CAPÍTULO II

Aprendizaje federado para una sociedad de máquinas

Senén Barro*
José Miguel Burés
Roberto Iglesias

La inteligencia artificial a menudo se concibe como el proyecto de diseñar agentes artificiales capaces de tomar decisiones de forma autónoma y actuar de forma eficiente en entornos cada vez más complejos. Cuanto más sofisticadas se hagan las tareas que los humanos deleguemos en dichos agentes, menos podremos supervisar sus decisiones y mayor será el impacto (positivo o negativo) que tendrán sus acciones en nuestras vidas. Hay una concienciación creciente de que la autonomía de los sistemas IA ha de desempeñarse de forma responsable y, en particular, que sus decisiones y acciones han de estar alineadas con los valores (éticos) de la sociedad. En este capítulo se van a repasar algunas técnicas para gobernar los sistemas multiagente, y presentar modelos recientes para que los agentes IA puedan desempeñar su autonomía de forma responsable, alineando sus acciones con los valores éticos de la sociedad.

Palabras clave: aprendizaje automático, aprendizaje federado, privacidad de los datos, datos no IID, deriva de concepto, personalización del aprendizaje.

* Este trabajo ha recibido apoyo financiero de la Xunta de Galicia — Consellería de Educación, Ciencia, Universidades e Formación Profesional (acreditación de Centro de investigación de Galicia 2024-2027 ED431G-2023/04 y acreditación de Grupo de Referencia Competitiva (2022-2025, Project ED431C 2022/19 CONSOLIDACIÓN 2022 GRC-GSI) y de la Unión Europea (Fondo Europeo de Desarrollo Regional — FEDER).

1. DEL RAZONAMIENTO AUTOMÁTICO AL APRENDIZAJE AUTOMÁTICO

A lo largo de su historia, la inteligencia artificial ha seguido dos grandes aproximaciones para abordar la resolución de problemas. La primera se apoya en la representación explícita del conocimiento y el razonamiento automático; la segunda, en el aprendizaje a partir de datos mediante métodos estadísticos y de optimización, fundamentalmente. Aunque a menudo se presentan como paradigmas contrapuestos, ambas aproximaciones han coexistido desde los orígenes del campo. Lo que ha cambiado con el tiempo es cuál de ellas ha ocupado una posición predominante en la investigación y en las aplicaciones prácticas.

En sus inicios, durante las décadas de 1950 y 1960, la IA se desarrolló fundamentalmente bajo una perspectiva simbólica. La hipótesis central era que el comportamiento inteligente podía entenderse como la manipulación de símbolos de acuerdo con reglas formales. Tuvieron una especial atención los problemas asociados a espacios de estados en los que la resolución de un problema significa encontrar un camino que nos lleve desde el estado de partida a otro estado, normalmente no único, que podamos considerar como una buena solución, si no la óptima, al problema. Los juegos de mesa son un ejemplo paradigmático, y dentro de estos, el ajedrez. En ese proceso de encontrar un estado solución, normalmente es inviable explorar computacionalmente todas las opciones, así que se hizo común el uso de heurísticas diseñadas por humanos para guiar la búsqueda de una solución. En este contexto se consolidaron conceptos como la resolución de problemas mediante búsqueda, la demostración automática de teoremas y el uso de la lógica como lenguaje de representación del conocimiento. Los trabajos fundacionales de Newell, Simon y McCarthy establecieron una visión de la IA como una disciplina orientada a modelar el razonamiento humano a través de estructuras simbólicas y procesos inferenciales (McCarthy, 1959; Newell y Simon, 1976).

Con el paso del tiempo, se hizo evidente que estos enfoques funcionaban bien en dominios pequeños y altamente estructurados, pero tenían dificultades para escalar a problemas del mundo real, donde el conocimiento relevante es ingente, incompleto y a menudo incierto. Esta constatación condujo, en los años setenta y principios de los ochenta, al auge de los sistemas expertos (a veces denominados también sistemas basados en conocimiento). En lugar de aspirar a un razonamiento general, estos sistemas se centraban en capturar el conocimiento específico de un dominio mediante formas computacionalmente tratables. Dicho conocimiento podía proceder de fuentes escritas, como libros o publicaciones científicas, o directamente de expertos humanos. La promesa era pragmática: si se lograba formalizar lo que saben los especialistas sobre un tema, el sistema podría replicar —al menos parcialmente— su capacidad de resolución de problemas en su ámbito de competencia. Durante este periodo, la IA simbólica alcanzó su mayor impacto industrial, con aplicaciones en diagnóstico médico, configuración de sistemas y soporte a la decisión (Shortliffe, 1976).

Sin embargo, este enfoque puso de manifiesto limitaciones estructurales importantes. La adquisición y el mantenimiento del conocimiento resultaron ser procesos costosos y complejos, y los sistemas tendían a comportarse de forma inadecuada ante situaciones no previstas en el conocimiento disponible, a menudo muy incompleto. Además, el tratamiento de la incertidumbre reveló las carencias de una lógica puramente determinista, lo que motivó la incorporación de modelos probabilísticos y métodos de razonamiento bajo incertidumbre, como las redes bayesianas (Pearl, 1988). Estas dificultades, combinadas con expectativas sobredimensionadas —algo común durante el desarrollo de la IA— desembocaron en periodos de desilusión y reducción de financiación, conocidos como "inviernos de la IA".

A finales de los años ochenta y durante la década de 1990 se produjo un cambio gradual pero profundo en la orientación del campo. El aumento de la capacidad de cómputo y la creciente disponibilidad de datos digitales favorecieron el desarrollo de métodos basados en el *aprendizaje automático*. En lugar de codificar explícitamente el conocimiento, estos enfoques se centraban en aprender modelos a partir de ejemplos, evaluando su calidad mediante criterios empíricos como el error de generalización. Se consolidó así una visión más cercana a la estadística y al reconocimiento de patrones, con técnicas como los árboles de decisión, los modelos probabilísticos y las máquinas de vectores soporte (Quinlan, 1993; Vapnik, 1995).

Las redes neuronales, presentes desde los orígenes de la disciplina, experimentaron en este periodo un nuevo impulso, especialmente en tareas de percepción como el reconocimiento de caracteres. Aun así, durante muchos años su uso estuvo limitado por dificultades de entrenamiento y por la falta de datos y cómputo suficientes (LeCun *et al.,* 1998). No fue hasta mediados de la década de 2000 cuando una serie de avances técnicos permitió entrenar modelos más grandes y complejos, sentando las bases de lo que posteriormente se conocería como aprendizaje profundo (Hinton *et al.,* 2006).

El punto de inflexión que marca el predominio actual del enfoque basado en datos suele situarse en torno a 2012, con resultados espectaculares en visión por computador obtenidos mediante redes neuronales profundas entrenadas a gran escala (Krizhevsky *et al*., 2012). Desde entonces, el aprendizaje profundo ha pasado a dominar áreas como la visión, el procesamiento del lenguaje natural y, en combinación con el aprendizaje por refuerzo, ciertos problemas de control y toma de decisiones. Sistemas capaces de aprender directamente a partir de percepciones de bajo nivel y de optimizar su comportamiento mediante la interacción con el entorno ilustran la potencia de esta aproximación (Mnih *et al., 2015;* Silver *et al.*, 2016).

No obstante, este predominio no implica la desaparición del enfoque simbólico. Por el contrario, las limitaciones de los sistemas puramente basados en datos —dependencia de grandes volúmenes de información, dificultades de interpretación, falta de garantías formales— han reavivado el interés por integrar conocimiento explícito, razonamiento y

aprendizaje. En la práctica, muchas de las arquitecturas más avanzadas combinan componentes aprendidos con mecanismos clásicos de búsqueda, planificación o restricción lógica, dando lugar a enfoques híbridos que tratan de aprovechar lo mejor de ambos paradigmas.

Desde una perspectiva histórica, puede afirmarse que la IA ha oscilado entre periodos de predominio simbólico y periodos dominados por el aprendizaje a partir de datos, sin que ninguno de los dos enfoques haya sido nunca completamente excluyente. La situación actual refleja más bien una convergencia: el reconocimiento de que la inteligencia artificial robusta y confiable probablemente requiera tanto de la capacidad de aprender de la experiencia como la de razonar con estructuras de conocimiento explícitas, dependiendo del problema y del contexto de aplicación.

2. ¿QUÉ ES EL APRENDIZAJE AUTOMÁTICO Y CUÁL ES SU ORIGEN?

El ámbito del aprendizaje automático está integrado en el de la inteligencia artificial, y se orienta al desarrollo de algoritmos que mejoran automáticamente a través de la experiencia. Me parece especialmente elegante la definición dada por Tom Mitchell, brillante investigador de la IA: "Un sistema aprende de la *experiencia* X con respecto a una *tarea* T y con una medida de *rendimiento* P, si su rendimiento para la tarea T, medido mediante P, mejora con la experiencia X" (Mitchell, 1997).

Abunda la idea equivocada de que el ámbito del aprendizaje automático es bastante reciente, pero no es así. Algunas de las bases matemáticas están establecidas desde hace un par de siglos. Sirva de ejemplo el ajuste de puntos a una recta por mínimos cuadrados (regresión lineal). La primera forma de regresión lineal documentada fue el método de los mínimos cuadrados publicado por Legendre en 1805. Es cierto que ahí están las bases matemáticas e implícito un algoritmo que aprende una función que permite predecir el valor de una variable independiente dado el valor de una o más variables dependientes, pero no se trata de una implementación en una máquina que automatice el proceso de aprender dicha función.

En todo caso, hace bastantes décadas que el aprendizaje automático como tal dio sus primeros pasos. El primer programa informático que empleó aprendizaje automático fue el juego de damas desarrollado por Arthur Samuel en 1952 (Samuel, 1959). Este programa, ejecutado en una computadora IBM 701, se considera pionero porque fue capaz de mejorar su desempeño a medida que jugaba más y más partidas.

Samuel diseñó su *software* para que ajustara sus estrategias en función de las partidas previas, afinando su capacidad de juego sin intervención humana directa. El programa no solo calculaba movimientos óptimos con reglas predefinidas, sino que también

empleaba técnicas de refinamiento basadas en la experiencia, un concepto fundamental en el aprendizaje automático (recordemos la definición de aprendizaje dada por Mitchell). Aunque rudimentario, su enfoque inspiró futuros algoritmos de aprendizaje por refuerzo, esenciales en la inteligencia artificial moderna, en particular en los modelos grandes de lenguaje o LLM.

El trabajo de Samuel sentó las bases del aprendizaje automático como disciplina científica, demostrando que los programas podían "aprender de la experiencia" en lugar de seguir reglas estrictamente programadas.

3. APRENDIZAJE SUPERVISADO, NO SUPERVISADO Y POR REFUERZO

El interés por el aprendizaje automático es tal que la mayoría de los investigadores en IA están implicados de un modo u otro en él. No es de extrañar, así que la mayor parte de las publicaciones científicas en los congresos de IA se sitúen en la órbita del aprendizaje automático (y dentro de este, de un modo notorio, en los modelos grandes de lenguaje, o LLM). Es cierto que en la mayor parte de los casos no se trata de propuestas realmente originales, sino de variantes, muchas veces de escasísimo valor, o ninguno, de modelos base en aprendizaje automático, como pueden ser los clasificadores (Fernández-Delgado *et al.*, 2014) o regresores (Fernández-Delgado *et al.*, 2019).

