que contiene vídeos de ensayos celulares relacionados con procesos de cáncer. En ambos casos se trata de conjuntos de datos de alta dimensión, donde se requiere de herramientas que faciliten al usuario la exploración de los *datasets*, en busca de información de valor acerca del estado y/o naturaleza de los sistemas bajo estudio.

En lo que respecta al primer caso de estudio, las políticas de ahorro y eficiencia energética en grandes edificios —impulsadas tanto a nivel nacional, como en coordinación con la Unión Europea— suponen un reto de gran importancia estratégica. Por ello, en busca de un futuro más sostenible, se trabaja en la actualidad en el desarrollo de nuevas medidas energéticas que favorezcan un funcionamiento más eficiente de los edificios. Habitualmente, se dispone de gran cantidad de datos acerca del consumo energético en estas instalaciones, con lo que el diseño de nuevas medidas suele comenzar con un análisis exploratorio de los datos, de manera que el usuario pueda adquirir una percepción del uso de la energía y sea así capaz de sugerir estrategias para una mayor eficiencia. Dado el gran tamaño de los conjuntos de trabajo, este análisis se lleva a cabo a través de herramientas de visualización de datos, capaces de proporcionar una representación compacta e intuitiva de los *datasets*, que permita al usuario descubrir y comprender los factores implicados en el consumo de energía [255, 256, 257]. En este contexto, cabe destacar que el consumo en grandes edificios ha demostrado estar particularmente vinculado con patrones de comportamiento humano —de frecuencia horaria, semanal, mensual, etc.— con lo que se ha dedicado especial interés en la literatura al estudio de la relación entre las variables temporales de contexto —hora del día, día de la semana, mes del año, etc.— y la evolución del consumo energético [228, 258].

En consecuencia, hemos propuesto explorar en esta tesis el potencial de los autoencoders profundos para la generación de un mapa 2D interpretable y con significado del consumo energético en las instalaciones del Hospital de León, el cual incorpore mecanismos de interacción que permitan explorar la influencia de las variables temporales de contexto en los patrones de consumo del Hospital. Como se expone en las próximas secciones, el mapa obtenido ha permitido identificar de forma intuitiva la presencia de *clusters* en los datos —asociados a ciertos rangos de horas y días de la semana— mejorando así la comprensión del usuario acerca del funcionamiento de la instalación. Adicionalmente, la exploración interactiva del mapa ha facilitado la detección de comportamientos anómalos en el consumo del Hospital.

Por su parte, la motilidad celular hace referencia a la habilidad de las células para moverse espontánea e independientemente, y representa un proceso dinámico crucial en un amplio abanico de procesos biológicos. En particular, este movimiento se encuentra relacionado con la supervivencia celular y, por tanto, con la metástasis, que representa en la actualidad uno de los grandes retos en el tratamiento clínico del cáncer. No obstante, la motilidad celular es un fenómeno complejo, afectado por el contexto fisiológico, el tipo de célula, su morfología, o las interacciones célula-célula. Además, las células pueden moverse en modo ameboide, mesenquimal o epitelial, como individuos o en grupos, etc. e incluso pueden

[4]ISPA: Instituto de Investigación Sanitaria del Principado de Asturias (`https://www.ispasturias.es`).

cambiar dinámicamente entre diferentes modos en respuesta a entornos cambiantes [259]. Por lo tanto, existe un gran interés en el estudio de los mecanismos que subyacen a todos los tipos de motilidad celular, con el objetivo final de identificar terapias que aumenten la motilidad de las células beneficiosas y bloqueen la propagación de las células dañinas [260].

En este contexto, cabe destacar que los avances en microscopía han desempeñado un papel clave en el estudio de los procesos celulares, pues han facilitado la adquisición de imágenes y/o vídeos de los cultivos celulares de interés, de manera que es posible analizar la evolución de los mismos a lo largo del tiempo. Como resultado, se generan conjuntos de datos de gran tamaño, cuya exploración podría aportar intuición acerca de nuevas hipótesis o líneas de investigación en el estudio de la motilidad celular. Sin embargo, la exploración manual de estos *datasets* —que pueden contener cientos, miles o, incluso, millones de imágenes— consume una gran cantidad de tiempo por parte del operador humano, para el cual, además, la tarea de encontrar relaciones de interés en tal vasta cantidad de datos resulta enormemente compleja. Ante esta situación, proponemos explorar un conjunto de datos constituido por vídeos de ensayos celulares, mediante su proyección en un espacio de baja dimensión, empleando para ello un autoencoder profundo. De esta manera, trasladaremos el enfoque propuesto para el análisis del consumo energético a un nuevo problema de trabajo, en este caso, de ámbito biomédico, con el propósito de encontrar patrones explicativos o con significado acerca de los mecanismos de motilidad celular.

Cabe recordar que, como se exponía en el Capítulo 1, las técnicas de aprendizaje profundo han demostrado un gran potencial en el manejo de datos complejos y el éxito que ya han tenido en otros campos —como el reconocimiento facial y de voz, o el procesamiento del lenguaje natural— ha comenzado a extenderse hacia nuevos ámbitos. En particular, se espera que en los próximos años el aprendizaje profundo tenga un gran impacto en los sistemas de ingeniería —motivación de esta tesis— así como en otros campos con problemáticas similares, como el de la biomedicina [261], donde la monitorización de la condición de los procesos también es crítica. Hasta el momento, las técnicas *deep learning* ya han sido empleadas con éxito, por ejemplo, en problemas de análisis de expresión genética, en la predicción estructural de proteínas, en la segmentación de imágenes médicas o en la predicción de trayectorias de migración celular [36, 262, 263]. No obstante, existe una amplia variedad de enfoques profundos en la literatura que podrían ser transferibles a este nuevo ámbito y cuya aplicabilidad en problemas biomédicos está aún por explorar.

En este escenario, los enfoques profundos empleados para la monitorización de la condición en procesos e instalaciones de ingeniería cobran especial relevancia, dado que podrían ser también útiles en el análisis de procesos biomédicos. En detalle, los enfoques planteados en esta tesis —detección y diagnóstico de fallos, detección de anomalías, generación de indicadores de salud, visualización de mapas de estados 2D— podrían ser potencialmente extrapolables a aplicaciones de detección y diagnóstico de enfermedades, detección de anomalías médicas, generación de indicadores de salud de pacientes, generación de mapas 2D de los procesos biológicos bajo estudio, etc. En consecuencia, hemos propuesto explorar en nuestra investigación el potencial de dichos enfoques —en particular, la visua-

lización de mapas de estados 2D de los procesos— sobre un problema biomédico, como es el estudio de la motilidad celular. Como se refleja a continuación, los resultados de la investigación han demostrado que el mapa 2D generado permite identificar diferentes patrones de movimiento en las células, proporcionado así una evaluación preliminar de las muestras de trabajo, que representa un valioso punto de partida para análisis posteriores. Estos resultados evidencian además el potencial de la transferencia de conocimiento entre estos dos ámbitos, estableciendo así una interesante línea de trabajo futuro.

6.2. Método propuesto

En esta sección se presentan las arquitecturas profundas empleadas para la generación de mapas 2D de los procesos. También se incluye la descripción de los conjuntos de datos utilizados en los experimentos.

6.2.1. Conjuntos de datos

6.2.1.1. Consumo eléctrico

En primer lugar, se ha considerado un *dataset* proporcionado por el grupo SUPPRESS de la Universidad de León, que contiene registros del consumo eléctrico en el Hospital del León. En detalle, estos datos proceden de un medidor colocado en el *Cuadro General de Baja Tensión 2*, que ha registrado el consumo en una de las alas del hospital —la cual contiene plantas de hospitalización, un centro de procesamiento de datos, salas con equipos de diagnóstico, etc.— durante un periodo de un año y con frecuencia de un minuto.

Este *dataset* ha sido normalizado mediante un escalado min-max [140] de rango $[0, 1]$ y, a continuación, ha sido enventanado utilizando ventanas de tamaño 60 elementos —equivalente a una hora de consumo— y con un solapamiento de 20 elementos —equivalente a un solapamiento de 20 minutos entre muestras. Las dimensiones del conjunto de trabajo resultante se indican en la Tabla 6.1. Cabe destacar también que este conjunto ha sido dividido aleatoriamente en dos subconjuntos, de entrenamiento y test (70 % y 30 % de las muestras, respectivamente), empleados para el entrenamiento y evaluación del autoencoder profundo presentado en la Sección 6.2.2.

Tabla 6.1: Conjunto de datos de consumo eléctrico (el tamaño está expresado como: número de muestras $\times$ número de elementos en las muestras).

	Conjunto de trabajo	Subconjunto de entrenamiento	Subconjunto de test
Tamaño:	26278 $\times$ 60	18394 $\times$ 60	7884 $\times$ 60

6.2.1.2. Motilidad celular

En segundo lugar, se ha considerado un *dataset* proporcionado por el grupo de investigación liderado por la Dra. María Dolores Chiara, perteneciente al área de investigación de Cáncer de Cabeza y Cuello del ISPA. Este conjunto de datos consta de cinco vídeos de cultivos celulares, caracterizados por niveles muy bajos —casi inexistentes— de SDHB. En la literatura se ha descrito que una menor expresión o actividad de la succinato deshidrogenasa B (SDHB), normalmente como consecuencia de mutaciones en el gen que la codifica, se relaciona con la patogénesis de numerosos carcinomas renales [264, 265, 266, 267, 268]. Por ello, existe un gran interés en el estudio de esta proteína, en busca de una mayor comprensión acerca de su implicación en la carcinogénesis y en la progresión de este tipo de tumores. El movimiento de las células podría desempeñar un papel clave en el desarrollo de dichos procesos, con lo que hemos propuesto una exploración de los vídeos de trabajo enfocada al análisis de su motilidad celular.

El procesamiento de los vídeos en busca de información de valor sobre el movimiento de las células supone una tarea compleja, incluso para las arquitecturas profundas. Por ello, se recurre habitualmente a un preprocesamiento previo de los datos, que permita extraer información relevante de los vídeos, con la que alimentar posteriormente al modelo profundo, simplificando —y orientando— así su aprendizaje. En particular, se ha llevado a cabo el cálculo del campo de velocidad de las células, dado que este proporciona una descripción detallada de su movimiento y se trata de un enfoque común en el análisis de la motilidad celular [269, 270]. Para ello, se ha empleado el algoritmo de flujo óptico de Gunnar Farnebäck [271], disponible en la librería OpenCV-Python[5]. Cabe destacar que los algoritmos de flujo óptico han empezado a ser empleados recientemente en el ámbito celular, pero ya han demostrado ser capaces de proporcionar campos de velocidad más precisos y robustos que otras técnicas más extendidas en el estado del arte, como las basadas en velocimetría de imágenes de partículas *(Particle Image Velocimetry, PIV)* [272, 273]. En el Vídeo 6.1 se muestra una visualización del campo de velocidad obtenido para uno de los vídeos de trabajo.

Vídeo 6.1

Como se indica en la Tabla 6.2.a, el conjunto de partida consta de 5 vídeos, de 539 fotogramas cada uno que, a su vez, tienen un tamaño de 2000×2000 píxeles, donde para cada píxel es conocida una variable —en este caso, su nivel de gris. Tras el cálculo de los campos de velocidad, se dispone de un nuevo conjunto (Tabla 6.2.b), donde para cada píxel son conocidas cuatro variables: las componentes horizontal y vertical de su velocidad (u, v) y la posición del píxel en la imagen (x, y). A continuación, los campos de velocidad han sido sometidos a un post-procesado que, como se aprecia en la Tabla 6.2.c, ha alterado tanto el número de fotogramas como el tamaño de los mismos. En primer lugar —y por recomendación del grupo que ha proporcionado los datos— han sido preservados los primeros 360 fotogramas de cada vídeo, mientras que el resto han sido descartados del estudio, al presentar una alta densidad celular y ser poco representativos del estado del proceso. En segundo lugar, se ha realizado un diezmado de los campos (con ven-

[5]El algoritmo ha sido aplicado utilizando los siguientes parámetros (expresados según la notación de la librería Object Tracking de OpenCV-Python): pyr_scale = 0.5, levels = 3, winsize = 60, iterations = 3, poly_n = 5, poly_sigma = 1.1, flags = cv2.OPTFLOW_FARNEBACK_GAUSSIAN.

tanas cuadradas de tamaño 4x4 y sin solapamiento), a fin de reducir el tamaño del conjunto y facilitar así su procesamiento; también, han sido descartados los bordes (superior e inferior) de los campos, al corresponderse con las franjas (superior e inferior) de los fotogramas, que no contienen ningún tipo de información celular. Como resultado, el campo de cada fotograma se ha visto reducido de un tamaño de 2000×2000 a uno de 355×500.

Tabla 6.2: Conjunto de datos de motilidad celular (a) y conjuntos derivados de su preprocesamiento (b, c, d, e). El nº de muestras de cada conjunto está expresado como: a) nº de vídeos × nº de fotogramas × ancho del fotograma × alto del fotograma; b y c) nº de vídeos × nº de fotogramas × ancho del campo × alto del campo; d y e) nº de vídeos × nº de fotogramas × nº de ventanas.

Conjunto de datos	Nº de muestras	Tamaño de las muestras
a) Datos de partida	(5×539×2000×2000)	(1)
b) Campos de velocidad	(5×539×2000×2000)	(4)
c) Campos de velocidad post-procesados	(5×360×355×500)	(4)
d) HOOF	(5×360×24)	(16)
e) HOOF temporal	(5×344×24)	(16×16)

A continuación, se ha procedido a la extracción de un descriptor representativo de los campos de velocidad, como es el histograma de flujo óptico orientado *(Histogram of Oriented Optical Flow, HOOF)* [274], ilustrado en la Figura 6.3. Estos histogramas han demostrado en la literatura un gran potencial como descriptores del movimiento en variadas aplicaciones [275, 276, 277], también con usos recientes en el estudio de la motilidad celular [278]. En nuestro caso, hemos dividido los fotogramas de trabajo en 24 ventanas —de tamaño 75×75 y sin solapamiento— y hemos calculado un HOOF de 16 elementos para cada una de ellas. Las dimensiones del *dataset* resultante se indican en la Tabla 6.2.d.

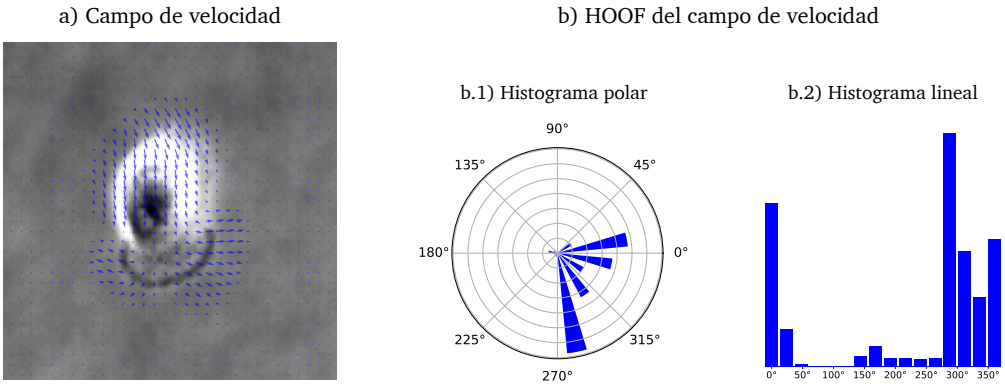


Figura 6.3: Ejemplo de HOOF para una ventana de trabajo. En (a) se observa la ventana, junto con su correspondiente campo de velocidad. En (b) se muestra el HOOF del campo de velocidad con diferentes representaciones: en forma de histograma polar (b.1) y en forma de histograma lineal (b.2). En este ejemplo se ha generado un HOOF de 16 *bins* o elementos.

Cabe destacar que las muestras de este conjunto de datos contienen valiosa información acerca del desplazamiento de las células dentro de cada ventana.

Sin embargo, esta información se limita al desplazamiento experimentado entre un fotograma y el siguiente. A fin de obtener muestras con un mayor contexto temporal de la evolución de las células, se ha fusionado —para cada ventana— el HOOF actual con los HOOF de los 15 fotogramas siguientes, dando lugar al conjunto de trabajo definitivo (Tabla 6.2.e) constituido por 41280 muestras — 5×344×24 muestras— de tamaño 16×16.

Finalmente, este *dataset* fue normalizado mediante un escalado min-max [140] de rango $[0, 1]$ y dividido aleatoriamente en dos subconjuntos, de entrenamiento y test (70 % y 30 % de las muestras, respectivamente) (Tabla 6.3), empleados para el entrenamiento y evaluación del autoencoder profundo presentado en la Sección 6.2.2.

Tabla 6.3: Conjunto de datos de motilidad celular (el tamaño está expresado como: número de muestras × tamaño de las muestras).

	Conjunto de trabajo	Subconjunto de entrenamiento	Subconjunto de test
Tamaño:	41280 × (16×16)	28896 × (16×16)	12384 × (16×16)

6.2.2. Generación de mapas 2D

Los *datasets* de trabajo no son directamente visualizables, al estar constituidos por muestras con un elevado número de dimensiones. Ante esta situación, proponemos llevar a cabo una reducción de la dimensión de los datos, que permita proyectar los *datasets* en un espacio de dimensión dos (2D), visualizable y fácil de explorar por parte del operador humano.

Esta reducción de la dimensión ha sido abordada por medio de autoencoders profundos, que proporcionan en su espacio latente una versión compacta y con significado de los datos de entrada. Cabe destacar que la dimensión de dicho espacio latente es, habitualmente, un hiperparámetro del modelo, particularmente elegido para optimizar el rendimiento del autoencoder —por ejemplo, en términos de su error de reconstrucción. Sin embargo, en este caso se trata de un parámetro prefijado, por motivos de visualización, a dos dimensiones.

En detalle, fueron entrenados dos *deep autoencoders* (uno por *dataset*) utilizando el algoritmo de descenso del gradiente [80] en combinación con el optimizador ADAM [97]. El número de épocas, el tamaño del *mini-batch* y la arquitectura de cada modelo se indican en la Tabla 6.4. En dicha tabla se observa que el autoencoder entrenado con el *dataset* de consumo eléctrico está constituido por capas densas; mientras, en el caso del *dataset* de motilidad celular, el autoencoder incluye también capas convolucionales y de submuestreo, especialmente recomendadas en la literatura para el procesamiento de datos 2D [62], como es el caso de las muestras que constituyen este *dataset*. En cuanto a las funciones de activación, se ha utilizado la función ReLU en todas las capas, excepto en la capa de salida y en el cuello de botella del modelo, donde se ha empleado la función de activación lineal.

En la tabla se observa también que el autoencoder entrenado con el *dataset* de motilidad celular no tiene un cuello de botella 2D —fácilmente visualizable— sino 10D. El entrenamiento del autoencoder se ha visto empobrecido con dimensiones

Tabla 6.4: Arquitectura del *deep autoencoder* para cada *dataset*. Todas las capas son de naturaleza densa, a excepción de las siguientes: Conv2D (capa convolucional 2D), MaxPooling2D (capa de submuestreo por valor máximo), Flatten (capa de conversión unidimensional), Reshape (capa de conversión multidimensional), Conv2DTranspose (capa de deconvolución). Las capas convolucionales y de submuestreo han sido configuradas con un relleno *(padding)* de tipo nulo y un paso *(stride)* de valor 1, salvo en el caso de la primera capa de deconvolución para la cual se ha empleado un paso de valor 2.

	Nº de épocas	Tamaño *mini-batch*	Nº de capas	Nº de neuronas en las capas		
				Encoder	Cuello de botella	Decoder
Consumo eléctrico	200	300	9	(60,90,180,90)	(2)	(90,180,90,60)
Motilidad celular	600	1000	15	(16×16×1, 14×14×8 Conv2D, 12×12×4 Conv2D, 10×10×2 Conv2D, 5×5×2 MaxPooling2D, 50 Flatten, 20)	(10)	(20, 50, 5×5×2 Reshape, 12×12×2 Conv2DTranspose, 14×14×4 Conv2DTranspose, 16×16×8 Conv2DTranspose, 16×16×1 Conv2D)

inferiores a 10D, con lo que se ha mantenido este número de dimensiones en el espacio latente y se ha llevado a cabo una reducción de la dimensión adicional, de 10D a 2D, utilizando la técnica UMAP, como se propone en otros trabajos de la literatura [248, 249, 250]. Para el entrenamiento del modelo UMAP se ha empleado un número de 5 vecinos y una distancia mínima de 0.5.

