E3.1 PLANTEAMIENTO TEÓRICO: MODELOS IA Y EXPLICABILIDAD

Versión 1.1

Información de la Documentación

Web del Proyecto	https://capsul-ia.es/
Fecha	30/12/2024
Nivel Diseminación	Público
Autor	Lucía García (GTD), Diego Gil (GTD)
Colaboradores	Alejandro Astruc (BSC), Axel Brando (BSC), Jokin Labaien (IKL)
Revisor	Jokin Labaien (IKL), Joel Frias (SML)
Palabras clave	Modelos IA, explicabilidad,

Project MIG-20232044, subsidized by the CDTI – Center for Industrial Technological Development, supported by the Ministry of Science and Innovation, and approved in the call for the year 2023 of the procedure for granting aid for "Science and Innovation Missions" within the framework of the TransMisiones 2023 initiative,

Registro de Modificaciones

Versión	Descripción del cambio
V0.1	Primer Borrador
V0.2	Primera versión
V1.0	Primera versión para revisar
V1.1	Se modifica el nivel de diseminación de confidencial a público

Índice

1 Introducción a la explicabilidad de la IA 5
1.1 Historia y Origen de la XAI 6
1.2 Objetivo 7
1.3 Alcance 7
2 Metodología de explicabilidad 8
2.1 Modelos autoexplicables 9
2.1.1 Ventajas e inconvenientes 9
2.2 Técnicas de explicabilidad 9
2.2.1 Técnicas de Explicabilidad Emergentes y Complementarias 11
2.2.2 Desafíos y consideraciones 11
3 Propuestas de explicabilidad 13
3.1 Cápsula 2: Visión Artificial 13
3.1.1 Algoritmos de explicabilidad para YOLO 15
3.1.2 PointNet y PointCloud Saliency Maps 19
3.2 Cápsula 3: Datos Tabulares y Series Temporales 22
3.2.1 Redes Neuronales Convolucionales (CNN) 22
3.2.2 Redes Neuronales Recurrentes (RNN) 25
3.2.3 Transformers 28
4 Conclusiones 31
5 Referencias 33

Índice de figuras

Figura 1. Evolución de la tasa de error en ImageNet. [26] 14

Figura 2. Tipos de segmentación de imágenes [30] 15

Figura 3. Comparativa entre un mapa de calor generado con CAM (no discriminante respecto a características) y uno generado con dCAM (discriminante en las dos dimensiones). [55] 24

Figura 4. Arquitectura de RETAIN 26

Figura 5. Arquitectura basada en LSTM y CNN que genera los pesos de atención y las predicciones [58] 27

Figura 6. Mecanismo basado en propagación hacia atrás que genera la atención en la dimensión de variables [58] 28

Figura 7. Arquitectura de DFSTrans [63] 29

Índice de ecuaciones

Ecuación 1. Ponderación αkc para el canal k [42] 16
Ecuación 2. Suma de activaciones ponderadas de todos los canales [42] 17
Ecuación 3. Centro del sistema de coordenadas 21
Ecuación 4. Gradiente de la pérdida L con respecto a r 21
Ecuación 5. El gradiente de L con respecto a ρi 21
Ecuación 6. Saliencia de xi 21

1 Introducción a la explicabilidad de la IA

La inteligencia artificial (IA) ha transformado significativamente la manera en que interactuamos con la tecnología, permitiendo avances en diversas áreas, desde la automatización de tareas cotidianas hasta la resolución de problemas complejos en sectores críticos. Sin embargo, a medida que los algoritmos y modelos se han vuelto más sofisticados, también han surgido preocupaciones sobre su falta de transparencia. Los modelos de aprendizaje profundo, en particular, a menudo operan como "cajas negras", dificultando a los usuarios y desarrolladores entender cómo se toman las decisiones [1], [2]. Este desafío ha llevado al desarrollo de la Inteligencia Artificial Explicable (XAI, por sus siglas en inglés), una subdisciplina que busca abordar la opacidad inherente de estos sistemas [3], [4].

La XAI se centra en proporcionar explicaciones claras y comprensibles sobre el funcionamiento interno de los modelos de IA o las razones detrás de sus predicciones. Este enfoque no solo mejora la confianza en los sistemas, sino que también facilita su adopción en sectores donde la precisión y la ética son esenciales, como la salud, la justicia penal y la ciberseguridad [5]. Además, la creciente presión regulatoria, como las cláusulas del Reglamento General de Protección de Datos (GDPR) en Europa, ha subrayado la importancia de la explicabilidad para garantizar el cumplimiento legal y ético en el uso de tecnologías avanzadas [1], [4].

En particular, las explicaciones que ofrece la XAI existen bajo alguna de las siguientes teorías de la explicación: *lógica, causal* o *funcional*. Si definimos la cuestión de generar explicaciones como respuesta a "¿por qué un modelo hace ciertas predicciones?", nos encontraríamos en el contexto teórico de explicaciones pragmáticas funcionales. La descripción del fenómeno a explicar (o explanandum) se define funcionalmente en torno a preguntas de "por qué", y la explicación (o explanans) consiste en estructuras lógicas que señalan un contexto explicativo y definen relaciones de relevancia con diferentes fenómenos [6]. Es en torno a esta pregunta que podemos deducir una serie de características de los métodos de explicabilidad [7]:

I. El modelo: considerando el aspecto del modelo en la pregunta, los métodos varían con el tipo de acceso dado a la hora de explicarlo, en qué grado y de qué modo podemos extraer información del modelo y sus elementos internos. A esta cualidad se la denomina **accesibilidad del modelo**, que abarca desde modelos cajas negra o **black-box**, sin acceso a ninguno de los elementos internos, tan solo a los datos de entrada y salida; hasta modelos caja blanca o **white-box** que permiten acceso a cualesquiera de los elementos internos constituyentes del modelo. Otra propiedad de los métodos de XAI en relación con el modelo es si el método está diseñado para un modelo **específico** (**model-specific**) o es independiente del modelo, esto es, de propósito general, **agnóstico** con respecto al modelo (**model-agnostic**). Una tercera propiedad es en qué **etapa** o momento en el proceso de modelado se incorporó la explicabilidad, o se tuvo en cuenta, bien **ante-hoc**, antes del entreno del modelo, incorporando interpretabilidad en su arquitectura, siendo esta **autoexplicable**, o **post-hoc**, tras haber entrenado un modelo sin condicionar su arquitectura o el proceso de entreno.

II. La profundidad de la explicación: la respuesta a una pregunta de por qué se da un fenómeno en principio no tiene límite de granularidad. Podríamos considerar tan solo los aspectos más relevantes de la explicación, pero diferentes escenarios requieren de un grado variable de granularidad. La mayoría de las publicaciones evitan esta problemática casi por completo y se restringe a los datos y el modelo, esto es, enfoques **modelo-céntricos**, sin considerar agentes con influencia más amplia u otras externalidades. Algunas publicaciones recientes abogan por explicaciones más informadas por cuestiones sociales que contextualicen las explicaciones en el mundo real, estas pasarían a ser **explicaciones socio-estructurales** [8].

III. **Rango** (**scope**): Otro aspecto de la profundidad de la explicación es si es aplicable al razonamiento general del modelo, o depende de la instancia específica a explicar. Esto es, una explicación puede ser aplicable de forma general al funcionamiento medio del modelo sobre la distribución de datos, una explicación **global**, o ajustada al comportamiento en torno a una región cercana o **local** de una instancia concreta.

IV. Casi todos los métodos (modelo-céntricos) de explicabilidad ponen énfasis o recaen en un aspecto concreto de los datos o el modelo para formular explicaciones. A este aspecto se le denomina **tipo de explicación** o **unidad de explicación**. Esta unidad puede ser **atributos de entrada** (**input features**), **ejemplos**, **elementos de la arquitectura del modelo**, **conceptos**, **interacciones entre atributos**, o una combinación de estos. Podemos presentar estas unidades de explicación de formas variadas, a esto se le denomina **forma de explicación**. Esta forma puede ser **visual**, **puntuaciones de importancia**, **lenguaje natural**, **diagramas**, etc.

En este documento, se exploran los avances, las metodologías clave y las aplicaciones actuales de la XAI, con el objetivo de ofrecer una visión completa de su estado del arte y los retos futuros.

1.1 Historia y Origen de la XAI

La evolución de la Inteligencia Artificial Explicable (XAI) está profundamente arraigada en la historia del aprendizaje automático y la necesidad de interpretabilidad en sistemas complejos. Durante la última década, el uso de algoritmos de aprendizaje profundo y redes neuronales ha crecido exponencialmente, gracias a su capacidad para manejar datos no estructurados y realizar tareas complejas con gran precisión. Sin embargo, estos modelos a menudo operan como "cajas negras", lo que significa que las razones detrás de sus decisiones no son directamente observables ni comprensibles [1], [2].

Los primeros enfoques hacia la interpretabilidad se centraron en modelos más simples, como las regresiones lineales y los árboles de decisión. A medida que se incorporaron redes neuronales profundas y modelos complejos como las máquinas de soporte vectorial, surgió la preocupación por la falta de transparencia, especialmente en aplicaciones críticas como la medicina, la conducción autónoma y la justicia penal [2], [3].

En respuesta a estos desafíos, el campo de la XAI emergió como una disciplina interdisciplinaria que combina inteligencia artificial, ética, derechos legales y diseño centrado en el usuario. La adopción del Reglamento General de Protección de Datos (GDPR) en la Unión Europea marcó un hito significativo al exigir explicaciones comprensibles para decisiones automatizadas, resaltando la importancia de la explicabilidad no solo como una cuestión técnica, sino también como un imperativo ético y social [1], [4].

1.2 Objetivo

El principal objetivo de la XAI es proporcionar a los usuarios, ya sean expertos técnicos o no, una comprensión clara de cómo los modelos de IA toman decisiones. Esto incluye diseñar modelos que sean interpretables desde su concepción o desarrollar herramientas post-hoc que permitan desentrañar la lógica detrás de un modelo ya entrenado. La XAI busca equilibrar dos aspectos fundamentales: la precisión predictiva y la interpretabilidad, lo cual resulta esencial en sectores como la salud, la ciberseguridad y la energía [3], [5].

Además, este documento tiene como objetivo sistematizar las metodologías actuales, identificando los avances más relevantes, las limitaciones de los enfoques actuales y las oportunidades para futuras investigaciones en este campo dinámico y crítico.

1.3 Alcance

Este estado del arte aborda dos aspectos principales de la XAI: los modelos autoexplicables y las técnicas de explicabilidad post-hoc. A través de ejemplos de aplicaciones concretas, se analiza cómo estas metodologías están transformando diversas industrias. También se presentan los desafíos persistentes, como la evaluación de la efectividad de las explicaciones y la integración de valores éticos en el diseño de los modelos. Finalmente, se discuten direcciones futuras, como el desarrollo de métodos de explicabilidad que sean dinámicos, robustos y adaptables a diferentes dominios [2], [4], [5].

2 Metodología de explicabilidad

La explicabilidad en los sistemas de inteligencia artificial es un componente crucial para garantizar la confianza y la transparencia en sus aplicaciones. A medida que los modelos de aprendizaje automático y profundo se vuelven más complejos, entender cómo estos generan sus predicciones o decisiones resulta esencial, especialmente en sectores críticos como la salud, la justicia y la ciberseguridad. La explicabilidad no solo permite identificar posibles sesgos y errores en los modelos, sino que también contribuye a cumplir con regulaciones legales y éticas [9].

La necesidad de explicabilidad surge de la complejidad inherente a muchos modelos de IA, especialmente aquellos basados en técnicas de aprendizaje profundo. Estos modelos, a menudo descritos como "cajas negras", pueden generar resultados precisos pero difíciles de interpretar. La falta de transparencia puede llevar a problemas de confianza y aceptación, así como a desafíos éticos y legales. Por lo tanto, desarrollar metodologías que permitan explicar el funcionamiento interno de estos modelos es una prioridad en la investigación y aplicación de la IA [10].