En general, casi todos los algoritmos y aplicaciones de aprendizaje automático se sitúan en tres categorías principales, dependiendo de cómo se guía el proceso de aprendizaje: aprendizaje supervisado, aprendizaje no supervisado y aprendizaje por refuerzo. Esta distinción es clásica, pero sigue siendo conceptualmente útil porque refleja tres maneras distintas de relacionar datos, objetivos y funciones a optimizar —aprendizaje propiamente dicho—, lo que ayuda a entender qué tipo de problemas puede abordar cada enfoque y cuáles son sus límites.

El *aprendizaje supervisado* es, históricamente y aún hoy, la forma más extendida y mejor comprendida. En este paradigma, el sistema dispone de un conjunto de ejemplos en los que cada entrada está asociada a una salida deseada, decidida de antemano y en coherencia con el problema a resolver. El aprendizaje se guía explícitamente por esa señal externa: el modelo ajusta sus parámetros para minimizar la discrepancia entre sus predicciones y la realidad buscada —etiquetas con la respuesta deseada para los datos de entrada—. Problemas de clasificación, regresión y reconocimiento de patrones encajan naturalmente en este marco, y gran parte del éxito reciente del aprendizaje profundo se apoya en formulaciones supervisadas a gran escala. Su fortaleza principal es la claridad del objetivo: se detalla lo que se espera del sistema. Su principal limitación es también evidente: requiere grandes cantidades de datos etiquetados, cuya obtención suele ser costosa, lenta o directamente inviable.

Un ejemplo claro y simple de problema resoluble con aprendizaje supervisado es la clasificación de correo electrónico (spam/no spam), en el que partiríamos de un conjunto de entrenamiento formado por cientos o miles de ejemplos de correos etiquetados, según el caso, como deseados o no deseados. El modelo aprende a generalizar a partir de estos ejemplos para clasificar correos nuevos, ya en condiciones reales de operación.

En contraste, el *aprendizaje no supervisado* prescinde de etiquetas explícitas. El sistema recibe únicamente los datos de entrada y debe descubrir por sí mismo regularidades, estructuras o representaciones internas que capturen aspectos relevantes de esos datos, útiles para abordar el problema en cuestión. Aquí, el proceso de aprendizaje no está guiado por una noción externa de "respuesta correcta", sino por criterios internos como la compacidad, la reconstrucción, la independencia o la probabilidad de los datos observados. Históricamente, este tipo de aprendizaje ha estado ligado a tareas como el agrupamiento, la reducción de dimensionalidad o el modelado de distribuciones. Conceptualmente, su importancia va más allá de estas aplicaciones concretas: el aprendizaje no supervisado encarna la idea de que un sistema puede organizar un micromundo —a través de los datos que lo representan— sin supervisión directa, lo que lo aproxima a ciertos mecanismos de aprendizaje humano y animal. Sin embargo, precisamente por carecer de una señal externa clara, evaluar y controlar lo que se aprende resulta más difícil, y los resultados pueden ser más dependientes de supuestos implícitos del modelo.

Un ejemplo de aplicación diseñable a través del aprendizaje no supervisado es la segmentación de clientes a partir de datos derivados de su comportamiento (compras, frecuencia, gasto). El sistema agrupa clientes con patrones similares sin que nadie defina previamente qué es un "tipo" de cliente. El resultado es una estructura útil para análisis o *marketing*, aunque no impuesta desde fuera. Será tras la identificación de los agrupamientos que se correspondan con distintos perfiles de clientes, cuando estos se asocien con una etiqueta o acción de utilidad en el dominio.

El *aprendizaje por refuerzo* (Sutton y Barto, 2018) introduce una tercera forma de aprendizaje. En este caso, el sistema aprende a partir de su interacción con un entorno, recibiendo señales de recompensa o castigo que evalúan las consecuencias positivas o negativas de sus acciones. A diferencia del aprendizaje supervisado, no se le indica qué acción es correcta en cada situación; solo se le proporciona una valoración escalar de su comportamiento, a menudo con un cierto retraso temporal. El proceso de aprendizaje se guía así por un objetivo global (maximizar la recompensa acumulada) y no por ejemplos individuales etiquetados. Este paradigma es especialmente adecuado para problemas de control, toma de decisiones secuenciales y planificación bajo incertidumbre. Su complejidad conceptual es mayor, ya que el agente debe enfrentarse simultáneamente a la exploración del entorno y a la explotación de lo ya aprendido, y los datos dependen de las propias decisiones del sistema. Hace décadas que en nuestro grupo de investigación usamos este tipo de aprendizaje para que los robots autónomos aprendan a moverse en entornos complejos y dinámicos y a resolver tareas en ellos.

Aunque estas tres formas suelen presentarse como categorías separadas, en la práctica sus fronteras son cada vez más permeables. Existen enfoques *semisupervisados*, que combinan pequeños conjuntos de datos etiquetados con grandes volúmenes de datos sin etiquetar; métodos *autosupervisados*[1], en los que la señal de aprendizaje se construye a partir de la propia estructura de los datos; y sistemas de aprendizaje por refuerzo que incorporan modelos supervisados o representaciones aprendidas de forma no supervisada. Esta hibridación refleja una comprensión más madura del problema: guiar el aprendizaje no es una decisión binaria, sino un diseño continuo que depende del tipo de señales o datos disponibles, del coste de obtenerlos y del nivel de autonomía deseado para el sistema.

Desde una perspectiva conceptual, estas tres formas de aprendizaje corresponden a tres respuestas distintas a una misma pregunta fundamental: ¿de dónde proviene la información que orienta la mejora del modelo? En el aprendizaje supervisado, procede de un maestro explícito; en el no supervisado, de la propia estructura de los datos; y en el aprendizaje por refuerzo, de la interacción con el entorno —físico o digital— y sus consecuencias. Entender esta distinción es clave para situar cualquier técnica concreta de aprendizaje automático dentro de un marco más amplio y para elegir, de manera informada, el paradigma adecuado para cada problema.

Seguramente el lector, aunque no sea un especialista en inteligencia artificial, habrá oído hablar del *aprendizaje profundo* y de su extraordinaria influencia en el *boom* actual de la IA. Es importante aclarar que no se trata de un nuevo tipo de aprendizaje en el mismo sentido que el aprendizaje supervisado, no supervisado o por refuerzo. Como hemos dicho, estas categorías describen cómo se guía el proceso de aprendizaje a través de distintas tipologías de datos (etiquetados, sin etiquetar y asociados a recompensas o penalizaciones, según el resultado de las decisiones tomadas por el sistema). En cambio, el aprendizaje profundo (LeCun *et al.*, 2015) describe una familia de modelos y representaciones, no el tipo de datos disponibles durante el aprendizaje. Se refiere al uso de redes neuronales con muchas capas que aprenden representaciones jerárquicas mediante optimización por gradiente (Rumelhart *et al.*, 1986). Desde este punto de vista, el aprendizaje profundo es una opción arquitectónica y algorítmica, no un paradigma de aprendizaje en sí mismo. De hecho, en el aprendizaje profundo pueden combinarse los distintos tipos de aprendizaje antes descritos.

[1] En el caso de los LLM, su fase inicial de entrenamiento se realiza mediante aprendizaje autosupervisado. En concreto, partiendo de grandes colecciones de texto sin etiquetar, se generan tareas de predicción interna, como predecir la siguiente palabra (o token) a partir del contexto previo, o bien reconstruir partes del texto ocultas. El texto original proporciona simultáneamente la entrada y la "señal de supervisión": no se añade información externa, pero el aprendizaje sí está guiado por un objetivo explícito.

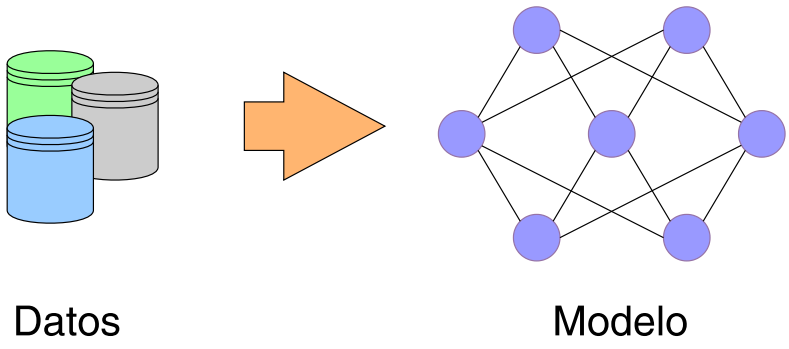
4. DEL APRENDIZAJE LOCAL AL FEDERADO

Pensemos ahora en las posibilidades de aplicación del aprendizaje automático en función de en dónde residen los datos y cómo se accede a ellos durante el entrenamiento. En este sentido, es posible trazar una tipología del aprendizaje automático que resulta especialmente útil para entender tanto las arquitecturas técnicas como las implicaciones prácticas (privacidad, escalabilidad, control, costes). Este eje es relativamente reciente en la historia del campo, pero hoy es central en muchos sistemas reales. En la forma más clásica de aprendizaje automático, los datos se encuentran centralizados en un único dispositivo o repositorio y el entrenamiento se realiza directamente sobre ese conjunto de entrenamiento. Este esquema, que durante décadas fue prácticamente el único considerado, presupone que los datos pueden recopilarse, almacenarse y procesarse en un único entorno computacional. Desde el punto de vista algorítmico, es el escenario ideal, por ser el más simple y el que en principio permite obtener mejores resultados: el modelo se entrena optimizando una función objetivo definida sobre todos los datos disponibles. Gran parte del aprendizaje supervisado, no supervisado y profundo se ha formulado históricamente bajo esta hipótesis. Su principal ventaja es la eficiencia estadística y computacional; su limitación, cada vez más relevante, es que no siempre es viable concentrar datos por razones de volumen, latencia, propiedad o privacidad.

El aprendizaje que puede tener acceso al conjunto entero de datos de entrenamiento, y también de validación, puede ser local, si hay una única fuente o repositorio de datos, o puede ser centralizado si estas son múltiples pero los datos en última instancia están disponibles sin limitaciones para que un sistema central haga su trabajo.

FIGURA 1

ESQUEMA DE APRENDIZAJE LOCAL. SE GENERA UN MODELO ESPECÍFICO PARA PROCESAR LOS DATOS LOCALES



Fuente: Elaboración propia.

Formalizando mínimamente el funcionamiento de un modelo de *aprendizaje local,* podemos partir de una entidad E, que opera en un mundo W, en el que realiza una tarea T, a través de un modelo M, aprendido a partir de su experiencia X, con una medida de rendimiento P:

$$E(X(t)) \rightarrow M(t) \quad [1]$$

El modelo para realizar T puede ser dinámico, buscando mejorar el rendimiento, $P(t)$, a través de nuevas experiencias, $X(t)$, en cuyo caso:

$$E(M(t), X(t)) \rightarrow M(t+\Delta), \text{ con } P(t+\Delta) \geq P(t) \quad [2]$$

Cuando los datos están distribuidos entre múltiples fuentes, pero pueden agregarse o compartirse, al menos parcialmente, aparecen esquemas de aprendizaje centralizado o distribuido.