En lo que respecta a la elección de los hiperparámetros de cada modelo (número de capas, número de neuronas en las capas, número de épocas, tamaño del *mini-batch*, etc.), se han ejecutado diferentes abanicos de experimentos y se han elegido aquellos hiperparámetros con los que el modelo ha demostrado un menor error de reconstrucción. Para cada *dataset*, dicho error de reconstrucción ha sido evaluado sobre el subconjunto de test, mientras que el modelo ha sido alimentado con el subconjunto de entrenamiento.

6.3. Resultados

En esta sección se exponen los mapas 2D generados para cada conjunto de trabajo. Adicionalmente, se presenta la exploración de estos mapas mediante su integración en herramientas de visualización interactiva.

6.3.1. Mapa 2D del *dataset* de consumo eléctrico

Una vez finalizado el entrenamiento del autoencoder, es posible proyectar cualquier nueva muestra entrante sobre su espacio latente. En este caso, hemos proyectado el conjunto de datos al completo, a fin de obtener una representación 2D del mismo que proporcione una visión global del consumo eléctrico en el hospital. Esta proyección o mapa 2D de los datos se expone en la Figura 6.4.

El mapa obtenido permite distinguir la presencia de *clusters* en los datos, re-

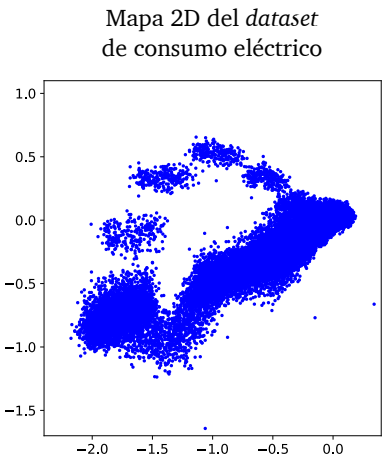


Figura 6.4: Proyección del *dataset* de consumo eléctrico sobre el espacio latente del *deep autoencoder* previamente entrenado.

velando así la existencia de diferentes patrones de consumo eléctrico a lo largo del *dataset*. Con el propósito de facilitar la exploración del mapa, en busca de relaciones de interés entre *clusters*, este ha sido integrado en una herramienta de visualización interactiva, expuesta en la Figura 6.5.

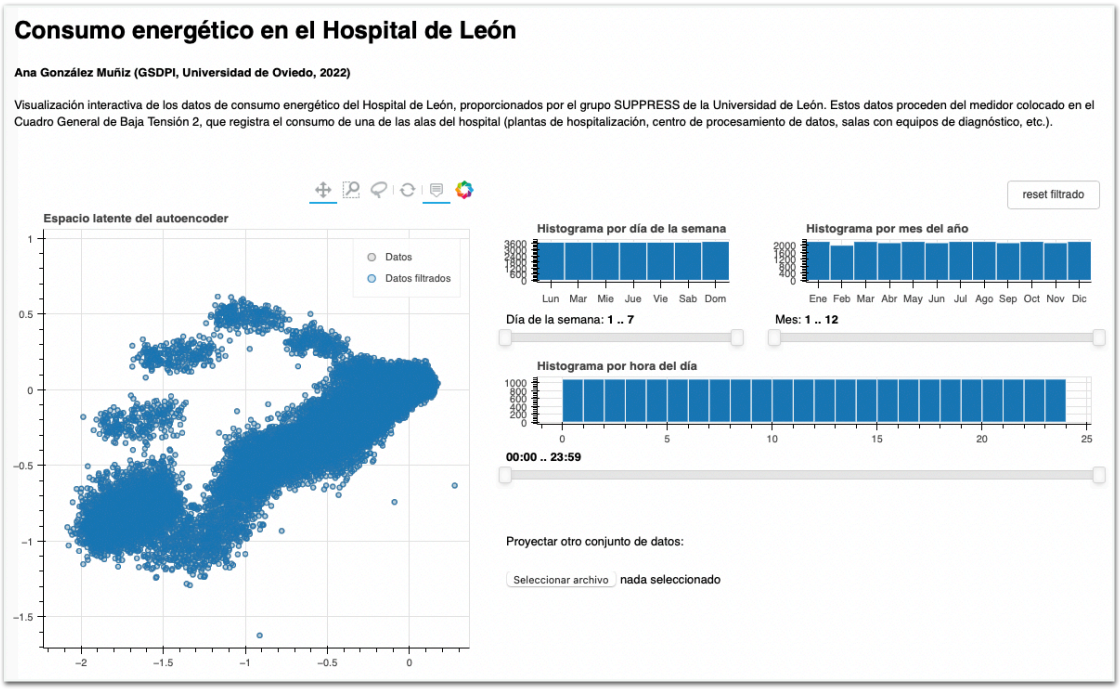


Figura 6.5: Visualización interactiva del mapa de consumo eléctrico.

Esta herramienta permite navegar por el mapa, gracias a elementos de desplazamiento y *zoom*. También permite visualizar la información de contexto asociada a cada muestra, en este caso, variables temporales (día de la semana, mes del año y hora del día a la que fue recogida la muestra), al posicionar el cursor sobre la misma. Adicionalmente, la herramienta incluye tres histogramas de los datos, uno para cada variable de contexto, a través de los cuales es posible filtrar las muestras visibles en el mapa. De igual manera, también es posible filtrar los propios histo-

gramas, seleccionando en el mapa el subconjunto deseado de muestras mediante la herramienta de lazo. Como resultado, la combinación de todos estos elementos da lugar a una herramienta de visualización interactiva, intuitiva y de fácil manejo para el usuario, que ha demostrado contribuir a mejorar la comprensión del mapa obtenido. En particular, esta herramienta ha permitido identificar diferentes patrones de consumo en el mapa, ayudando también a detectar la presencia de patrones anómalos en los datos. El uso de la herramienta se expone en el Vídeo 6.2 y los resultados de la exploración del mapa se detallan en las Figuras 6.6 y 6.8.

Vídeo 6.2

Como se aprecia en la Figura 6.6, la exploración del mapa ha permitido encontrar relaciones de interés entre los *clusters* y las variables temporales de contexto. En detalle, el consumo del hospital durante los fines de semana —señalado en color gris en el mapa— se encuentra restringido a una pequeña zona del espacio latente. Mientras, el consumo de lunes a viernes se extiende por el resto del espacio, conformando *clusters* asociados a franjas horarias concretas: los consumos en horas tempranas del día —de seis a siete de la mañana— dan lugar a cuatro *clusters* fácilmente identificables y separados del resto de muestras; a continuación, el consumo de la mañana —de siete a una del mediodía— se manifiesta en un único *cluster*; en último lugar, los consumos de la tarde y la noche se agrupan en un gran conjunto, del cual forman parte también los consumos del fin de semana. En vista de estos resultados, el consumo del hospital parece estar estrechamente vinculado a la actividad humana, que es elevada al comienzo del día —arranque de equipos de diagnóstico y comienzo del turno de día en plantas de hospitalización— y que se va reduciendo progresivamente hasta la noche, cuando la actividad disminuye en las plantas de hospitalización y se limita a labores de vigilancia y mantenimiento en el resto de instalaciones, al igual que ocurre en periodos de fin de semana.

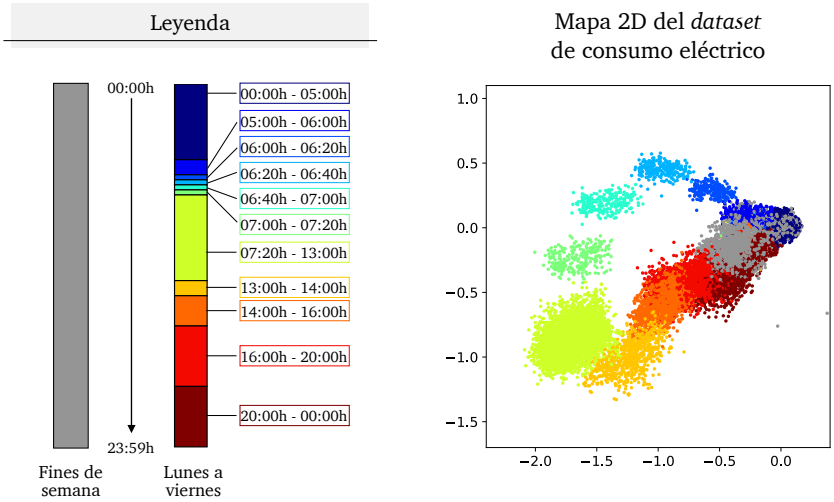


Figura 6.6: Mapa de consumo eléctrico etiquetado por franjas horarias.

También se expone en la Figura 6.7 una comparativa del mapa proporcionado por el autoencoder profundo y otro obtenido empleando la técnica de reducción de la dimensión UMAP[6], donde se observa que ambos enfoques son capaces de encontrar *clusters* en los datos. Sin embargo, gracias a su habilidad para aprender

[6]Para el entrenamiento del modelo UMAP se ha empleado un número de 5 vecinos y una distancia mínima de 0.5.

representaciones desenredadas de los datos *(disentangled representations)* [243, 244], el *deep autoencoder* proporciona un mapa más regularizado y coherente con respecto al perfil de consumo horario del hospital. Estos resultados concuerdan con otros trabajos en la literatura, donde se apunta la capacidad de los modelos profundos para proporcionar mapas con mayor significado acerca de la naturaleza de los procesos, que aquellos obtenidos a través de otros métodos DR [27, 247].

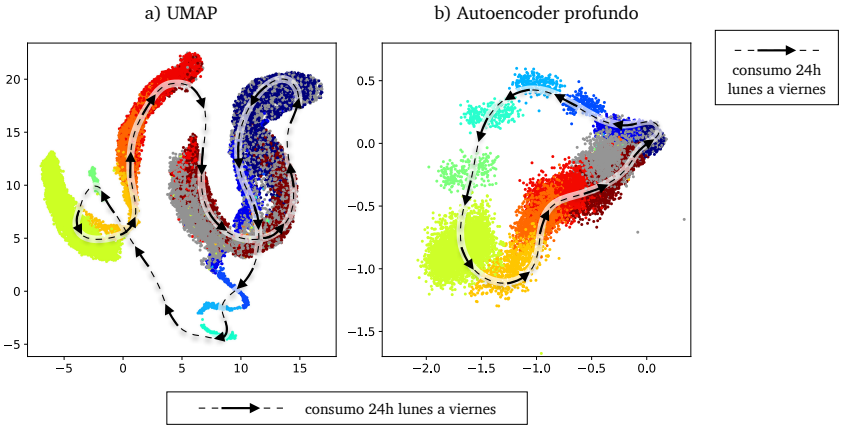


Figura 6.7: Comparativa de mapas de consumo eléctrico: a) mapa generado usando la técnica UMAP; b) mapa proporcionado por el autoencoder profundo.

La exploración interactiva del mapa también permite detectar de forma intuitiva la presencia de muestras anómalas en los datos. Un ejemplo de ello se presenta en la Figura 6.8, donde las muestras del mapa han sido filtradas —a través de los histogramas— para visualizar tan solo aquellas correspondientes con el consumo a las seis de la mañana, de lunes a viernes. Como se indicaba previamente, dicho patrón de consumo conforma un *cluster* fácilmente distinguible en el mapa, pero se observan también muestras alejadas de este *cluster*, que aparecen proyectadas en la zona del espacio asociada a los fines de semana. Estas muestras presentan, por tanto, un patrón de consumo diferente al esperado y pueden ser consideradas anómalas. Al explorar la información de contexto de dichas muestras, se observa que se corresponden con días festivos, en los que la actividad del hospital es reducida y se asemeja a la de los fines de semana, con lo que la proyección de estas muestras anómalas sobre la zona del mapa asociada a los fines de semana es coherente con la actividad del hospital. Cabe destacar que estas muestras anómalas han sido detectadas también en el resto de franjas horarias.

En último lugar, la herramienta incluye la posibilidad de proyectar un nuevo conjunto de datos sobre el mapa. Esta funcionalidad ha sido ilustrada en el vídeo de la herramienta, pero no ha sido explorada en nuestros experimentos.

6.3.2. Mapa 2D del *dataset* de motilidad celular

El *dataset* de motilidad celular ha sido proyectado en un espacio de dimensión dos, empleando para ello una reducción de la dimensión en dos pasos: en

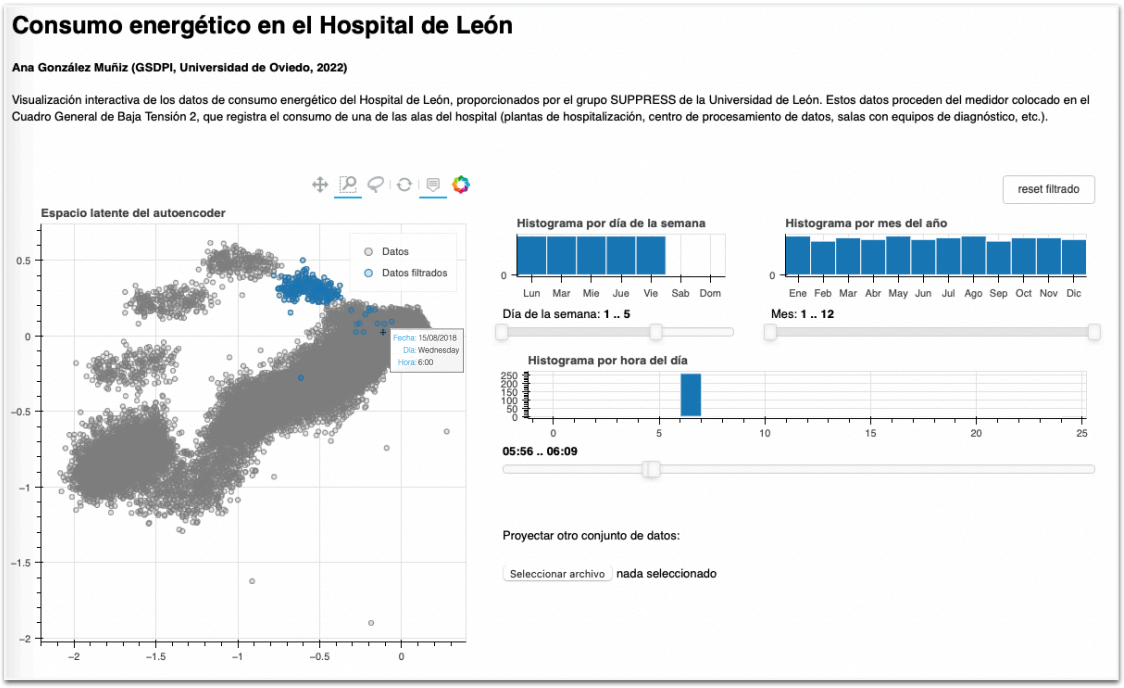


Figura 6.8: Detección de anomalías en el hospital a través de la visualización interactiva del mapa (el cursor está posicionado sobre una de las muestras anómalas, correspondiente al día 15 de agosto).

primer lugar, la dimensión original del *dataset* ha sido reducida de 256D[7] a 10D[8], mediante su proyección en el espacio latente de un autoencoder profundo; a continuación, los datos han sido reducidos de 10D a 2D, utilizando la técnica UMAP. Como resultado, se ha obtenido el mapa 2D de los datos que se presenta en la Figura 6.9.

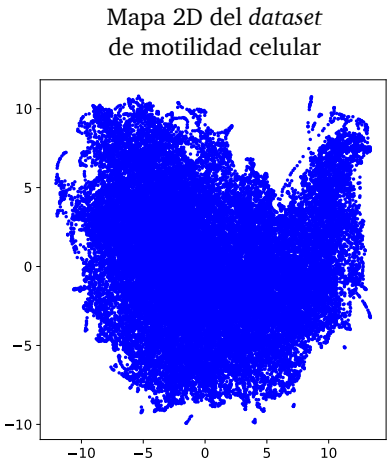


Figura 6.9: Proyección 2D del *dataset* de motilidad celular.

A fin de facilitar la exploración del mapa, en busca de posibles patrones de movimiento, este ha sido integrado en una herramienta de visualización interac-

[7]Las muestras de trabajo tienen un tamaño de 16×16 elementos, que es equivalente a 256D.

[8]Como se indica en la Sección 6.2.2, el autoencoder propuesto tiene un cuello de botella de 10D (dicho número de dimensiones fue elegido para minimizar el error de reconstrucción del modelo).

tiva, expuesta en la Figura 6.10. Esta herramienta permite navegar por el mapa, gracias a elementos de desplazamiento y *zoom*, así como visualizar la información de contexto asociada a cada muestra —en este caso, el nombre del vídeo al que pertenece la muestra, su número de ventana y fotogramas que representa— al posicionar el cursor sobre la misma. Adicionalmente, la herramienta permite seleccionar una muestra cualquiera en el mapa y visualizar tanto la muestra, como su clip de vídeo correspondiente. Como resultado, la combinación de todos estos elementos da lugar a una herramienta de visualización interactiva, intuitiva y de fácil manejo para el usuario, que ha demostrado contribuir a mejorar la comprensión del mapa obtenido. En particular, esta herramienta ha permitido identificar diferentes patrones de movimiento en el mapa. El uso de la herramienta se expone en el Vídeo 6.3 y los resultados de la exploración del mapa se detallan en la Figura 6.11.

Vídeo 6.3

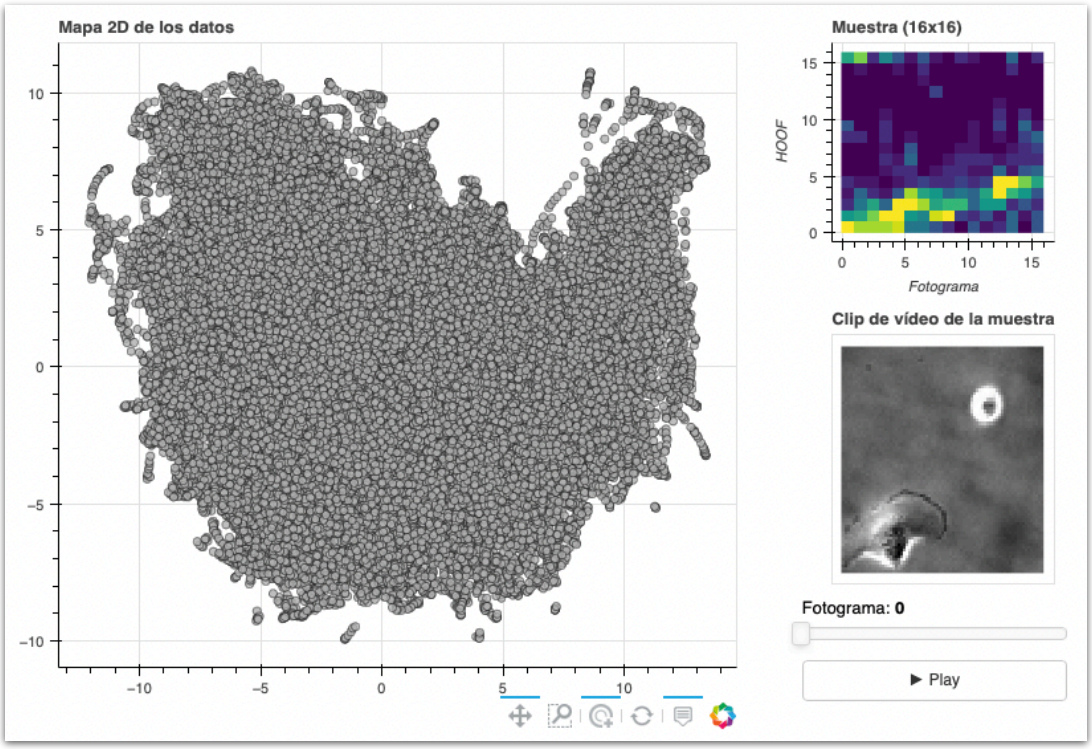


Figura 6.10: Visualización interactiva del mapa de motilidad celular. Cabe destacar que para la visualización de las muestras de trabajo se ha empleado un mapa de calor con la paleta de color *viridis*.