Si un modelo se diseña para ser autoexplicable, o con la explicabilidad en mente durante o antes del entrenamiento, este caería dentro de la categoría ante-hoc. Como tal, las arquitecturas de modelos autoexplicables está constreñidas con el fin de ser interpretables. Este enfoque hacia la explicabilidad existe dentro del paradigma intrínseco. La complejidad del modelo, su rendimiento y explicabilidad están relacionados. Un modelo más simple es considerado más explicable pero incapaz de conseguir un rendimiento comparable al de opciones más complejas.

En contraste con este enfoque está el enfoque post-hoc que toma un modelo ya entrenado e intenta explicarlo. Bajo este enfoque podemos beneficiarnos de la flexibilidad que conlleva no restringir el diseño del modelo, porque las explicaciones se generan a partir del modelo entrenado. Además, hay una preferencia notable en la industria por modelos estándares disponibles de propósito general, con lo cual la explicabilidad queda relegada a un plano secundario, como un añadido. Otra ventaja del enfoque post-hoc es que se puede aplicar a modelos de propósito general, y modelos que se desarrollen en el futuro.

Ambos paradigmas dentro de la XAI poseen limitaciones propias, la autoexplicabilidad es un concepto disputado con ejemplos como el debate sobre la atención que aún no se ha resuelto [11], y los métodos o técnicas de explicabilidad (post-hoc) requieren de más recursos computacionales y tienden a generar explicaciones contradictorias dependiendo del método.

En esta sección, se presentan dos enfoques fundamentales en la XAI: los modelos autoexplicables, que son comprensibles por diseño, y las técnicas de explicabilidad post-hoc, que buscan interpretar modelos ya entrenados. Cada enfoque tiene sus ventajas y limitaciones, dependiendo del contexto de aplicación y las necesidades específicas del usuario final. A continuación, se profundiza en estas metodologías, proporcionando ejemplos concretos y analizando sus implicaciones prácticas y teóricas.

2.1 Modelos autoexplicables

Los modelos autoexplicables, o de caja blanca, están diseñados para ofrecer interpretabilidad intrínseca sin necesidad de métodos adicionales para su explicación. Entre los modelos más comunes en esta categoría se encuentran los siguientes:

1. **Regresiones lineales y logísticas**: Estos modelos son transparentes debido a su naturaleza matemática simple, donde los coeficientes proporcionan una interpretación directa de la relación entre variables independientes y el resultado [2]. En la regresión lineal, cada coeficiente indica el cambio esperado en la variable dependiente por un cambio unitario en la variable independiente, manteniendo constantes las demás variables.
2. **Árboles de decisión**: Estos modelos representan decisiones y sus posibles consecuencias en una estructura jerárquica que es intuitivamente comprensible para los usuarios. Cada nodo del árbol representa una característica específica, y cada rama representa un resultado posible, lo que permite rastrear cómo se llegó a una predicción [3], [5].
3. **Modelos de reglas difusas**: Aunque más complejos, los sistemas basados en reglas pueden diseñarse para ser comprensibles si las reglas son explícitas y las combinaciones de condiciones se mantienen manejables. Estos modelos utilizan lógica difusa para manejar la incertidumbre y la imprecisión, permitiendo una interpretación más flexible de las reglas [2], [12].

2.1.1 Ventajas e inconvenientes

Los modelos autoexplicables destacan por su simplicidad y claridad, lo que los hace ideales para aplicaciones donde la confianza del usuario es crucial, como en la medicina y las finanzas. Su naturaleza transparente permite a los usuarios entender fácilmente cómo se generan las predicciones, lo que facilita la identificación de posibles sesgos y errores [9]. Sin embargo, su desempeño puede ser limitado en tareas con datos altamente no lineales o donde las interacciones entre características son complejas, ya que estos modelos a menudo no capturan adecuadamente tales relaciones [10].

Esto ha llevado al desarrollo de enfoques híbridos, como los modelos de caja gris, que buscan equilibrar precisión e interpretabilidad al combinar elementos de modelos autoexplicables y modelos de caja negra [13]. Estos enfoques híbridos permiten aprovechar la alta precisión de los modelos complejos mientras se mantiene un nivel aceptable de interpretabilidad, adaptándose mejor a una variedad de aplicaciones prácticas [13].

2.2 Técnicas de explicabilidad

Para modelos más complejos, como las redes neuronales profundas, se requieren técnicas adicionales para lograr la explicabilidad. Estas técnicas pueden clasificarse en explicaciones **locales** o **globales**, dependiendo de si se enfocan en explicar una predicción específica o el comportamiento general del modelo.

- **Explicaciones locales:** Estas técnicas buscan interpretar el porqué de una predicción en particular, ayudando a entender cómo el modelo toma decisiones en casos específicos. Para este tipo de explicaciones, se pueden utilizar métodos como:
    - **LIME (Local Interpretable Model-agnostic Explanations):** LIME crea un modelo interpretable simplificado (como una regresión lineal o un árbol de decisión) que aproxima el comportamiento del modelo complejo en torno a una predicción específica. Genera perturbaciones en los datos de entrada y observa cómo cambian las predicciones, construyendo un modelo simple que representa el comportamiento del modelo complejo en esa vecindad [2], [3], [9].
    - **SHAP (SHapley Additive exPlanations):** Basada en la teoría de juegos, SHAP asigna valores a cada característica de entrada [2], [9], indicando su contribución a una predicción específica. Aunque es ampliamente utilizado para explicaciones locales, también puede ser aplicado para obtener explicaciones globales (promediando las contribuciones de las características a través de múltiples predicciones).
    - **GradCAM:** GradCAM [14] es una técnica de visualización de mapas de calor utilizada en redes neuronales convolucionales (CNN) para entender qué regiones de una imagen son importantes para la predicción de una clase específica. Grad-CAM funciona calculando los gradientes de la clase objetivo con respecto a las características de la última capa convolucional del modelo. En la literatura existen variantes métodos de visualización de mapas de calor (Grad-CAM++ [15], HiResCAM [16], EigenCAM [17], etc.), las cuales ofrecen una explicación intuitiva del razonamiento local del modelo. [2], [10].

- **Explicaciones globales**: Estas técnicas proporcionan una visión general del comportamiento del modelo, permitiendo entender patrones más amplios en sus predicciones. Para este tipo de explicaciones, se pueden utilizar métodos como:
    - **SHAP:** Cuando se calculan valores de Shapley en un conjunto amplio de datos, es posible obtener una perspectiva global sobre qué características son más relevantes para las predicciones del modelo en promedio.
    - **Análisis de sensibilidad:** Evalúa cómo las variaciones en las entradas afectan las salidas, revelando la importancia de cada característica y sus interacciones. Es útil para identificar qué variables tienen mayor peso en el comportamiento global del modelo [3], [4], [10].
    - **Partial Dependence Plots (PDP):** Muestran cómo una o dos características seleccionadas afectan las predicciones promedio del modelo, manteniendo las demás constantes. Estas gráficas ayudan a visualizar relaciones globales no lineales o interacciones entre variables [18].

Es aparente como muchos de los métodos (post-hoc) de explicabilidad más populares tienden a estar en desacuerdo. Cada método resuelve un problema de optimización diferente bajo diferentes criterios, y por tanto obtienen explicaciones contradictorias. En [19] estudian esta problemática y definen métricas para medir el desacuerdo entre distintas explicaciones. Además, realizan una serie de encuestas a profesionales de la industria de la IA mostrando cuan arbitrarias son las decisiones tomadas en torno a qué método de XAI emplear. **El problema del desacuerdo** se acentúa por la incapacidad de falsear objetivamente las explicaciones. En muchos casos el criterio al que se recurre para validar o falsear explicaciones es evaluación humana, práctica que ha sido criticada por

ser problemática e incluso peligrosa [20]. Hay una ausencia de métodos objetivos para evaluar la validez de las explicaciones.

2.2.1 Técnicas de Explicabilidad Emergentes y Complementarias

Además de las técnicas mencionadas, existen técnicas que están ganando relevancia por su capacidad para abordar limitaciones de los métodos comentados anteriormente:

- **Métodos basados en atención**: Utilizados ampliamente en modelos de procesamiento de lenguaje natural (NLP), en visión por computadora y en datos secuenciales (series temporales, por ejemplo), estos métodos permiten visualizar qué partes de una entrada (como palabras en un texto o regiones de una imagen) son consideradas más relevantes para una predicción. Los mecanismos de atención permiten visualizar qué palabras o frases influyen más en la decisión del modelo [2], [13].
- **Descomposición de modelos**: Esta técnica implica descomponer un modelo complejo en componentes más simples que son más fáciles de interpretar. Por ejemplo, una red neuronal profunda puede ser descompuesta en capas y neuronas individuales para analizar su contribución a la predicción final [13].
- **Métodos basados en prototipos y críticas:** Este enfoque identifica ejemplos prototípicos que son representativos del comportamiento del modelo y ejemplos críticos que son atípicos o limitan la confianza del modelo. Esto puede ser útil en conjuntos de datos complejos como imágenes o texto, donde los prototipos ayudan a ilustrar patrones aprendidos por el modelo [21], [22].
- **Explicaciones mediante IA generativa:** Explicaciones generadas por modelos de lenguaje avanzados (GPT-4, LLaMa, Claude…) ayudan a interpretar y contextualizar los resultados de modelos más complejos. Por ejemplo, pueden generar explicaciones narrativas que transforman análisis estadísticos o decisiones algorítmicas en lenguaje natural accesible para usuarios no técnicos [23].
- **Contrafactuales**: Estas explicaciones proponen cambios mínimos necesarios en las características de entrada para alterar la predicción. Por ejemplo, en un modelo de crédito, un contrafactual podría mostrar que, si el ingreso de un solicitante fuera ligeramente mayor, su solicitud de crédito sería aprobada [3], [4], [13].

2.2.2 Desafíos y consideraciones

Un desafío crítico para las técnicas de explicabilidad post-hoc es garantizar que las explicaciones sean tanto fieles al modelo como comprensibles para los usuarios. Esto incluye abordar problemas de robustez, escalabilidad y adaptabilidad a diferentes dominios [5], [12].

La robustez se refiere a la capacidad de las explicaciones para mantenerse consistentes y precisas frente a perturbaciones en los datos de entrada [9]. La escalabilidad es otro

desafío importante, ya que las técnicas de explicabilidad deben ser capaces de manejar grandes volúmenes de datos y modelos complejos sin perder eficiencia [24]. Además, la adaptabilidad a diferentes dominios implica que las técnicas deben ser flexibles y aplicables a una variedad de contextos y tipos de datos, lo cual es esencial para su implementación en sectores diversos como la medicina, la justicia y la ciberseguridad [13].

Un componente clave para abordar estos desafíos es la evaluación de las explicaciones mediante métricas que permitan medir tanto su calidad técnica como su utilidad práctica. La fidelidad, es decir, qué tan bien las explicaciones reflejan el comportamiento real del modelo subyacente, se evalúa mediante métricas como infidelity y sensitivity [25]. Además, la coherencia local es útil para validar si las explicaciones son consistentes con las predicciones del modelo en pequeñas variaciones de los datos de entrada [26].

Por otro lado, la comprensibilidad se puede evaluar mediante métodos cualitativos y cuantitativos. Las métricas cuantitativas incluyen la complejidad de las explicaciones (como el número de reglas en un modelo explicativo basado en reglas o la longitud de una descripción textual) y el nivel de abstracción necesario para comprenderlas [27]. A nivel práctico, las pruebas con usuarios reales son fundamentales, midiendo factores como la facilidad de uso, el tiempo necesario para entender una explicación y su impacto en la toma de decisiones. Esto es, los desarrolladores de modelos suelen requerir explicaciones técnicas detalladas, mientras que los usuarios finales, como médicos o jueces, prefieren explicaciones más intuitivas y orientadas a sus dominios específicos [28].

Estos desafíos subrayan la necesidad de un enfoque equilibrado que combine precisión y claridad en las explicaciones, promoviendo así la confianza y la aceptación de los sistemas de IA en aplicaciones críticas [24].