El aprendizaje centralizado se asume que es posible concentrar los datos. Si es posible juntarlos todos para poder aprender sobre ellos sin más, casi siempre va a ser la mejor opción, salvo si hacerlo de ese modo añade costes muy significativos de comunicación, memoria o demora en la obtención de una primera aproximación a la resolución del problema, pongamos por caso.

En el *aprendizaje centralizado* un conjunto de entidades, $ES = \{E_1, E_2 \cdots E_S\}$, comparte sus experiencias $XS = \{X_1, X_2 \cdots X_S\}$, de modo que es posible calcular directamente un modelo global centralizado, MC, de resolución de la tarea T, a partir de las mismas:

$$EC(\{X_s(t), s = 1, \cdots, S\}) \rightarrow MC(t), \quad [3]$$

tal que, idealmente $PC(t) \geq P_s(t), \forall s = 1, \cdots, S$.

El aprendizaje centralizado para resolver una tarea se calcula generalmente en una arquitectura cliente-servidor y suele seguir el siguiente procedimiento (ver figura 2):

Paso 1. Los clientes envían sus datos al servidor (que se gestionan específicamente para añadir privacidad y seguridad, si es necesario).

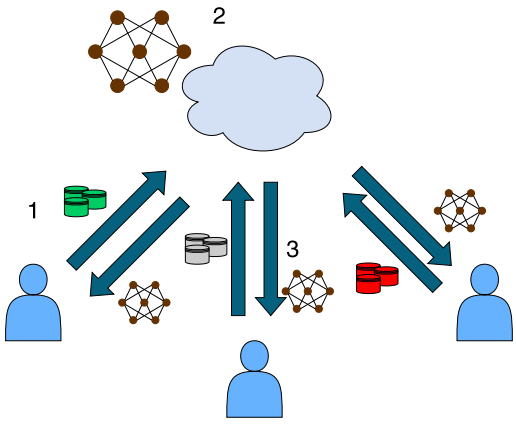
Paso 2. El servidor utiliza todos los datos para aprender una solución para resolver la tarea.

Paso 3. El servidor envía la solución a los clientes (la solución puede adaptarse o no a cada uno de ellos según sus especificidades).

Paso 4. Los clientes adoptan la solución proporcionada por el servidor y operan con ella.

FIGURA 2

APRENDIZAJE CENTRALIZADO, MEDIANTE UNA ARQUITECTURA CLIENTE-SERVIDOR: (1) LOS CLIENTES ENVÍAN SUS DATOS AL SERVIDOR; (2) ESTE CREA UNA SOLUCIÓN GLOBAL CON TODA LA INFORMACIÓN Y (3) LA DISTRIBUYE DE VUELTA A LOS CLIENTES



Fuente: Elaboración propia.

En el *aprendizaje distribuido* el entrenamiento se reparte entre varios nodos de cómputo, que procesan subconjuntos de los datos y sincronizan periódicamente los parámetros del modelo. Esta distribución puede responder a razones puramente técnicas (acelerar el entrenamiento de modelos grandes) o a la naturaleza misma del sistema (datos generados en múltiples localizaciones). Aunque conceptualmente sigue tratándose de un aprendizaje "global", como el centralizado, en este caso no existe un único punto donde residen todos los datos durante el proceso de construcción del modelo solución. Este enfoque se ha vuelto esencial en el entrenamiento de modelos a gran escala, pero no resuelve por sí mismo todas las restricciones que puedan existir en ciertos problemas, singularmente los de privacidad, ya que los datos o sus derivados siguen siendo visibles para la infraestructura central.

Un paso muy significativo en este sentido, y el punto al que queríamos llegar por ser el tema central de este texto, lo constituye el *aprendizaje federado* (McMahan *et al.*, 2017), en el que los datos permanecen en los dispositivos o lugares de generación de los mismos, de modo que en el aprendizaje sobre ellos solo se intercambia información de lo aprendido (funciones o modelos resultantes durante el aprendizaje) en ningún

caso los datos de entrenamiento. Este paradigma responde a restricciones cada vez más frecuentes en contextos como dispositivos personales o aplicaciones en el ámbito de la salud o las finanzas, donde la privacidad de los datos es una cuestión legal, o al menos ética.

Dado un conjunto de entidades $ES = \{E_1, E_2 \cdots E_S\}$ que resuelven la tarea T mediante modelos locales $MS = \{M_1, M_2 \cdots M_S\}$, obtenidos a partir de sus experiencias particulares $MS = \{M_1, M_2 \cdots M_S\}$, nos planteamos obtener un modelo global, MF, mediante una estrategia de aprendizaje federado, que mejore o ayude a mejorar a cualquier modelo local:

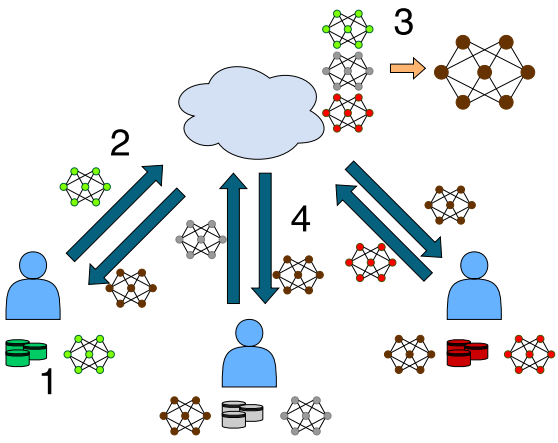
$$EF(\{M_s(t), s = 1, \cdots, S\}) \rightarrow MF(t), \qquad [4]$$

tal que, idealmente, $PF(t) \geq P_s(t), \forall s = 1, \cdots, S$, si t es suficientemente grande.

Las arquitecturas cliente-servidor son especialmente adecuadas para el aprendizaje federado que funciona con conjuntos de datos que comparten el mismo conjunto de variables o características; por ejemplo, los que provienen de un conjunto de estaciones meteorológicas distribuidas por un territorio o de una misma aplicación de móviles instalada en muchos de estos dispositivos. A continuación se describe el procedimiento general para una arquitectura cliente-servidor federada (ver figura 3):

FIGURA 3

ARQUITECTURA CLIENTE-SERVIDOR FEDERADA: 1) ENTRENAMIENTO LOCAL EN CADA CLIENTE, (2) ENVÍO DE MODELOS AL SERVIDOR, (3) COMBINACIÓN PARA CREAR UN MODELO GLOBAL Y (4) DISTRIBUCIÓN DE LA SOLUCIÓN A LOS CLIENTES



Fuente: Elaboración propia.

Paso 1. N clientes aprenden a resolver una tarea en sus conjuntos de datos (normalmente a partir de un modelo inicial proporcionado por el servidor y que suele ser el mismo para todos los clientes, al menos cuando todos ellos tienen que resolver la misma tarea).

Paso 2. Los modelos locales resultantes del aprendizaje local, $\{M_n, n = 1, \cdots, N\}$, se envían al servidor (después de ser tratados específicamente para añadir más privacidad y seguridad, si es necesario).

Paso 3. El servidor utiliza todos los modelos locales para aprender un modelo mejor para resolver la tarea.

Paso 4. El servidor envía la solución a los N clientes (la solución puede adaptarse o no a cada uno de ellos en función de sus especificidades y/o de las tareas de las que son responsables).

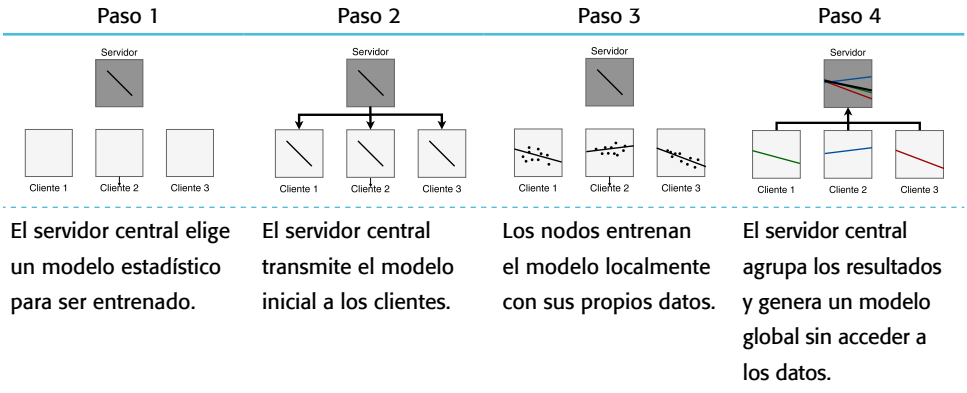
Paso 5. Los clientes asumen la solución proporcionada por el servidor y operan con ella (la solución puede ser común o no a todos ellos).

Normalmente el procedimiento indicado se repite, de modo que los clientes continúan mejorando sus modelos para resolver las tareas de las que son responsables, hasta que se verifica un criterio de parada, como haber alcanzado un rendimiento local y/o colectivo determinado.

Por si no le ha quedado claro cuál es la base del aprendizaje federado, le pondremos un ejemplo simple que creemos que le ayudará a entenderlo mejor. Gráficamente se describe en la figura 4. Imagínese que tiene tres fuentes de datos y quiere aprender la función lineal que los ajusta en cada caso a una recta. En este caso los datos no están disponibles de partida, sino que se trata de un sistema dinámico en el que van entrando continuamente nuevos datos y la función aprendida ha de irse ajustando a ellos a lo largo del tiempo. Además, los datos locales no pueden compartirse por una cuestión de privacidad. En primer lugar, para poder operar integrando las funciones aprendidas y no los datos locales, la función a aprender tendrá que ser única (paso 1). A partir de ahí, todos los aprendedores —clientes— adoptarán esa función como punto de partida previo al ajuste de la misma a los datos locales de los que dispone cada aprendedor (paso 2). Los aprendedores ya operarán con sus respectivas funciones locales (paso 3), que en este caso serán progresivamente ajustadas a los nuevos datos locales. Tras un cierto tiempo de reajuste local continuo de las funciones, los aprendedores envían sus

FIGURA 4

ILUSTRACIÓN DE LA IDEA BASE DEL APRENDIZAJE FEDERADO, EN LA QUE LOS CLIENTES LOCALES AJUSTAN CON SUS DATOS UNA FUNCIÓN OBJETIVO, LA COMPARTEN PARA OBTENER UNA FUNCIÓN DE SÍNTESIS GLOBAL, QUE DE NUEVO UTILIZAN Y AJUSTAN A SUS DATOS LOS APRENDEDORES LOCALES, SIGUIENDO UN PROCESO ITERATIVO



Fuente: Elaboración propia.

respectivas funciones al servidor y este hará una composición de las mismas para obtener una función promedio (paso 4).

Fijémonos que en este ejemplo se asume que los datos en cada sistema local no están dados de antemano como un conjunto único y estático, sino que llegan secuencialmente, y el sistema debe actualizar su modelo de manera incremental. En muchos escenarios reales —sistemas de recomendación, detección de fraude, control— los datos se generan continuamente y no pueden almacenarse o reprocesarse en bloque. En estos casos, el aprendizaje está estrechamente ligado al flujo de datos y a las decisiones sobre qué información conservar, descartar o resumir. Desde esta perspectiva, el "lugar" de los datos es efímero: existen en el momento en que son observados y, tras su uso, pueden desaparecer. Después volveremos a tratar este tema del aprendizaje continuo, tan importante para un sinfín de aplicaciones.