Como se aprecia en la Figura 6.11, la exploración del mapa ha permitido encontrar diferentes patrones de movimiento en los datos, representados en color en la figura (cabe destacar que el mapa no ha sido explorado en su totalidad, dado el gran volumen de muestras a manejar; las muestras no exploradas se representan en color gris). En detalle, se han identificado zonas con ausencia/presencia de células, zonas donde predominan los movimientos de rotación/desplazamiento, zonas de interacción entre células, etc. Por tanto, el mapa ha demostrado proporcionar una representación compacta y con significado de los datos, cuya semántica ha podido ser explorada gracias a su integración en la herramienta de visualización interactiva propuesta. En el Vídeo 6.4 se expone una animación del mapa,

Vídeo 6.4

donde se han incluido varias muestras de ejemplo para cada uno de los patrones de motilidad identificados.

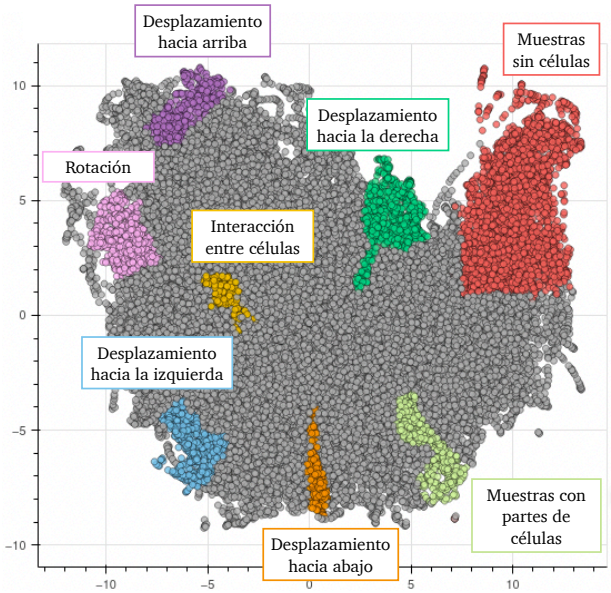


Figura 6.11: Mapa de motilidad celular etiquetado por patrones de movimiento (las muestras que no han sido etiquetadas se presentan en color gris).

Cabe destacar que este mapa 2D de los datos no solo constituye una valiosa herramienta para la exploración del *dataset* de trabajo, sino que proporciona también evidencia del potencial que los enfoques basados en el análisis de la motilidad celular podrían tener en la investigación de los procesos de cáncer. En particular, el mapa obtenido permite establecer potenciales líneas de trabajo futuro en el estudio de los mecanismos que relacionan la proteína SDHB con los procesos de desarrollo y evolución de los carcinomas renales. A partir del mapa sería posible analizar, por ejemplo, las diferencias de comportamiento entre los cultivos celulares con niveles bajos de SDHB —*dataset* de trabajo— y otros cultivos de diferente naturaleza —como podría ser el caso de cultivos de control o con niveles *normales* de SDHB— mediante su proyección sobre el mapa de trabajo. Adicionalmente, la identificación de diferentes zonas de motilidad en el mapa permite establecer también potenciales líneas de trabajo futuro relativas al etiquetado automático —no supervisado— de las muestras. Dicho etiquetado permitiría llevar a cabo un análisis detallado del conjunto de datos, descartando aquellas muestras sin información acerca del proceso —muestras sin células— o restringiendo el análisis a tan solo ciertos patrones de interés presentes en el mapa.

También se exponen en la Figura 6.12 los mapas obtenidos al realizar una reducción de la dimensión de los datos en un único paso —de 256D a 2D— utilizando tanto la técnica UMAP (subfigura b)[9] como un autoencoder profundo (subfigura c)[10], en comparación con el mapa propuesto (subfigura a), obtenido al combinar el uso de ambas técnicas. En esta comparativa se observa que, tanto la

[9]Para el entrenamiento del modelo UMAP se ha empleado un número de 10 vecinos y una distancia mínima de 0.5.

[10]El modelo profundo empleado es idéntico al propuesto, salvo por la dimensión de su cuello de botella que, en este caso, es 2D.

técnica UMAP, como el autoencoder profundo, presentan dificultades para capturar la estructura del *dataset* de trabajo en un espacio de tan baja dimensionalidad (2D). Sin embargo, la combinación de ambos enfoques proporciona una representación regularizada de los datos, que presenta también una mayor separabilidad de los diferentes patrones de motilidad celular. Como se indica en otros trabajos de la literatura [279, 280], el éxito de esta combinación se debe a que explota: por un lado, la capacidad de los autoencoders profundos para generar representaciones latentes desenredadas y de baja dimensión —en nuestro caso 10D— de los datos; y, por otro lado, la habilidad de la técnica UMAP para proporcionar representaciones compactas de dichas representaciones latentes, visualizables (2D) y con buenas propiedades de separabilidad, lo que las convierte en representaciones especialmente útiles en aplicaciones de *clustering*. En línea con dichos trabajos, los resultados obtenidos demuestran que, aún cuando la elevada dimensionalidad intrínseca de los datos impide capturar su estructura en un espacio de dimensión dos, es posible generar mapas 2D de calidad y con significado acerca de los procesos gracias a la combinación de ambas técnicas DR.

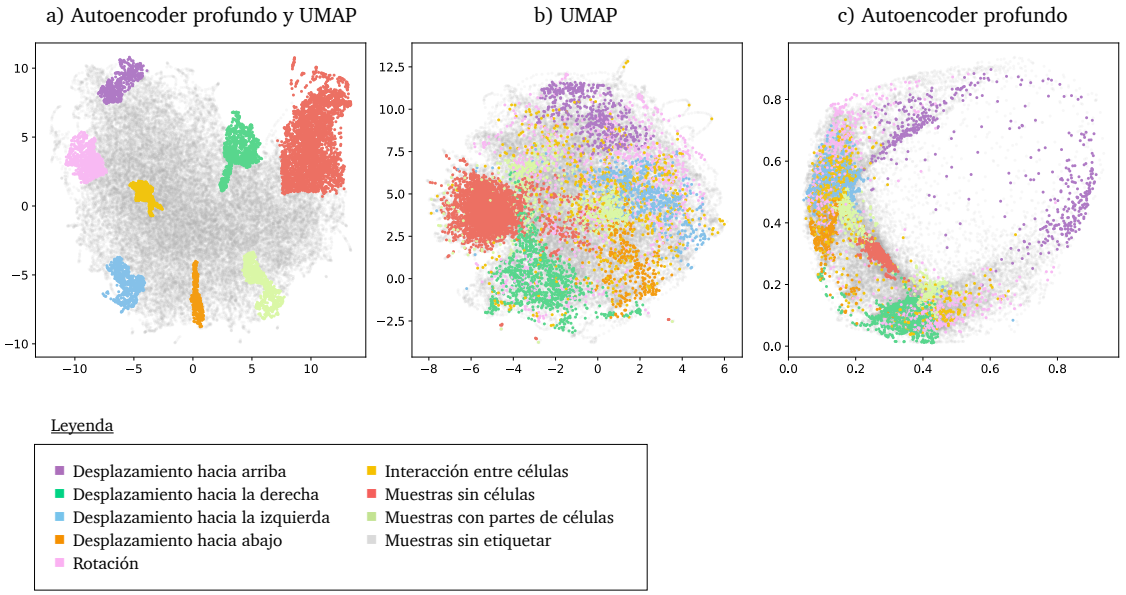


Figura 6.12: Comparativa de mapas de motilidad celular: a) mapa generado usando un autoencoder profundo en combinación con la técnica UMAP; b) mapa proporcionado por la técnica UMAP; c) mapa proporcionado por un autoencoder profundo.

6.4. Conclusiones

A lo largo de este capítulo se ha explorado el potencial de las arquitecturas profundas para la generación de mapas 2D de los procesos. En detalle, se ha recurrido al uso de autoencoders profundos para la obtención de representaciones latentes compactas, visualizables y con significado de los datos, que han sido utilizadas como mapas 2D de los procesos y que, a su vez, han sido comparadas con las representaciones proporcionadas por la técnica UMAP, que se trata de un enfoque popular en la literatura. Adicionalmente, estos mapas han sido integrados

en herramientas de visualización interactiva, facilitando así al usuario el análisis exploratorio de los datos.

En cuanto a los conjuntos de datos considerados en los experimentos, este enfoque ha sido evaluado sobre dos *datasets*, asociados a diferentes ámbitos de aplicación. En primer lugar, se ha considerado un *dataset* con registros del consumo eléctrico en el Hospital del León, que ha permitido explorar el potencial de nuestra propuesta en el ámbito de la ingeniería y, en particular, en la monitorización de la condición de una gran instalación, como es el Hospital de León. El mapa 2D generado para este *dataset* ha revelado la existencia de diferentes patrones de consumo en los datos, presentes en forma de *clusters* en el mapa. Adicionalmente, la exploración interactiva del mapa ha permitido relacionar dichos patrones con ciertas variables temporales de contexto, como el día de la semana, el mes del año o la hora del día. Como resultado, ha sido posible vincular el consumo en el hospital, en ciertas horas del día y días de la semana, con regiones específicas del mapa de trabajo. En último lugar, la exploración del mapa ha permitido detectar también la presencia de muestras anómalas en los datos, correspondientes con el consumo en el hospital en días festivos.

En segundo lugar, se ha trasladado este enfoque al ámbito biomédico, donde, al igual que ocurre en los sistemas de ingeniería, la monitorización de la condición de los procesos también es crítica. En particular, se ha generado un mapa 2D para el análisis de la motilidad celular en procesos de cáncer, a partir de un *dataset* constituido por vídeos de cultivos celulares, caracterizados por niveles muy bajos de SDHB —lo cual se ha relacionado en la literatura con la patogénesis de numerosos carcinomas renales. La exploración del mapa obtenido ha permitido identificar distintos patrones de movimiento en las células, cada uno asociado a una zona diferente del mapa. En detalle, se han identificado zonas con ausencia/presencia de células, zonas donde predominan los movimientos de rotación/desplazamiento, zonas de interacción entre células, etc. Por tanto, el enfoque propuesto ha permitido caracterizar los diferentes estados de movimiento que atraviesan las células a lo largo de los cultivos de estudio, proporcionando así evidencia del potencial que los enfoques profundos podrían tener en el análisis de la motilidad celular, la cual se encuentra estrechamente vinculada con la supervivencia celular y, en consecuencia, con los procesos de cáncer.

En conclusión, el enfoque profundo empleado ha demostrado ser capaz de capturar la estructura subyacente en los datos, para ambos *datasets*, generando como resultado mapas interpretables y con significado de los procesos bajo estudio. Adicionalmente, la integración de estos mapas en herramientas de visualización interactiva ha permitido explorar los datos de una forma sencilla e intuitiva para el usuario, contribuyendo a una mejora de la comprensión de los procesos. Finalmente, se han establecido potenciales conexiones entre el ámbito de los sistemas de ingeniería y el de la biomedicina, trasladando a este último, con éxito, el enfoque propuesto.

CAPÍTULO 7

Conclusiones y trabajo futuro

Este capítulo final recoge las conclusiones de nuestra investigación, las contribuciones de esta tesis en la literatura y posibles líneas de trabajo futuro.

7.1. Discusión y conclusiones finales

Esta tesis ha explorado las posibilidades de aplicación de las técnicas *deep learning* en el análisis y mejora de la eficiencia de los sistemas de ingeniería. Para ello, se ha analizado el rendimiento de distintos tipos de redes profundas sobre diferentes problemas y contextos de ingeniería, evaluando con ello sus potenciales contribuciones en este ámbito.

En el Capítulo 1 se ha introducido el propósito de esta investigación, exponiendo el impacto que las técnicas *deep learning* han tenido ya en otros campos y los posibles beneficios que podría conllevar su uso en el ámbito de la ingeniería y, en particular, en la monitorización de la condición de los sistemas. Las arquitecturas profundas han revolucionado por completo ámbitos como el procesamiento de imagen y vídeo o el procesamiento del lenguaje, con resultados sorprendentes —superiores a otros enfoques del estado del arte— en una amplia variedad de aplicaciones. Este éxito podría ser extrapolable a otros campos, como el de la ingeniería, que ha experimentado una transformación digital en los últimos años —de la mano de conceptos como la Industria 4.0 o el Internet de las Cosas— la cual ha dado lugar a la generación de cantidades masivas de datos, que requieren de potentes herramientas de análisis capaces de transformar estos datos en información de valor para las empresas. Ante esta necesidad y teniendo en cuenta la escalabilidad de las técnicas *deep learning* (cuyo rendimiento se incrementa cuanto mayor es el volumen de datos a manejar), se espera que los modelos profundos protagonicen en el futuro notables avances en el análisis y mejora de la eficiencia en procesos e instalaciones de ingeniería. En consecuencia, se ha llevado a cabo en la presente tesis un estudio del estado del arte actual y de las potenciales contribuciones de las técnicas *deep learning* en el ámbito de los sistemas de ingeniería.

Para ello, se ha comenzado en el Capítulo 2 por una revisión de la evolución de

las técnicas *deep learning*, desde sus inicios hasta nuestros días. A continuación, se han descrito los fundamentos de los modelos profundos y se han expuesto cuatro tipos de arquitecturas ampliamente empleadas en la literatura: redes *feedforward*, redes convolucionales, *deep autoencoders* y *variational autoencoders*. A lo largo del resto de capítulos de la tesis, se ha explorado el potencial de dichas arquitecturas en el ámbito de la monitorización de la condición de los sistemas, considerando para ello un amplio rango de problemas de ingeniería: clasificación (Capítulo 3), detección de anomalías (Capítulo 4), generación de indicadores de salud (Capítulo 5) y visualización de mapas de estados de los procesos (Capítulo 6).

En el Capítulo 3 se ha abordado la detección de fallos en máquinas rotativas y se ha evaluado el rendimiento de las arquitecturas profundas convolucionales en este ámbito. En detalle, se ha puesto especial interés en la capacidad de dichas arquitecturas para extraer características de forma automática *(feature learning)* a partir de los datos brutos de trabajo. Esta capacidad —compartida por todas las redes profundas— reside en su arquitectura composicional, que les confiere la habilidad de aprender representaciones intermedias de los datos de entrada *(representation learning)* y que, en el caso de las redes convolucionales, se ve potenciada gracias a la particular habilidad de sus filtros de convolución para capturar la presencia de patrones relevantes en los datos. En este contexto, se ha propuesto el uso de un modelo convolucional para la clasificación del estado de funcionamiento de una máquina a partir de datos crudos de operación —vibraciones y corrientes— de la misma. Gracias a este modelo, ha sido posible integrar en una única arquitectura las dos etapas de procesamiento presentes en los enfoques tradicionales de detección de fallos: extracción de características y clasificación de las características extraídas. Esto se debe a que los modelos profundos tienen la habilidad de aprender por sí mismos una representación óptima de los datos de entrada para, en base a ella, llevar a cabo la clasificación, eliminando así la necesidad de recurrir a métodos de ingeniería de características, que requieren de experiencia y conocimiento de dominio del sistema para llegar a monitorizar su estado de funcionamiento.

La arquitectura convolucional propuesta ha sido utilizada para clasificar el estado de una máquina con siete posibles estados de funcionamiento, y se ha comparado su rendimiento con el de otros clasificadores —entre ellos, las redes *feedforward*— basados en características extraídas manualmente. Los resultados obtenidos han indicado que el enfoque propuesto ha sido capaz de determinar la condición de la máquina con un acierto del 98 %, mostrando un rendimiento semejante al de los enfoques tradicionales pero siendo capaz de clasificar el estado del sistema sin necesidad de ningún tipo de conocimiento previo acerca del mismo. Adicionalmente, se ha llevado a cabo un análisis de los filtros de convolución del modelo, que ha permitido identificar las características aprendidas por los mismos durante el proceso de entrenamiento de la arquitectura. De esta manera, se ha proporcionado al usuario valiosa información acerca de la máquina, revelando aquellas características presentes en los datos que el modelo ha considerado relevantes para la clasificación de su estado de funcionamiento. Con ello, además, se ha aportado luz a las transformaciones internas de los datos que tienen lugar en los modelos profundos —a menudo considerados *cajas negras*—, favoreciendo la confianza del usuario en los resultados obtenidos y atenuando así una de sus prin-

cipales desventajas respecto a los métodos tradicionales de análisis, habitualmente más intuitivos para el usuario al estar construidos sobre su propio conocimiento del proceso. En último lugar, este enfoque ha sido empleado en la monitorización de otra máquina rotativa, donde también ha obtenido buenos resultados de clasificación.

En conclusión, el modelo profundo expuesto en el Capítulo 3 ha demostrado ser capaz de monitorizar con éxito la condición del sistema, sin necesidad de experiencia o conocimiento de dominio del problema y proporcionando, además, información sobre la máquina. Este enfoque también se ha mostrado competitivo en la monitorización de una máquina diferente, demostrando que el éxito del modelo podría ser fácilmente extrapolable a nuevos contextos de trabajo. Estos resultados son, en definitiva, una evidencia del potencial de las arquitecturas profundas como valiosas herramientas para la detección y diagnóstico de fallos en sistemas de ingeniería.

A lo largo del Capítulo 4 se ha explorado el potencial de los autoencoders profundos en el ámbito de la detección de anomalías. Para ello, se ha investigado un enfoque de redundancia analítica ampliamente utilizado en la literatura, que combina el uso de *deep autoencoders* junto con técnicas de análisis de residuos. En particular, se ha propuesto una extensión de este enfoque, que incluye un análisis novedoso de los residuos y que, en lugar del autoencoder tradicional, incorpora un autoencoder variacional (VAE). Los VAEs heredan la estructura de los *deep autoencoders*, incluyendo una restricción adicional en el cuello de botella que les obliga a aprender una distribución de probabilidad del espacio latente y gracias a la cual transforman la arquitectura determinista del autoencoder en un modelo probabilístico. En los últimos años, esta arquitectura ha ido creciendo en popularidad y ha demostrado resultados prometedores en aplicaciones de generación de imagen, creación de pistas musicales o diseño de nuevos compuestos moleculares. En detalle, el enfoque propuesto consta de: (1) un VAE, previamente entrenado para reconstruir muestras del comportamiento normal del proceso bajo estudio, de tal manera que los residuos de las nuevas muestras entrantes se conviertan en una medida de su desviación respecto al comportamiento normal esperado; (2) un algoritmo de clasificación en dos pasos o *two-step*, encargado de clasificar las muestras entrantes a partir de sus residuos, y capaz de determinar tanto su naturaleza normal/anómala, como la de sus componentes. Este enfoque ha sido evaluado en tres contextos de ingeniería diferentes (una máquina rotativa, un sistema hidráulico y un sistema de monitorización del movimiento humano) y su rendimiento ha sido comparado con el de otros enfoques de la literatura, tanto en términos de técnicas de reconstrucción (autoencoder variacional vs. *deep autoencoder*) como de clasificadores (clasificador *two-step* vs. clasificadores convencionales).