3 Propuestas de explicabilidad

En este apartado, se presentan diversas propuestas de técnicas de explicabilidad adaptadas a los algoritmos de inteligencia artificial utilizados: YOLO, PointNet, CNN, LSTM y Transformers. La selección de la técnica adecuada es crucial, ya que cada tipo de algoritmo posee características y complejidades únicas que requieren enfoques específicos para garantizar una interpretación precisa y comprensible de sus resultados. La explicabilidad en IA no solo mejora la transparencia y la confianza en los modelos, sino que también facilita la identificación de posibles sesgos y errores, y asegura el cumplimiento de regulaciones legales y éticas. Además, la capacidad de explicar las decisiones de los modelos es esencial para su aceptación en sectores críticos como la medicina, la justicia y la ciberseguridad.

Para los algoritmos de visión artificial como YOLO y PointNet, es fundamental utilizar técnicas que permitan visualizar y entender cómo estos modelos procesan y analizan las imágenes y nubes de puntos. Por otro lado, para los algoritmos utilizados en un optimizador multicriterio, como CNN, LSTM y Transformers, se requieren técnicas que puedan descomponer y explicar las decisiones basadas en múltiples criterios y secuencias temporales. A continuación, se analizan las técnicas recomendadas para estos algoritmos, destacando cómo pueden ayudar a desentrañar las decisiones de modelos complejos y proporcionar una guía práctica para su implementación efectiva.

A través de estos análisis, se pretende proporcionar una guía práctica para seleccionar y aplicar las técnicas de explicabilidad más adecuadas en función del algoritmo utilizado y del contexto de aplicación.

3.1 Cápsula 2: Visión Artificial

La visión artificial es una vertiente de la visión por computador donde se utilizan técnicas de inteligencia artificial y machine learning para conseguir sistemas capaces de interpretar y comprender la información representada en imágenes o videos y, en caso de ser necesario, tomar decisiones basadas en esa información.

Aunque el campo de la visión por computador ha tenido su auge desde la década de los 2010s, su arquitectura básica, las redes neuronales convolucionales, tienen su origen en el Neocognitron de K. Fukushima [29]. Esta arquitectura fue posteriormente mejorada por el equipo de Yann LeCun, y el campo de la visión por computador creció en popularidad tras la competición ImageNet de 2015 cuando se consiguió una tasa de error en esta tarea de clasificación por debajo del error humano con la arquitectura ResNet.

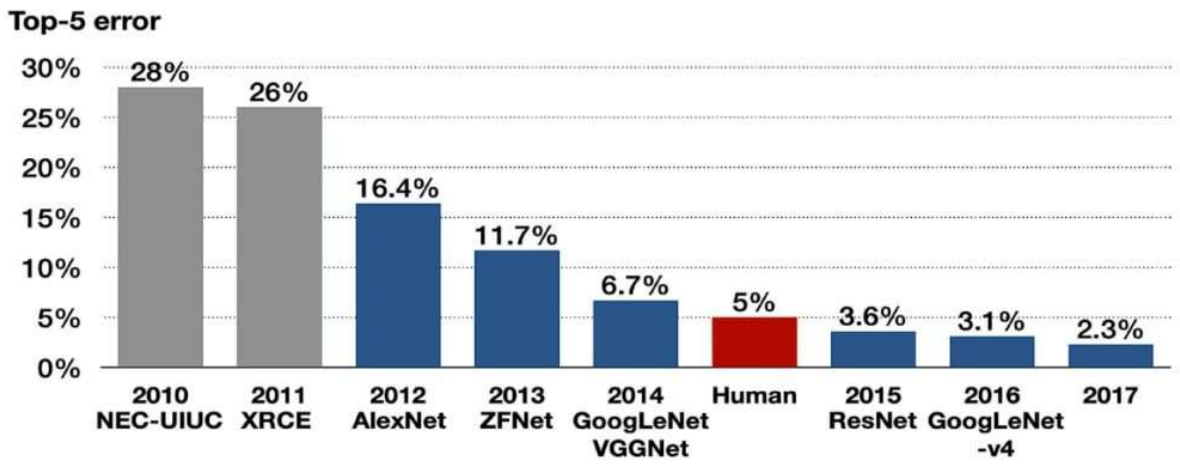

Figura 1. Evolución de la tasa de error en ImageNet. [30]

Actualmente, a parte de las aproximaciones ya tradicionales en base a CNNs, se investiga la utilización de la arquitectura Transformer [31], originalmente pensada para procesamiento del lenguaje natural, en el campo de la visión por computador, dando lugar a nuevas arquitecturas como los Vision Transformers (ViT) [32].

Dentro del campo de la visión artificial existen tres objetivos básicos para estos sistemas de visión artificial: clasificación, detección de objetos y segmentación.

La clasificación de imágenes consiste en asignar una o más etiquetas de clase a una determinada imagen. Esta tarea tiene una amplia gama de aplicaciones, como el reconocimiento de objetos en fotografías, detección de enfermedades en imágenes biomédicas o la inspección visual de productos manufacturados.

Por otro lado, la detección de objetos en imagen consiste, no solo en identificar la presencia de objetos, sino también localizar cada objeto dentro de esa imagen.

Por último, la segmentación [33] es una técnica que trata de dividir una imagen en distintos grupos. Dentro de esta técnica existen diferentes tipos.

- **Segmentación semántica**: que tiene como objetivo asignar una etiqueta de clase a cada píxel de la imagen, indicando la clase del objeto al que pertenece.
- **Segmentación de instancias:** similar a la segmentación semántica pero además distingue entre diferentes instancias del mismo objeto.
- **Segmentación panóptica**: combina ambas aproximaciones anteriores, etiquetando todos los píxeles de la imagen y distinguiendo entre diferentes instancias de objetos.

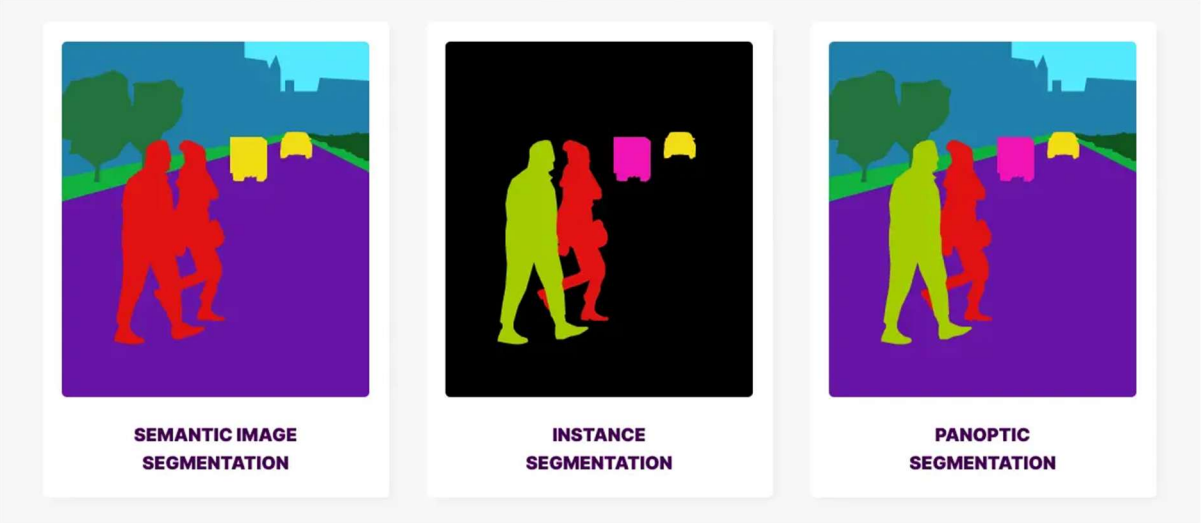

Figura 2. Tipos de segmentación de imágenes ***[34]***

Estas técnicas son igualmente aplicables tanto a imágenes como a videos y, dependiendo de las prestaciones del equipo donde se ejecute el sistema de visión artificial, son aplicables también a video en *streaming*.

3.1.1 Algoritmos de explicabilidad para YOLO

El modelo YOLOv8 (You Only Look Once, versión 8), desarrollado por Ultralytics, se ha convertido en un referente en el campo de la visión por computadora, especialmente en tareas de detección de objetos, clasificación de imágenes y segmentación de instancias [35]. YOLOv8 se basa en los avances de sus predecesores, incorporando mejoras significativas en la arquitectura del modelo y en la experiencia del desarrollador, lo que lo hace más eficiente y fácil de usar [36].

El algoritmo YOLO se originó en 2015 con la publicación de YOLOv1 por Joseph Redmon. Desde entonces, ha evolucionado considerablemente, con versiones posteriores que han mejorado tanto en precisión como en velocidad. YOLOv8, en particular, introduce optimizaciones que permiten un rendimiento superior en términos de precisión y velocidad de inferencia, lo que lo hace ideal para aplicaciones en tiempo real [35]. A diferencia de otros modelos de detección de objetos que siguen un enfoque de dos etapas, YOLO realiza la detección en una sola pasada, lo que contribuye a su rapidez [37].

Entre las ventajas de YOLOv8 se encuentran su capacidad para ser entrenado y desplegado en hardware de bajo costo, su alta precisión y su velocidad de inferencia rápida [38]. Estas características lo hacen adecuado para una amplia gama de aplicaciones, desde la vigilancia y la conducción autónoma hasta la realidad aumentada y la robótica. Sin embargo, YOLOv8 también tiene algunas limitaciones. Por ejemplo, puede tener dificultades para detectar objetos pequeños en imágenes de alta resolución y puede ser menos preciso en comparación con modelos más complejos como Mask R-CNN en tareas de segmentación de instancias [39].

En resumen, YOLOv8 representa un equilibrio entre precisión y eficiencia, ofreciendo una solución robusta para muchas aplicaciones de visión por computadora. A continuación, se

discutirán las técnicas de explicabilidad que se pueden utilizar con YOLOv8 para mejorar la comprensión de sus decisiones y aumentar la confianza en sus predicciones.

Para el algoritmo YOLO, se recomiendan técnicas como Grad-CAM [14] y sus variantes (Grad-CAM++ [15], HiResCAM [16], EigenCAM [17]) que permiten visualizar las áreas de la imagen que más influyen en las decisiones del modelo. Estas técnicas ayudan a identificar qué partes de la imagen están siendo consideradas por el modelo al realizar una detección, proporcionando una interpretación visual intuitiva. A continuación, se detalla el funcionamiento del método Grad-CAM por ser uno de los métodos más empleados en la literatura, y posteriormente se analizan sus principales variantes para comprender su evolución y mejoras.

3.1.1.1 Funcionamiento de Grad-CAM

Grad-CAM (Gradient-weighted Class Activation Mapping) es una técnica de visualización utilizada en redes neuronales convolucionales (CNN) para entender qué regiones de una imagen son importantes para la predicción de una clase específica. Grad-CAM funciona calculando los gradientes de la clase objetivo con respecto a las características de la última capa convolucional del modelo. Estos gradientes se utilizan para ponderar las activaciones de la capa convolucional, generando un mapa de calor que resalta las áreas de la imagen que más influyen en la decisión del modelo [40], [41].

A continuación, se detalla el funcionamiento técnico de Grad-CAM:

1. **Cálculo de Gradientes** [42]:

Primero, se realiza una pasada hacia adelante a través de la red para obtener las activaciones de la última capa convolucional y la puntuación de la clase objetivo.

Luego, se calcula el gradiente de la puntuación de la clase objetivo (y^c) con respecto a las activaciones de la última capa convolucional (A^k). Este gradiente indica cómo cambia la puntuación de la clase objetivo cuando se modifican ligeramente las activaciones de la última capa convolucional.

Matemáticamente, esto se expresa como ($\frac{\partial y^c}{\partial A^k}$), donde (k) denota el índice del canal en la última capa convolucional

2. **Ponderación de Activaciones** [42]:

Los gradientes calculados se promedian globalmente para obtener una ponderación de importancia para cada canal de la última capa convolucional. Esta ponderación refleja la importancia de cada canal para la clase objetivo.

La ponderación (α_k^c) para el canal (k) se calcula como:

$$\left[\alpha_k^c = \frac{1}{Z}\sum_i\sum_j \frac{\partial y^c}{\partial A_{ij}}\right]$$

Ecuación 1. Ponderación α_k^c para el canal k [42]

donde (Z) es el número de píxeles en el mapa de características (es decir, el producto de las dimensiones espaciales del mapa de características), y (i) y (j) son los índices espaciales [42].