4.1. ¿Qué ocurre cuando no se quiere o no se pueden compartir los datos?

Tenemos muy cerca en el tiempo la pandemia. Seguramente recordará el lector las aplicaciones de seguimiento de contagios (*contact tracing apps*). En general, hubo dos métodos con arquitecturas de gestión de datos diferentes, centralizadas y no centralizadas (Troncoso *et al.,* 2022), en función de garantizar, respectivamente, una mayor utilidad en salud pública o una mayor privacidad de los datos del usuario. España asumió una aplicación denominada Radar COVID, basada en el modelo descentralizado de notificación de exposiciones, alineado con las recomendaciones europeas de privacidad

y con el enfoque adoptado por la mayoría de los países de la Unión Europea (UE). Es cierto que su escasa utilización la hizo inútil en la práctica.

En las arquitecturas no centralizadas, los móviles próximos entre sí intercambiaban identificadores anónimos. Cada móvil disponía así de la huella dejada por aquellos otros móviles que en un momento dado estuvieron cercanos a aquel. Cada persona que enfermase de COVID-19 y dispusiese de esta aplicación en su móvil, debería notificarlo a un servidor central, que se encargaría de ir registrando los identificadores generados por los móviles de las personas enfermas. Todos los móviles con la aplicación instalada se encargaban periódicamente de acceder al servidor para chequear si tenían en su memoria alguno de esos identificadores, lo que supondría el haber estado en contacto (en proximidad) con una persona enferma.

En el método centralizado, los identificadores que un móvil recoge de aquellos de los que estuvo próximo son enviados al servidor central. Cuando alguien da positivo, lo notifica al servidor central y es este el que se encarga directamente de notificar a todos aquellos móviles que hubiesen estado suficientemente próximos al de la persona enferma, y durante un tiempo suficiente, de esta circunstancia.

Fijémonos en que ni durante una pandemia se impuso el modelo centralizado, a pesar del valor incuestionable que podría tener para hacer estudios epidemiológicos, en particular los asociados a la forma y evolución de los contagios. Por eso, disponer de arquitecturas de datos y de aprendizaje automático que preserven su propiedad y, sobre todo, su privacidad, es cada vez más importante. El Reglamento General de Protección de Datos (RGPD) de la Unión Europea es paradigmático en este sentido. Cuando no existe una solución fácil o viable para reunir los datos, necesitamos otros enfoques distintos del aprendizaje centralizado. Eso explica por qué, aunque el aprendizaje federado se encuentra todavía en sus primeras etapas, en realidad en su infancia, está recibiendo mucha atención no solo por parte de la comunidad científica, sino también por parte de las empresas.

5. APRENDIZAJE FEDERADO: ORIGEN, FUNDAMENTOS Y EJEMPLOS

El aprendizaje federado (FL, *Federated Learning*) designa un conjunto de técnicas para entrenar modelos de aprendizaje automático cuando los datos no pueden o no deben centralizarse. La idea esencial es desplazar el cómputo hacia donde residen los datos (dispositivos u organizaciones) y agregar actualizaciones de los modelos aprendidos localmente (no los datos en bruto) para construir un modelo común, que pasa a ser compartido entre los aprendedores locales. Esta formulación reordena prioridades clásicas del aprendizaje automático: junto a la precisión del modelo que se aprende, pasan a primer plano la privacidad, la gobernanza del dato, el coste de comunicación y la robustez frente a la heterogeneidad, entre otras cuestiones menos relevantes en los sistemas centralizados.

El aprendizaje federado surge como respuesta a un problema cada vez más frecuente en sistemas modernos: los datos valiosos para entrenar modelos (texto tecleado en los móviles, sensores de dispositivos de vestir, historias clínicas o transacciones financieras, por ejemplo) son sensibles, voluminosos y, a menudo, están sujetos a restricciones legales y contractuales. Centralizarlos en una base de datos estructurada o incluso en un lago de datos (*data lake*) puede ser inviable o simplemente no recomendable. En ese contexto, la publicación seminal (McMahan *et al.,* 2017; McMahan *et al.,* 2016) que acuñó el término "Federated Learning" propone explícitamente entrenar modelos sin mover los datos, coordinando una "federación" de clientes —a los que a veces nos referimos como aprendedores— con un servidor que agrega actualizaciones de los modelos aprendidos localmente.

Los autores de este artículo que da inicio al campo del aprendizaje federado utilizan redes profundas (DNN, o *Deep Neural Networks*) como modelo de aprendizaje. Aunque este tipo de modelos es muy popular en el aprendizaje profundo, no es necesario ni mucho menos, ya que el aprendizaje federado es agnóstico al modelo o modelos de aprendizaje considerados —pueden aplicarse modelos de regresión logística, modelos lineales u otros enfoques—[2]. El utilizar redes profundas en este caso y en muchos otros puede facilitar el diseño de clasificadores y predictores con datos no IID (no independientes ni idénticamente distribuidos), especialmente complejos de tratar, como luego veremos. Por ejemplo, si varios hospitales colaboran para entrenar un modelo que prediga enfermedades, cada centro atiende a pacientes con perfiles distintos: algunos con mayor población anciana, otros con más jóvenes o con distintas patologías previas. Aunque todos midan lo mismo, los datos locales presentan distribuciones diferentes, lo que hace que el aprendizaje global sea más complicado.

La generación de un modelo global a partir de los modelos locales obtenidos por los aprendedores es relativamente sencilla y escalable. En este caso, los autores propusieron el ahora muy popular algoritmo *Federated Averaging* (FedAvg), una generalización del *stochastic gradient descent* para redes neuronales profundas entrenadas de forma distribuida.

En el apéndice 8 se incluye y explica el algoritmo original, FedAvg, por ser el propuesto para el caso del artículo fundacional del campo y ser, todavía, en sus diferentes versiones, ampliamente utilizado.

[2] En nuestro grupo de investigación hemos propuesto la arquitectura ECFL (*Ensemble and Continual Federated Learning*), pensada para escenarios más "reales" que los que puede abordar el aprendizaje federado clásico: múltiples dispositivos, datos en flujo continuo y cambios de distribución (*concept drift*), que luego analizaremos con detalle. La idea central es que, en lugar de intentar entrenar un único modelo global común (como en FedAvg), el modelo global sea un comité (*ensemble*) formado por varios modelos locales entrenados de manera independiente en cada cliente. Esto permite agregar modelos heterogéneos (distintas familias algorítmicas o estructuras), porque la agregación ya no requiere promediar parámetros, sino combinar predicciones (Casado *et al.,* 2023).

Pero el propio marco teórico del artículo deja claro que FL es agnóstico al modelo: en principio, puede aplicarse a regresión logística, modelos lineales u otros enfoques, siempre que el entrenamiento se base en actualizaciones de parámetros agregables.

Uno de los primeros casos de uso publicados sobre aprendizaje federado se orientó a la mejora de las sugerencias del teclado virtual Gboard de Google (Yang *et al.,* 2018). En concreto, el aprendizaje federado se usó para mejorar las sugerencias de consulta y de palabras (*query suggestions/next-word prediction*) que Gboard muestra mientras el usuario escribe. Cuando Gboard muestra una consulta sugerida a un usuario, su teléfono almacena localmente información sobre el contexto actual y si ha hecho clic en la sugerencia. El aprendizaje federado procesa ese historial en el dispositivo para sugerir mejoras a la siguiente iteración del modelo de sugerencia de consultas de Gboard. El objetivo era entrenar modelos de lenguaje a partir de lo que los usuarios teclean realmente, sin enviar ese texto —altamente sensible— a los servidores de Google.

La idea es aprender palabras fuera de vocabulario (OOV, *Out Of Vocabulary*) basándose en las palabras escritas con frecuencia por los usuarios de dispositivos móviles. OOV se refiere a palabras que no están incluidas en el vocabulario del dispositivo móvil de un usuario. Las palabras que faltan en el vocabulario no pueden predecirse mediante la sugerencia de teclado, la autocorrección o la escritura por gestos. Este es un claro ejemplo en el que no es posible compartir datos (palabras OOV) de los distintos usuarios (datos personales y sensibles), ni es factible aprender de la experiencia individual de cada usuario, porque es muy estrecha y extraordinariamente lenta. El aprendizaje federado es particularmente útil para resolver esta tarea, entrenando un modelo compartido de generación de OOV basado en los datos de todos los usuarios de móviles sin necesidad de transmitir datos sensibles a un servidor centralizado.

Durante este proceso, el modelo de generación de OOV en el móvil de cada usuario se actualiza constantemente, mientras que los datos de entrenamiento permanecen en el dispositivo. Como resultado, cada dispositivo móvil acaba teniendo un potente modelo de generación de OOV.

Otro ejemplo desarrollado, en este caso en nuestro grupo de investigación, consiste en haber usado el aprendizaje federado para el reconocimiento de la actividad de caminar a partir de los sensores inerciales de los móviles (Casado *et al.,* 2020), pero en condiciones realistas donde el móvil puede ir en mano, bolsillo, mochila, etc., y su orientación cambia constantemente. De nuevo se trata de una aplicación donde los datos recabados por los sensores de los móviles de los distintos usuarios son sensibles y privados.

En general, acudir a arquitecturas de aprendizaje federado responde al interés o necesidad de preservar la privacidad de datos obtenidos o distribuidos de múltiples fuentes, pero conviene subrayar que el aprendizaje federado no equivale automáticamente a una garantía plena de privacidad. El diseño original se apoya en la minimización de

datos (no subir datos brutos), pero el hecho de compartir actualizaciones de parámetros puede aportar información que permita en algunos casos recuperar datos o información de origen que siga siendo sensible (Suliman y Leith, 2023). Por eso muchas veces se incorporan medidas adicionales al aprendizaje federado, como el cifrado de datos o la agregación segura (Bonawitz *et al.,* 2017).

Creemos importante destacar que la contribución seminal al campo del aprendizaje federado se hizo por investigadores de Google. Pero aún más relevante es que Google no se limitó a proponer un algoritmo, sino que desarrolló y describió una arquitectura de sistema para ejecutar aprendizaje federado en producción con millones de dispositivos, abordando problemas que apenas aparecen en entornos de laboratorio (selección de clientes, fallos, sincronización por rondas, despliegue continuo, restricciones energéticas y de conectividad) (Bonawitz *et al.*, 2019). Este salto —del algoritmo a la plataforma operativa— es una de las razones por las que el aprendizaje federado se convirtió en un área propia y no en una simple variante del aprendizaje distribuido.

Hay que reconocer a las grandes compañías tecnológicas haber hecho algunas de las contribuciones más relevantes de los últimos años a las ciencias de la computación y a la IA en particular. De hecho, también es una aportación de Google la arquitectura de transformadores (Vaswani *et al.,* 2017), central en los LLM modernos. El paralelismo no es superficial: tanto en el aprendizaje federado como en los transformadores se observa un patrón recurrente de contribución "de plataforma".

La literatura sobre FL ha crecido de forma muy rápida desde 2017. Un estudio bibliométrico basado en Web of Science, en el que se analizan 3.107 artículos (Algorabi *et al.,* 2024) pone de manifiesto el rápido crecimiento en publicaciones en el campo, y el protagonismo de China, seguido a gran distancia por EE. UU. De hecho, entre las organizaciones con un mayor número de publicaciones en el período analizado (2017-2023), las diez primeras son chinas (Hong Kong incluido), si bien las principales revistas que recogen publicaciones de aprendizaje federado son del IEEE (por ejemplo, *IEEE Internet of Things Journal* e *IEEE Access*).