Los resultados de la investigación han demostrado la capacidad de nuestra propuesta para detectar anomalías con éxito en los tres contextos y con un rendimiento superior al de los enfoques convencionales. Esto sugiere que la habilidad del VAE para aprender la función de densidad de probabilidad (PDF) de los datos normales le confiere un rendimiento superior al obtenido por el *deep autoencoder*. Los *deep autoencoders* tienen la capacidad de modelar la geometría de los datos en el espacio de entrada, pero no su densidad, lo que les hace precisos en la reconstrucción de las muestras normales, pero también de algunas muestras anómalas;

los VAEs, en cambio, restringen la reconstrucción de los datos al soporte de la PDF aprendida, lo que les convierte en herramientas más eficientes y con un gran potencial en contextos de detección de anomalías. Por su parte, el clasificador propuesto también ha demostrado un mejor rendimiento que el resto de enfoques de la comparativa, con la ventaja adicional de que este proporciona al usuario no solo la clasificación de las muestras, sino también la de sus componentes. Por ello, se ha incluido adicionalmente un análisis visual de la clasificación por componentes de las muestras, que ha ilustrado la contribución del clasificador *two-step* a la mejora de la comprensión de los procesos bajo estudio, al proporcionar al usuario intuición acerca de la decisión de anomalía y facilitar también así el diagnóstico de la misma.

En conclusión, el enfoque profundo propuesto en el Capítulo 4 ha permitido detectar anomalías con éxito en tres contextos de ingeniería diferentes, proporcionando valiosa información adicional acerca de las muestras de trabajo, y sin necesidad de ninguna información de contexto acerca de los procesos o de anomalías previas en los mismos. Por tanto, estos resultados son una evidencia del potencial de las arquitecturas profundas como valiosas herramientas para la detección y diagnóstico de anomalías en sistemas de ingeniería.

En el Capítulo 5 se ha abordado la generación de indicadores de salud de los procesos y se ha evaluado el rendimiento de las arquitecturas profundas en este ámbito. En particular, se ha propuesto una variante de los enfoques tradicionales de análisis de residuos, en la que los residuos no han sido calculados en el espacio original de los datos, sino en un espacio de menor dimensión que, en nuestro caso, ha sido el espacio latente de un autoencoder profundo. Los *deep autoencoders* proporcionan en sus espacios latentes representaciones de baja dimensión, compactas y con significado de los datos, que capturan la estructura subyacente en los mismos. Nuestro enfoque ha tratado de explotar esta particularidad de los autoencoders profundos, para obtener así residuos con mayor significado acerca del estado de los procesos que aquellos calculados en el espacio de entrada de los datos. Por tanto, se ha propuesto emplear el error de reconstrucción latente de los autoencoders profundos como indicador de salud de las máquinas y su rendimiento ha sido evaluado ante tres conjuntos de datos diferentes, considerando dos tipos de autoencoders (*deep autoencoder* y *variational autoencoder*) y comparando su rendimiento con el de otros enfoques del estado del arte: el error de reconstrucción convencional, en el que los residuos son calculados en el espacio de entrada del autoencoder; el enfoque RaPP —recientemente presentado en la literatura— en el que los residuos son calculados a lo largo de los espacios ocultos del autoencoder; y varios descriptores estadísticos habitualmente empleados en la literatura como indicadores de salud de los procesos.

Los resultados obtenidos han demostrado que esta propuesta ha sido capaz de superar en rendimiento a sus competidores, en los tres conjuntos de datos considerados, e independientemente del tipo de autoencoder empleado para la generación de los residuos. Por tanto, el error latente de los *deep autoencoders* se ha revelado como un valioso indicador de salud de los procesos que, en esta investigación, ha demostrado ser más preciso y coherente con respecto al proceso de degradación de las máquinas, que el error calculado tanto en el espacio de entrada como a lo largo de todos los espacios ocultos del autoencoder. Esto sugiere que la

capacidad de los espacios latentes para aprender representaciones desenredadas de los datos, les permite capturar con éxito los principales modos de variación presentes en los mismos, como, por ejemplo, la degradación de las máquinas.

En conclusión, el enfoque profundo propuesto en el Capítulo 5 ha sido capaz de capturar fielmente el grado de degradación de las máquinas bajo estudio, ante tres conjuntos de datos diferentes, y sin necesidad de ninguna información de contexto acerca de los sistemas. Por tanto, estos resultados son una evidencia del potencial de los espacios latentes como valiosas herramientas para la monitorización de la salud en sistemas de ingeniería.

Finalmente, en el Capítulo 6 se ha explorado el potencial de las arquitecturas profundas para la generación de visualizaciones interpretables de los procesos. En detalle, se ha recurrido al uso de autoencoders profundos para la obtención de representaciones latentes compactas, visualizables (2D) y con significado de los datos, que han sido utilizadas como mapas 2D de los procesos y que han sido comparadas con las representaciones proporcionadas por la técnica de reducción de la dimensión UMAP. Adicionalmente, estos mapas han sido integrados en herramientas de visualización interactiva, facilitando así al usuario el análisis exploratorio de los datos. Este enfoque ha sido empleado en la monitorización de la condición de una gran instalación, el Hospital de León, a partir de un conjunto de datos constituido por registros del consumo eléctrico en el hospital. La exploración interactiva de este mapa ha revelado la existencia de diferentes patrones de consumo en los datos que han sido vinculados con ciertas variables temporales de contexto. Adicionalmente, la exploración del mapa también ha permitido identificar la presencia de muestras anómalas en los datos.

Complementariamente, dicho enfoque ha sido trasladado al ámbito biomédico, donde se espera que el aprendizaje profundo tenga un gran impacto en los próximos años, al igual que en los sistemas de ingeniería. En ambos campos, la monitorización de la condición de los procesos es crítica, con lo que los enfoques profundos empleados para la monitorización de la condición en procesos e instalaciones de ingeniería podrían ser también útiles en el análisis de procesos biomédicos. En consecuencia, se ha explorado el potencial de estos enfoques —en particular, la visualización de mapas de estados 2D de los procesos— sobre un problema biomédico, como es el estudio de la motilidad celular. Para ello, se ha analizado un conjunto de datos constituido por vídeos de cultivos celulares caracterizados por niveles muy bajos de SDHB —lo cual se ha relacionado en la literatura con la patogénesis de numerosos carcinomas renales. La exploración del mapa obtenido ha permitido identificar distintos patrones de movimiento en las células, proporcionado así una evaluación preliminar de las muestras de trabajo, que representa un valioso punto de partida para posteriores análisis.

En conclusión, el enfoque profundo expuesto en el Capítulo 6 ha demostrado ser capaz de capturar la estructura subyacente en los datos, para ambos *datasets*, generando como resultado mapas interpretables y con significado de los procesos bajo estudio. Dichos mapas han presentado, además, una mejor interpretabilidad y una mayor regularización que aquellos proporcionados por la técnica UMAP. Adicionalmente, la integración de estos mapas en herramientas de visualización interactiva ha permitido explorar los datos de una forma sencilla e intuitiva para

el usuario, contribuyendo a una mejora de la comprensión de los procesos y facilitando al operador humano la identificación de relaciones relevantes en los datos. De esta manera, se han obtenido potentes herramientas de análisis, resultado de combinar los beneficios de la interacción, la visualización de datos y el aprendizaje profundo en un único enfoque. Finalmente, se han establecido potenciales conexiones entre el ámbito de los sistemas de ingeniería y el de la biomedicina, evidenciando el potencial de la transferencia de conocimiento entre estos dos ámbitos.

En conjunto, esta investigación ha puesto de relevancia la habilidad de las técnicas *deep learning* para extraer información de valor acerca de los procesos, a partir de datos brutos de operación de los mismos. Estas técnicas han demostrado ser capaces de aprender valiosas representaciones intermedias de los datos de entrada —versiones compactas y *desenredadas* de los datos originales— encargadas de capturar aquella información relevante para la resolución del problema objetivo. Esta habilidad para extraer características de los datos de forma automática, también conocida como *aprendizaje de características*, ha permitido abordar un amplio rango de problemas a lo largo de esta tesis sin necesidad de recurrir a etapas previas de *ingeniería de características* —propias de los enfoques tradicionales de análisis de datos— cuyo diseño consume habitualmente una gran cantidad de tiempo y requiere de experiencia y/o conocimiento previo acerca de los sistemas. En este contexto, las arquitecturas de tipo convolucional han demostrado una especial habilidad para la extracción automática de características, gracias a la capacidad de sus filtros de convolución para capturar patrones relevantes en los datos. No obstante, las arquitecturas de tipo denso —constituidas por capas no convolucionales— han sido ampliamente utilizadas a lo largo de la investigación, demostrando también un gran desempeño en este ámbito.

Cabe destacar también que, para extraer las características de los datos y procesarlas hasta obtener la salida objetivo, los modelos profundos llevan a cabo un elevado número de transformaciones simples de los datos de entrada, lo que les ha convertido en potentes herramientas de análisis. Sin embargo, estas transformaciones internas de los datos son difíciles de interpretar por parte del usuario, lo cual genera habitualmente desconfianza en los resultados obtenidos y, especialmente, en el ámbito de la ingeniería, tradicionalmente vinculado al diseño manual de características. Ante esta situación, se ha tratado de aportar luz a las transformaciones de los datos ejecutadas en los modelos propuestos a lo largo de la investigación. Para ello, se ha llevado a cabo una introspección en los modelos y se han visualizado ciertos parámetros internos de sus arquitecturas, así como algunas de las proyecciones intermedias de los datos disponibles en sus espacios ocultos. En este contexto destacan, por ejemplo: la visualización de los filtros de convolución aprendidos por el modelo de detección de fallos (Figura 3.11), la visualización de la clasificación por componentes de las muestras en el modelo de detección de anomalías (Figura 4.6), la visualización de las trayectorias de degradación de las máquinas en el modelo propuesto para la generación de indicadores de salud de los procesos (Figura 5.8) o la propia visualización de los mapas de estados 2D de los procesos (Figuras 6.6 y 6.11).

Con estas visualizaciones hemos demostrado que es posible mejorar la comprensión del usuario acerca de las transformaciones de los datos llevadas a cabo

por los modelos profundos, favoreciendo su interpretabilidad y atenuando así su condición de *cajas negras*. Esta introspección en los modelos ha proporcionado además información de valor acerca de los procesos bajo estudio, permitiendo al usuario incrementar su conocimiento de dominio de los sistemas, así como contrastar el ya existente. Además, cabe puntualizar que los modelos profundos no representan tan solo una alternativa a los métodos tradicionales de ingeniería de características, sino que estos pueden ser también combinados, aprovechando los beneficios de ambos. Una muestra de ello se ha presentado en el análisis de la motilidad celular (Sección 6.2.1.2), donde, para la generación de un mapa 2D de los datos, se ha empleado un autoencoder profundo precedido de una etapa previa de extracción manual de características —basada en la extracción del campo de velocidades y el histograma HOOF de los vídeos de trabajo— que ha permitido simplificar y orientar el aprendizaje del modelo profundo.

Otro aspecto de relevancia en esta tesis ha sido la versatilidad demostrada por los *deep autoencoders*. Estas arquitecturas han permitido abordar tres tipos de problemas (detección de anomalías, generación de indicadores de salud y visualización de mapas 2D de los procesos) explotando en cada uno de los casos una particularidad diferente de los autoencoders profundos. En detalle, los residuos de estos modelos han demostrado portar valiosa información acerca del estado de los procesos, que ha sido empleada para la detección de anomalías en los mismos; mientras, su error de reconstrucción latente ha demostrado capturar fielmente el grado de desviación de las muestras respecto a su comportamiento normal esperado, con lo que ha sido utilizado como indicador del nivel de degradación de los procesos; en último lugar, sus espacios latentes han demostrado capturar representaciones compactas y con significado de los datos, que han sido utilizadas como mapas 2D de los procesos. Por tanto, la habilidad de los autoencoders profundos para modelar la geometría de los datos de entrenamiento —y, en el caso de los VAEs, también su densidad— proporciona como resultado valiosas representaciones intermedias de los datos, disponibles a lo largo de su arquitectura. En esta investigación se ha demostrado que la exploración de dichas representaciones ha permitido abordar con éxito la monitorización de la condición de los sistemas en tres contextos diferentes, proporcionando así una evidencia del potencial de los autoencoders profundos en este ámbito.

Adicionalmente, a lo largo de esta tesis se ha puesto también en valor el uso de los *deep autoencoders* como valiosas herramientas para la exploración de conjuntos de datos no etiquetados, como es el caso de los *datasets* de consumo eléctrico y de motilidad celular expuestos en el Capítulo 6. Al tratarse de modelos de aprendizaje no supervisado y dada su capacidad para generar mapas 2D de los datos, los autoencoders profundos han permitido explorar el contenido de dichos *datasets* y, además, de una forma intuitiva para el usuario. De esta manera, los *deep autoencoders* han demostrado su potencial como valiosas herramientas de análisis, especialmente en aquellos contextos donde el usuario pretende llevar a cabo un análisis preliminar de los datos, que le permita familiarizarse con los conjuntos de trabajo, para generar así nuevas hipótesis y formular las preguntas a resolver en análisis posteriores.

En último lugar, cabe destacar que las arquitecturas profundas empleadas a lo largo de esta investigación constan de un elevado número de hiperparámetros a

definir por el usuario, como: el número de capas del modelo, el número de neuronas en cada capa, el tamaño del filtro o el número de filtros de convolución (en el caso de las capas convolucionales), el tamaño de la máscara o el tipo de submuestreo (en el caso de las capas de submuestreo), el tipo de función de activación, el número de épocas, el tamaño del *mini-batch*, etc. La selección de dichos hiperparámetros se ha basado en la ejecución de diferentes abanicos de experimentos y se han elegido aquellos hiperparámetros con los que los modelos propuestos han demostrado mejores resultados. En este contexto cabe puntualizar que el rendimiento de los modelos es altamente sensible a la configuración de parámetros elegida y se ha constatado experimentalmente que las arquitecturas convolucionales y los *variational autoencoders* han sido más sensibles a dicha selección que las arquitecturas *feedforward* y los *deep autoencoders*, respectivamente.

En conclusión, a lo largo de esta tesis se ha expuesto un recorrido por la historia y los fundamentos de las arquitecturas profundas, y se ha explorado su aplicabilidad en la monitorización de la condición en sistemas de ingeniería. Para ello, se ha considerado un amplio rango de problemas y contextos de ingeniería, en los que las arquitecturas profundas han logrado un gran rendimiento, demostrando así su potencial en este ámbito y su capacidad para transformar los datos de operación de los procesos en información de valor para el usuario.

7.2. Contribuciones de la tesis

Las contribuciones de esta tesis se resumen en los siguientes puntos:

- Se ha presentado un recorrido por la historia y los fundamentos de las técnicas *deep learning*.
- Se ha explorado el potencial de algunas de las arquitecturas profundas más populares en la literatura (redes *feedforward*, redes convolucionales, *deep autoencoders*, *variational autoencoders*) en diferentes problemas de ingeniería (clasificación, detección de anomalías, generación de indicadores de salud, visualización de mapas de estados de los procesos).
- Se ha propuesto un enfoque profundo convolucional para la detección de fallos en máquinas rotativas que ha combinado en una única arquitectura las dos etapas de procesamiento presentes en los enfoques tradicionales de detección de fallos: extracción de características y clasificación de las características extraídas.
- Se han visualizado y analizado las características aprendidas por los filtros de convolución del modelo de detección de fallos, proporcionando información a priori desconocida acerca de la máquina bajo estudio.
- Se ha propuesto un enfoque profundo de detección de anomalías —constituido por un *variational autoencoder* y un clasificador de dos pasos— capaz de detectar no solo la naturaleza normal/anómala de las muestras, sino también la de sus componentes, proporcionando así una imagen detallada del estado de los procesos.

- Se ha propuesto un indicador de salud de los procesos —una variante del enfoque RaPP— que ha sido construido en el espacio latente de un *deep autoencoder* y que ha sido empleado como indicador del nivel de degradación de las máquinas.
- Se ha propuesto un enfoque profundo para la generación de mapas 2D de los procesos, que ha combinado el uso de *deep autoencoders* y técnicas de analítica visual, dando como resultado mapas intuitivos e interactivos de los procesos.
- Se ha trasladado la generación de mapas 2D de los procesos al análisis de un proceso biomédico —el estudio de la motilidad celular— demostrando el potencial de la transferencia de conocimiento entre este ámbito y el de los sistemas de ingeniería.

7.3. Trabajo futuro

Los enfoques propuestos en esta tesis doctoral sugieren nuevas líneas de investigación y trabajo futuro, entre las que hemos destacado las siguientes:

- El estudio de mecanismos de *Explainable Artificial Intelligence (XAI)* [281] que favorezcan la explicabilidad e interpretabilidad de las arquitecturas convolucionales empleadas para la clasificación del estado de los procesos. En este contexto destacan enfoques como el método SHAP *(Shapley additive explanations)* [282] o los mapas de activación de clases *(Class Activation Maps, CAM)* [283], que permiten identificar qué patrones o características de una muestra son discriminantes para su clasificación en una determinada clase.
- La investigación de métodos para la estimación de la dimensionalidad intrínseca de los datos [284] que, en el contexto de los autoencoders profundos, faciliten al usuario la selección de un hiperparámetro óptimo para la dimensionalidad de su espacio latente.
- La exploración del potencial de arquitecturas profundas no empleadas en esta tesis, como las redes neuronales recurrentes *(Recurrent Neural Networks, RNNs)*, que destacan por su elevado rendimiento en el tratamiento de datos secuenciales y que han obtenido un gran éxito en tareas predictivas en los ámbitos del procesamiento de texto, audio y vídeo [285]. Estas arquitecturas podrían ser empleadas, por ejemplo, para la predicción del tiempo de vida de las máquinas *(Remaining Useful Life, RUL)* a partir del indicador de salud propuesto en el Capítulo 5.
- El estudio de la semántica de los espacios latentes de los *deep autoencoders*, en busca de direcciones con significado acerca de la naturaleza de los procesos [92]. En este contexto, cobra también especial interés el análisis comparativo de los espacios proporcionados por los *deep autoencoders* y aquellos obtenidos a través de los *variational autoencoders* (VAEs), para explorar así el impacto de la naturaleza generativa de los VAEs en la semántica de los

espacios latentes. Una posible línea de trabajo en este ámbito consistiría en replicar, utilizando *variational autoencoders*, los mapas 2D presentados en el Capítulo 6 de esta tesis y analizar los resultados obtenidos.

- La exploración del potencial de los enfoques empleados en esta investigación (clasificación, detección de anomalías, generación de indicadores de salud) en el ámbito biomédico.
- En último lugar, cabe destacar que los VAEs han demostrado un mejor rendimiento que los *deep autoencoders* en el ámbito de la detección de anomalías (Capítulo 4), al emplear el error de reconstrucción del autoencoder como indicador de anomalía. Mientras, en la generación de indicadores de salud de los procesos (Capítulo 5), donde se ha empleado como indicador de salud el error de reconstrucción latente del autoencoder, estas arquitecturas han demostrado un rendimiento semejante. Ante estos resultados, cabe explorar si el espacio elegido para el cálculo de los residuos ha atenuado las diferencias de rendimiento entre las dos arquitecturas.

APÉNDICE A

Publicaciones

En este apéndice se enumeran todas las publicaciones relacionadas con nuestra investigación. Se dividen en dos categorías: *1) Publicaciones principales*, que incluyen los resultados presentados en este documento en los Capítulos 3, 4 y 5; y *2) Publicaciones relacionadas*, en las que hemos colaborado durante el desarrollo de esta investigación.

A.1. Publicaciones principales

Heliyon 6 (2020) e03395

Contents lists available at ScienceDirect

Heliyon

www.elsevier.com/locate/heliyon

Research article

DCNN for condition monitoring and fault detection in rotating machines and its contribution to the understanding of machine nature

Ana González-Muñiz *, Ignacio Díaz, Abel A. Cuadrado

Electrical Engineering Dept., University of Oviedo, Edif. Dept., Campus de Viesques s/n, 33204, Gijón, Spain

ARTICLE INFO

Keywords:
Industrial engineering
Systems engineering
Artificial neural networks
Artificial intelligence
Signal processing
System fault detection
Process monitoring
Condition monitoring
Feature learning
Vibration analysis
Fault detection
Convolutional neural network

ABSTRACT

Rotating machines are critical equipment in many processes, and failures in their operation can have serious implications. Consequently, fault detection in rotating machines has been widely investigated. Conventional detection systems include two blocks: feature extraction and classification. These systems are based on manually engineered features (ball pass frequencies, RMS value, kurtosis, crest factor, etc.) and therefore require a high level of human expertise (it is a human who designs and selects the most appropriate set of features to perform the classification). Instead, we propose a system for condition monitoring and fault detection in rotating machines based on a 1-D deep convolutional neural network (1D DCNN), which merges the tasks of feature extraction and classification into a single learning body. The proposed system has been designed for use on a rotating machine with seven possible operating states and it proves to be able to determine the operating condition of the machine almost as accurately as conventional feature-engineered classifiers, but without the need for prior knowledge of the machine. The proposed system has also reported good classification on a bearing fault dataset from another machine, thus demonstrating its capability to monitor the condition of different machines. Finally, the analysis of the features learned by the deep model has revealed valuable and previously unknown machine information, such as the rotational speed of the machine or the number of balls in the bearings. In this way, our results illustrate not only the good performance of CNNs, but also their versatility and the valuable information they could provide about the monitored machine.