3. **Generación del Mapa de Calor** [42]:

Las activaciones de la última capa convolucional se ponderan utilizando las ponderaciones calculadas (α_k^c). Esto se hace multiplicando cada canal de activación (A^k) por su ponderación correspondiente (α_k^c).

Luego, se suman las activaciones ponderadas a través de todos los canales para obtener un mapa de calor:

$$L^c_{\text{Grad-CAM}}: \left[L^c_{\text{Grad-CAM}} = \text{ReLU}\left(\sum_k \alpha_k^c A^k \right) \right]$$

Ecuación 2. Suma de activaciones ponderadas de todos los canales [42]

La función ReLU (Rectified Linear Unit) se aplica para mantener solo las contribuciones positivas, ya que solo las características que tienen un efecto positivo en la clase objetivo son de interés.

4. **Superposición del Mapa de Calor** [41]:

El mapa de calor generado se redimensiona para que coincida con el tamaño de la imagen de entrada y se superpone a la imagen original. Esto permite visualizar las regiones de la imagen que más influyen en la predicción de la clase objetivo.

La superposición se realiza típicamente utilizando una combinación de colores (por ejemplo, un colormap de tipo jet) para resaltar las áreas importantes.

Grad-CAM es una técnica poderosa porque no requiere modificaciones en la arquitectura del modelo ni reentrenamiento, lo que facilita su implementación en modelos ya entrenados. Además, proporciona una visualización intuitiva que ayuda a los usuarios a entender y confiar en las decisiones del modelo.

3.1.1.2Ventajas y limitaciones de Grad-CAM

Las ventajas que presenta esta técnica son:

- **Visualización intuitiva**: Grad-CAM genera mapas de calor que son fáciles de interpretar, mostrando claramente las regiones de la imagen que influyen en las decisiones del modelo [41].
- **Compatibilidad con diferentes modelos**: Grad-CAM puede ser utilizado con una variedad de arquitecturas de redes neuronales, incluyendo CNNs y Transformers de Visión [43].
- **No requiere modificaciones del modelo**: Grad-CAM no necesita cambios en la arquitectura del modelo ni reentrenamiento, lo que facilita su implementación en modelos ya entrenados [44].

Las limitaciones de este tipo de técnica, por otro lado:

- **Resolución limitada**: Los mapas de calor generados por Grad-CAM pueden carecer de detalles a nivel de píxel, lo que puede ser un inconveniente en aplicaciones que requieren alta precisión [41].
- **Dependencia de la última capa convolucional**: La efectividad de Grad-CAM depende de la calidad de las activaciones de la última capa convolucional, lo que puede limitar su aplicabilidad en modelos con arquitecturas más complejas [44].

- **Sensibilidad a perturbaciones**: Grad-CAM puede ser sensible a pequeñas perturbaciones en los datos de entrada, lo que puede afectar la consistencia de las visualizaciones [41].

3.1.1.3 Variantes de Grad-CAM

El desarrollo continuo de técnicas de explicabilidad ha llevado a la creación de múltiples algoritmos de generación de mapas de calor. Durante el análisis de estos algoritmos, se han identificado varios que se pueden aplicar a YOLO como, por ejemplo: Grad-CAM++, EigenCAM, EigenGradCAM, XGradCAM, HiResCAM o LayerCAM.

Cada uno de estos algoritmos está diseñado para abordar limitaciones específicas del método original. Grad-CAM++ [24] mejora el método original Grad-CAM introduciendo ponderaciones más precisas para los mapas de características. A diferencia de Grad-CAM, que asigna un único peso por canal, Grad-CAM++ calcula pesos específicos para cada píxel dentro de cada canal usando gradientes de segundo orden, permitiendo una localización más precisa de las regiones relevantes, especialmente útil en modelos con múltiples instancias de una misma clase en una imagen. Otra variante de GradCAM, XGradCAM, escala los gradientes mediante la normalización de las activaciones. Este método introduce un enfoque más refinado para ponderar las contribuciones de cada característica, mejorando la capacidad de interpretación del mapa de calor al considerar tanto la magnitud de los gradientes como la intensidad de las activaciones.

Por otro lado, HiResCAM [16] aborda una de las principales limitaciones de Grad-CAM: su baja resolución espacial. Esta variante utiliza técnicas de multiplicación elemento a elemento entre activaciones y gradientes, garantizando una fidelidad comprobable en ciertos modelos. Al preservar más información espacial, HiResCAM proporciona visualizaciones más precisas y detalladas de las áreas influyentes en la decisión del modelo.

EigenCAM [17] introduce un enfoque completamente diferente basado en el análisis de componentes principales (PCA). En lugar de utilizar gradientes, EigenCAM descompone las activaciones de la capa convolucional utilizando técnicas de álgebra lineal, extrayendo el primer componente principal. Aunque carece de discriminación de clases, este método ha demostrado generar resultados visuales muy consistentes y estables. Por otro lado, EigenGradCAM combina los principios de EigenCAM con la discriminación de clases. Calcula el primer componente principal de la multiplicación entre activaciones y gradientes, lo que genera mapas de calor similares a Grad-CAM pero con una representación más limpia y estructurada.

LayerCAM [45] proporciona una perspectiva más profunda al considerar activaciones de múltiples capas convolucionales, no solo de la última. Pondera espacialmente las activaciones utilizando gradientes positivos, lo que resulta especialmente efectivo en capas más profundas de la red neuronal. De esta manera, el enfoque permite capturar información de diferentes niveles de abstracción.

Cada una de estas variantes representa un avance en la comprensión y visualización de las decisiones de modelos de visión por computadora. No existe un método universalmente óptimo, y la elección dependerá del modelo específico, la tarea y los requisitos de interpretabilidad.

3.1.1.4 Conclusiones

En cuanto a su implementación, las técnicas de generación de mapas de calor se han integrado en diversas bibliotecas y frameworks de aprendizaje profundo. La biblioteca 'pytorch-grad-cam' o 'Yolov8 Explainer' en GitHub, por ejemplo, proporciona implementaciones avanzadas que soportan múltiples algoritmos para CNNs, Transformers de Visión, y tareas como clasificación, detección de objetos y segmentación[43]. Frameworks como Keras también ofrecen herramientas para visualizar activaciones de clase en modelos de clasificación de imágenes [46].

Las variantes de mapas de calor, como Grad-CAM, Grad-CAM++, EigenCAM y LayerCAM, han demostrado ser herramientas que ayudan a proporcionar explicabilidad a modelos como YOLOv8. Dado que estos son modelos de detección de objetos en tiempo real, comprender qué partes de la imagen influyen en cada detección es fundamental para validar y mejorar su rendimiento [47].

En resumen, las técnicas de generación de mapas de calor representan un avance en la interpretabilidad de modelos de visión por computadora. Cada variante aborda limitaciones específicas, desde la precisión en la localización hasta la estabilidad en la generación de mapas de activación. A pesar de sus restricciones individuales, estas técnicas proporcionan herramientas para mejorar la transparencia, reducir sesgos y aumentar la confianza en las predicciones de modelos de inteligencia artificial.

3.1.2 PointNet y PointCloud Saliency Maps

PointNet es una arquitectura de redes neuronales diseñada para procesar nubes de puntos 3D. Las nubes de puntos son conjuntos de puntos en espacio 3D que representan objetos o escenas. PointNet utiliza conjuntos de puntos sin orden, a diferencia de los métodos tradicionales basados en matrices de vóxeles (imágenes 3D) reduciendo el volumen de datos. Esta arquitectura permite realizar tareas de clasificación, segmentación de partes y segmentación semántica.

Las técnicas de explicabilidad incluyen la visualización de funciones de puntos, la generación de mapas de atención o saliencia. Estas técnicas permiten entender cómo el modelo representa diferentes clases y qué información de la nube de puntos está utilizando para tomar decisiones. El método PointCloud Saliency Maps [48] se describirá en mayor detalle.

3.1.2.1 Fundamentos de PointCloud Saliency Maps [48]

El objetivo del método es crear mapas de saliencia que identifiquen las partes más importantes de una nube de puntos para una tarea específica como clasificación o segmentación.

La saliencia de un punto de la nube se puede definir como la contribución al resultado de la tarea (por ejemplo, qué tan relevante es un punto para clasificar correctamente un objeto). Se hace por tanto una proyección del concepto de saliencia del contexto imagen-píxel al contexto nube-punto.

La implementación original del método utiliza la red DGCNN, que está mayormente basada en PointNet y permite realizar tareas de clasificación, segmentación semánticas y segmentación de partes.

Se ha encontrado otra implementación [49], que introduce algunos cambios en el algoritmo original y utiliza la arquitectura PointNet en lugar de DGCNN. En este documento se describe la implementación original.

DGCNN introduce una operación novedosa en PointNet, EdgeConv, que captura estructuras geométricas locales manteniendo la invariancia de la red a permutaciones de puntos. Específicamente, genera características que describen relaciones entre puntos vecinos creando un grafo de grupos locales y aplicando operaciones sobre aristas que conectan pares de puntos vecinos.

3.1.2.2 Desplazamiento de Puntos [48]

El método se basa en la técnica de Desplazamiento de Puntos (Point Shifting), en lugar de la técnica de la Eliminación de Puntos (Point Dropping) usada en otros métodos.

Point Dropping es una técnica que evalúa la validez de un mapa de saliencia. Si el mapa es preciso, entonces eliminar los puntos con mayor puntuación de saliencia mejorará el desempeño en la tarea de clasificación.

Point Shifting, por otra parte, se basa no en eliminar los puntos con mejor puntuación, sino en desplazarlos hacia el centro de la nube. La lógica detrás es que los puntos externos de una nube de puntos son los que más influyen en la clasificación o segmentación, ya que contienen la información sobre la forma de los objetos. En cambio, los puntos cercanos al centro tienen un impacto casi nulo en el desempeño del reconocimiento. Concretamente, los puntos externos se refieren a aquellos que permanecen en su posición original, sin desplazarse hacia el centro. Por lo tanto, eliminar un punto genera un efecto similar al de moverlo hacia el centro, al reducir su influencia en el resultado de la clasificación. Una justificación más precisa de esta idea es que, tras la traslación de coordenadas, los puntos centrales de todas las nubes de puntos quedan en la misma posición, lo que hace que su contribución al reconocimiento sea insignificante.

3.1.2.3 Cálculo del Mapa de Saliencia [48]

Con base en lo descrito en 3.1.2.2, se aproxima la contribución de un punto al gradiente de pérdida (loss), es decir la diferencia entre las pérdidas en la predicción de dos nubes de puntos incluyendo o excluyendo el punto que es objeto de la operación de desplazamiento. Los puntos se representan el un Sistema de Coordenadas Esférico, donde cada punto es representado como (r, ψ, φ), siendo r la distancia al centro de la esfera. Por su parte, ψ, φ son los ángulos con respecto al origen. En este sistema de coordenadas, desplazar un punto hacia el centro con un desplazamiento δ incrementa la pérdida L por $-\frac{\partial L}{\partial r}\delta$.