Tras el artículo seminal, el progreso en el campo se ha centrado más en ir realizando aportaciones algorítmicas y metodológicas, cuyo valor se mide más sobre problemas de laboratorio o conjuntos de datos abiertos pero sencillos (como MNIST), lo que está muy alejado de la solución a problemas reales de cierta complejidad. Esto denota que el campo todavía está más asentando conceptos, aportando recursos y herramientas computacionales y explorando las limitaciones de la tecnología de aprendizaje federado actual, que usándola en la resolución de problemas del mundo real. Sin duda, faltan algunos años para darle la suficiente madurez al campo.

De hecho, si analizamos algunos sectores como salud, finanzas o robótica personal e industrial, en los que el aprendizaje federado está llamado a tener un gran prota-

gonismo, la brecha entre expectativas y realidades es todavía muy grande. En salud, por ejemplo, gran parte de la evidencia publicada corresponde a escenarios *cross-silo* (hospitales o centros) y a menudo se limita a prototipos o estudios con un número reducido de instituciones, con desafíos persistentes de interoperabilidad y heterogeneidad clínica (Shah *et al.,* 2025). En finanzas, además de la heterogeneidad y la latencia, entran con fuerza cuestiones de auditoría, explicabilidad y cumplimiento normativo; la bibliografía reciente insiste en que la adopción práctica se ve condicionada por gobernanza, riesgos operacionales y amenazas de seguridad específicas (Kennedy *et al.*, 2025). En robótica y sistemas ciberfísicos, la dificultad crece porque el aprendizaje suele ser secuencial, sensible al tiempo real y frecuentemente requiere manejar gran cantidad de datos (en general procedentes de sensores que aportan información de la interacción de los robots con el mundo físico) y políticas de control con garantías: el aprendizaje federado encaja peor cuando la comunicación es cara y la distribución no estacionaria es la norma.

En conjunto, la conclusión razonable tras casi una década de desarrollo del aprendizaje federado es doble: (i) ha demostrado viabilidad operativa en algunos productos a gran escala (especialmente móviles); y (ii) su generalización a dominios "duros" y altamente regulados progresa, pero con un ritmo más lento y una dependencia fuerte de ingeniería, gobernanza y robustez.

6. QUEDA MÁS CAMINO QUE EL ANDADO

Como en cualquier nuevo campo científico-tecnológico, en el aprendizaje federado las preguntas son muchas más que las repuestas. Son muchos los problemas sin resolver que suponen barreras importantes a su aplicación generalizada a problemas del mundo real. Aunque no es una lista exhaustiva de los mismos, los problemas recurrentes y que explican por qué muchos éxitos siguen siendo "de laboratorio" o de alcance limitado, pueden agruparse en cinco frentes.

6.1. Comunicación, latencia y disponibilidad

Una de las limitaciones estructurales del aprendizaje federado es que el proceso de entrenamiento depende críticamente de la comunicación entre los participantes y el servidor coordinador. A diferencia del aprendizaje centralizado, donde el acceso a los datos es inmediato y continuo, en entornos federados la comunicación es costosa, limitada y heterogénea, tanto en ancho de banda como en fiabilidad (McMahan *et al.,* 2017; Bonawitz *et al.,* 2019).

En muchos escenarios reales —especialmente en configuraciones *cross-device*[3]— los participantes presentan disponibilidad intermitente. Dispositivos móviles pueden estar apagados, sin batería o sin conexión; robots o sistemas ciberfísicos pueden encontrarse fuera de servicio; y nodos industriales u hospitalarios pueden tener ventanas de comunicación restringidas por razones operativas o de seguridad (Yang *et al.,* 2018). Como consecuencia, en cada ronda federada solo una fracción de los participantes puede contribuir efectivamente al entrenamiento, lo que introduce variabilidad y ralentiza la convergencia.

Este problema se agrava cuando el modelo a entrenar es grande o complejo, como ocurre con redes neuronales profundas. En estos casos, cada ronda implica el intercambio de millones de parámetros, y la frecuencia de actualización necesaria para mantener un aprendizaje estable puede hacer que el coste de comunicación domine completamente el proceso. Incluso algoritmos diseñados para ser eficientes, como FedAvg, pueden volverse impracticables si el volumen de datos transmitidos y la latencia de sincronización superan los límites del sistema.

La latencia añade un segundo nivel de dificultad. El aprendizaje federado suele organizarse en rondas sincronizadas, lo que implica esperar a que un conjunto suficiente de participantes complete su entrenamiento local antes de agregar los resultados. En entornos con nodos lentos o poco fiables, este mecanismo introduce retrasos significativos o fuerza a descartar participantes, reduciendo la eficiencia estadística del aprendizaje. Existen alternativas asíncronas, pero introducen a su vez problemas de estabilidad y de consistencia del modelo global.

En conjunto, la combinación de coste de comunicación, latencia y disponibilidad variable no es un simple inconveniente de implementación, sino un factor que condiciona qué problemas pueden abordarse de forma realista con aprendizaje federado. Mientras estas limitaciones no se mitiguen de forma sistemática —mediante compresión, reducción de frecuencia de comunicación, arquitecturas jerárquicas o infraestructuras dedicadas—, el uso del aprendizaje federado en problemas complejos y a gran escala seguirá siendo selectivo y dependiente del contexto operativo.

6.2. Privacidad, seguridad y amenaza adversarial

Aunque el aprendizaje federado se presenta a menudo como una solución "intrínsecamente privada", en realidad introduce nuevos riesgos de seguridad y privacidad que no aparecen —o lo hacen de forma distinta— en el aprendizaje centralizado. El hecho de que los datos no se compartan directamente no implica que el sistema sea inmune a

[3] Suele usarse la referencia *cross-device* para hablar de sistemas con muchos dispositivos individuales, no coordinados, con alta variabilidad. Por el contrario, cuando son pocos participantes (organizaciones), estables y bien conectados se habla de *cross-silo*.

ataques: el intercambio de actualizaciones de modelo abre vectores de ataque específicos que deben considerarse explícitamente (Bonawitz *et al.,* 2017).

Uno de los problemas más estudiados es la corrupción o envenenamiento del modelo (*model poisoning*). En este tipo de ataque, uno o varios participantes maliciosos envían actualizaciones manipuladas con el objetivo de degradar el rendimiento global o forzar comportamientos indeseados (Suliman y Leith, 2023). Una variante especialmente peligrosa es la introducción de puertas traseras (*backdoors*), donde el modelo funciona correctamente en la mayoría de los casos, pero responde de forma errónea ante patrones específicos elegidos por el atacante. En entornos federados, detectar estos ataques resulta difícil, ya que no se tiene acceso a los datos locales y las actualizaciones pueden parecer estadísticamente plausibles.

Otro frente crítico es la integridad de la agregación. El servidor coordinador asume que las actualizaciones recibidas reflejan entrenamiento legítimo, pero en ausencia de mecanismos adicionales no puede verificar si los participantes son confiables ni la corrección de los gradientes. Incluso en escenarios sin participantes maliciosos, errores de implementación o fallos en dispositivos pueden introducir ruido sistemático que sesgue el modelo global.

Desde el punto de vista de la privacidad, se ha demostrado que las actualizaciones de modelo pueden filtrar información sensible. Los ataques de inferencia permiten, en ciertos casos, reconstruir características de los datos locales o inferir la pertenencia de un individuo a un conjunto de entrenamiento, a partir de gradientes o parámetros compartidos (Zhu *et al.,* 2019). Para mitigar estos riesgos se han propuesto técnicas como la agregación segura, que impide al servidor observar actualizaciones individuales, y la privacidad diferencial, que limita formalmente la información que puede inferirse sobre un participante concreto (Troncoso *et al*., 2022).

Sin embargo, estas defensas no son gratuitas. La agregación segura introduce sobrecoste computacional y de comunicación, además de complejidad criptográfica. La privacidad diferencial requiere añadir ruido a las actualizaciones o a los resultados agregados, lo que suele traducirse en una pérdida de precisión o en una necesidad mayor de datos y rondas de entrenamiento para alcanzar un rendimiento comparable. En problemas complejos, este compromiso entre privacidad, seguridad y utilidad se vuelve especialmente delicado.

En conjunto, la seguridad y la privacidad en aprendizaje federado no son propiedades automáticas, sino objetivos de diseño que deben equilibrarse cuidadosamente con la eficiencia y la calidad del modelo. La presencia de amenazas adversariales realistas implica que el aprendizaje federado, lejos de eliminar el problema de la confianza, lo

reformula: ya no se trata solo de confiar en una infraestructura central, sino de gestionar la desconfianza entre múltiples participantes y el propio coordinador del sistema.

6.3. Evaluación y validación clínica/regulatoria

En dominios de alto impacto social y económico, como la salud o las finanzas, el éxito de un sistema de aprendizaje automático no se mide únicamente por su rendimiento predictivo. En estos contextos es imprescindible cumplir requisitos adicionales de trazabilidad, validación externa, control de sesgos, reproducibilidad y auditoría, que están estrechamente ligados a marcos regulatorios y a la gestión del riesgo (Shah *et al.,* 2025). El aprendizaje federado introduce dificultades específicas para satisfacer estos requisitos, ya que rompe muchos de los supuestos habituales de evaluación en entornos centralizados.

Uno de los principales retos es la validación externa. En escenarios federados, los datos permanecen distribuidos entre múltiples instituciones o dispositivos, lo que dificulta la construcción de conjuntos de validación independientes y representativos. Cada nodo puede evaluar el modelo con sus propios datos, pero estas evaluaciones no son necesariamente comparables, ya que las distribuciones locales difieren y las métricas pueden reflejar realidades muy distintas. En salud, por ejemplo, un modelo puede funcionar bien en un hospital concreto y fallar sistemáticamente en otro con una población diferente, sin que exista un mecanismo sencillo para detectar este problema de forma global.

La trazabilidad y la reproducibilidad constituyen otro punto crítico. En un sistema federado, el modelo global es el resultado de múltiples rondas de entrenamiento con subconjuntos variables de participantes, datos que evolucionan en el tiempo y posibles actualizaciones asíncronas. Reconstruir exactamente cómo se obtuvo una versión concreta del modelo —qué nodos participaron, con qué datos y bajo qué condiciones— es mucho más complejo que en un entorno centralizado, lo que complica tanto la auditoría técnica como el cumplimiento normativo.

El control de sesgos se ve igualmente afectado. En ausencia de una visión global de los datos, identificar sesgos sistemáticos hacia determinados subgrupos resulta difícil. En sectores regulados, esta limitación es especialmente problemática, ya que las autoridades suelen exigir evidencias claras de equidad y ausencia de discriminación. El aprendizaje federado requiere, por tanto, nuevas estrategias para evaluar y mitigar sesgos sin acceder directamente a los datos subyacentes (He *et al.,* 2020).

Por último, la evaluación continua en producción plantea retos adicionales. En entornos clínicos o financieros, los modelos deben monitorizarse para detectar degradaciones de rendimiento o cambios de comportamiento que puedan tener consecuencias graves.