1. Introduction

Detection and diagnosis of faults are useful to optimize and guarantee the safety in the operation of machines, leading to higher productivity and process efficiency, with benefits such as reduced operating costs, longer machine life or improved operating uptime [1]. Bearings are essential components in rotating machines and their failure is one of the most common causes of machinery breakdown [1, 2]. The presence of these elements induces inherent system vibrations, which are generated not only under normal operating conditions but also under fault conditions (external raceway faults, internal raceway faults, rolling element faults, cage faults, imbalances, misalignments, etc.). Vibration analysis is therefore often used to monitor the operation of rotating machines and thus to detect system faults [3, 4].

Frequency domain analysis is a widely adopted technique [5, 6, 7] for the study of system vibrations. It requires knowledge about the fundamental frequencies of the system and proposes to monitor the amplitude of vibrations at such frequencies in order to detect anomalies. Although this technique provides good results, it has significant disadvantages. For example, manually designed frequencies may differ slightly from the real frequencies of the machine, as the bearings operate under a combination of rolling and sliding [8]. Another source of error is the simultaneous presence of different types of faults as well as the interference from additional sources of vibration, both of which can obscure important frequencies in the spectrum [6]. Finally, some defects, such as lubrication ones, do not manifest themselves as a new frequency, making them very difficult to detect across the frequency spectrum [9].

These drawbacks reveal the weaknesses of methods based on manually engineered features, whose performance depends to a large extent on the quality of the selected features. This situation has led to numerous studies on the optimal choice of features and it has been demonstrated that, in certain contexts [10, 11, 12, 13], fault features can be successfully extracted by using traditional methods of analysis. However, efficient fault detection remains a challenge when systems are very complex. In such cases, choosing the right features is still a diffi-

* Corresponding author.
E-mail address: anaglezmuniz@gmail.com (A. González-Muñiz).

https://doi.org/10.1016/j.heliyon.2020.e03395
Received 26 April 2019; Received in revised form 13 November 2019; Accepted 5 February 2020

Figura A.1: Artículo publicado en la revista *Heliyon* [101] con los resultados presentados en el Capítulo 3.

Computers and Electrical Engineering 101 (2022) 108065

Contents lists available at ScienceDirect

Computers and Electrical Engineering

journal homepage: www.elsevier.com/locate/compeleceng

Two-step residual-error based approach for anomaly detection in engineering systems using variational autoencoders☆

Ana González-Muñiz [a,*], Ignacio Díaz [a], Abel A. Cuadrado [a], Diego García-Pérez [a], Daniel Pérez [b]

[a] Electrical Engineering Department, University of Oviedo, Edif. Departmental Oeste 2, Campus de Viesques s/n, Gijón 33204, Spain
[b] SUPPRESS Research Group, University of León, Escuela de Ingenierías, Campus de Vegazana, León 24007, Spain

ARTICLE INFO

Keywords:
Anomaly detection
Novelty detection
Deep autoencoder
Variational autoencoder
Engineering systems

ABSTRACT

Anomaly detection is a crucial task in the engineering systems field. However, there is usually little or no information about all possible abnormal modes in systems. Hence, a common approach is to build a model of healthy behaviour, based on normal operation data, so that anomaly detection would depend on how well new data fit this model. According to this idea, we propose a residual-error based approach consisting of: a variational autoencoder, used to model the probability density function of the system's healthy behaviour; and a two-step classification algorithm, which classifies the incoming samples based on their residuals, and reports not only their normal/anomalous nature but also that of their components. We have tested this proposal in three different engineering contexts and we have compared its performance with that of state-of-the-art approaches, demonstrating its capability to successfully detect and characterize anomalies.

1. Introduction

Anomaly detection consists in finding those patterns in data that do not conform to expected normal behaviour. Such patterns, commonly known as anomalies or outliers, represent deviations from normal behaviour, so their detection is not only of great value, but is often critical in a wide variety of real-world applications, like intrusion detection [1], fraud detection [2] or medical anomaly detection [3]. In the context of industrial processes, anomaly detection is also a topic of great importance since both detection and diagnosis of faults are crucial to optimize and guarantee safety in the operation of machines, leading to higher productivity and process efficiency, with benefits such as reduced operating costs, longer operating life or improved operating uptime [4].

Given the relevance of anomaly detection, there has been extensive research on this topic: we can find several approaches in the state of the art, depending on the application domain and even the dataset [5]. In the typical scenario, we deal with datasets containing mostly normal behaviour data and few, if any, examples of anomalous behaviour, which sometimes are not even representative of all possible abnormal modes. This makes traditional multi-class classification schemes unsuitable for anomaly detection [6].

Instead, a common approach in the literature is to construct a model of normal behaviour (based on available normal data), so that anomaly detection would depend on how well new data fit this model [6]. In other words, the aim of this approach is to build an analytical model of the process under study (key concept in the digital twin approach [7]), in such a way that its comparison

☆ This paper is accepted by Associate Editor Dr. Shadi Aljawarneh.

* Corresponding author.
E-mail addresses: gonzalezmunana@uniovi.es (A. González-Muñiz), idiaz@uniovi.es (I. Díaz), cuadrado@isa.uniovi.es (A.A. Cuadrado), diegogarcia@isa.uniovi.es (D. García-Pérez), dperl@unileon.es (D. Pérez).

https://doi.org/10.1016/j.compeleceng.2022.108065
Received 1 July 2021; Received in revised form 28 April 2022; Accepted 1 May 2022

Figura A.2: Artículo publicado en la revista *Computers and Electrical Engineering* [142] con los resultados presentados en el Capítulo 4.

Reliability Engineering and System Safety 224 (2022) 108482

Contents lists available at ScienceDirect

Reliability Engineering and System Safety

journal homepage: www.elsevier.com/locate/ress

Health indicator for machine condition monitoring built in the latent space of a deep autoencoder

Ana González-Muñiz *, Ignacio Díaz, Abel A. Cuadrado, Diego García-Pérez

Electrical Engineering Department, University of Oviedo, Edif. Departmental Oeste 2, Campus de Viesques s/n, Gijón 33204, Spain

ARTICLE INFO

Keywords:
Health indicator
Deep autoencoder
Latent space
Anomaly detection
Engineering systems

ABSTRACT

The construction of effective health indicators plays a key role in the engineering systems field: they reflect the degradation degree of the system under study, thus providing vital information for critical tasks ranging from anomaly detection to remaining useful life estimation, with benefits such as reduced maintenance costs, improved productivity or increased machine availability. The reconstruction error of deep autoencoders has been widely used in the literature for this purpose, but this approach does not fully exploit the hierarchical nature of deep models. Instead, we propose to take advantage of the disentangled representations of data that are available in the latent space of autoencoders, by using the latent reconstruction error as machine health indicator. We have tested our proposal on three different datasets, considering two types of autoencoders (deep autoencoder and variational autoencoder), and comparing its performance with that of state-of-the-art approaches in terms of well-known quality metrics. The results of the research demonstrate the capability of our health indicator to outperform conventional approaches, in the three datasets, and regardless of the type of autoencoder used to generate the residuals. In addition, we provide some intuition on the suitability of latent spaces for the monitoring of machinery condition.

1. Introduction

In the last decade there has been a growing interest in the monitoring of machinery condition [1–3]. Machines are critical equipment in a wide variety of domains, and it has become crucial to optimize and guarantee safety in their operation, searching for higher productivity and process efficiency, and benefits such as reduced operating costs, longer machine life or improved operating uptime [4].

Condition monitoring systems in the literature have traditionally been focused on knowledge-based [5] or physical-based approaches [6]. However, with the rapid development of smart instruments, digital communication networks and computing techniques, there is nowadays an increasing availability of condition monitoring data, which has brought research attention to data-driven methods for the monitoring of machine health [7].

In the study of these methods, special efforts have been dedicated to the construction of health indicators (HIs) [8–10]. HIs reflect the degradation degree of the system under study, providing valuable input information for tasks such as fault diagnosis, anomaly detection (AD) or remaining useful life (RUL) estimation. For example, in the field of AD, the state of a system is usually declared anomalous when its HI (or HIs) meets certain conditions such as exceeding a predetermined threshold [11,12]. In prognostic tasks, such as RUL estimation, HIs are also used in many ways: similarity methods estimate the RUL of machines by comparison of degradation profiles that are built on one or more HIs [13]; other approaches propose to build degradation models, based on HI data, to predict the future condition of the machines [14], etc. The performance of these tasks heavily depends on the quality of the health indicator [15,16], so the construction of effective HIs plays a key role in the context of machine health monitoring.

In traditional knowledge-based or physical-based approaches, HIs are typically manually built, requiring expertise or prior knowledge of the system that can sometimes be difficult or even impossible to obtain. Also, these HIs are often designed for a specific degradation process, and it can be tough to make them generalize well to others. Hence, constructing a good HI is a challenging work that in data-driven methods does not rely on the human, but on powerful machine learning techniques able to extract HIs automatically from process data, also known as *feature learning* [17].

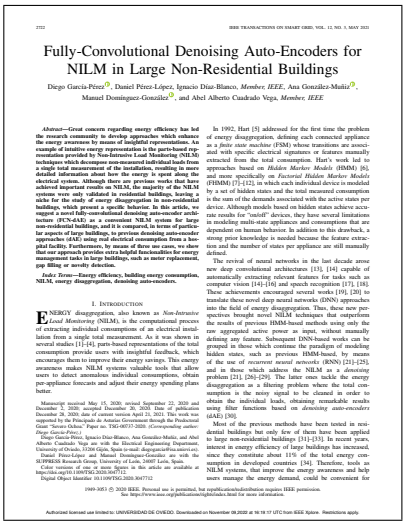
As a branch of machine learning, deep learning models have set in recent years an exciting new trend with a big impact on big data analysis [18], showing promising results in fields such as image or voice recognition [19,20], but also in system health management [21].

* Corresponding author.
E-mail addresses: anaglezmuniz@gmail.com (A. González-Muñiz), idiaz@uniovi.es (I. Díaz), cuadrado@isa.uniovi.es (A.A. Cuadrado), diegogarcia@isa.uniovi.es (D. García-Pérez).

https://doi.org/10.1016/j.ress.2022.108482
Received 15 June 2021; Received in revised form 9 February 2022; Accepted 20 March 2022
Available online 2 April 2022

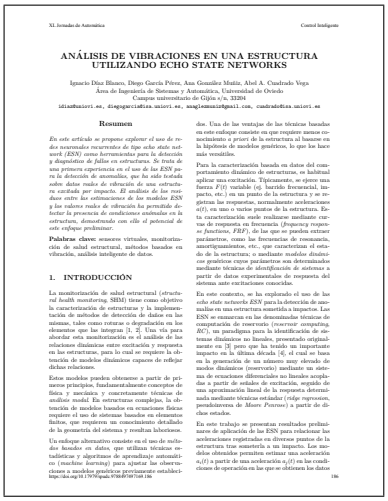
Figura A.3: Artículo publicado en la revista *Reliability Engineering & System Safety* [175] con los resultados presentados en el Capítulo 5.

A.2. Publicaciones relacionadas



Fully-Convolutional Denoising Auto-Encoders for NILM in Large Non-Residential Buildings

Figura A.4: Artículo publicado en la revista *IEEE Transactions on Smart Grid* [286]. Este artículo propone un enfoque profundo de tipo *deep autoencoder* para abordar un problema de ingeniería diferente a los considerados en esta tesis: la desagregación no intrusiva de la demanda eléctrica *(Non-Intrusive Load Monitoring, NILM)*, que consiste en estimar el consumo individual de los diversos dispositivos conectados a la red eléctrica, a partir de una lectura agregada de su consumo.



ANÁLISIS DE VIBRACIONES EN UNA ESTRUCTURA UTILIZANDO ECHO STATE NETWORKS

Figura A.5: Artículo presentado en las *XL Jornadas de Automática* [287]. Este artículo está relacionado con el problema de detección de anomalías en sistemas de ingeniería (Capítulo 4) que, en este caso, ha sido abordado mediante el análisis de los residuos de una arquitectura no profunda de tipo *echo state network (ESN)*.

XVII Simposio CEA de Control Inteligente

Modelado de series temporales mediante echo state networks para aplicaciones de analítica visual

Figura A.6: Artículo presentado en el *XVII Simposio CEA de Control Inteligente* [288]. Este artículo está relacionado con la generación de mapas 2D de los procesos (Capítulo 6) que, en este caso, ha sido abordada mediante el uso de una *echo state network (ESN)* en combinación con la técnica de reducción de la dimensión PCA.

Bioinformatics, 37(11), 2021, 1571–1580
doi: 10.1093/bioinformatics/btaa989
Advance Access Publication Date: 27 November 2020
Original Paper

Data and text mining

Morphing projections: a new visual technique for fast and interactive large-scale analysis of biomedical datasets

Ignacio Díaz[1,*], José M. Enguita[1], Ana González[1], Diego García[1], Abel A. Cuadrado[1], María D. Chiara[2,3] and Nuria Valdés[4]

[1]Department of Electrical Engineering, University of Oviedo, Gijón 33204, Spain, [2]Institute of Sanitary Research of the Principado de Asturias, Hospital Universitario Central de Asturias, Oviedo 33011, Spain, [3]CIBERONC (Network of Biomedical Research in Cancer), Madrid 28029, Spain and [4]Department of Internal Medicine, Section of Endocrinology and Nutrition, Hospital Universitario de Cabueñes, Gijón 33204, Spain

*To whom correspondence should be addressed.
Associate Editor: Jonathan Wren
Received on May 13, 2020; revised on November 9, 2020; editorial decision on November 13, 2020; accepted on November 16, 2020

Abstract

Motivation: Biomedical research entails analyzing high dimensional records of biomedical features with hundreds or thousands of samples each. This often involves using also complementary clinical metadata, as well as a broad user domain knowledge. Common data analytics software makes use of machine learning algorithms or data visualization tools. However, they are frequently *one-way* analyses, providing little room for the user to reconfigure the steps in light of the observed results. In other cases, reconfigurations involve large latencies, requiring a retraining of algorithms or a large pipeline of actions. The complex and multiway nature of the problem, nonetheless, suggests that user interaction feedback is a key element to boost the cognitive process of analysis, and must be both *broad* and *fluid*.

Results: In this article, we present a technique for biomedical data analytics, based on blending meaningful views in an efficient manner, allowing to provide a natural smooth way to transition among different but complementary representations of data and knowledge. Our hypothesis is that the confluence of diverse complementary information from different domains on a highly interactive interface allows the user to discover relevant relationships or generate new hypotheses to be investigated by other means. We illustrate the potential of this approach with three case studies involving gene expression data and clinical metadata, as representative examples of high dimensional, multidomain, biomedical data.

Availability and implementation: Code and demo app to reproduce the results available at https://gitlab.com/idiaz-blanco/morphing-projections-demo-and-dataset-preparation.

Contact: idiaz@uniovi.es

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Biomedical data is growing at astonishing rates with the broadening of access to massive analyses, including laboratory tests, medical image or gene expression data, bearing precise information about the underlying biological state of the subject. However, analysis of high dimensional biomedical datasets is rarely directly managed by doctors, even though they encompass a unique set of challenges, hard important problems and huge potential sanitary impact.

At the same time as quality and availability of biomedical data increased, accordingly did its application in biomedical research. Most interestingly, there has also been a huge research activity in the field of visualization of this high-dimensional data, as a very powerful tool for researchers, providing better data interpretation, easier detection of patterns and generation of new hypotheses. Related surveys about the use of these techniques can be found in Kamal *et al.* (2014) and O'Donoghue *et al.* (2018).

In this article, we set the focus on the analysis of genomic data for cancer research, which is a paradigmatic example of such problems. Many processes related to the onset, spreading and evolution of cancer, are strongly related to biological pathways that involve complex chains of regulation/deregulation mechanisms acting on

(a) Artículo publicado en la revista *Bioinformatics* [253].

ESANN 2022 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 5-7 October 2022, i6doc.com publ., ISBN 978287587084-1.
Available from http://www.i6doc.com/en/.

Interactive dual projections for gene expression analysis

Ignacio Díaz[1], José M. Enguita[1], Diego García[1],
Ana González[1], Abel A. Cuadrado[1], María D. Chiara[2] and Nuria Valdés[3] *

1- University of Oviedo - Dept of Electrical Engineering
Edificio Torres Quevedo, módulo 2, Campus de Gijón 33204 - SPAIN
2- Institute of Sanitary Research of the Principado de Asturias
Hospital Universitario Central de Asturias, Oviedo 33011 - SPAIN.
3- Department of Internal Medicine, Section of Endocrinology and Nutrition
Hospital Universitario de Cabueñes, Gijón 33204 - SPAIN.

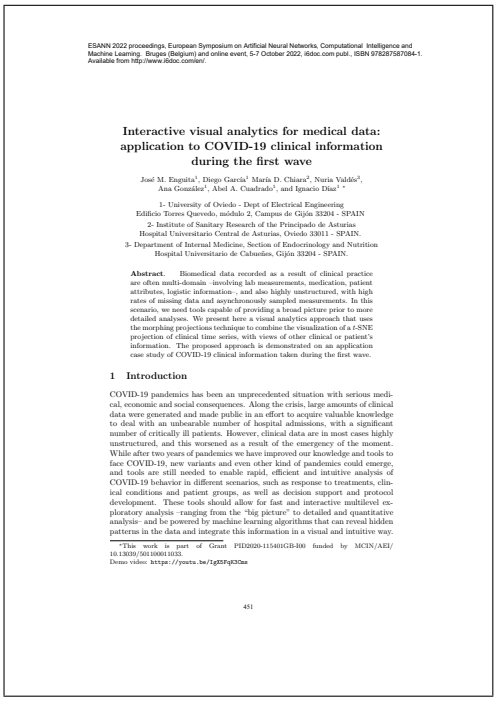
Abstract. We present an application of interactive dimensionality reduction (DR) for exploratory analysis of gene expression data that produces two lively updated projections, a sample map and a gene map, by rendering intermediate results of a *t*-SNE. The user can condition the projections "on the fly" by subsets of genes or samples, so updated views reveal co-expression patterns for different cancer types or gene groups.

1 Introduction

Gene expression data analysis is one of the key tools in biomedical research. The underlying processes taking place in cancer involve complex interactions among the genes and other factors, with up or down regulations at different transcriptional stages. As a result, the levels of expressions in the genes form a transcriptomic signature that can be used to study, differentiate, detect and diagnose cancer. Typically, the levels of expressions of thousands of genes are measured for hundreds of samples taken under different biological conditions, such as types of cancers. A widely used approach consists in using DR techniques like the *t-distributed stochastic neighbor embedding* (*t*-SNE) to visualize the samples in two dimensions [1] so near points represent transcriptomically similar samples. While most papers focus on projecting samples, genes can also be projected, as in [2], where a dual *t*-SNE is used for transcriptome-wide and samples-wide exploration of gene expression data. However, existing approaches assume transcriptomic similarities in all genes and samples, potentially ignoring associations of groups of genes with groups of samples. In this paper, we propose a method based on interactive DR [3], to produce simultaneous free-running sample and gene maps, allowing *human in the loop* user selection of groups of genes or samples to update both maps. We present encouraging results of our approach through three case studies. See demo at `https://gadpi.edv.uniovi.es/GEM-iDR`.