De forma que se calcula la contribución de un punto mediante el gradiente negativo de la pérdida L con respecto a r, es decir, $-\frac{\partial L}{\partial r}$. Para calcular $\frac{\partial L}{\partial r}$ para una cierta nube de puntos, se utiliza la mediana de los puntos de la nube como centro del sistema de coordenadas esférico, denotado como denotado como x_c:

$$x_{cj} = mediana(xij|xi \in X)(j = 1,2,3)$$

Ecuación 3. Centro del sistema de coordenadas

En consecuencia, $\frac{\partial L}{\partial r}$ se puede calcular mediante los gradientes en las coordenadas ortogonales originales como:

$$\frac{\partial L}{\partial r_i} = \sum_{j=1}^{3} \frac{\partial L}{\partial x_{ij}} \cdot \frac{(x_{ij} - x_{cj})}{r_i}$$

Ecuación 4. Gradiente de la pérdida L con respecto a r

donde $r_i = \sqrt{\sum_{j=1}^{3} \left(x_{i_j} - x_{c_j}\right)^2}$. Se aplica también un cambio de variable $\rho_i = r_i^{-\alpha} \quad (\alpha > 0)$ escalando por α la nube de puntos para permitir mayor flexibilidad en la construcción del mapa de saliencia. El gradiente de L con respecto a ρ_i se calcula

$$\frac{\partial \mathcal{L}}{\partial \rho_i} = -\frac{1}{\alpha} \frac{\partial \mathcal{L}}{\partial r_i} r_i^{1+\alpha}$$

Ecuación 5. El gradiente de L con respecto a ρ_i

Siendo $\frac{\delta \rho}{\delta r}$ el paso diferencial sobre ρ/r, y dado que se da $\delta \rho = -\alpha r^{-(\alpha+1)} \delta r$, desplazar un punto por $-\delta r$ equivale a desplazar por $\delta \rho$. De modo que ignorando el factor positivo $\alpha r^{-(\alpha+1)}$, se aproxima la variación de la pérdida por $\frac{\partial \mathcal{L}}{\partial \rho} \delta \rho$. De forma que se mide la contribución de un punto x_i como $\frac{\partial \mathcal{L}}{\partial \rho_i}$. Con lo que la saliencia de x_i viene dada por

$$s_i = -\frac{1}{\alpha} \frac{\partial L}{\partial r_i} r_i^{1+\alpha}$$

Ecuación 6. Saliencia de x_i

3.1.2.4Ventajas y desventajas del método

Las ventajas que se han identificado en el uso de esta técnica son:

- **Generalidad**. Puede aplicarse a diferentes arquitecturas de redes neuronales, como PointNet y DGCNN.
- **Eficacia respecto a otras técnicas**: El método basado en Desplazamiento de Puntos ha conseguido mejores resultados que el basado en eliminación de puntos.
- **Disponibilidad de recursos**. Se han encontrado diferentes implementaciones del método, frente a otros métodos de los que no hay implementaciones conocidas para las arquitecturas más utilizadas.

Algunas desventajas de la técnica son:

- **Costo computacional**: El método basado en desplazamientos puede ser computacionalmente costoso, ya que implica modificar y reevaluar puntos o regiones de la nube repetidamente.

- **Falta de evaluación estandarizada**: No hay métricas universales para evaluar objetivamente la calidad de los mapas de saliencia en nubes de puntos, lo que dificulta la comparación con otros métodos.

3.2 Cápsula 3: Datos Tabulares y Series Temporales

En el análisis de datos tabulares y series temporales, la XAI juega un papel fundamental debido a la complejidad inherente de estas estructuras. Los datos tabulares, organizados en filas y columnas, representan relaciones entre múltiples. Por otro lado, las series temporales consisten en observaciones de una o más variables registradas en un intervalo de tiempo continuo o discreto. Estas pueden clasificarse como series temporales univariables (UTS por sus siglas en inglés), cuando representan una única variable dependiente del tiempo (por ejemplo, la temperatura diaria de una ciudad), o multivariables (MTS), cuando involucran múltiples variables interrelacionadas que evolucionan con el tiempo [50].

Para analizar este tipo de datos, se utilizan modelos como las redes neuronales convolucionales (CNN), las redes neuronales de memoria a largo y corto plazo (LSTM) y los transformadores. Las CNN son particularmente útiles en series temporales para detectar patrones locales o características específicas dentro de ventanas temporales. Las LSTM, diseñadas para manejar dependencias a largo plazo, son ideales para capturar relaciones temporales complejas en secuencias extensas. Finalmente, los T ransformers, con su mecanismo de atención, sobresalen en el manejo de series multivariables y en la identificación de interacciones importantes entre diferentes variables y a lo largo de múltiples escalas de tiempo.

Sin embargo, la complejidad de estos modelos plantea desafíos de interpretabilidad, ya que a menudo actúan como "cajas negras". Aquí es donde la explicabilidad resulta esencial, permitiendo a los usuarios entender qué variables, ventanas temporales o patrones influyen más en las predicciones. Los métodos de explicabilidad aplicables dependen en algunos casos del modelo utilizado, y existen técnicas específicas para el caso concreto de las Series Temporales.

Un enfoque destacado en el ámbito de la explicabilidad para series temporales es TSInterpret [51], un *framework* que proporciona un conjunto de herramientas integradas que facilitan la aplicación de métodos de explicabilidad específicamente diseñados para modelos de clasificación de series temporales. TSInterpret es compatible con una amplia gama de arquitecturas modernas, como redes LSTM, CNN y transformadores, ofreciendo soporte para la interpretación de modelos que trabajan tanto con series univariables como multivariables. Soporta las librerías *scikit-learn*, *tensorflow* y *pytorch*.

3.2.1 Redes Neuronales Convolucionales (CNN)

Las redes neuronales convolucionales (CNN) son modelos de aprendizaje profundo diseñados para procesar datos con una estructura de cuadrícula, como las imágenes. En el contexto del análisis de series temporales, las CNN pueden capturar patrones espaciales y temporales mediante la aplicación de filtros convolucionales que detectan características locales en los datos [52]. Su capacidad para manejar grandes volúmenes

de datos y extraer características relevantes las hace ideales para tareas de predicción y clasificación en series temporales [53].

Para las redes neuronales convolucionales (CNN), se utilizan comúnmente técnicas como la visualización de filtros y mapas de características, así como mapas de saliencia y mapas de activación de clase (CAM) [54]. Estas técnicas destacan las características más importantes que el modelo ha aprendido en diferentes capas, proporcionando una visión clara de cómo se procesan las imágenes. En [55] utilizan Grad-CAM (ver 3.1.1.1) para la explicabilidad de este tipo de redes. Sin embargo, existe dCAM [56], que se explica a continuación y funciona especialmente bien es este caso de uso.

3.2.1.1dCAM

Pese a que las CNNs tienen buen desempleño en tareas sobre series temporales, los métodos de explicabilidad más habitualmente usados en estas redes no proporcionan resultados tan satisfactorios en el caso específico de las series temporales multivariable. El algoritmo dCAM [56] soluciona este hecho teniendo en cuenta la información discriminante temporal y dimensional a la hora de obtener CAMs. El repositiorio del método incluye implementaciones para arquitecturas estado del arte basadas en CNN, tales como ResNet o InceptionTime.

Fundamentos de dCAM

Para entender el beneficio de usar dCAM en lugar de las técnicas tradicionales para generar CAMs, hay que entender que las MTS, a diferencia de las UTS, tienen dos dimensiones, la temporal y la dimensión de variables:

- La dimesión temporal representa la ordenación de los datos en el tiempo. Las observaciones se realizan en puntos de tiempo sucesivos, que pueden estar espaciados uniformemente (por ejemplo, diario, por hora o anualmente) o de manera irregular.
- La dimensión de variables es la que se extiende por las diferentes variables (o características) que se observan simultáneamente en cada punto de tiempo. Estas variables pueden ser independientes o dependientes, presentando en muchos casos interrelaciones que pueden ser capturadas por los modelos de Deep Learning.

Cada una de estas dos dimensiones puede presentar características discriminantes:

Las características discriminantes son a tributos o propiedades derivadas de los datos que son especialmente útiles para distinguir entre diferentes clases, categorías o condiciones. Estas características se emplean en tareas de clasificación o agrupamiento y tienen como objetivo capturar patrones en la serie temporal que sean más relevantes para las diferencias entre los grupos objetivo.

Los métodos tradicionales para calcular CAMs resaltan correctamente la información más relevante a lo largo de la dimensión temporal, pero no a lo largo de la dimensión de variables. Por tanto, aunque suficientes para las UTS, no lo son para las MTS.

Como respuesta a esta limitación, dCAM proporciona una forma de obtener CAMs multivariables señalando características discriminantes en cada dimensión. Para hacerlo posible, se debe utilizar en combinación con arquitecturas de red compatibles, como dCNN, dResNet o dInceptionTime, basadas en las arquitecturas originales de estas redes.

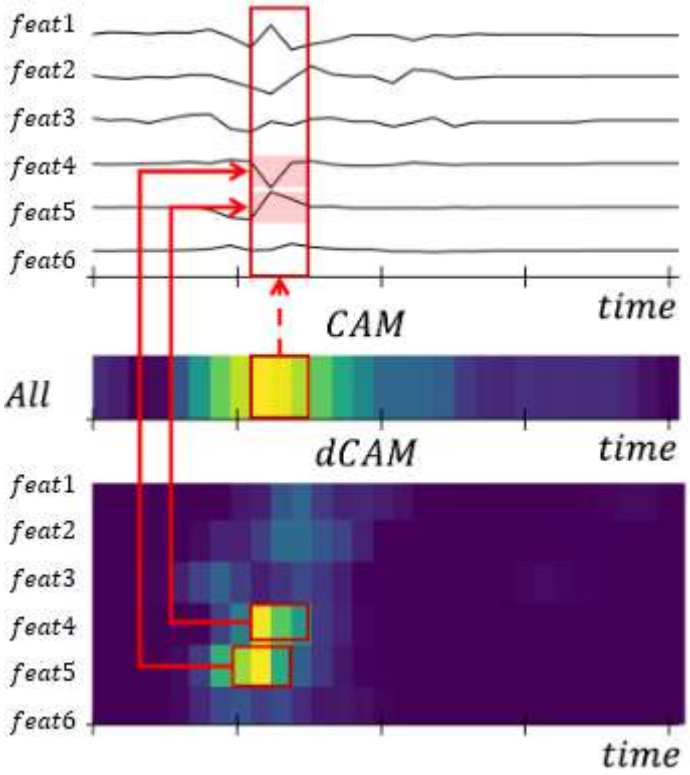

Figura 3. Comparativa entre un mapa de calor generado con CAM (no discriminante respecto a características) y uno generado con dCAM (discriminante en las dos dimensiones). [56]

Arquitecturas de red atentas a dimensiones

La debilidad de las técnicas tradicionales de CAMs ante la multidimensionalidad se origina en que las arquitecturas clásicas de CNNS mezclan las dimensiones en la primera capa convolucional. De forma que el CAM es una serie de datos univariable y permite saber qué dimensión es la discriminante. Esto se soluciona usando una CNN de dos dimensiones, con dos kernels que recorren cada una de las dos dimensiones. Sin embargo, dado que cada kernel recibe como entrada cada dimensión independientemente, una arquitectura así no permite aprender características que dependen de múltiples dimensiones.

Arquitectura dCNN

La arquitectura dCNN (y de manera similar las arquitecturas dResNet y dInceptionTime) combina lo mejor de ambos mundos, transformando el input en un cubo, donde cada fila contiene una cierta combinación de las tres dimensiones. Esto permite aprender características en múltiples dimensiones simultáneamente. El CAM resultante es una MTS, donde cada fila corresponde a una cierta combinación de las dimensiones. Esto se traduce en el siguiente formato de entrada de los datos:

$$C(T) = \begin{pmatrix} T^{(D-1)} & T^{(0)} & \dots & T^{(D-3)} & T^{(D-2)} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ T^{(1)} & T^{(2)} & \dots & T^{(D-1)} & T^{(0)} \\ T^{(0)} & T^{(1)} & \dots & T^{(D-2)} & T^{(D-1)} \end{pmatrix}$$

Cada fila y columna contiene todas las dimensiones, y una dimensión $T^{(i)}$ no está nunca en la misma posición en las filas.

3.2.2 Redes Neuronales Recurrentes (RNN)

Las redes neuronales recurrentes (RNN, por sus siglas en inglés) son un tipo de red neuronal diseñada para procesar datos secuenciales, como series temporales o texto. A diferencia de las redes neuronales tradicionales, las RNN tienen conexiones recurrentes que les permiten mantener información de pasos previos, lo que es útil para modelar dependencias temporales o contextuales.

Los siguientes son algunos de los tipos más relevantes de RNN:

- **RNN estándar**: La versión básica, que utiliza una función de activación recurrente para mantener un estado oculto que captura información de la entrada anterior. Estas redes sufren el problema del desvanecimiento del gradiente. Este consiste en que, si el estado previo que influencia la predicción actual no pertenece al pasado más reciente, la RNN pueden no predecir adecuadamente. En definitiva, estas redes pueden ser inefectivas a la hora de aprender interdependencias temporales a largo plazo entre características.
- **Long Short-Term Memory (LSTM)**: Una mejora sobre las RNN estándar. Utilizan una arquitectura basada en "celdas" que incluyen compuertas de entrada, olvido y salida. Esta arquitectura controla la información que se almacena en el “estado de la celda”, vector que conserva la información más relevante de estados previos, y no solo del último estado. Esto soluciona el problema del desvanecimiento del gradiente y permite a estas redes aprender dependencias a largo plazo.
- **Gated Recurrent Unit (GRU)**: Similar a las LSTM, pero con una estructura más simple. Utilizan estados ocultos en lugar del “estado de la celda”. Tienen dos compuertas principales (actualización y reinicio), frente de las tres de las LSTM. Esto las hace más eficientes computacionalmente.
- **Bidireccionales (BiRNN)**: Procesan la secuencia en ambas direcciones (hacia adelante y hacia atrás) para capturar información tanto del pasado como del futuro en la serie de datos.