En un sistema federado, esta monitorización debe realizarse de forma distribuida y respetando las restricciones de privacidad, lo que ha motivado líneas de investigación específicas sobre evaluación federada, métricas agregadas seguras y protocolos de validación distribuidos (Kairouz *et al.,* 2021).

En conjunto, la dificultad de evaluar y validar modelos en aprendizaje federado no es un problema accesorio, sino un factor limitante clave para su adopción en dominios regulados. Superar esta barrera requiere no solo avances algorítmicos, sino también marcos metodológicos y organizativos que permitan garantizar confianza, cumplimiento y responsabilidad en sistemas donde los datos no pueden centralizarse.

6.4. MLOps federado y gobernanza del ciclo de vida

MLOps (*Machine Learning Operations*) es el conjunto de prácticas, procesos y herramientas que permiten desarrollar, desplegar, monitorizar y mantener modelos de aprendizaje automático en producción de forma fiable y reproducible. Por tanto, entrenar un modelo mediante aprendizaje automático —y federado, en particular— es solo una parte del problema.

Una síntesis útil es que el aprendizaje federado funciona mejor cuando coinciden tres condiciones que afectan a los datos, al problema y al sistema: muchos datos útiles pero no exportables; el problema a resolver debe tener un objetivo común claro entre los participantes —aprendedores—, de modo que todos optimizan esencialmente la misma tarea (por ejemplo, predicción de texto, detección de fraude, diagnóstico); una infraestructura capaz de orquestar el entrenamiento. El aprendizaje federado no es solo un algoritmo, sino un sistema distribuido complejo. Requiere seleccionar participantes, gestionar comunicación, tolerar fallos, asegurar privacidad, versionar modelos y monitorizar su evolución. Sin una infraestructura que coordine estas operaciones de forma robusta y automatizada, los costes operativos superan rápidamente los beneficios teóricos del enfoque.

Cuando falla alguna de estas condiciones, el beneficio puede diluirse frente a alternativas (aprendizaje centralizado con fuertes controles, enclaves seguros, intercambio de datos sintéticos o colaboración mediante modelos ya preentrenados).

6.5. Heterogeneidad no IID y desalineación de objetivos

Quizás este sea el problema más importante y complejo de resolver. En nuestro grupo de investigación está siendo uno de nuestros retos de investigación más relevantes. El problema surge cuando los datos entre clientes/instituciones difieren en distribución,

calidad, codificación y sesgos. Además, los objetivos pueden no estar alineados (por ejemplo, distintos hospitales con protocolos distintos). Esto tensiona tanto la convergencia como la equidad del rendimiento entre participantes, y obliga a técnicas de personalización y/o modelos por dominio que complican el despliegue.

Vamos a tratar de exponer la complejidad del problema.

6.5.1. Datos no independientes y no idénticamente distribuidos

Como ya hemos dicho, una de las diferencias fundamentales entre el aprendizaje automático clásico y el aprendizaje federado es que, en este último, los datos no se recogen ni se mezclan en un único repositorio, sino que proceden de múltiples fuentes autónomas (dispositivos, usuarios, instituciones). Esta característica rompe, en muchos casos, la hipótesis estadística habitual de que los datos son independientes e idénticamente distribuidos (IID), sobre la que se apoyan gran parte de los algoritmos de aprendizaje.

Para comprender las implicaciones de esta ruptura, conviene analizar por separado ambos conceptos: independencia e identidad de distribución, ya que describen propiedades distintas de los datos.

6.5.2. Independencia o no de los datos

Dos conjuntos de datos son independientes cuando la observación de uno no proporciona información sobre el otro. En el contexto del aprendizaje federado, esto se traduce en que los datos generados por una fuente (cliente o aprendedor) no influyen causalmente en los datos generados por otra.

Un ejemplo sencillo de datos independientes sería el de varios sensores de temperatura instalados en ubicaciones geográficas alejadas y sin interacción entre sí. Cada sensor genera sus mediciones de forma autónoma, y conocer las lecturas de uno no permite inferir las de otro, ni de forma aproximada. De manera similar, en un sistema federado con usuarios independientes que escriben textos privados en sus móviles, los datos de un usuario no afectan directamente a los de los demás.

Por el contrario, los datos no independientes aparecen cuando existe correlación o dependencia entre las fuentes. Un ejemplo típico es una red social, donde el comportamiento de un usuario influye en el de otros (mensajes, tendencias, reacciones). En un escenario federado, esto implica que los datos de distintos clientes están acoplados, lo que viola la hipótesis de independencia y complica tanto el entrenamiento como la evaluación de los modelos.

6.5.3. Distribución idéntica o no de los datos

Los datos son idénticamente distribuidos cuando todos proceden de la misma distribución estadística, incluso aunque sean independientes. En aprendizaje federado, esto significa que todos los clientes observan datos con características similares y en proporciones comparables.

Un ejemplo de datos idénticamente distribuidos sería un conjunto de dispositivos idénticos operando en condiciones controladas, como sensores industriales calibrados que miden la misma variable en procesos repetitivos. En este caso, cada cliente ve esencialmente el mismo tipo de datos, y los algoritmos de aprendizaje funcionan de forma similar a un entorno centralizado.

En cambio, los datos no idénticamente distribuidos son aquellos en los que cada fuente observa una distribución distinta. Este es el caso más común en aprendizaje federado real. Por ejemplo, distintos hospitales atienden a poblaciones de pacientes con características demográficas y clínicas diferentes; o distintos usuarios de móviles escriben en idiomas, registros y estilos distintos. Aunque los datos puedan ser independientes entre sí, sus distribuciones difieren, lo que introduce heterogeneidad estadística entre clientes.

6.5.4. Combinaciones habituales en aprendizaje federado

Al combinar ambas propiedades, se obtienen distintos escenarios relevantes en sistemas federados. El caso independiente e idénticamente distribuido (IID) se da cuando cada cliente o aprendedor recibe una muestra aleatoria de un mismo conjunto global de datos. Este escenario es común en experimentos controlados, pero poco representativo de aplicaciones reales.

Un escenario más realista es el de datos independientes pero no idénticamente distribuidos, donde los clientes no influyen entre sí, pero cada uno observa una distribución distinta. Este es el caso típico en aplicaciones como salud, finanzas o dispositivos personales, y constituye una de las principales dificultades del aprendizaje federado.

También pueden darse situaciones en las que los datos son no independientes, pero sí idénticamente distribuidos, como redes de sensores cercanos que miden el mismo fenómeno físico y presentan correlaciones espaciales o temporales.

Finalmente, el escenario más complejo corresponde a datos no independientes y no idénticamente distribuidos, donde existen tanto dependencias entre clientes como diferencias profundas en sus distribuciones: por ejemplo, en sistemas sociales, económicos o de tráfico.

En resumen, en aprendizaje federado, el término no IID engloba una variedad de situaciones muy distintas. Identificar correctamente si la falta de IID se debe a dependencias entre fuentes, a diferencias de distribución, o a ambas, es esencial para diseñar algoritmos adecuados y para interpretar de forma correcta los resultados obtenidos en escenarios reales.

6.5.5. Deriva de concepto

En los sistemas de aprendizaje federado que operan de forma continuada en el tiempo, una de las principales fuentes de degradación del rendimiento es la *deriva de concepto (concept drift)*. Este fenómeno hace referencia a cambios en el proceso generador de los datos que invalidan, total o parcialmente, las hipótesis bajo las cuales se entrenó el modelo. En contextos federados, la deriva de concepto adquiere una relevancia especial, ya que la adaptación del modelo es más costosa y compleja que en escenarios centralizados.

Desde un punto de vista probabilístico, cualquier problema de aprendizaje supervisado puede describirse mediante la distribución conjunta $P(x, y)$, que puede descomponerse como:

$$P(x, y) = P(x)P(y \mid x) \quad [5]$$

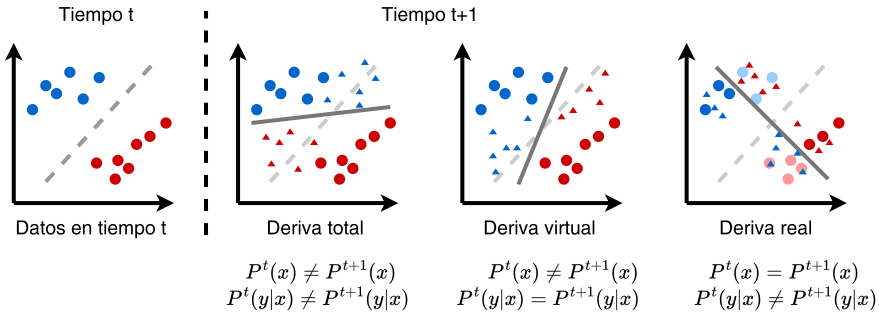
donde $P(x)$ representa la distribución de las entradas observadas y $P(y \mid x)$ la relación entre dichas entradas y las etiquetas. La deriva de concepto se produce cuando esta distribución conjunta cambia con el tiempo, es decir, cuando $P^t(x, y) \neq P^{t+1}(x, y)$. La utilidad de esta descomposición radica en que permite distinguir qué parte del problema está cambiando, lo cual tiene implicaciones directas sobre los mecanismos de detección y adaptación necesarios. En el estado inicial del sistema, correspondiente al instante t, los datos de entrada y las etiquetas mantienen una relación estable. Un modelo entrenado en estas condiciones puede aprender una frontera de decisión adecuada que separa correctamente las clases. Sin embargo, a medida que el sistema evoluciona y se recopilan nuevos datos en instantes posteriores, pueden aparecer distintos tipos de deriva.

Vamos a describir los distintos tipos de deriva, que gráficamente se muestran en la figura 5.

El caso más general y severo es la denominada *deriva total*, en la que cambian simultáneamente la distribución de las entradas y la relación entrada–salida:

FIGURA 5

CLASIFICACIÓN DE LA DERIVA DE CONCEPTO: COMPARATIVA ENTRE LA DERIVA TOTAL (CAMBIO EN ESPACIO DE ENTRADA Y EN LA RELACIÓN MODELO-ETIQUETA), VIRTUAL (CAMBIO SOLO EN EL ESPACIO DE ENTRADA) Y REAL (CAMBIO EN LA RELACIÓN MODELO-ETIQUETA)



Fuente: Elaboración propia.

$$P^t(x) \neq P^{t+1}(x), \qquad P^t(y \mid x) \neq P^{t+1}(y \mid x) \qquad [6]$$

En este escenario, los datos no solo se concentran en regiones distintas del espacio de entrada, sino que además el significado de las etiquetas respecto a esas entradas se ha modificado. Desde la perspectiva del aprendizaje federado, este es el caso más difícil de gestionar, ya que exige mecanismos de adaptación profunda del modelo y, habitualmente, la incorporación de nueva supervisión distribuida para evitar el deterioro progresivo del rendimiento.

Un caso más benigno es la *deriva virtual*, en la que únicamente cambia la distribución de las entradas, mientras que la relación entrada–salida permanece inalterada:

$$P^t(x) \neq P^{t+1}(x), \qquad P^t(y \mid x) = P^{t+1}(y \mid x) \qquad [7]$$

Aquí, el "concepto" subyacente no ha cambiado, pero los datos se presentan de forma distinta. En la práctica, este tipo de deriva puede estar causado por cambios en sensores —debido a su degradación, por ejemplo—, condiciones ambientales o patrones de uso. En sistemas federados, la deriva virtual suele poder abordarse mediante técnicas de adaptación más ligeras, como reentrenamientos parciales o ajustes en la ponderación de los clientes, sin necesidad de redefinir completamente el modelo.