*This work is part of Grant PID2020-115401GB-I00 funded by MCIN/AEI/10.13039/501100011033. The results shown here are based upon data generated by the TCGA Research Network: `https://www.cancer.gov/tcga`.

(b) Artículo presentado en el *30th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning* [289].



ESANN 2022 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 5-7 October 2022, i6doc.com publ., ISBN 978287587084-1.
Available from http://www.i6doc.com/en/.

Interactive visual analytics for medical data: application to COVID-19 clinical information during the first wave

José M. Enguita[1], Diego García[1] María D. Chiara[2], Nuria Valdés[3],
Ana González[1], Abel A. Cuadrado[1], and Ignacio Díaz[1] *

1- University of Oviedo - Dept of Electrical Engineering
Edificio Torres Quevedo, módulo 2, Campus de Gijón 33204 - SPAIN
2- Institute of Sanitary Research of the Principado de Asturias
Hospital Universitario Central de Asturias, Oviedo 33011 - SPAIN.
3- Department of Internal Medicine, Section of Endocrinology and Nutrition
Hospital Universitario de Cabueñes, Gijón 33204 - SPAIN.

Abstract. Biomedical data recorded as a result of clinical practice are often multi-domain –involving lab measurements, medication, patient attributes, logistic information–, and also highly unstructured, with high rates of missing data and asynchronously sampled measurements. In this scenario, we need tools capable of providing a broad picture prior to more detailed analyses. We present here a visual analytics approach that uses the morphing projections technique to combine the visualization of a *t*-SNE projection of clinical time series, with views of other clinical or patient's information. The proposed approach is demonstrated on an application case study of COVID-19 clinical information taken during the first wave.

1 Introduction

COVID-19 pandemics has been an unprecedented situation with serious medical, economic and social consequences. Along the crisis, large amounts of clinical data were generated and made public in an effort to acquire valuable knowledge to deal with an unbearable number of hospital admissions, with a significant number of critically ill patients. However, clinical data are in most cases highly unstructured, and this worsened as a result of the emergency of the moment. While after two years of pandemics we have improved our knowledge and tools to face COVID-19, new variants and even other kind of pandemics could emerge, and tools are still needed to enable rapid, efficient and intuitive analysis of COVID-19 behavior in different scenarios, such as response to treatments, clinical conditions and patient groups, as well as decision support and protocol development. These tools should allow for fast and interactive multilevel exploratory analysis –ranging from the "big picture" to detailed and quantitative analysis– and be powered by machine learning algorithms that can reveal hidden patterns in the data and integrate this information in a visual and intuitive way.

*This work is part of Grant PID2020-115401GB-I00 funded by MCIN/AEI/10.13039/501100011033.
Demo video: `https://youtu.be/1gXSFqX3Cms`

(c) Artículo presentado en el *30th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning* [290].

Figura A.7: Estos artículos está relacionados con el uso de herramientas de visualización interactiva para el análisis de datos y, también, con la idea de la transferencia de conocimiento entre los ámbitos de la ingeniería y de la biomedicina, que son aspectos abordados en el Capítulo 6 de esta tesis.

Bibliografía

[1] Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006.

[2] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *nature*, 550(7676):354–359, 2017.

[3] John Tromp. The number of legal go positions. In *International Conference on Computers and Games*, pages 183–190. Springer, 2016.

[4] Murray Campbell, A Joseph Hoane Jr, and Feng-hsiung Hsu. Deep blue. *Artificial intelligence*, 134(1-2):57–83, 2002.

[5] Athanasios Voulodimos, Nikolaos Doulamis, Anastasios Doulamis, and Eftychios Protopapadakis. Deep learning for computer vision: A brief review. *Computational intelligence and neuroscience*, 2018, 2018.

[6] Daniel W Otter, Julian R Medina, and Jugal K Kalita. A survey of the usages of deep learning for natural language processing. *IEEE transactions on neural networks and learning systems*, 32(2):604–624, 2020.

[7] Amitha Mathew, P Amudha, and S Sivakumari. Deep learning techniques: an overview. In *International Conference on Advanced machine learning technologies and applications*, pages 599–608. Springer, 2020.

[8] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

[9] John D Kelleher and Brendan Tierney. *Data science*. MIT Press, 2018.

[10] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[11] Francois Chollet. *Deep learning with Python*. Simon and Schuster, 2021.

[12] Steve Bako, Thijs Vogels, Brian McWilliams, Mark Meyer, Jan Novák, Alex Harvill, Pradeep Sen, Tony Derose, and Fabrice Rousselle. Kernel-predicting convolutional networks for denoising monte carlo renderings. *ACM Trans. Graph.*, 36(4):97–1, 2017.

[13] Andrey Ignatov, Nikolay Kobyshev, Radu Timofte, Kenneth Vanhoey, and Luc Van Gool. Wespe: weakly supervised photo enhancer for digital cameras. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 691–700, 2018.

[14] Mingming He, Dongdong Chen, Jing Liao, Pedro V Sander, and Lu Yuan. Deep exemplar-based colorization. *ACM Transactions on Graphics (TOG)*, 37(4):1–16, 2018.

[15] Manuel Ruder, Alexey Dosovitskiy, and Thomas Brox. Artistic style transfer for videos and spherical images. *International Journal of Computer Vision*, 126(11):1199–1219, 2018.

[16] Xin Wang, Wenhu Chen, Jiawei Wu, Yuan-Fang Wang, and William Yang Wang. Video captioning via hierarchical reinforcement learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4213–4222, 2018.

[17] Ruben Tolosana, Ruben Vera-Rodriguez, Julian Fierrez, Aythami Morales, and Javier Ortega-Garcia. Deepfakes and beyond: A survey of face manipulation and fake detection. *Information Fusion*, 64:131–148, 2020.

[18] Yu-Siang Huang and Yi-Hsuan Yang. Pop music transformer: Beat-based modeling and generation of expressive pop piano compositions. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 1180–1188, 2020.

[19] Andreja Rojko. Industry 4.0 concept: Background and overview. *International Journal of Interactive Mobile Technologies*, 11(5), 2017.

[20] Angelos Angelopoulos, Emmanouel T Michailidis, Nikolaos Nomikos, Panagiotis Trakadas, Antonis Hatziefremidis, Stamatis Voliotis, and Theodore Zahariadis. Tackling faults in the industry 4.0 era—a survey of machine-learning solutions and key aspects. *Sensors*, 20(1):109, 2019.

[21] John D Kelleher. *Deep learning*. MIT press, 2019.

[22] Samir Khan and Takehisa Yairi. A review on the application of deep learning in system health management. *Mechanical Systems and Signal Processing*, 107:241–265, 2018.

[23] Yoshua Bengio, Aaron C Courville, and Pascal Vincent. Unsupervised feature learning and deep learning: A review and new perspectives. *CoRR, abs/1206.5538*, 1(2665):2012, 2012.

[24] Zewen Li, Fan Liu, Wenjie Yang, Shouheng Peng, and Jun Zhou. A survey of convolutional neural networks: analysis, applications, and prospects. *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[25] Finale Doshi-Velez and Been Kim. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*, 2017.

[26] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507, 2006.

[27] Yasi Wang, Hongxun Yao, and Sicheng Zhao. Auto-encoder based dimensionality reduction. *Neurocomputing*, 184:232–242, 2016.

[28] Xianxu Hou, Linlin Shen, Ke Sun, and Guoping Qiu. Deep feature consistent variational autoencoder. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1133–1141. IEEE, 2017.

[29] Jinwon An and Sungzoon Cho. Variational autoencoder based anomaly detection using reconstruction probability. *Special Lecture on IE*, 2(1):1–18, 2015.

[30] Matt J Kusner, Brooks Paige, and José Miguel Hernández-Lobato. Grammar variational autoencoder. In *International conference on machine learning*, pages 1945–1954. PMLR, 2017.

[31] Dong Yul Oh and Il Dong Yun. Residual error based anomaly detection using auto-encoder in smd machine sound. *Sensors*, 18(5):1308, 2018.

[32] Tom White. Sampling generative networks. *arXiv preprint arXiv:1609.04468*, 2016.

[33] Shan Carter and Michael Nielsen. Using artificial intelligence to augment human intelligence. *Distill*, 2(12):e9, 2017.

[34] Ki Hyun Kim, Sangwoo Shim, Yongsub Lim, Jongseob Jeon, Jeongwoo Choi, Byungchan Kim, and Andre S Yoon. Rapp: Novelty detection with reconstruction along projection pathway. In *International Conference on Learning Representations*, 2019.

[35] Andrew J Lew and Markus J Buehler. Encoding and exploring latent design space of optimal material structures via a vae-lstm model. *Forces in Mechanics*, 5:100054, 2021.

[36] Chensi Cao, Feng Liu, Hai Tan, Deshou Song, Wenjie Shu, Weizhong Li, Yiming Zhou, Xiaochen Bo, and Zhi Xie. Deep learning and its applications in biomedicine. *Genomics, proteomics & bioinformatics*, 16(1):17–32, 2018.

[37] Warren S McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4): 115–133, 1943.

[38] Donald Olding Hebb. *The organization of behavior: a neuropsychological theory*. Science editions, 1949.

[39] Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.

[40] B Widrow and ME Hoff. Ire wescon convention record, volume 4, chapter adaptive switching circuits. *IRE, New York*, 1960.

[41] Marvin Minsky and Seymour Papert. An introduction to computational geometry. *Cambridge tiass., HIT*, 479:480, 1969.

[42] Stanley J Farlow. The gmdh algorithm of ivakhnenko. *The American Statistician*, 35(4):210–215, 1981.

[43] John J Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, 79(8):2554–2558, 1982.

[44] Paul Werbos. Beyond regression:"new tools for prediction and analysis in the behavioral sciences. *Ph. D. dissertation, Harvard University*, 1974.

[45] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

[46] Sepp Hochreiter. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 6(02):107–116, 1998.

[47] Geoffrey E Hinton et al. Learning distributed representations of concepts. In *Proceedings of the eighth annual conference of the cognitive science society*, volume 1, page 12. Amherst, MA, 1986.

[48] Kunihiko Fukushima and Sei Miyake. Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition. In *Competition and cooperation in neural nets*, pages 267–285. Springer, 1982.

[49] David H Hubel and Torsten N Wiesel. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of physiology*, 160(1):106, 1962.

[50] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

[51] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[52] Jeffrey L Elman. Finding structure in time. *Cognitive science*, 14(2):179–211, 1990.

[53] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[54] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[55] Yoshua Bengio, Pascal Lamblin, Dan Popovici, and Hugo Larochelle. Greedy layer-wise training of deep networks. *Advances in neural information processing systems*, 19, 2006.

[56] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256. JMLR Workshop and Conference Proceedings, 2010.

[57] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 315–323. JMLR Workshop and Conference Proceedings, 2011.

[58] Andrew L Maas, Awni Y Hannun, Andrew Y Ng, et al. Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml*, volume 30(1), page 3. Citeseer, 2013.

[59] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[60] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*, 2015.

[61] Chigozie Nwankpa, Winifred Ijomah, Anthony Gachagan, and Stephen Marshall. Activation functions: Comparison of trends in practice and research for deep learning. *arXiv preprint arXiv:1811.03378*, 2018.

[62] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[63] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[64] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2414–2423, 2016.

[65] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27, 2014.

[66] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26, 2013.

[67] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057. PMLR, 2015.

[68] Tusty Nadia Maghfira, T Basaruddin, and Adila Krisnadhi. Infant cry classification using cnn–rnn. In *Journal of Physics: Conference Series*, volume 1528(1), page 012019. IOP Publishing, 2020.

[69] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103, 2008.

[70] Adam Roberts, Jesse Engel, and Douglas Eck. Hierarchical variational autoencoders for music. In *NIPS Workshop on Machine Learning for Creativity and Design*, volume 3, 2017.

[71] Rafael Gómez-Bombarelli, Jennifer N Wei, David Duvenaud, José Miguel Hernández-Lobato, Benjamín Sánchez-Lengeling, Dennis Sheberla, Jorge Aguilera-Iparraguirre, Timothy D Hirzel, Ryan P Adams, and Alán Aspuru-Guzik. Automatic chemical design using a data-driven continuous representation of molecules. *ACS central science*, 4(2):268–276, 2018.

[72] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[73] Pavel Korshunov and Sébastien Marcel. Deepfakes: a new threat to face recognition? assessment and detection. *arXiv preprint arXiv:1812.08685*, 2018.

[74] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.

[75] Renlong Hang, Qingshan Liu, Danfeng Hong, and Pedram Ghamisi. Cascaded recurrent neural networks for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 57(8):5384–5394, 2019.

[76] Abdulaziz M Alayba, Vasile Palade, Matthew England, and Rahat Iqbal. A combined cnn and lstm model for arabic sentiment analysis. In *International cross-domain conference for machine learning and knowledge extraction*, pages 179–191. Springer, 2018.

[77] Thanh Thi Nguyen, Quoc Viet Hung Nguyen, Cuong M Nguyen, Dung Nguyen, Duc Thanh Nguyen, and Saeid Nahavandi. Deep learning for deepfakes creation and detection: A survey. *arXiv preprint arXiv:1909.11573*, 2019.

[78] Sagnik Sarkar, Shaashwat Agrawal, Thar Baker, Praveen Kumar Reddy Maddikunta, and Thippa Reddy Gadekallu. Catalysis of neural activation functions: Adaptive feed-forward training for big data applications. *Applied Intelligence*, pages 1–20, 2022.

[79] Luis Perez and Jason Wang. The effectiveness of data augmentation in image classification using deep learning. *arXiv preprint arXiv:1712.04621*, 2017.

[80] Sebastian Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016.

[81] Ossama Abdel-Hamid, Abdel-rahman Mohamed, Hui Jiang, Li Deng, Gerald Penn, and Dong Yu. Convolutional neural networks for speech recognition. *IEEE/ACM Transactions on audio, speech, and language processing*, 22(10): 1533–1545, 2014.

[82] Dennis M Dimiduk, Elizabeth A Holm, and Stephen R Niezgoda. Perspectives on the impact of machine learning, deep learning, and artificial intelligence on materials, processes, and structures engineering. *Integrating Materials and Manufacturing Innovation*, 7(3):157–172, 2018.

[83] Vincent Dumoulin and Francesco Visin. A guide to convolution arithmetic for deep learning. *arXiv preprint arXiv:1603.07285*, 2016.

[84] Y-Lan Boureau, Jean Ponce, and Yann LeCun. A theoretical analysis of feature pooling in visual recognition. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 111–118, 2010.

[85] Hervé Bourlard and Yves Kamp. Auto-association by multilayer perceptrons and singular value decomposition. *Biological cybernetics*, 59(4):291–294, 1988.

[86] Geoffrey E Hinton and Richard Zemel. Autoencoders, minimum description length and helmholtz free energy. *Advances in neural information processing systems*, 6, 1993.

[87] Shenghua Gao, Yuting Zhang, Kui Jia, Jiwen Lu, and Yingying Zhang. Single sample face recognition via learning deep supervised autoencoders. *IEEE transactions on information forensics and security*, 10(10):2108–2118, 2015.

[88] Ayush Tewari, Michael Zollhofer, Hyeongwoo Kim, Pablo Garrido, Florian Bernard, Patrick Perez, and Christian Theobalt. Mofa: Model-based deep convolutional face autoencoder for unsupervised monocular reconstruction. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 1274–1283, 2017.

[89] Fang Zhao, Jiashi Feng, Jian Zhao, Wenhan Yang, and Shuicheng Yan. Robust lstm-autoencoders for face de-occlusion in the wild. *IEEE Transactions on Image Processing*, 27(2):778–790, 2017.

[90] Giuseppina Andresini, Annalisa Appice, Nicola Di Mauro, Corrado Loglisci, and Donato Malerba. Exploiting the auto-encoder residual error for intrusion detection. In *2019 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, pages 281–290. IEEE, 2019.

[91] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[92] Yang Liu, Eunice Jun, Qisheng Li, and Jeffrey Heer. Latent space cartography: Visual analysis of vector space embeddings. In *Computer Graphics Forum*, volume 38(3), pages 67–78. Wiley Online Library, 2019.

[93] Ali Moradi Vartouni, Saeed Sedighian Kashi, and Mohammad Teshnehlab. An anomaly detection method to detect web attacks using stacked auto-encoder. In *2018 6th Iranian Joint Congress on Fuzzy and Intelligent Systems (CFIS)*, pages 131–134. IEEE, 2018.

[94] Mohit Sewak, Sanjay K Sahay, and Hemant Rathore. An overview of deep learning architecture of deep neural networks and autoencoders. *Journal of Computational and Theoretical Nanoscience*, 17(1):182–188, 2020.

[95] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(7), 2011.

[96] Tijmen Tieleman, Geoffrey Hinton, et al. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 4(2):26–31, 2012.

[97] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[98] François Chollet et al. Keras: Deep learning library for theano and tensorflow. *URL: https://keras. io/k*, 7(8):T1, 2015.

[99] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

[100] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015.

[101] Ana González-Muñiz, Ignacio Díaz, and Abel A Cuadrado. Dcnn for condition monitoring and fault detection in rotating machines and its contribution to the understanding of machine nature. *Heliyon*, 6(2):e03395, 2020.

[102] R Keith Mobley. *An introduction to predictive maintenance*. Elsevier, 2002.

[103] Brian P Graney and Ken Starry. Rolling element bearing analysis. *Materials Evaluation*, 70(1):78, 2012.

[104] Michael J Devaney and Levent Eren. Detecting motor bearing faults. *IEEE Instrumentation & Measurement Magazine*, 7(4):30–50, 2004.

[105] Steve J Lacey. An overview of bearing vibration analysis. *Maintenance & asset management*, 23(6):32–42, 2008.

[106] Ingo Mierswa and Katharina Morik. Automatic feature extraction for classifying audio data. *Machine learning*, 58(2):127–149, 2005.

[107] Pankaj Gupta and MK Pradhan. Fault detection analysis in rolling element bearing: A review. *Materials Today: Proceedings*, 4(2):2085–2094, 2017.

[108] RBW Heng and Mohd Jailani Mohd Nor. Statistical analysis of sound and vibration signals for monitoring rolling element bearing condition. *Applied Acoustics*, 53(1-3):211–226, 1998.

[109] Subhasis Nandi and Hamid A Toliyat. Condition monitoring and fault diagnosis of electrical machines-a review. In *Conference Record of the 1999 IEEE Industry Applications Conference. Thirty-Forth IAS Annual Meeting (Cat. No. 99CH36370)*, volume 1, pages 197–204. IEEE, 1999.

[110] Sanna Poyhonen, Pedro Jover, and Heikki Hyotyniemi. Signal processing of vibrations for condition monitoring of an induction motor. In *First International Symposium on Control, Communications and Signal Processing, 2004.*, pages 499–502. IEEE, 2004.

[111] Peter J Tavner. Review of condition monitoring of rotating electrical machines. *IET electric power applications*, 2(4):215–247, 2008.

[112] Fadi Al-Badour, Mehmet Sunar, and Lahouari Cheded. Vibration analysis of rotating machinery using time–frequency analysis and wavelet techniques. *Mechanical Systems and Signal Processing*, 25(6):2083–2101, 2011.

[113] Robert B Randall and Jerome Antoni. Rolling element bearing diagnostics—a tutorial. *Mechanical systems and signal processing*, 25(2):485–520, 2011.

[114] Czeslaw T Kowalski and Teresa Orlowska-Kowalska. Neural networks application for induction motor faults diagnosis. *Mathematics and computers in simulation*, 63(3-5):435–448, 2003.

[115] M Saimurugan, KI Ramachandran, V Sugumaran, and NR Sakthivel. Multi component fault diagnosis of rotational mechanical system based on decision tree and support vector machine. *Expert Systems with Applications*, 38 (4):3819–3826, 2011.

[116] Bo-Suk Yang, Xiao Di, and Tian Han. Random forests classifier for machine fault diagnosis. *Journal of mechanical science and technology*, 22(9):1716–1725, 2008.