Los métodos de explicabilidad para redes recurrentes incluyen desde métodos comunes a múltiples arquitecturas de RNN hasta métodos específicos de cada arquitectura.

3.2.2.1 RETAIN

REverse Time AttentIoN (RETAIN) [57] es un método de explicabilidad para múltiples arquitecturas de RNN, como LSTM y GRU, que modifica la arquitectura de estas para conseguir explicabilidad local.

RETAIN se basa en el paradigma de red basada en atención, que, como parte del proceso de generar la predicción, utiliza pesos de atención para dar diferente importancia a cada elemento de cada vector de entrada. Esto se traduce en atención temporal-discriminante y característica-discriminante.

Concretamente, se utilizan dos conjuntos de pesos, uno para atención temporal-discriminante, y otro para atención característica-discriminante. Para el vector de entrada x_i, representado por el *embedding* $v_i = W_{\text{emb}} x_i$, los escalares $\alpha 1, \dots, \alpha i$ son los pesos de atención temporal-discriminante que controlan la influencia de cada *embedding* $v_1, \dots, v_i$. Por otro lado, los vectores $\beta 1, \dots, \beta i$ son los pesoso de atención característica-

discriminante que controlan la influencia de cada coordenada del *embedding* $v_{1,1}, v_{1,2}, \dots, v_{1,m}, \dots, v_{i,1}, v_{i,2}, \dots, v_{i,m}$.

Se usan dos redes RNNs, RNN_α y RNN_β, para generar ambos conjuntos de pesos.

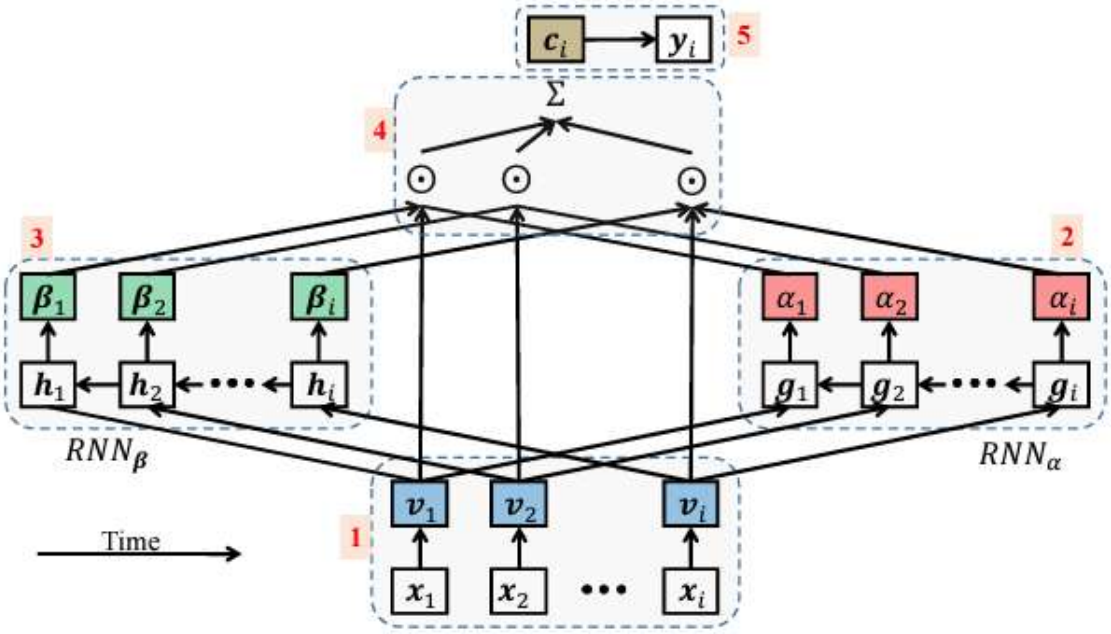

Figura 4. Arquitectura de RETAIN [57]

3.2.2.2Explicabilidad sobre LSTMs

Las redes neuronales recurrentes (RNN) LSTM (*Long Short-Term Memory*) están diseñadas para manejar dependencias a largo plazo en datos secuenciales. Las LSTM superan las limitaciones de las RNN tradicionales al mantener información relevante durante largos periodos, lo que es crucial para el análisis de series temporales [44]. Estas redes son especialmente efectivas en la predicción de secuencias y en la modelización de datos temporales complejos, como los datos financieros y meteorológicos [50].

En cuanto a los modelos LSTM, que son populares en el procesamiento de secuencias temporales, las técnicas de explicabilidad incluyen mecanismos de atención, LIME y SHAP [51].

Añadir mecanismos de atención sobre este tipo de redes es una práctica habitual para obtener interpretabilidad en LSTMs. En [58] los autores investigan la estructura de las redes neuronales LSTM con mecanismos de atención para aprender estados ocultos específicos de cada variable, con el objetivo de capturar diferentes dinámicas en series temporales multivariantes y distinguir la contribución de cada variable a la predicción.

Siguiendo esta línea, en [59] se propone una aproximación que incorpora un mecanismo de atención en redes neuronales LSTM para mejorar la interpretabilidad y el rendimiento en tareas de predicción complejas. El enfoque se utiliza en el ámbito de la salud, donde el modelo LSTM con atención permite identificar qué características de los datos influyen en las decisiones del modelo a lo largo del tiempo.

Los mecanismos de atención permiten visualizar qué partes de la secuencia son más influyentes en las predicciones del modelo, mientras que métodos como LIME y SHAP ayudan a descomponer las predicciones en contribuciones individuales de cada característica.

Mecanismo de atención basado en LSTM [60]

Se trata de una arquitectura ensemble LSTM con una capa CNN que proporciona interpretabilidad local tanto en la dimensión temporal como en la dimensión de variables. La atención temporal se consigue con un mecanismo de atención integrado en la arquitectura basada en LSTM que genera las predicciones, mientras que la atención que discrimina entre características se consigue mediante una segunda arquitectura de propagación hacia atrás.

Primero, la serie temporal de tamaño n se aumenta mediante una capa convolucional que genera nuevas series temporales mediante transformaciones de la serie original. La nueva obtenida se introduce en una capa LSTM que genera un estado oculto h_i para cada fila de la subserie X de tamaño w (la ventana temporal). Los estados ocultos pasan a una capa densa que aprende el contexto. A partir de este contexto se generan los pesos de atención $\alpha = \{\alpha_{t-W}, \alpha_{t-W+1}, \dots, \alpha_{t-1}\}$ que proporcionan la interpretabilidad temporal. Por último, una capa dense genera la predicción.

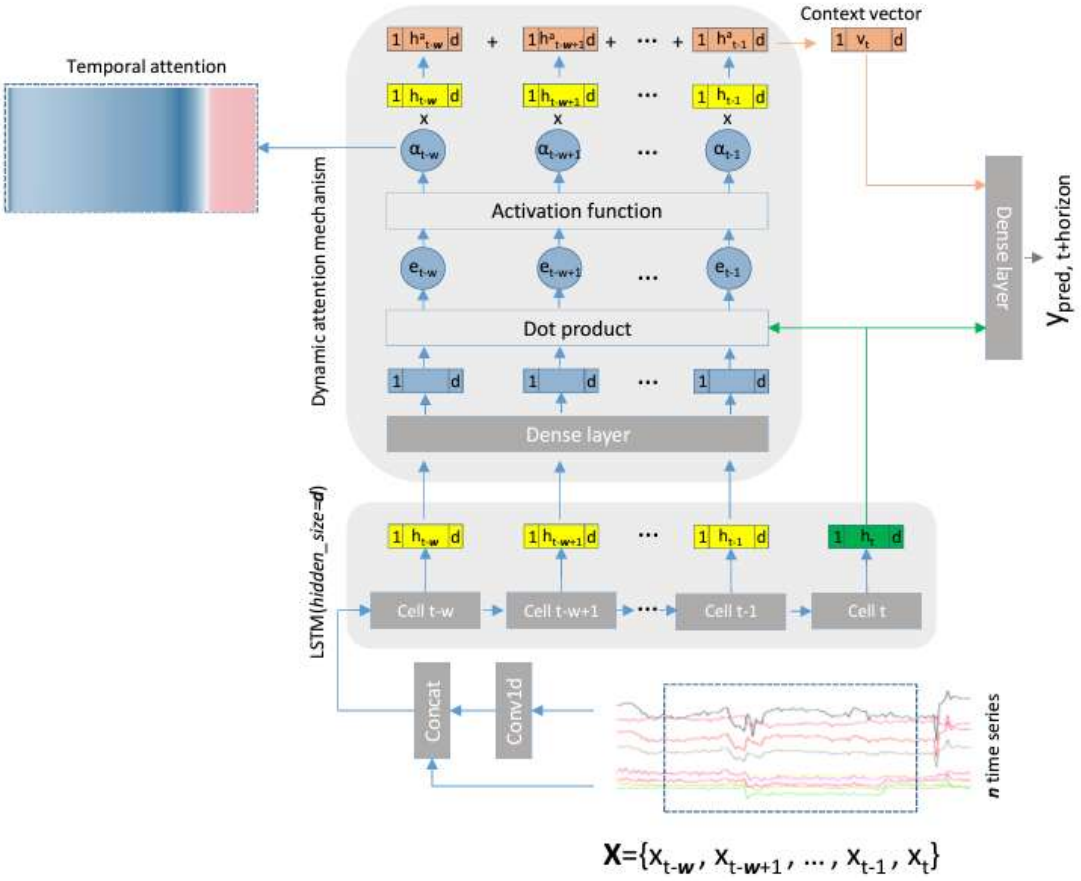

Figura 5. Arquitectura basada en LSTM y CNN que genera los pesos de atención y las predicciones [60]

Por otra parte, se aplica la propagación hacia atrás en cada salto temporal t a partir cada estado oculto h_i^a modificado por el mecanismo de atención y el vector de entrada x_i para todo i en $[t-w, \dots, t]$. De esta forma se resalta la característica del vector x_i que induce un gradiente en el estado oculto h_i^a.

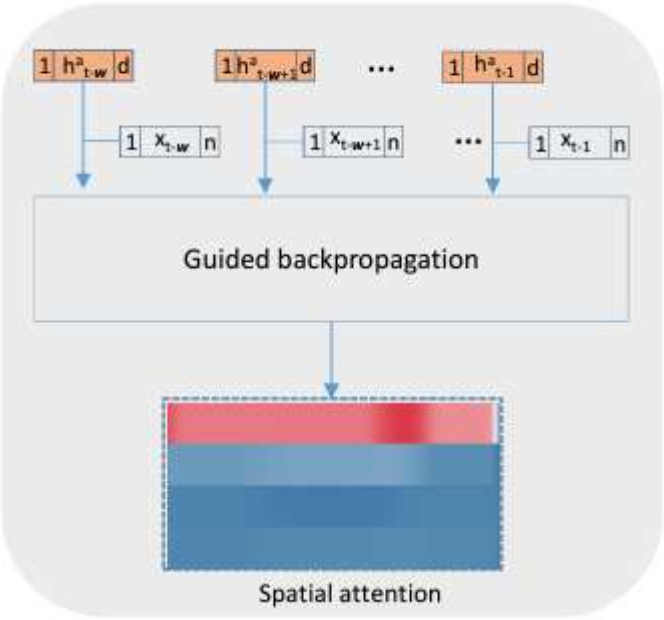

Figura 6. Mecanismo basado en propagación hacia atrás que genera la atención en la dimensión de variables ***[60]***

3.2.3 Transformers

Los Transformers son una arquitectura de red neuronal que ha revolucionado el procesamiento de lenguaje natural y se está aplicando cada vez más al análisis de series temporales. A diferencia de las RNN, los Transformers utilizan mecanismos de atención que permiten procesar secuencias de datos en paralelo, mejorando la eficiencia y la capacidad de capturar relaciones a largo plazo [61]. Esta arquitectura es particularmente útil para tareas que requieren la integración de información de múltiples fuentes y la identificación de patrones complejos en los datos temporales [62].

