Requisitos y diseño de la arquitectura y casos de uso (final)

MLEDGE - Aprendizaje automático en la nube y en el borde

(Cloud and Edge Machine Learning)

Diciembre de 2023

Información sobre el entregable

Nombre del documento:

Requisitos y diseño de la arquitectura y casos de uso (final)

Versión actual: 1.0

Proyecto: MLEDGE - Aprendizaje automático en la nube y en el borde (Cloud and Edge Machine Learning)

Paquete de trabajo: P1 - Análisis de requisitos y diseño de la arquitectura y casos de uso

Tarea: A1.1: Requisitos y casos de uso

Entregable: E1.2 - M12 - Requisitos y diseño de la arquitectura y casos de uso (final)

Autores: Santiago Andrés (IMDEA)

Revisores: Nikolaos Laoutaris (IMDEA)

Historial de Versiones

Versión	Fecha	Resumen de modificaciones
Version 1.0	31-12-2023	Versión inicial del documento

Índice

1. Introducción

En diciembre de 2022 fue adjudicado a IMDEA Networks el proyecto “MLEDGE - Aprendizaje automático en la nube y en el borde (Cloud and Edge Machine Learning)” (REGAGE22e00052829516, en adelante el ‘Proyecto’ o MLEDGE) por parte del Ministerio de Asuntos Económicos y Transformación Digital del Gobierno de España, con fondos de la Unión Europea dentro del Plan de Recuperación, Transformación y Resiliencia (European Union - NextGenerationEU/PRTR). El proyecto tiene como objetivo habilitar un ecosistema próspero de servicios FL en el borde seguros y eficientes capaces de facilitar el uso de datos personales y B2B confidenciales para entrenar modelos de ML para consumidores mientras se protege la privacidad de los datos y de sus propietarios.

El presente documento se corresponde con el entregable E1.2 de título “Requisitos y diseño de la arquitectura y casos de uso (final)” y tiene como objetivos: la Investigación del estado del arte en áreas de interés para alcanzar los objetivos del proyecto, presentar un primer diseño de la arquitectura y definición de los requisitos para cada caso de uso, incluyendo la definición de métricas específicas de evaluación e indicadores clave de rendimiento (KPI).

La estructura del documento será la siguiente:

- En la sección 2 presentamos el problema que busca solucionar MLEDGE y las principales áreas de interés y tecnologías que se emplearán para ayudar en su objetivo.
- En la sección 3 se presenta un estado del arte a nivel tecnológico y científico detallado para cada una de las áreas de interés y tecnologías identificadas como relevantes para el proyecto.
- La sección 4 presenta algunas soluciones comerciales existentes en las áreas anteriormente
- La sección 5 desarrolla los requisitos de los diferentes módulos que se desarrollarán dentro del proyecto y de los casos de uso contemplados en su desarrollo.
- La sección 6 resume las conclusiones del análisis de requisitos y diseño realizado en este entregable y apunta a los siguientes pasos dentro del proyecto.

2. Definición del problema

En este apartado se presenta el problema que trata de resolver MLEDGE, comenzando por su contexto. Se identifican además las áreas de interés que van a ser objetivos científicos del proyecto, y para las cuales se va a realizar un profundo estudio sobre el estado del arte en los apartados posteriores.

2.1. Contexto

La toma de decisiones basada en datos e impulsada por algoritmos de *Aprendizaje Automático* o *Machine Learning (ML)* está cambiando la forma en que funcionan la sociedad y la economía, y está teniendo un profundo impacto positivo en nuestra vida cotidiana. De hecho, las aplicaciones de ML se están volviendo aún más ubicuas e integradas, a menudo de manera invisible, en nuestras actividades diarias, logrando un impacto directo en cosas como la forma en la que nos orientamos en una ciudad, el cómo decidimos qué comprar o dónde comer, mientras que al mismo tiempo nos podemos mantener a salvo de fraudes financieros, o disponemos de herramientas que nos recuerdan tomar los medicamentos o que nos sugieren nuevos hábitos personalizados para un estilo de vida más saludable.

Sin embargo, para que las soluciones basadas en ML sean efectivas en tales tareas, **a menudo se tienen que procesar datos cerca del usuario final**. Además, **dichos datos pueden ser privados y de naturaleza confidencial***.* El *Aprendizaje Distribuido* y, en particular, *Federado* (*FL: Federated Learning*) emerge como un paradigma líder dentro de la rama del ML satisfaciendo estas dos propiedades. FL ha crecido en paralelo con la expansión de la nube hacia el borde (*CloudEdge*) pero, curiosamente, ambos paradigmas se han desarrollado en su mayoría de forma independiente, a pesar de su paralelismo natural y sus posibles ganancias sinérgicas.