[117] Wade A Smith and Robert B Randall. Rolling element bearing diagnostics using the case western reserve university data: A benchmark study. *Mechanical systems and signal processing*, 64:100–131, 2015.

[118] Pavle Boškoski, Janko Petrovčič, Bojan Musizza, and Đani Juričić. Detection of lubrication starved bearings in electrical motors by means of vibration analysis. *Tribology international*, 43(9):1683–1692, 2010.

[119] Ruqiang Yan, Robert X Gao, and Xuefeng Chen. Wavelets for fault diagnosis of rotary machines: A review with applications. *Signal processing*, 96:1–15, 2014.

[120] P Konar and P Chattopadhyay. Bearing fault detection of induction motor using wavelet and support vector machines (svms). *Applied Soft Computing*, 11(6):4203–4211, 2011.

[121] Bing Li, Pei-lin Zhang, Dong-sheng Liu, Shuang-shan Mi, Guo-quan Ren, and Hao Tian. Feature extraction for rolling element bearing fault diagnosis utilizing generalized s transform and two-dimensional non-negative matrix factorization. *Journal of Sound and Vibration*, 330(10):2388–2399, 2011.

[122] Xiaofeng Liu, Lin Ma, and Joseph Mathew. Machinery fault diagnosis based on fuzzy measure and fuzzy integral data fusion techniques. *Mechanical Systems and Signal Processing*, 23(3):690–700, 2009.

[123] Pavan Kumar Kankar, Satish C Sharma, and Suraj Prakash Harsha. Fault diagnosis of ball bearings using machine learning methods. *Expert Systems with applications*, 38(3):1876–1886, 2011.

[124] Michael E Tipping and Christopher M Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61(3):611–622, 1999.

[125] Bruno A Olshausen and David J Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381 (6583):607–609, 1996.

[126] Radford M Neal. Connectionist learning of belief networks. *Artificial intelligence*, 56(1):71–113, 1992.

[127] Bo Li, M-Y Chow, Yodyium Tipsuwan, and James C Hung. Neural-network-based motor rolling bearing fault diagnosis. *IEEE transactions on industrial electronics*, 47(5):1060–1069, 2000.

[128] GF Bin, JJ Gao, XJ Li, and BS Dhillon. Early fault diagnosis of rotating machinery based on wavelet packets—empirical mode decomposition feature extraction and neural network. *Mechanical Systems and Signal Processing*, 27:696–711, 2012.

[129] Faisal AlThobiani, Andrew Ball, et al. An approach to fault diagnosis of reciprocating compressor valves using teager–kaiser energy operator and deep belief networks. *Expert Systems with Applications*, 41(9):4113–4122, 2014.

[130] Feng Jia, Yaguo Lei, Jing Lin, Xin Zhou, and Na Lu. Deep neural networks: A promising tool for fault characteristic mining and intelligent diagnosis of rotating machinery with massive data. *Mechanical systems and signal processing*, 72:303–315, 2016.

[131] Meng Gan, Cong Wang, et al. Construction of hierarchical diagnosis network based on deep learning and its application in the fault pattern recognition of rolling element bearings. *Mechanical Systems and Signal Processing*, 72:92–104, 2016.

[132] Yann LeCun, Yoshua Bengio, et al. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10):1995, 1995.

[133] Tara N Sainath, Brian Kingsbury, Abdel-rahman Mohamed, George E Dahl, George Saon, Hagen Soltau, Tomas Beran, Aleksandr Y Aravkin, and Bhuvana Ramabhadran. Improvements to deep convolutional neural networks for lvcsr. In *2013 IEEE workshop on automatic speech recognition and understanding*, pages 315–320. IEEE, 2013.

[134] Dan Claudiu Cireşan, Ueli Meier, Luca Maria Gambardella, and Jürgen Schmidhuber. Deep, big, simple neural nets for handwritten digit recognition. *Neural computation*, 22(12):3207–3220, 2010.

[135] Rui Zhao, Ruqiang Yan, Zhenghua Chen, Kezhi Mao, Peng Wang, and Robert X Gao. Deep learning and its applications to machine health monitoring. *Mechanical Systems and Signal Processing*, 115:213–237, 2019.

[136] Feng Jia, Yaguo Lei, Na Lu, and Saibo Xing. Deep normalized convolutional neural network for imbalanced fault classification of machinery and its understanding via visualization. *Mechanical Systems and Signal Processing*, 110:349–367, 2018.

[137] Antonio Criminisi, Jamie Shotton, Ender Konukoglu, et al. Decision forests: A unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning. *Foundations and trends® in computer graphics and vision*, 7(2–3):81–227, 2012.

[138] Chih-Wei Hsu and Chih-Jen Lin. A comparison of methods for multiclass support vector machines. *IEEE transactions on Neural Networks*, 13(2):415–425, 2002.

[139] Ignacio Díaz Blanco, Abel Alberto Cuadrado Vega, Ana González Muñiz, Diego García Pérez, et al. Dataicann: datos de vibración y corriente de un motor de inducción. 2019.

[140] Josh Patterson and Adam Gibson. *Deep learning: A practitioner's approach*. .ºReilly Media, Inc.", 2017.

[141] Eric Bechhoefer. Condition based maintenance fault database for testing diagnostics and prognostic algorithms. *MFPT Data*, 2013.

[142] Ana González-Muñiz, Ignacio Díaz, Abel A Cuadrado, Diego García-Pérez, and Daniel Pérez. Two-step residual-error based approach for anomaly detection in engineering systems using variational autoencoders. *Computers and Electrical Engineering*, 101:108065, 2022.

[143] Khaled Alrawashdeh and Carla Purdy. Toward an online anomaly intrusion detection system based on deep learning. In *2016 15th IEEE international conference on machine learning and applications (ICMLA)*, pages 195–200. IEEE, 2016.

[144] Mohiuddin Ahmed, Abdun Naser Mahmood, and Md Rafiqul Islam. A survey of anomaly detection techniques in financial domain. *Future Generation Computer Systems*, 55:278–288, 2016.

[145] Thomas Schlegl, Philipp Seeböck, Sebastian M Waldstein, Ursula Schmidt-Erfurth, and Georg Langs. Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In *International conference on information processing in medical imaging*, pages 146–157. Springer, 2017.

[146] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):1–58, 2009.

[147] Marco AF Pimentel, David A Clifton, Lei Clifton, and Lionel Tarassenko. A review of novelty detection. *Signal processing*, 99:215–249, 2014.

[148] Rolf Isermann. Model-based fault-detection and diagnosis–status and applications. *Annual Reviews in control*, 29(1):71–85, 2005.

[149] A Balasubramanian and Ranganath Muthu. Model based fault detection and diagnosis of doubly fed induction generators–a review. *Energy Procedia*, 117:935–942, 2017.

[150] Moazzam Nazir, Abdul Qayyum Khan, Ghulam Mustafa, and Muhammad Abid. Robust fault detection for wind turbines using reference model-based approach. *Journal of King Saud University-Engineering Sciences*, 29(3):244–252, 2017.

[151] Raghavendra Chalapathy and Sanjay Chawla. Deep learning for anomaly detection: A survey. *arXiv preprint arXiv:1901.03407*, 2019.

[152] Marco Martinelli, Enrico Tronci, Giovanni Dipoppa, and Claudio Balducelli. Electric power system anomaly detection using neural networks. In *International conference on knowledge-based and intelligent information and engineering systems*, pages 1242–1248. Springer, 2004.

[153] Mayu Sakurada and Takehisa Yairi. Anomaly detection using autoencoders with nonlinear dimensionality reduction. In *Proceedings of the MLSDA 2014 2nd workshop on machine learning for sensory data analysis*, pages 4–11, 2014.

[154] Hoang Anh Dau, Vic Ciesielski, and Andy Song. Anomaly detection using replicator neural networks trained on examples of one class. In *Asia-Pacific Conference on Simulated Evolution and Learning*, pages 311–322. Springer, 2014.

[155] Chong Zhou and Randy C Paffenroth. Anomaly detection with robust deep autoencoders. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 665–674, 2017.

[156] Yves Nsoga Nguimbous, Riadh Ksantini, and Adel Bouhoula. Anomaly-based intrusion detection using auto-encoder. In *2019 International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, pages 1–5. IEEE, 2019.

[157] Fouzi Harrou, Abdelkader Dairi, Bilal Taghezouit, and Ying Sun. An unsupervised monitoring procedure for detecting anomalies in photovoltaic systems using a one-class support vector machine. *Solar Energy*, 179:48–58, 2019.

[158] Jeongsu Lee, Young Chul Lee, and Jeong Tae Kim. Fault detection based on one-class deep learning for manufacturing applications limited to an imbalanced database. *Journal of Manufacturing Systems*, 57:357–366, 2020.

[159] Adam Roberts, Jesse Engel, Colin Raffel, Curtis Hawthorne, and Douglas Eck. A hierarchical latent vector model for learning long-term structure in music. In *International conference on machine learning*, pages 4364–4373. PMLR, 2018.

[160] Yunchen Pu, Zhe Gan, Ricardo Henao, Xin Yuan, Chunyuan Li, Andrew Stevens, and Lawrence Carin. Variational autoencoder for deep learning of images, labels and captions. *Advances in neural information processing systems*, 29, 2016.

[161] Haowen Xu, Wenxiao Chen, Nengwen Zhao, Zeyan Li, Jiahao Bu, Zhihan Li, Ying Liu, Youjian Zhao, Dan Pei, Yang Feng, et al. Unsupervised anomaly detection via variational auto-encoder for seasonal kpis in web applications. In *Proceedings of the 2018 world wide web conference*, pages 187–196, 2018.

[162] Yaxiang Fan, Gongjian Wen, Deren Li, Shaohua Qiu, Martin D Levine, and Fei Xiao. Video anomaly detection and localization via gaussian mixture fully convolutional variational autoencoder. *Computer Vision and Image Understanding*, 195:102920, 2020.

[163] Yuda Gao, Bin Shi, Bo Dong, Yan Chen, Lingyun Mi, Zhiping Huang, and Yuanyuan Shi. Rvae-abfa: robust anomaly detection for highdimensional data using variational autoencoder. In *2020 IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC)*, pages 334–339. IEEE, 2020.

[164] Milad Memarzadeh, Bryan Matthews, and Ilya Avrekh. Unsupervised anomaly detection in flight data using convolutional variational auto-encoder. *Aerospace*, 7(8):115, 2020.

[165] Jiaxin Zhou and Takashi Komuro. Recognizing fall actions from videos using reconstruction error of variational autoencoder. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 3372–3376. IEEE, 2019.

[166] Peng Luo, Buhong Wang, Tengyao Li, and Jiwei Tian. Ads-b anomaly data detection model based on vae-svdd. *Computers & Security*, 104:102213, 2021.

[167] Gavneet Singh Chadha, Arfyan Rabbani, and Andreas Schwung. Comparison of semi-supervised deep neural networks for anomaly detection in industrial processes. In *2019 IEEE 17th International Conference on Industrial Informatics (INDIN)*, volume 1, pages 214–219. IEEE, 2019.

[168] Shuyu Lin, Ronald Clark, Robert Birke, Sandro Schönborn, Niki Trigoni, and Stephen Roberts. Anomaly detection for time series using vae-lstm hybrid model. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4322–4326. Ieee, 2020.

[169] Ignacio Diaz and Jaakko Hollmen. Residual generation and visualization for understanding novel process conditions. In *Proceedings of the 2002 International Joint Conference on Neural Networks. IJCNN'02 (Cat. No. 02CH37290)*, volume 3, pages 2070–2075. IEEE, 2002.

[170] Nikolai Helwig, Eliseo Pignanelli, and Andreas Schütze. Condition monitoring of a complex hydraulic system using multivariate statistics. In *2015 IEEE International Instrumentation and Measurement Technology Conference (I2MTC) Proceedings*, pages 210–215. IEEE, 2015.

[171] Oresti Banos, Rafael Garcia, Juan A Holgado-Terriza, Miguel Damas, Hector Pomares, Ignacio Rojas, Alejandro Saez, and Claudia Villalonga. mhealthdroid: a novel framework for agile development of mobile health applications. In *International workshop on ambient assisted living*, pages 91–98. Springer, 2014.

[172] Paweł Daniluk, M Goździewski, Sławomir Kapka, and M Kośmider. Ensemble of auto-encoder based and wavenet like systems for unsupervised anomaly detection. *Challenge on Detection and Classification of Acoustic Scenes and Events (DCASE 2020 Challenge), Tech. Rep*, 2020.

[173] Davíð Steinar Ásgrímsson, Ignacio González, Giampiero Salvi, and Raid Karoumi. Bayesian deep learning for vibration-based bridge damage detection. In *Structural Health Monitoring Based on Data Science Techniques*, pages 27–43. Springer, 2022.

[174] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[175] Ana González-Muñiz, Ignacio Díaz, Abel A Cuadrado, and Diego García-Pérez. Health indicator for machine condition monitoring built in the latent space of a deep autoencoder. *Reliability Engineering & System Safety*, 224: 108482, 2022.

[176] Luyang Jing, Ming Zhao, Pin Li, and Xiaoqiang Xu. A convolutional neural network based feature learning and fault diagnosis method for the condition monitoring of gearbox. *Measurement*, 111:1–10, 2017.

[177] Adrian Stetco, Fateme Dinmohammadi, Xingyu Zhao, Valentin Robu, David Flynn, Mike Barnes, John Keane, and Goran Nenadic. Machine learning methods for wind turbine condition monitoring: A review. *Renewable energy*, 133:620–635, 2019.

[178] James Wakiru, Liliane Pintelon, Peter N Muchiri, Peter K Chemweno, and Stanley Mburu. Towards an innovative lubricant condition monitoring strategy for maintenance of ageing multi-unit systems. *Reliability Engineering & System Safety*, 204:107200, 2020.

[179] Paul Phillips and Dominic Diston. A knowledge driven approach to aerospace condition monitoring. *Knowledge-Based Systems*, 24(6):915–927, 2011.

[180] Dawn An, Nam H Kim, and Joo-Ho Choi. Practical options for selecting data-driven or physics-based prognostics algorithms with reviews. *Reliability Engineering & System Safety*, 133:223–236, 2015.

[181] Yaguo Lei, Naipeng Li, Liang Guo, Ningbo Li, Tao Yan, and Jing Lin. Machinery health prognostics: A systematic review from data acquisition to rul prediction. *Mechanical systems and signal processing*, 104:799–834, 2018.

[182] Kaixiang Peng, Ruihua Jiao, Jie Dong, and Yanting Pi. A deep belief network based health indicator construction and remaining useful life prediction using improved particle filter. *Neurocomputing*, 361:19–28, 2019.

[183] Wasim Ahmad, Sheraz Ali Khan, MM Manjurul Islam, and Jong-Myon Kim. A reliable technique for remaining useful life estimation of rolling element bearings using dynamic regression models. *Reliability Engineering & System Safety*, 184:67–76, 2019.

[184] Yapeng Zhou, Miaohua Huang, Yupu Chen, and Ye Tao. A novel health indicator for on-line lithium-ion batteries remaining useful life prediction. *Journal of Power Sources*, 321:1–10, 2016.

[185] Wennian Yu, II Yong Kim, and Chris Mechefske. An improved similarity-based prognostic algorithm for rul estimation using an rnn autoencoder scheme. *Reliability Engineering & System Safety*, 199:106926, 2020.

[186] Pengfei Wen, Shuai Zhao, Shaowei Chen, and Yong Li. A generalized remaining useful life prediction method for complex systems based on composite health indicator. *Reliability Engineering & System Safety*, 205:107241, 2021.

[187] Liang Guo, Naipeng Li, Feng Jia, Yaguo Lei, and Jing Lin. A recurrent neural network based health indicator for remaining useful life prediction of bearings. *Neurocomputing*, 240:98–109, 2017.

[188] Lei Xiao, Junxuan Tang, Xinghui Zhang, Eric Bechhoefer, and Siyi Ding. Remaining useful life prediction based on intentional noise injection and feature reconstruction. *Reliability Engineering & System Safety*, 215:107871, 2021.

[189] Maryam M Najafabadi, Flavio Villanustre, Taghi M Khoshgoftaar, Naeem Seliya, Randall Wald, and Edin Muharemagic. Deep learning applications and challenges in big data analytics. *Journal of big data*, 2(1):1–21, 2015.

[190] Ali Bou Nassif, Ismail Shahin, Imtinan Attili, Mohammad Azzeh, and Khaled Shaalan. Speech recognition using deep neural networks: A systematic review. *IEEE access*, 7:19143–19165, 2019.

[191] Geert Litjens, Thijs Kooi, Babak Ehteshami Bejnordi, Arnaud Arindra Adiyoso Setio, Francesco Ciompi, Mohsen Ghafoorian, Jeroen Awm Van Der Laak, Bram Van Ginneken, and Clara I Sánchez. A survey on deep learning in medical image analysis. *Medical image analysis*, 42:60–88, 2017.

[192] Liang Guo, Yaguo Lei, Naipeng Li, and Saibo Xing. Deep convolution feature learning for health indicator construction of bearings. In *2017 Prognostics and System Health Management Conference (PHM-Harbin)*, pages 1–6. IEEE, 2017.

[193] Longting Chen, Guanghua Xu, Sicong Zhang, Wenqiang Yan, and Qingqiang Wu. Health indicator construction of machinery based on end-to-end trainable convolution recurrent neural networks. *Journal of Manufacturing Systems*, 54:1–11, 2020.

[194] Wei Zhang, Xiang Li, Hui Ma, Zhong Luo, and Xu Li. Transfer learning using deep representation regularization in remaining useful life prediction across operating conditions. *Reliability Engineering & System Safety*, 211: 107556, 2021.

[195] Liang Guo, Yaguo Lei, Naipeng Li, Tao Yan, and Ningbo Li. Machinery health indicator construction based on convolutional neural networks considering trend burr. *Neurocomputing*, 292:142–150, 2018.

[196] Xiang Li, Qian Ding, and Jian-Qiao Sun. Remaining useful life estimation in prognostics using deep convolution neural networks. *Reliability Engineering & System Safety*, 172:1–11, 2018.

[197] Sheng Xiang, Yi Qin, Jun Luo, Huayan Pu, and Baoping Tang. Multicellular lstm-based deep learning model for aero-engine remaining useful life prediction. *Reliability Engineering & System Safety*, 216:107927, 2021.

[198] Wihan Booyse, Daniel N Wilke, and Stephan Heyns. Deep digital twins for detection, diagnostics and prognostics. *Mechanical Systems and Signal Processing*, 140:106612, 2020.

[199] Fan Xu, Zhelin Huang, Fangfang Yang, Dong Wang, and Kwok Leung Tsui. Constructing a health indicator for roller bearings by using a stacked autoencoder with an exponential function to eliminate concussion. *Applied Soft Computing*, 89:106119, 2020.

[200] Dingliang Chen, Yi Qin, Yi Wang, and Jianghong Zhou. Health indicator construction by quadratic function-based deep convolutional auto-encoder and its application into bearing rul prediction. *ISA transactions*, 114:44–56, 2021.

[201] Pankaj Malhotra, Vishnu Tv, Anusha Ramakrishnan, Gaurangi Anand, Lovekesh Vig, Puneet Agarwal, and Gautam Shroff. Multi-sensor prognostics using an unsupervised health index based on lstm encoder-decoder. *arXiv preprint arXiv:1608.06154*, 2016.

[202] Seung Yeop Shin and Han-joon Kim. Extended autoencoder for novelty detection with reconstruction along projection pathway. *Applied Sciences*, 10(13):4497, 2020.

[203] Gabriel Michau and Olga Fink. Domain adaptation for one-class classification: monitoring the health of critical systems under limited information. *arXiv preprint arXiv:1907.09204*, 2019.

[204] Gabriel Michau, Yang Hu, Thomas Palmé, and Olga Fink. Feature learning for fault detection in high-dimensional condition monitoring signals. *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*, 234(1):104–115, 2020.