Por último, la deriva real se produce cuando la distribución de las entradas se mantiene estable, pero cambia la relación entre entradas y etiquetas:

$$P^t(x) = P^{t+1}(x), \qquad P^t(y \mid x) = P^{t+1}(y \mid x) \qquad [8]$$

Este tipo de deriva es particularmente problemática, ya que no resulta evidente al analizar únicamente las entradas. El cambio afecta al significado de las clases o a la regla de decisión, lo que implica que un modelo entrenado previamente puede seguir observando datos "familiares", pero producir predicciones sistemáticamente erróneas. En aprendizaje federado, detectar y corregir una deriva real suele requerir acceso a nuevas etiquetas o señales de validación distribuidas, lo que introduce importantes retos operativos y de coordinación.

En conjunto, esta tipología pone de manifiesto que no toda deriva de concepto es equivalente. En sistemas federados y continuos, identificar si el cambio afecta a $P(x)$, a $P(y \mid x)$, o a ambos, es un paso esencial, y nada trivial, para diseñar estrategias de adaptación eficaces. Ignorar esta distinción conduce a soluciones parciales que funcionan en escenarios controlados, pero que fallan cuando el sistema se enfrenta a la complejidad y no estacionalidad propias de aplicaciones reales.

En nuestro grupo hemos abordado este problema de la no estacionalidad temporal de los datos (Casado *et al.*, 2022). Aunque FedAvg funciona razonablemente bien con heterogeneidad entre clientes, no está diseñado para adaptarse cuando la distribución de datos cambia con el tiempo, algo frecuente en dispositivos inteligentes (móviles, dispositivos de vestir, robots) y que provoca degradación de rendimiento si no se detecta y gestiona adecuadamente. En nuestro caso hemos propuesto el algoritmo CDA-FedAvg (*Concept-Drift-Aware Federated Averaging*), una extensión de FedAvg orientada a aprendizaje federado continuo en un único objetivo compartido (*single-task*), pero con posibles cambios de distribución en el tiempo. A diferencia de FedAvg "por rondas" estrictas, planteamos un esquema asíncrono en el que los clientes ganan autonomía para decidir cuándo entrenar y qué datos usar, mientras el servidor actúa sobre todo como orquestador y agregador cuando recibe actualizaciones.

Por otra parte, en 2022 publicamos un artículo de revisión (Criado *et al*., 2022) cuyo objetivo era ordenar y clarificar la heterogeneidad estadística de los datos no IID, tanto entre clientes (cada dispositivo/institución tiene datos distintos) como a lo largo del tiempo (los datos cambian, y nos enfrentamos al *concept drift*). Evidenciamos que la mayor parte de las soluciones obviaban el problema del no IID y que, a menudo, cuando lo hacían, no precisaban con suficiente cuidado qué tipo de heterogeneidad se asumía, lo que dificulta comparar métodos y entender cuándo funcionan realmente.

La contribución central de nuestro trabajo fue conceptual y taxonómica: proponiendo una clasificación formal de la heterogeneidad y revisando estrategias de aprendizaje federado destacadas en función de esa clasificación (por ejemplo, métodos orientados a mitigar la

divergencia por datos no IID, técnicas de personalización, ajustes en agregación/optimización, etc.). En paralelo, el artículo introduce de forma explícita el puente con el aprendizaje continuo (*Continual Learning* o CL): subraya que muchas situaciones federadas reales no solo son "no IID entre clientes", sino también no estacionarias en el tiempo, y que herramientas clásicas de aprendizaje continuo (por ejemplo, mecanismos para manejar deriva y reducir olvido) pueden adaptarse al entorno federado para mejorar robustez.

La tabla 1 organiza los distintos tipos de heterogeneidad de datos en aprendizaje federado, atendiendo a dos ejes conceptuales ortogonales:

- Heterogeneidad espacial, es decir, diferencias estadísticas entre los datos de distintos participantes en un mismo instante, y
- Heterogeneidad temporal, es decir, cambios en la distribución de los datos a lo largo del tiempo.

TABLA 1

TAXONOMÍA DE LA HETEROGENEIDAD ESPACIAL Y TEMPORAL EN APRENDIZAJE FEDERADO.

		Heterogeneidad espacial			
		Sin cambios (datos IID)	Cambios en el espacio de entrada entre clientes	Cambios en el comportamiento entre clientes	Cambios entrada y comportamiento entre clientes
Heterogeneidad temporal	Sin cambios (datos IID)				
	Deriva de concepto virtual		NO ABORDADO HASTA AHORA		
	Deriva de concepto real				
	Deriva total de concepto				

Nota: La zona sombreada indica combinaciones de heterogeneidad que no han sido abordadas de forma efectiva por los métodos existentes. En el eje horizontal se muestra la heterogeneidad espacial, es decir, diferencias estadísticas entre los datos de distintos participantes en un mismo instante, y en el eje vertical, heterogeneidad temporal, es decir, cambios en la distribución de los datos a lo largo del tiempo.

En el eje horizontal se representan los distintos grados de heterogeneidad espacial, desde el caso ideal de datos IID hasta escenarios en los que existen diferencias entre

clientes en el espacio de entrada $P_i(x)$, en el comportamiento o relación entrada–salida $P_i(y \mid x)$ o en ambos simultáneamente. En el eje vertical se representan los distintos grados de heterogeneidad temporal, desde datos estacionarios hasta situaciones de deriva de concepto, distinguiendo entre deriva virtual (cambios en), deriva real (cambios en) y deriva total. En el eje horizontal se representan los distintos grados de heterogeneidad espacial, desde el caso ideal de datos IID hasta escenarios en los que existen diferencias entre clientes en el espacio de entrada $P_i(x)$, en el comportamiento o relación entrada–salida $P_i(y \mid x)$ o en ambos simultáneamente. En el eje vertical se representan los distintos grados de heterogeneidad temporal, desde datos estacionarios hasta situaciones de deriva de concepto, distinguiendo entre deriva virtual (cambios en $P(x)$), deriva real (cambios en $P(y \mid x)$) y deriva total.

Las celdas sombreadas indican regiones del espacio del problema que, hasta ahora, no han sido abordadas de forma satisfactoria por la literatura en aprendizaje federado. En particular, cuando coexisten heterogeneidad espacial en el comportamiento y deriva temporal, los métodos actuales muestran limitaciones claras, ya sea por problemas de convergencia, degradación del rendimiento o falta de mecanismos de adaptación. La tabla pone de manifiesto que buena parte de la investigación se ha centrado en escenarios parciales (por ejemplo, no IID espacial sin deriva temporal), mientras que los escenarios más realistas —donde ambas heterogeneidades coexisten— siguen siendo en gran medida un reto abierto.

La tabla 2 amplía el análisis anterior incorporando un elemento clave para la aplicabilidad real del aprendizaje federado: las restricciones prácticas sobre la disponibilidad de datos etiquetados. Manteniendo los mismos ejes de heterogeneidad espacial y temporal que en la tabla anterior, esta tabla indica qué supuestos adicionales son necesarios para que los métodos actuales puedan funcionar en cada escenario.

Las leyendas distinguen entre cuatro situaciones: ausencia de restricciones, disponibilidad de datos etiquetados de un solo participante en múltiples momentos (1P-MT), disponibilidad de datos etiquetados de todos los participantes en un único momento inicial (TP-1T) y disponibilidad de datos etiquetados de todos los participantes de forma periódica (TP-MT). Estas restricciones no describen algoritmos concretos, sino hipótesis implícitas que muchos trabajos asumen —a menudo de forma no explícita— para poder manejar la heterogeneidad y la deriva.

La tabla muestra que, a medida que aumenta la complejidad del escenario (heterogeneidad espacial combinada con deriva temporal), los métodos existentes requieren supuestos de supervisión cada vez más fuertes, como la necesidad de etiquetado periódico o la existencia de un participante "ancla" que proporcione datos de referencia. Esto pone de relieve una brecha importante entre los entornos experimentales y los despliegues reales, donde dichas condiciones suelen ser costosas, difíciles de mantener o directamente inviables.

TABLA 2

RESTRICCIONES PRÁCTICAS SOBRE LA DISPONIBILIDAD DE DATOS ETIQUETADOS NECESARIOS PARA ABORDAR DISTINTOS ESCENARIOS DE HETEROGENEIDAD ESPACIAL Y TEMPORAL EN APRENDIZAJE FEDERADO Y CONTINUO

<table>
<tr><td colspan="2" rowspan="2"></td><td colspan="4">Heterogeneidad espacial</td></tr>
<tr><td>Sin cambios (datos IID)</td><td>Cambios en el espacio de entrada entre clientes</td><td>Cambios en el comportamiento entre clientes</td><td>Cambios entrada y comportamiento entre clientes</td></tr>
<tr><td rowspan="4">Heterogeneidad temporal</td><td>Sin cambios (datos IID)</td><td></td><td></td><td colspan="2" rowspan="2">Suficientes datos etiquetados de cada participante, al inicio. Restricción TP-1T</td></tr>
<tr><td>Deriva de concepto virtual</td><td></td><td></td></tr>
<tr><td>Deriva de concepto real</td><td colspan="2" rowspan="2">Suficientes datos etiquetados de un participante, de forma periódica. Restricción 1P-MT</td><td colspan="2" rowspan="2">Suficientes datos etiquetados de cada participante, de forma periódica. Restricción TP-MT</td></tr>
<tr><td>Deriva total de concepto</td></tr>
</table>

Nota: En el eje horizontal se muestra la heterogeneidad espacial, es decir, diferencias estadísticas entre los datos de distintos participantes en un mismo instante, y en el eje vertical, heterogeneidad temporal, es decir, cambios en la distribución de los datos a lo largo del tiempo.

En conjunto, la tabla sugiere que el cuello de botella no es solo algorítmico, sino también organizativo y operativo: muchos enfoques actuales funcionan únicamente si se acepta un coste elevado en términos de etiquetado y coordinación entre participantes.

6.5.6. Datos ruidosos y de calidad desigual en los clientes

Además de las diferencias en la distribución de los datos entre clientes, los sistemas de aprendizaje federado reales suelen presentar heterogeneidad en los tipos de datos o en la calidad de la información aportada por cada cliente o aprendedor. No todos los clientes obtienen sus datos en las mismas condiciones ni con el mismo nivel de fiabilidad.

Por ejemplo, en una red de sensores o dispositivos inteligentes, algunos nodos pueden estar correctamente calibrados y operar de forma estable, mientras que otros pueden sufrir fallos intermitentes, ruido elevado o condiciones de operación singulares. En aplicaciones centradas en usuarios, normalmente hay participantes que generan datos abundantes y consistentes, mientras que otros contribuyen de forma esporádica o con patrones muy particulares. Aunque estas situaciones no impliquen necesariamente un comportamiento malicioso, su efecto sobre el aprendizaje conjunto puede ser similar.