De hecho, es bien reconocido que una de las cargas de trabajo más exigentes para la computación en la nube moderna, incluidas sus extensiones hacia el borde, tiene que ver con ***el alojamiento de modelos de aprendizaje automático que operan con datos confidenciales de usuarios finales individuales u organizaciones administrativamente independientes que colaboran bajo diferentes suposiciones de confianza (de total a nula, y cualquier nivel intermedio).*** En este contexto, CloudEdge tiene dos grandes ventajas:

1. **Está cerca del usuario final**, ya sean individuos u organizaciones y, por lo tanto, puede ofrecer un bajo retraso para las aplicaciones, por ejemplo, de realidad aumentada impulsadas por ML, automóviles autónomos, aplicaciones de control relacionadas con IoT, aplicaciones para hogares inteligentes, etc.
2. **Puede mantener los datos confidenciales en el entorno local de los usuarios finales**, para que se procesen localmente sin transferirlos a nubes centralizadas u otros dominios, y evitando el procesamiento en los dispositivos del cliente que a menudo adolecen de importantes limitaciones en la capacidad de proceso.

Al mismo tiempo, CloudEdge afronta también ciertos desafíos:

1. **Capacidades de procesamiento** menores en los nodos de borde,
2. **Mayor sobrecarga de comunicación** en comparación con nubes centralizadas y,
3. **Múltiples organizaciones administrativas** colaborando bajo suposiciones de confianza no uniformes

El paradigma del *Aprendizaje Federado o Federated Learning (FL)* aprovecha las ventajas de CloudEdge. FL permite a diferentes clientes entrenar sus modelos localmente sin revelar datos locales que puedan ser confidenciales, permitiendo al mismo tiempo colaborar con otros clientes compartiendo las actualizaciones de sus respectivos modelos, las cuales se pueden combinar para construir un modelo global superior [Konečny16, McMahan17]. Además, lo hace cerca de los usuarios, de forma que estos se benefician del modelo entrenado para sus condiciones locales con una menor latencia que si se hiciera de forma centralizada en la nube.

Adicionalmente, **FL se adapta a los desafíos que impone el procesamiento en el borde de la nube**. La menor capacidad de procesamiento se soslaya con una distribución de las cargas de trabajo entre los diferentes nodos de borde, que de esta manera son capaces de entrenar modelos complejos usando sólo datos locales. Al intercambiar solo las actualizaciones de los modelos (gradientes) en lugar de datos sin procesar, FL reduce en gran medida los gastos generales de comunicación en comparación con otros paradigmas en los que todos los datos sin procesar deben enviarse a un nodo centralizado, o paradigmas distribuidos donde se intercambian datos sin procesar reales, como la computación *Peer-to-Peer (P2P)*. Al procesar etiquetas locales localmente sin revelarlas a partes remotas que no son de confianza, o que son de confianza parcial, FL resuelve los desafíos de seguridad y privacidad relacionados con el intercambio de datos sin procesar entre diferentes nodos de CloudEdge que pertenecen a diferentes entidades administrativas.

2.2. Objetivos

En el presente proyecto, **aprendizaje automático en la nube y en el borde (MLEDGE)**, trabajaremos para implementar FL como una capa intersectorial independiente pero optimizada sobre CloudEdge, y utilizaremos aplicaciones y datos del mundo real para demostrar que esta sinergia puede producir grandes beneficios para todos. MLEDGE tiene como objetivo **habilitar un ecosistema próspero de servicios FL en el borde seguros y eficientes** capaces de facilitar el uso de datos personales y B2B confidenciales para entrenar modelos de ML para consumidores mientras se protege la privacidad de los datos y de sus propietarios. Estudios recientes en el ámbito de la “Estrategia europea de datos” estimaron que la economía de los datos alcanzará un impacto de 827.000 millones de euros para los UE27 ya en 2025. Sin embargo, aún hoy en día las preocupaciones sobre la privacidad y la propiedad obstaculizan su pleno desarrollo. MLEDGE contribuirá decisivamente a aumentar estas proyecciones en el período 2025-2030.