Para modelos Transformers, ampliamente utilizados en procesamiento de lenguaje natural, visión por computador y series temporales, las técnicas de explicabilidad más efectivas incluyen principalmente la visualización de la atención [63], aunque también se han empleado técnicas como la propagación de relevancia capa por capa (LRP) [64]. La visualización de la atención facilita la interpretación al destacar qué elementos de la secuencia son más relevantes para las predicciones del modelo. Por su parte, la propagación de relevancia capa por capa (LRP) permite descomponer la decisión del modelo para entender cómo se distribuyen las contribuciones a través de las diferentes capas del Transformer, proporcionando una interpretación más detallada y jerárquica de sus decisiones.

3.2.3.1DFSTrans

El modelo *Diagnostic Spatio-temporal Transformer* (DFSTrans) [65] emplea un enfoque de codificación posicional basado en la Transformada Discreta de Fourier (DFT) sobre los embeddings extraídos por redes convolucionales para series multivariantes. Después aplica Transformers espaciotemporales que garantizan la fidelidad en la representación temporal, evitando sesgos hacia dependencias de largo plazo y mejorando la captura de patrones de corto y mediano plazo. Para la explicabilidad, utiliza matrices de atención espacial y temporal que cuantifican las relaciones entre las dimensiones espaciales y los segmentos temporales. Además, define métricas específicas, como las puntuaciones de relevancia espacial y temporal, que permiten identificar qué variables y momentos tienen mayor influencia en la detección de anomalías. Estas métricas son particularmente útiles para proporcionar interpretaciones locales y globales de los patrones anómalos detectados.

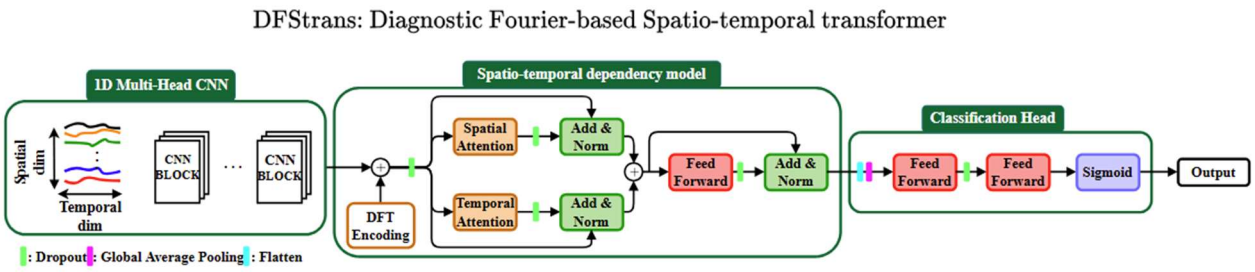

Figura 7. Arquitectura de DFSTrans ***[65]***

3.2.3.2Explicabilidad sobre AutoFormer

Aparte del diagnóstico de anomalías, la explicabilidad también ha sido aplicada en modelos de Transformers diseñados para la predicción de series temporales, como es AutoFormer.

En [66] se introduce un marco basado en Modelos de Cuello de Botella de Conceptos (*Concept Bottleneck Models*) para mejorar la interpretabilidad de los Transformers aplicados a series temporales. El enfoque modifica el objetivo de entrenamiento para fomentar que el modelo desarrolle representaciones similares a conceptos interpretables predefinidos, como características temporales y modelos autorregresivos simples.

Para medir esta alineación entre las representaciones del modelo y los conceptos interpretables, se emplea la Alineación de Núcleo Centrado (*Centered Kernel Alignment*). Este marco ha sido aplicado al modelo Autoformer, demostrando que es posible mantener el rendimiento predictivo mientras se mejora significativamente la interpretabilidad. Además, permite realizar intervenciones locales en escenarios específicos sin necesidad de reentrenamiento, lo que lo hace especialmente útil para la toma de decisiones en aplicaciones prácticas de predicción de series temporales.

3.2.3.3TFT

El *Temporal Fusion Transformer* (TFT) [67] es una arquitectura basada en atención diseñada para la predicción multihorizonte en series temporales. Combina capas recurrentes para el procesamiento local y capas de autoatención interpretables para capturar dependencias a largo plazo. TFT incorpora componentes especializados para

seleccionar características relevantes y utiliza mecanismos de compuerta para suprimir elementos innecesarios, lo que permite un rendimiento elevado en diversos escenarios. Desde una perspectiva de explicabilidad, el TFT destaca por proporcionar interpretaciones detalladas a través de análisis de importancia de características y dinámicas temporales. Esto permite a los usuarios entender qué variables y patrones específicos han influido más en las predicciones del modelo.

4 Conclusiones

En esta sección, se resumen y sintetizan las conclusiones alcanzadas a lo largo del documento. Como se ha podido observar a lo largo de todo el texto, existen diversas técnicas de explicabilidad. Se utilizarán unas u otras dependiendo de, principalmente, el modelo que se pretenda explicar y los resultados que se deseen obtener.

Dentro de los casos de usos aplicables en nuestro contexto encontramos dos capsulas (la Cápsula 2 y la Cápsula 3) a las que se quiere aplicar explicabilidad: la de visión artificial y la que trata datos tabulares y series temporales. Ambas capsulas tienen naturalezas diferentes y, por tanto, utilizan modelos de Inteligencia Artificial distintos. Es por este motivo por el que se han propuesto varias técnicas de explicabilidad para aplicarlas a los distintos modelos que se pretenden usar. Para la cápsula de visión existen dos modelos diferentes YOLO y PointNet, mientras que la cápsula 3 puede utilizar Redes Neuronales Convolucionales (CNN), Redes Neuronales Recurrentes (RNN) y Transformers.

En primer lugar, se ha visto YOLO. Si bien es cierto, que existen diversos métodos que abarcan la explicabilidad de esta red, el más usado y extenso en la literatura es Grad-CAM. Grad-CAM es una técnica de explicabilidad que genera un mapa de calor que permite al usuario observar qué zonas de la imagen han sido más influyentes a la hora de obtener los resultados de YOLO. Esta técnica, aparte de ser conocida y ampliamente utilizada por la comunidad, proporciona una solución visual fácilmente entendible para el usuario.

Seguidamente, PointNet constituye una arquitectura de redes neuronales para procesar nubes de puntos en 3D. Un método de explicabilidad equiparable a lo que es Grad-CAM para YOLO es PointCluod Saliency Maps que permite entender cómo el modelo representa las distintas clases y qué información de la nube de puntos está utilizando en su toma de decisiones.

En cuanto a las opciones para la tercera cápsula, la primera presentada son las Redes Neuronales Convolucionales (CNN). Estas redes, no se utilizan únicamente con el propósito que aquí se presenta, sino que son las más utilizadas en el campo de la visión. Por ello, pueden usarse técnicas como las explicadas para el apartado anterior, que generen mapas de calor. En este caso, se propone dCAM, que funciona especialmente bien en este caso de uso, donde se pretende utilizar la CNN con series temporales multivariables.

Como segunda opción de modelo para la cápsula 3 están las Redes Neuronales Recurrentes (RNN). Existen de varios tipos como: las RNN estándar, Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU) o las Bidireccionales (BiRNN). Entre las técnicas de explicabilidad aplicables a estas redes destacan *REverse Time AttentIoN* (RETAIN), LIME y SHAP. Por su parte, RETAIN se utiliza para dar explicabilidad local a diferentes tipos de redes RNN como LSTM o GRU. Por otro lado, LIME y SHAP son técnicas utilizadas para aportar aplicabilidad tanto global como local a una amplia gama de modelos, tanto de RNN como de cualquier otra naturaleza.

Finalmente, la última opción de modelos de inteligencia artificial para la tercera cápsula son los transformers. Estos suponen una arquitectura de red revolucionaria que permite procesar secuencias de datos, lo que mejora la eficiencia y la capacidad de capturar relaciones a largo plazo. Para este tipo de modelos, las técnicas de explicabilidad más

eficaces es, principalmente, la visualización de la atención. Sin embargo, existen otras técnicas que han resultado efectivas como la relevancia por capa (LRP). Mientras que la primera facilita la interpretación destacando los elementos más relevantes para la predicción, la segunda descompone la decisión del modelo y proporciona una interpretación más detallada y jerárquica.

5 Referencias

[1] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi, “A survey of methods for explaining black box models,” *ACM Comput Surv*, vol. 51, no. 5, Sep. 2018, doi: 10.1145/3236009.

[2] S. Ali *et al.*, “Explainable Artificial Intelligence (XAI): What we know and what is left to attain Trustworthy Artificial Intelligence,” *Information Fusion*, vol. 99, p. 101805, Nov. 2023, doi: 10.1016/J.INFFUS.2023.101805.

[3] L. Longo *et al.*, “Explainable Artificial Intelligence (XAI) 2.0: A manifesto of open challenges and interdisciplinary research directions,” Jun. 01, 2024, *Elsevier B.V.* doi: 10.1016/j.inffus.2024.102301.

[4] Z. Zhang, H. Al Hamadi, E. Damiani, C. Y. Yeun, and F. Taher, “Explainable Artificial Intelligence Applications in Cyber Security: State-of-the-Art in Research.”

[5] Z. Chen, F. Xiao, F. Guo, and J. Yan, “Interpretable machine learning for building energy management: A state-of-the-art review,” Feb. 01, 2023, *Elsevier Ltd*. doi: 10.1016/j.adapen.2023.100123.

[6] C. Grimsley, E. Mayfield, and J. R. S. Bursten, “Why Attention is Not Explanation: Surgical Intervention and Causal Reasoning about Neural Models,” 2020.

[7] Q. Lyu, M. Apidianaki, and C. Callison-Burch, “Towards Faithful Model Explanation in NLP: A Survey,” 2024, doi: 10.1162/coli.

[8] A. Smart and A. Kasirzadeh, “Beyond Model Interpretability: Socio-Structural Explanations in Machine Learning,” Sep. 2024, doi: 10.1007/s00146-024-02056-1.

[9] “¿Qué es la IA explicable?” Accessed: Dec. 10, 2024. [Online]. Available: https://www.ibm.com/es-es/topics/explainable-ai

[10] “Explicabilidad.” Accessed: Dec. 10, 2024. [Online]. Available: https://www.educaopen.com/digital-lab/metaterminos/e/explicabilidad

[11] A. Bibal *et al.*, “Is Attention Explanation? An Introduction to the Debate,” in *60th Annual Meeting of the Association for Computational Linguistics*, Long Papers, May 2022, pp. 3889–3900.

[12] K. Fiok, F. V. Farahani, W. Karwowski, and T. Ahram, “Explainable artificial intelligence for education and training,” *Journal of Defense Modeling and Simulation*, vol. 19, no. 2, pp. 133–144, Apr. 2022, doi: 10.1177/15485129211028651.

[13] SHREERAJ, “Unveiling the Spectrum of Explainable AI: A Deep Dive into XAI Techniques.” Accessed: Dec. 10, 2024. [Online]. Available: https://medium.com/@shreeraj260405/unveiling-the-spectrum-of-explainable-ai-a-deep-dive-into-xai-techniques-1ccfa856ac96

[14] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, “Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization,” Oct. 2016, doi: 10.1007/s11263-019-01228-7.

[15] A. Chattopadhyay, A. Sarkar, P. Howlader, and V. N. Balasubramanian, “Grad-CAM++: Improved Visual Explanations for Deep Convolutional Networks,” Oct. 2017, doi: 10.1109/WACV.2018.00097.

[16] R. L. Draelos and L. Carin, “Use HiResCAM instead of Grad-CAM for faithful explanations of convolutional neural networks,” Nov. 2020, [Online]. Available: http://arxiv.org/abs/2011.08891

[17] M. B. Muhammad and M. Yeasin, “Eigen-CAM: Class Activation Map using Principal Components,” *IEE*, 2020.