[205] Abhinav Saxena, Kai Goebel, Don Simon, and Neil Eklund. Damage propagation modeling for aircraft engine run-to-failure simulation. In *2008 international conference on prognostics and health management*, pages 1–9. IEEE, 2008.

[206] A Agogino and K Goebel. Mill data set. best lab, uc berkeley. nasa ames prognostics data repository. 2007.

[207] Zhe Yang, Sameer Al-Dahidi, Piero Baraldi, Enrico Zio, and Lorenzo Montelatici. A novel concept drift detection method for incremental learning in nonstationary environments. *IEEE transactions on neural networks and learning systems*, 31(1):309–320, 2019.

[208] Geoffrey E Hinton, Peter Dayan, and Michael Revow. Modeling the manifolds of images of handwritten digits. *IEEE transactions on Neural Networks*, 8(1):65–74, 1997.

[209] H Sebastian Seung and Daniel D Lee. The manifold ways of perception. *science*, 290(5500):2268–2269, 2000.

[210] Charles Fefferman, Sanjoy Mitter, and Hariharan Narayanan. Testing the manifold hypothesis. *Journal of the American Mathematical Society*, 29(4): 983–1049, 2016.

[211] Mahmood Yousefi-Azar, Vijay Varadharajan, Len Hamey, and Uday Tupakula. Autoencoder-based feature learning for cyber security applications. In *2017 International joint conference on neural networks (IJCNN)*, pages 3854–3861. IEEE, 2017.

[212] Stefan Haufe, Sven Dähne, and Vadim V Nikulin. Dimensionality reduction for the analysis of brain oscillations. *NeuroImage*, 101:583–597, 2014.

[213] Martin Becker, Jens Lippel, André Stuhlsatz, and Thomas Zielke. Robust dimensionality reduction for data visualization with deep neural networks. *Graphical Models*, 108:101060, 2020.

[214] Jamie Baalis Coble. Merging data sources to predict remaining useful life–an automated method to identify prognostic parameters. 2010.

[215] Hongyu Yang, Joseph Mathew, and Lin Ma. Vibration feature extraction techniques for fault diagnosis of rotating machinery: a literature survey. In *Asia-pacific vibration conference*, number 42460, pages 801–807, 2003.

[216] Pangun Park, Mingyu Jung, and Piergiuseppe Di Marco. Remaining useful life estimation of bearings using data-driven ridge regression. *Applied Sciences*, 10(24):8977, 2020.

[217] Racha Khelif, Brigitte Chebel-Morello, Simon Malinowski, Emna Laajili, Farhat Fnaiech, and Noureddine Zerhouni. Direct remaining useful life estimation based on support vector regression. *IEEE Transactions on industrial electronics*, 64(3):2276–2285, 2016.

[218] Thomas Laloix, Benoit Iung, Alexandre Voisin, and Eric Romagne. Parameter identification of health indicator aggregation for decision-making in predictive maintenance: Application to machine tool. *CIRP Annals*, 68(1): 483–486, 2019.

[219] Zeqi Zhao, Bin Liang, Xueqian Wang, and Weining Lu. Remaining useful life prediction of aircraft engine based on degradation pattern learning. *Reliability Engineering & System Safety*, 164:74–83, 2017.

[220] Wenkai Hu, Ahmad W Al-Dabbagh, Tongwen Chen, and Sirish L Shah. Design of visualization plots of industrial alarm and event data for enhanced alarm management. *Control Engineering Practice*, 79:50–64, 2018.

[221] Mark Joswiak, You Peng, Ivan Castillo, and Leo H Chiang. Dimensionality reduction for visualizing industrial chemical process data. *Control Engineering Practice*, 93:104189, 2019.

[222] Laurens Van Der Maaten, Eric Postma, Jaap Van den Herik, et al. Dimensionality reduction: a comparative. *J Mach Learn Res*, 10(66-71):13, 2009.

[223] Shaeela Ayesha, Muhammad Kashif Hanif, and Ramzan Talib. Overview and comparative study of dimensionality reduction techniques for high dimensional data. *Information Fusion*, 59:44–58, 2020.

[224] Martin Steiger, Jürgen Bernard, Sebastian Mittelstädt, Hendrik Lücke-Tieke, Daniel Keim, Thorsten May, and Jörn Kohlhammer. Visual analysis of time-series similarities for anomaly detection in sensor networks. In *Computer graphics forum*, volume 33(3), pages 401–410. Wiley Online Library, 2014.

[225] Jiazhi Xia, Yuchen Zhang, Jie Song, Yang Chen, Yunhai Wang, and Shixia Liu. Revisiting dimensionality reduction techniques for visual cluster analysis: An empirical study. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):529–539, 2021.

[226] Daniel Pérez, Serafín Alonso, Antonio Morán, Miguel A Prada, Juan José Fuertes, and Manuel Domínguez. Comparison of network intrusion detection performance using feature representation. In *International Conference on Engineering Applications of Neural Networks*, pages 463–475. Springer, 2019.

[227] Manuel Domínguez, Serafín Alonso, Antonio Morán, Miguel A Prada, and Juan J Fuertes. Dimensionality reduction techniques to analyze heating systems in buildings. *Information Sciences*, 294:553–564, 2015.

[228] Antonio Moran, Juan J Fuertes, Miguel A Prada, Serafín Alonso, Pablo Barrientos, Ignacio Díaz, and Manuel Domínguez. Analysis of electricity consumption profiles in public buildings with dimensionality reduction techniques. *Engineering Applications of Artificial Intelligence*, 26(8):1872–1880, 2013.

[229] Satyaki Bhattacharjee and Karel Matouš. A nonlinear manifold-based reduced order model for multiscale analysis of heterogeneous hyperelastic materials. *Journal of Computational Physics*, 313:635–653, 2016.

[230] Yue Liu, Tianlu Zhao, Wangwei Ju, and Siqi Shi. Materials discovery and design using machine learning. *Journal of Materiomics*, 3(3):159–177, 2017.

[231] Chuanfang Zhang, Kaixiang Peng, and Jie Dong. A pt-sne and mmempm based quality-related process monitoring method for a variety of hot rolling processes. *Control Engineering Practice*, 89:1–11, 2019.

[232] Tao Zeng, Caizhi Zhang, Zhiyu Huang, Hao Liu, Siew Hwa Chan, Jingrui Chen, Ruju Li, and Anjian Zhou. Fast identification of power change rate of pem fuel cell based on data dimensionality reduction approach. *International Journal of Hydrogen Energy*, 44(38):21101–21109, 2019.

[233] NR Sakthivel, Binoy B Nair, M Elangovan, V Sugumaran, and S Saravanmurugan. Comparison of dimensionality reduction techniques for the fault diagnosis of mono block centrifugal pump using vibration signals. *Engineering Science and Technology, an International Journal*, 17(1):30–38, 2014.

[234] Sam T Roweis and Lawrence K Saul. Nonlinear dimensionality reduction by locally linear embedding. *science*, 290(5500):2323–2326, 2000.

[235] Harold Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of educational psychology*, 24(6):417, 1933.

[236] Karl Pearson. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin philosophical magazine and journal of science*, 2(11):559–572, 1901.

[237] Zhiqiang Ge, Zhihuan Song, Steven X Ding, and Biao Huang. Data mining and analytics in the process industry: The role of machine learning. *Ieee Access*, 5:20590–20616, 2017.

[238] Parisa Hajibabaee, Farhad Pourkamali-Anaraki, and Mohammad Amin Hariri-Ardebili. An empirical evaluation of the t-sne algorithm for data visualization in structural engineering. In *2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 1674–1680. IEEE, 2021.

[239] Ruixue Jia, Jing Wang, and Jinglin Zhou. Fault diagnosis of industrial process based on the optimal parametric t-distributed stochastic neighbor embedding. *Science China Information Sciences*, 64(5):1–3, 2021.

[240] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

[241] Etienne Becht, Leland McInnes, John Healy, Charles-Antoine Dutertre, Immanuel WH Kwok, Lai Guan Ng, Florent Ginhoux, and Evan W Newell. Dimensionality reduction for visualizing single-cell data using umap. *Nature biotechnology*, 37(1):38–44, 2019.

[242] Hamid Gadirov, Gleb Tkachev, Thomas Ertl, and Steffen Frey. Evaluation and selection of autoencoders for expressive dimensionality reduction of spatial ensembles. In *International Symposium on Visual Computing*, pages 222–234. Springer, 2021.

[243] Pratik Prabhanjan Brahma, Dapeng Wu, and Yiyuan She. Why deep learning works: A manifold disentanglement perspective. *IEEE transactions on neural networks and learning systems*, 27(10):1997–2008, 2015.

[244] Guido F Montufar, Razvan Pascanu, Kyunghyun Cho, and Yoshua Bengio. On the number of linear regions of deep neural networks. *Advances in neural information processing systems*, 27, 2014.

[245] CJ Battey, Gabrielle C Coffing, and Andrew D Kern. Visualizing population structure with variational autoencoders. *G3*, 11(1):jkaa036, 2021.

[246] Mohammed Ali, Rita Borgo, and Mark W Jones. Concurrent time-series selections using deep learning and dimension reduction. *Knowledge-Based Systems*, 233:107507, 2021.

[247] Feng Gao, Wei Zhang, Andrea A Baccarelli, and Yike Shen. Predicting chemical ecotoxicity by learning latent space chemical representations. *Environment International*, 163:107224, 2022.

[248] Mohammed Ali, Mark W Jones, Xianghua Xie, and Mark Williams. Timecluster: dimension reduction applied to temporal data for visual analytics. *The Visual Computer*, 35(6):1013–1026, 2019.

[249] Yongjian Wang, Zhenyuan Yu, and Zhe Wang. A temporal clustering method fusing deep convolutional autoencoders and dimensionality reduction methods and its application in air quality visualization. *Chemometrics and Intelligent Laboratory Systems*, 227:104607, 2022.

[250] Alex Morehead, Watchanan Chantapakul, and Jianlin Cheng. Semi-supervised graph learning meets dimensionality reduction. *arXiv preprint arXiv:2203.12522*, 2022.

[251] Dominik Sacha, Leishi Zhang, Michael Sedlmair, John A Lee, Jaakko Peltonen, Daniel Weiskopf, Stephen C North, and Daniel A Keim. Visual interaction with dimensionality reduction: A structured literature analysis. *IEEE transactions on visualization and computer graphics*, 23(1):241–250, 2016.

[252] Alex Endert, William Ribarsky, Cagatay Turkay, BL William Wong, Ian Nabney, I Díaz Blanco, and Fabrice Rossi. The state of the art in integrating machine learning into visual analytics. In *Computer Graphics Forum*, volume 36(8), pages 458–486. Wiley Online Library, 2017.

[253] Ignacio Díaz, José M Enguita, Ana González, Diego García, Abel A Cuadrado, María D Chiara, and Nuria Valdés. Morphing projections: a new visual technique for fast and interactive large-scale analysis of biomedical datasets. *Bioinformatics*, 37(11):1571–1580, 2021.

[254] Jarke J Van Wijk. The value of visualization. In *VIS 05. IEEE Visualization, 2005.*, pages 79–86. IEEE, 2005.

[255] Ahmed Aleshinloye, Muhammad Asif Manzoor, and Abdul Bais. Evaluation of dimensionality reduction techniques for load profiling application in smart grid environment. *IEEE Canadian Journal of Electrical and Computer Engineering*, 44(1):41–49, 2021.

[256] Halldór Janetzko, Florian Stoffel, Sebastian Mittelstädt, and Daniel A Keim. Anomaly detection for visual analytics of power consumption data. *Computers & Graphics*, 38:27–37, 2014.

[257] Serafín Alonso, Miguel A Prada, Juan J Fuertes, Ignacio Díaz, and Manuel Domínguez. Analysis of electricity bill data using interactive dimensionality reduction. In *Proceedings of the 16th International Conference on Engineering Applications of Neural Networks (INNS)*, pages 1–7, 2015.

[258] Damien Fay, John V Ringwood, Marissa Condon, and Michael Kelly. 24-h electrical load data—a sequential or partitioned time series? *Neurocomputing*, 55(3-4):469–498, 2003.

[259] Veronika Te Boekhorst, Luigi Preziosi, and Peter Friedl. Plasticity of cell migration in vivo and in silico. *Annu Rev Cell Dev Biol*, 32(1):491–526, 2016.

[260] Christina H Stuelten, Carole A Parent, and Denise J Montell. Cell motility in cancer invasion and metastasis: insights from simple model organisms. *Nature Reviews Cancer*, 18(5):296–312, 2018.

[261] Michael Wainberg, Daniele Merico, Andrew Delong, and Brendan J Frey. Deep learning in biomedicine. *Nature biotechnology*, 36(9):829–838, 2018.

[262] A Mencattini, D Di Giuseppe, MC Comes, P Casti, F Corsi, FR Bertani, L Ghibelli, L Businaro, C Di Natale, MC Parrini, et al. Discovering the hidden messages within cell trajectories using a deep learning approach for in vitro evaluation of cancer drug treatments. *Scientific reports*, 10(1):1–11, 2020.

[263] Jacob C Kimmel, Andrew S Brack, and Wallace F Marshall. Deep convolutional and recurrent neural networks for cell motility discrimination and prediction. *IEEE/ACM transactions on Computational Biology and Bioinformatics*, 18(2):562–574, 2019.

[264] W Marston Linehan, Laura S Schmidt, Daniel R Crooks, Darmood Wei, Ramaprasad Srinivasan, Martin Lang, and Christopher J Ricketts. The metabolic basis of kidney cancerthe metabolic basis of kidney cancer. *Cancer Discovery*, 9(8):1006–1021, 2019.

[265] Cissy Yong, Grant D Stewart, and Christian Frezza. Oncometabolites in renal cancer. *Nature Reviews Nephrology*, 16(3):156–172, 2020.

[266] Kristine M Cornejo, Min Lu, Ping Yang, Shulin Wu, Chao Cai, Aria Olumi, Robert H Young, Chin-Lee Wu, et al. Succinate dehydrogenase b: a new prognostic biomarker in clear cell renal cell carcinoma. *Human pathology*, 46(6):820–826, 2015.

[267] Jing Yang, Yi Zhou, Yanchun Li, Wanye Hu, Chen Yuan, Shida Chen, Gaoqi Ye, Yuzhou Chen, Yunyi Wu, Jing Liu, et al. Functional deficiency of succinate dehydrogenase promotes tumorigenesis and development of clear cell renal cell carcinoma through weakening of ferroptosis. *Bioengineered*, 13 (4):11187–11207, 2022.

[268] Zhiyu Fang, Qiang Sun, Huihui Yang, and Junfang Zheng. Sdhb suppresses the tumorigenesis and development of ccrcc by inhibiting glycolysis. *Frontiers in oncology*, 11:639408, 2021.

[269] Brian A Camley and Wouter-Jan Rappel. Physical models of collective cell motility: from cell to tissue. *Journal of physics D: Applied physics*, 50(11): 113002, 2017.

[270] Benoit Ladoux and René-Marc Mège. Mechanobiology of collective cell behaviours. *Nature reviews Molecular cell biology*, 18(12):743–757, 2017.

[271] Gunnar Farnebäck. Two-frame motion estimation based on polynomial expansion. In *Scandinavian conference on Image analysis*, pages 363–370. Springer, 2003.

[272] Dhruv K Vig, Alex E Hamby, and Charles W Wolgemuth. On the quantification of cellular velocity fields. *Biophysical journal*, 110(7):1469–1475, 2016.

[273] Emi Hoshikawa, Taisuke Sato, Kenta Haga, Ayako Suzuki, Ryota Kobayashi, Koichi Tabeta, and Kenji Izumi. Cells/colony motion of oral keratinocytes determined by non-invasive and quantitative measurement using optical flow predicts epithelial regenerative capacity. *Scientific reports*, 11(1):1–12, 2021.

[274] Rizwan Chaudhry, Avinash Ravichandran, Gregory Hager, and René Vidal. Histograms of oriented optical flow and binet-cauchy kernels on nonlinear dynamical systems for the recognition of human actions. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1932–1939. IEEE, 2009.

[275] Michiel Verburg and Vlado Menkovski. Micro-expression detection in long videos using optical flow and recurrent neural networks. In *2019 14th IEEE International conference on automatic face & gesture recognition (FG 2019)*, pages 1–6. IEEE, 2019.

[276] Yong-Jin Liu, Jin-Kai Zhang, Wen-Jing Yan, Su-Jing Wang, Guoying Zhao, and Xiaolan Fu. A main directional mean optical flow feature for spontaneous micro-expression recognition. *IEEE Transactions on Affective Computing*, 7(4):299–310, 2015.

[277] Rensso Victor Hugo Mora Colque, Carlos Caetano, Matheus Toledo Lustosa de Andrade, and William Robson Schwartz. Histograms of optical flow orientation and magnitude and entropy to detect anomalous events in videos. *IEEE Transactions on Circuits and Systems for Video Technology*, 27(3): 673–682, 2016.

[278] Yali Huang, Lei Hao, Heng Li, Zhiwen Liu, and Peiguang Wang. Quantitative analysis of intracellular motility based on optical flow model. *Journal of healthcare engineering*, 2017, 2017.

[279] Ryan McConville, Raul Santos-Rodriguez, Robert J Piechocki, and Ian Craddock. N2d:(not too) deep clustering via clustering the local manifold of an autoencoded embedding. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 5145–5152. IEEE, 2021.

[280] Mebarka Allaoui, Nour El-Houda Sayah Ben Aissa, Abdellah Ben Belghith, and Mohammed Lamine Kherfi. A machine learning-based tool for exploring covid-19 scientific literature. In *2021 International Conference on Recent Advances in Mathematics and Informatics (ICRAMI)*, pages 1–7. IEEE, 2021.

[281] Pantelis Linardatos, Vasilis Papastefanopoulos, and Sotiris Kotsiantis. Explainable ai: A review of machine learning interpretability methods. *Entropy*, 23(1):18, 2020.

[282] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017.

[283] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929, 2016.

[284] Jonathan Bac, Evgeny M Mirkes, Alexander N Gorban, Ivan Tyukin, and Andrei Zinovyev. Scikit-dimension: A python package for intrinsic dimension estimation. *Entropy*, 23(10):1368, 2021.

[285] Yong Yu, Xiaosheng Si, Changhua Hu, and Jianxun Zhang. A review of recurrent neural networks: Lstm cells and network architectures. *Neural computation*, 31(7):1235–1270, 2019.

[286] Diego Garcia-Perez, Daniel Pérez-López, Ignacio Diaz-Blanco, Ana González-Muñiz, Manuel Domínguez-González, and Abel Alberto Cuadrado Vega. Fully-convolutional denoising auto-encoders for nilm in large non-residential buildings. *IEEE Transactions on Smart Grid*, 12(3):2722–2731, 2020.

[287] Ignacio Díaz Blanco, Diego García Pérez, Ana González Muñiz, and Abel A Cuadrado Vega. Análisis de vibraciones en una estructura utilizando echo state networks. In *XL Jornadas de Automática*, pages 186–191. Universidade da Coruña, Servizo de Publicacións, 2019.

[288] Ignacio Díaz-Blanco, José María Enguita-González, Diego García-Perez, Abel Cuadrado-Vega, Ana González-Muñiz, and Manuel Domínguez. Modelado de series temporales mediante echo state networks para aplicaciones de analítica visual. In *XVII Simposio CEA de Control Inteligente: Reunión anual del grupo de Control Inteligente del comité español de automática (CEA)*, 2022.

[289] Ignacio Díaz-Blanco, José María Enguita-González, Diego García-Perez, Ana González-Muñiz, Abel Cuadrado-Vega, María Dolores Chiara-Romero, and Nuria Valdés-Gallego. Interactive dual projections for gene expression analysis. In *ESANN*, 2022.

[290] José María Enguita-González, Diego García-Perez, María Dolores Chiara-Romero, Nuria Valdés-Gallego, Ana González-Muñiz, Abel Cuadrado-Vega, and Ignacio Díaz-Blanco. Interactive visual analytics for medical data: Application to covid-19 clinical information during the first wave. In *ESANN*, 2022.