En el aprendizaje federado clásico, el objetivo es entrenar un único modelo global que funcione razonablemente bien para todos los clientes. Bajo este enfoque, se asume implícitamente que las contribuciones de los distintos participantes son compatibles entre sí y que las diferencias observadas reflejan la variabilidad razonable del entorno o del uso del sistema. Este supuesto, aunque útil desde un punto de vista conceptual, resulta limitado cuando algunos clientes generan datos muy ruidosos, poco representativos o sistemáticamente inconsistentes.

El aprendizaje federado personalizado surge como una extensión natural de este planteamiento. En lugar de buscar una única solución común, estos métodos permiten que cada cliente disponga de un modelo parcialmente adaptado a sus características locales. Esta distinción es clave: mientras que el aprendizaje federado estándar prioriza la generalización global, el aprendizaje federado personalizado pone el énfasis en capturar la diversidad entre clientes sin perjudicar el rendimiento del conjunto.

Un reto fundamental en este contexto es que, debido a las restricciones de privacidad inherentes al aprendizaje federado, no es posible inspeccionar directamente los datos de los clientes para evaluar su compatibilidad o calidad. El servidor central no puede saber si dos clientes observan fenómenos similares, si sus datos son coherentes entre sí o si uno de ellos introduce ruido de forma sistemática. Esta limitación obliga a diseñar mecanismos indirectos que permitan inferir estas propiedades a partir del comportamiento del proceso de entrenamiento, sin violar la privacidad de los datos locales.

Uno de los trabajos desarrollados en nuestro grupo de investigación (Burés *et al.*, 2025) aborda este problema analizando cómo contribuyen los distintos clientes al aprendizaje conjunto a lo largo del tiempo, con el objetivo de identificar patrones consistentes y discrepantes. De forma intuitiva, el sistema aprende a diferenciar entre clientes cuyas aportaciones resultan compatibles con el objetivo común y aquellos cuya información introduce inestabilidad. Esto permite ajustar su influencia sin necesidad de acceder a los datos originales ni excluir explícitamente a ningún participante.

Los estudios realizados en entornos de simulación muestran que este enfoque conduce a mejoras consistentes en el rendimiento y la estabilidad del aprendizaje, especialmente en escenarios con datos heterogéneos, presencia de ruido o clientes con objetivos maliciosos. Estos resultados ponen de manifiesto la importancia de tratar explícitamente la compatibilidad entre clientes y refuerzan la idea de que la personalización y la robustez son componentes clave para el despliegue efectivo del aprendizaje federado en aplicaciones reales.

7. MENSAJE FINAL

El mensaje final —coherente con el subtítulo de nuestro trabajo de 2022: "A long road ahead"— es deliberadamente crítico: aunque ha habido avances significativos en estos años en el aprendizaje federado, identificamos importantes retos abiertos persistentes en condiciones realistas (heterogeneidad múltiple, evolución temporal, restricciones de comunicación, etc.). El ámbito del aprendizaje federado necesita separar la paja cada vez más abundante del grano, bases de datos de evaluación y mecanismos de validación más rigurosos y representativos de la complejidad real de las aplicaciones realmente útiles y, por supuesto, conseguir avances claros en todos los aspectos antes apuntados, y que suponen de facto una seria limitación a la resolución de problemas del mundo real mediante aprendizaje automático.

En todo caso, es incontestable que el aprendizaje federado supone una gran oportunidad para empresas y organizaciones de todo tipo al aportar privacidad a los datos, poder mejorar colectivamente el aprendizaje de múltiples aprendedores, con o sin personalización de lo aprendido, y otras ventajas que hemos ido comentando a lo largo de este texto. Sectores como salud, banca, movilidad, industria 4.0 o los servicios digitales pueden beneficiarse de modelos que aprendan de múltiples fuentes de manera coordinada y segura, reduciendo costes, mejorando la calidad de los servicios y acelerando la innovación. En este sentido, aunque el camino sigue siendo largo, los avances recientes y las estrategias emergentes de operación en contextos no IID, la personalización y la mayor robustez y seguridad de las nuevas arquitecturas, muestran que el aprendizaje federado puede convertirse en un recurso indispensable en el mundo empresarial y tecnológico.

Referencias

Algorabi, Ö. *et al.* (2024). A Bibliometric Analysis on Federated Learning. *Journal of Advanced Research in Natural and Applied Sciences*, 10, 875–898.

Bonawitz, K. *et al.* (2017). Practical Secure Aggregation for Privacy-Preserving Machine Learning. En: *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, (1175–1191).

Bonawitz, K. *et al.* (2019). Towards Federated Learning at Scale: System Design. *arXiv*:1902.01046.

Burés, J. *et al.* (2025). Out of Distribution Detection and Adaptive Interpolation for Personalized Federated Learning. *Frontiers in Artificial Intelligence and Applications* (1873–1879).

Casado, F. *et al.* (2023). Ensemble and continual federated learning for classification tasks. *Machine Learning*, 112, 1–41.

Casado, F. E. *et al.* (2020). Walking Recognition in Mobile Devices. *Sensors*, *20*(4), art. 1189.

Casado, F. E. *et al.* (2022). Concept drift detection and adaptation for federated and continual learning. *Multimedia Tools Appl.*, 81(3), 3397–3419.

Criado, M. F. *et al.* (2022). Non-IID data and Continual Learning processes in Federated Learning: A long road ahead. *Information Fusion*, 88, 263-280.

Fernández-Delgado, M. *et al.* (2014). Do we Need Hundreds of Classifiers to Solve Real World Classification Problems? *Journal of Machine Learning Research*, 15, 3133–3181.

Fernández-Delgado, M. *et al.* (2019). An extensive experimental survey of regression methods. *Neural Networks*, 111, 11-34.

He, C. *et al.* (2020). FedML: A Research Library and Benchmark for Federated Machine Learning. *arXiv:2007.02945*.

Hinton, G. E., Osindero, S., y Teh, Y. W. (2006). A fast learning algorithm for deep belief nets. *Neural Comput.*, *18*(7), 1527–1554.

Kairouz, P. *et al.* (2021). Advances and Open Problems in Federated Learning. *Foundations and Trends in Machine Learning*, *14*(1–2), 1–210.

Kennedy, C., Hilal, A., y Momeni, M. (2025). The Role of Federated Learning in Improving Financial Security: A Survey. En: *GCAIoT 2025* (1-8).

Krizhevsky, A., Sutskever, I., y Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. En: *Proceedings of NIPS'12* (1097–1105).

LeCun, Y. *et al.* (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278–2324.

LeCun, Y., Yere, Y., y Hinton, G. (2015). Deep Learning. *Nature*, 521, 436-444.

McCarthy, J. (1959). Programs with Common Sense. En: *Proceedings of the Teddington Conference on the Mechanization of Thought Processes* (75–91). London.

McMahan, B. *et al.* (2017). Communication-Efficient Learning of Deep Networks from Decentralized Data. En: *Proceedings of AISTATS* (1273–1282).

McMahan, H. B. *et al.* (2016). Federated Learning of Deep Networks using Model Averaging. *CoRR*, abs/1602.05629.

Mitchell, T. M. (1997). *Machine learning*. New York: McGraw-hill.

Mnih, V. *et al.* (2015). Human-level control through deep reinforcement learning. *Nature*, *518*(7540), 529–533.

Newell, A., y Simon, H. A. (1976). Computer science as empirical inquiry: symbols and search. *Commun. ACM, 19*(3), 113–126.

Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Mateo, CA: Morgan Kaufmann.

Quinlan, J. R. (1993). *C4.5: programs for machine learning*. San Francisco, CA: Morgan Kaufmann.

Rumelhart, D. E., Hinton, G. E., y Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, 323, 533–536.

Samuel, A. L. (1959). Some Studies in Machine Learning Using the Game of Checkers. *IBM Journal of Research and Development, 3*(3), 210–229.

Shah, S. T. *et al.* (2025). Federated Learning in Public Health: A Systematic Review. *Healthcare*, *13*(21), art. 2760.

Shortliffe, E. (1976). Computer-based medical consultations: MYCIN. *Artificial Intelligence - AI*, 388, 1–20.

Silver, D. *et al.* (2016). Mastering the Game of Go with Deep Neural Networks and Tree Search. *Nature*, *529*(7587), 484–489.

Suliman, M., y Leith, D. (2023). Two Models are Better Than One. En: *Proceedings of ESORICS 2023* (105–122). Berlin: Springer.

Sutton, R. S., y Barto, A. G. (2018). *Reinforcement learning: an introduction*. Cambridge, MA: The MIT Press.

Troncoso, C. *et al.* (2022). Deploying decentralized, privacy-preserving proximity tracing. *Communications of the ACM*, 65, 48–57.

Vapnik, V. N. (1995). *The nature of statistical learning theory*. New York, NY: Springer-Verlag.

Vaswani, A. *et al.* (2017). Attention is all you need. En: *Advances in Neural Information Processing Systems*, (5998–6008).

YANG, T. *ET AL.* (2018). Applied Federated Learning: Improving Google Keyboard Query Suggestions. *CoRR*, abs/1812.02903.

ZHU, L., LIU, Z., Y HAN, S. (2019). Deep Leakage from Gradients. En: *Advances in Neural Information Processing Systems (NeurIPS)*, 32.

APÉNDICE A. ALGORITMO FEDAVG

Algoritmo 1: Federated Averaging

Inicializar el modelo global w_0

Para cada ronda $t = 1, 2, \cdots, T$

  El servidor selecciona un subconjunto de clientes S_t

  Para cada cliente $k \in S_t$ (en paralelo):

    $w_{t+1}^k \leftarrow$ ClienteActualiza (k, w_t)

  El servidor agrega:

  $w_{t+1} \leftarrow \sum_k \frac{n_k}{n} \cdot w_{t+1}^k$

Función ClienteActualiza (k, w)

  $w_{local} \leftarrow w$;

  Para cada época local $i = 1, \cdots, E$:

    Para cada minibatch $b \in D_k$:

      $w_{local} \leftarrow w_{local} - \eta \cdot \nabla \ell(w_{local}; b)$

  Devuelvor w_{local} ;

Notación:

- w_t : parámetros del modelo global en la ronda t
- S_t : subconjunto de clientes seleccionados en la ronda t
- D_k : datos locales del cliente k

- n_k : número de ejemplos en D_k
- $n = \sum_k n_K$: total de ejemplos de los clientes participantes
- E : número de épocas locales
- η : tasa de aprendizaje
- ℓ : función de pérdida

FedAvg es una generalización del SGD distribuido (*Distributed Stochastic Gradient Descent*, es una extensión del descenso de gradiente estocástico en la que el entrenamiento de un modelo se reparte entre varios nodos de cómputo que procesan datos en paralelo), adaptada al escenario federado, donde los datos están distribuidos y no se comparten. La idea es la siguiente:

1. *Inicialización global*: El servidor mantiene un modelo global común a todos los clientes.

2. *Selección de clientes*: En cada ronda solo participa una fracción de clientes, reflejando disponibilidad intermitente (por ejemplo, móviles conectados).

3. *Entrenamiento local*: Cada cliente entrena el modelo sobre sus propios datos, realizando varias épocas de SGD local. Este paso reduce drásticamente la comunicación, ya que no se envían gradientes en cada *minibatch*.

4. *Agregación ponderada*: El servidor promedia los modelos locales, ponderándolos por el número de datos de cada cliente. El resultado es el nuevo modelo global.