[18] J. H. Friedman, “Greedy Function Approximation: A Gradient Boosting Machine,” *The Annals of Statistics*, vol. 29, no. 5, pp. 1189–1232, 2001, [Online]. Available: http://www.jstor.org/stable/2699986

[19] S. Krishna, T. Han, A. Gu, S. Wu, S. Jabbari, and H. Lakkaraju, “The Disagreement Problem in Explainable Machine Learning: A Practitioner’s Perspective,” Feb. 2022, [Online]. Available: http://arxiv.org/abs/2202.01602

[20] A. Jacovi and Y. Goldberg, “Towards Faithfully Interpretable NLP Systems: How should we define and evaluate faithfulness?,” Apr. 2020, [Online]. Available: http://arxiv.org/abs/2004.03685

[21] B. Kim, R. Khanna, and O. Koyejo, “Examples are not Enough, Learn to Criticize! Criticism for Interpretability.”

[22] K. S. Gurumoorthy, A. Dhurandhar, G. Cecchi, and C. Aggarwal, “Efficient Data Representation by Selecting Prototypes with Importance Weights,” Jul. 2017, [Online]. Available: http://arxiv.org/abs/1707.01212

[23] Y. He, S. Cao, Y. Shi, Q. Chen, K. Xu, and N. Cao, “Leveraging Foundation Models for Crafting Narrative Visualization: A Survey,” Jan. 2024, [Online]. Available: http://arxiv.org/abs/2401.14010

[24] P. Parada Torralba, “Qué es la Explicabilidad en la Inteligencia Artificial (XAI) y cómo funciona.” Accessed: Dec. 10, 2024. [Online]. Available: https://www.iebschool.com/blog/que-es-la-explicabilidad-en-la-inteligencia-artificial-xai-y-como-funciona-inteligencia-artificial/

[25] M. Sundararajan, A. Taly, and Q. Yan, “Axiomatic Attribution for Deep Networks,” 2017.

[26] D. Alvarez-Melis and T. S. Jaakkola, “On the Robustness of Interpretability Methods,” Jun. 2018, [Online]. Available: http://arxiv.org/abs/1806.08049

[27] F. Doshi-Velez and B. Kim, “Towards A Rigorous Science of Interpretable Machine Learning,” Feb. 2017, [Online]. Available: http://arxiv.org/abs/1702.08608

[28] F. Poursabzi-Sangdeh, D. G. Goldstein, and J. M. Hofman, “Manipulating and measuring model interpretability,” in *Conference on Human Factors in Computing Systems - Proceedings*, Association for Computing Machinery, May 2021. doi: 10.1145/3411764.3445315.

[29] J. Schmidhuber, “Annotated History of Modern AI and Deep Learning,” Dec. 2022, [Online]. Available: http://arxiv.org/abs/2212.11279

[30] D.-Y. Kang, H. P. Duong, and J.-C. Park, “Application of Deep Learning in Dentistry and Implantology,” *The Korean Academy of Oral and Maxillofacial Implantology*, vol. 24, no. 3, pp. 148–181, Sep. 2020, doi: 10.32542/implantology.202015.

[31] A. Vaswani *et al.*, “Attention Is All You Need,” Jun. 2017, [Online]. Available: http://arxiv.org/abs/1706.03762

[32] A. Dosovitskiy *et al.*, “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale,” Oct. 2020, [Online]. Available: http://arxiv.org/abs/2010.11929

[33] S. Minaee, Y. Boykov, F. Porikli, A. Plaza, N. Kehtarnavaz, and D. Terzopoulos, “Image Segmentation Using Deep Learning: A Survey,” Jan. 2020, [Online]. Available: http://arxiv.org/abs/2001.05566

[34] “What is Image Segmentation: The Basics and Key Techniques.” Accessed: Dec. 16, 2024. [Online]. Available: https://mindy-support.com/news-post/what-is-image-segmentation-the-basics-and-key-techniques/

[35] J. Solawetz and Francesco, “What is YOLOv8? A Complete Guide.” Accessed: Dec. 10, 2024. [Online]. Available: https://blog.roboflow.com/what-is-yolov8/

[36] “Ultralytics YOLOv8.” Accessed: Dec. 10, 2024. [Online]. Available: https://docs.ultralytics.com/es/models/yolov8/#can-i-benchmark-yolov8-models-for-performance

[37] “MMYOLO Authors,” 2023. Accessed: Dec. 10, 2024. [Online]. Available: https://www.labelvisor.com/yolov8-vs-mask-r-cnn-in-depth-analysis-and-comparison/

[38] M. Dupont, “YOLOv8 vs Mask R-CNN: In-depth Analysis and Comparison.” Accessed: Dec. 10, 2024. [Online]. Available: https://www.labelvisor.com/yolov8-vs-mask-r-cnn-in-depth-analysis-and-comparison/

[39] “YOLO-NAS vs YOLOV8: A Comparison of Pros and Cons.” Accessed: Dec. 10, 2024. [Online]. Available: https://www.toolify.ai/ai-news/yolonas-vs-yolov8-a-comparison-of-pros-and-cons-2065044

[40] Melanie, “What is the Grad CAM method?” Accessed: Dec. 10, 2024. [Online]. Available: https://datascientest.com/en/what-is-the-grad-cam-method

[41] N. Vishwakarma, “A Guide to Grad-CAM in Deep Learning.” Accessed: Dec. 10, 2024. [Online]. Available: https://www.analyticsvidhya.com/blog/2023/12/grad-cam-in-deep-learning/

[42] K. Dawn, “GradCAM – Enhancing Neural Network Interpretability in the Realm of Explainable AI.” Accessed: Dec. 10, 2024. [Online]. Available: https://learnopencv.com/intro-to-gradcam/

[43] S. Kumar, A. A. Abdelhamid, and Z. Tarek, “Visualizing the Unseen: Exploring GRAD-CAM for Interpreting Convolutional Image Classifiers,” *Journal of Artificial Intelligence and Metaheuristics*, vol. 4, no. 1, pp. 34–42, 2023, doi: 10.54216/JAIM.040104.

[44] “Visualizing Model Insights: A Guide to Grad-CAM in Deep Learning.” Accessed: Dec. 10, 2024. [Online]. Available: https://datadance.ai/deep-learning/visualizing-model-insights-a-guide-to-grad-cam-in-deep-learning/

[45] P.-T. Jiang, C.-B. Zhang, Q. Hou, M.-M. Cheng, and Y. Wei, “LayerCAM: Exploring Hierarchical Class Activation Maps for Localization,” *JOURNAL OF LATEX CLASS FILES*, vol. 14, no. 8, Aug. 2015, [Online]. Available: https://mmcheng.net/layercam/.

[46] J. Gildenblat and contributors, “PyTorch library for CAM methods,” 2021, *GitHub*. Accessed: Dec. 10, 2024. [Online]. Available: https://github.com/jacobgil/pytorch-grad-cam

[47] F. Chollet, “Grad-CAM class activation visualization.”

[48] T. Zheng, C. Chen, J. Yuan, B. Li, and K. Ren, “PointCloud Saliency Maps,” Nov. 2018, [Online]. Available: http://arxiv.org/abs/1812.01687

[49] D. Li, “This is a re-implement version of ‘Point Cloud Saliency Maps’.”

[50] C. Gonzalo, “Series Temporales Multivariantes: Una Profundización en el Análisis Complejo.” Accessed: Dec. 11, 2024. [Online]. Available: https://www.carlosgonzalo.es/analisis-

de-series-temporales/series-temporales-multivariantes-una-profundizacion-en-el-analisis-complejo/

[51] J. Höllig, C. Kulbach, and S. Thoma, “TSInterpret: A unified framework for time series interpretability,” 2022. [Online]. Available: https://arxiv.org/abs/2208.05280

[52] “¿Cómo se puede entrenar y optimizar una CNN con TensorFlow y cuáles son algunas métricas de evaluación comunes para evaluar su desempeño?” Accessed: Dec. 11, 2024. [Online]. Available: https://es.eitca.org/inteligencia-artificial/eitc-ai-dltf-aprendizaje-profundo-con-tensorflow/redes-neuronales-convolucionales-en-tensorflow/redes-neuronales-convolucionales-con-tensorflow/examen-revisar-redes-neuronales-convolucionales-con-tensorflow/c%C3%B3mo-se-puede-entrenar-y-optimizar-una-cnn-usando-tensorflow-y-cu%C3%A1les-son-algunas-m%C3%A9tricas-de-evaluaci%C3%B3n-comunes-para-evaluar-su-desempe%C3%B1o/

[53] E. Roch Moraguez, “Redes Neuronales Recurrentes LSTM: Predicción de Series Temporales.” Accessed: Dec. 11, 2024. [Online]. Available: https://lovtechnology.com/redes-neuronales-recurrentes-lstm-prediccion-de-series-temporales/

[54] RoX818, “Explainable CNNs: Making Neural Networks Transparent.” Accessed: Dec. 10, 2024. [Online]. Available: https://aicompetence.org/explainable-cnns-making-networks-transparent/

[55] H. Ismail Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P. A. Muller, “Deep learning for time series classification: a review,” *Data Min Knowl Discov*, vol. 33, no. 4, pp. 917–963, Jul. 2019, doi: 10.1007/s10618-019-00619-1.

[56] P. Boniol, M. Meftah, E. Remy, and T. Palpanas, “dCAM: Dimension-wise Class Activation Map for Explaining Multivariate Data Series Classification,” in *Proceedings of the 2022 International Conference on Management of Data*, in SIGMOD/PODS ’22. ACM, Jun. 2022, pp. 1175–1189. doi: 10.1145/3514221.3526183.

[57] E. Choi, M. T. Bahadori, J. A. Kulas, A. Schuetz, W. F. Stewart, and J. Sun, “RETAIN: An Interpretable Predictive Model for Healthcare using Reverse Time Attention Mechanism,” Aug. 2016, [Online]. Available: http://arxiv.org/abs/1608.05745

[58] T. Guo, T. Lin, and N. Antulov-Fantulin, “Exploring Interpretable LSTM Neural Networks over Multi-Variable Data.”

[59] I. Gandin, A. Scagnetto, S. Romani, and G. Barbati, “Interpretability of time-series deep learning models: A study in

cardiovascular patients admitted to Intensive care unit," *J Biomed Inform*, vol. 121, Sep. 2021, doi: 10.1016/j.jbi.2021.103876.

[60] C. Schockaert, P. Wurth, S. A. Luxembourg, R. Leperlier, and A. Moawad, "Attention Mechanism for Multivariate Time Series Recurrent Model Interpretability Applied to the Ironmaking Industry."

[61] S. Karzhev, "Previsión de Series Temporales AI: Guía para principiantes." Accessed: Dec. 11, 2024. [Online]. Available: https://www.datacamp.com/es/blog/ai-time-series-forecasting

[62] L. A. Peña Florez, "Modelo de optimización multicriterio para la evaluación y la selección de proveedores en cadenas de suministro," Universidad Distral Francisco José de Caldas, Bogotá, 2015.

[63] H. Chefer, S. Gur, and L. Wolf, "Transformer Interpretability Beyond Attention Visualization." [Online]. Available: https://github.com/hila-

[64] P. Fantozzi and M. Naldi, "The Explainability of Transformers: Current Status and Directions," *Computers*, vol. 13, no. 4, Apr. 2024, doi: 10.3390/computers13040092.

[65] J. Labaien, T. Idé, P.-Y. Chen, E. Zugasti, and X. De Carlos, "Diagnostic Spatio-temporal Transformer with Faithful Encoding," May 2023, doi: 10.1016/j.knosys.2023.110639.

[66] A. van Sprang, E. Acar, and W. Zuidema, "Enforcing Interpretability in Time Series Transformers: A Concept Bottleneck Framework," Oct. 2024, [Online]. Available: http://arxiv.org/abs/2410.06070

[67] B. Lim, S. Arık, N. Loeff, and T. Pfister, "Temporal Fusion Transformers for interpretable multi-horizon time series forecasting," *Int J Forecast*, vol. 37, no. 4, pp. 1748–1764, Oct. 2021, doi: 10.1016/j.ijforecast.2021.03.012.