Los **objetivos generales del proyecto** se pueden resumir en los siguientes:

1. Hacer del aprendizaje federado una funcionalidad accesible y de fácil uso en el borde mediante el desarrollo de una capa de software intermedio y componentes que escondan la complejidad del procesamiento y el intercambio de datos que supone.
2. Resolver problemas técnicos asociados al aprendizaje federado en el borde de la nube.
3. Demostrar esta funcionalidad en casos de uso que reflejen problemas reales de la industria que pueden ser resueltos con estas tecnologías.
4. Explotar los resultados del proyecto involucrando a agentes externos y comunicar los hallazgos al público potencial en general.

Para alcanzar estos objetivos, se han catalogado una serie de **objetivos técnicos específicos** del proyecto que se resumen en los siguientes:

1. Diseñar un marco de desarrollo de servicios de aprendizaje federado (FLaaS) en el borde de la nube y componentes que ayuden a popularizar este tipo de servicios
2. Diseñar y desarrollar soluciones de seguridad (FedSecure) contra ataques de envenenamiento o inferencia lanzados desde servidores de borde rebeldes y/o nodos de agregación “honestos pero curiosos”.
3. Gestionar los desafíos de la portabilidad de datos en el borde de la red (DataEdge)
4. Crear un esquema de marca de agua (FedWM) para proteger contra la redistribución de los datos o metadatos que se intercambien entre servidores en el borde en el marco del FLaaS.
5. Crear una capa de lógica económica y de negocio (FLaaS Manager) que implemente una distribución justa de costes e ingresos entre las partes cuando colaboren en el entrenamiento de modelos de ML.
6. Soporte a DevOps y al desarrollo continuo de servicios de aprendizaje automático en la nube, optimizando los costes mediante su monitoreo, predicción y asignación inteligente y energéticamente eficiente de los trabajos de computación.

Finalmente, es uno de los objetivos básicos del proyecto diseñar, implementar y hacer públicos demostradores que trabajen con datos sensibles de individuos, y alimenten modelos de aprendizaje automático en el borde de la red en áreas clave de la economía tradicional y de la economía digital como FinTech, identidad, salud, transporte, control de acceso, etc. A tal fin, en la primera parte del proyecto se han diseñado pliegos técnicos y administrativos con el objeto de encargar a empresas externas el desarrollo de la plataforma FLaaS y el monitoreo de costes de computación, así como el diseño e implementación de casos de uso de negocio reales que se beneficien del aprendizaje distribuido en el borde de la nube.

2.3. Modelo conceptual

Para allanar el camino a la adopción del FL en el borde de la red para un creciente número de aplicaciones que empleen modelos de ML, **MLEDGE persigue el desarrollo de técnicas, librerías y componentes que permitan poner en marcha más ágilmente estos servicios.** Por tanto, el proyecto desarrollará tecnologías para habilitar servicios de FL en infraestructuras de nube híbrida, con diversos clientes MLEDGE contribuyendo con modelos entrenados con datos locales a un modelo central gestionado por un agregador MLEDGE. También se investigarán arquitecturas distribuidas que prescindan de este agregador y permitan el entrenamiento distribuido de modelo mediante técnicas P2P (peer-to-peer) entre clientes MLEDGE. Se dotará al sistema de capas de seguridad que protejan a los usuarios ante diversos tipos de ataques al proceso de aprendizaje federado (envenenamiento, inferencia, etc.), así como de sistemas que protejan mediante marcas de agua contra la redistribución no autorizada de modelos o de datos intercambiados por los nodos MLEDGE para uso interno. Adicionalmente, la capa de gestión de transacciones proporcionará funciones y componentes que permitan una adecuada distribución de costos y beneficios del entrenamiento federado entre las partes involucradas creando un modelo económico atractivo para todas ellas.

Por último, se demostrarán las funcionalidades de los componentes desarrollados mediante pruebas de concepto (TRL 4, tecnología validada en laboratorio), prototipos que serán la semilla para posteriores desarrollos y adopción en el mercado. Estos casos de uso representan el principal activo del proyecto ya que permitirá validar de forma experimental los resultados y hallazgos del proyecto. En particular, estos casos de uso emplearán de forma activa los componentes de MLEDGE a fin de implementar casos prácticos de aprendizaje federado que resuelvan problemas reales. Para ello, se desarrollarán en estrecha colaboración con la industria, de forma que se asegure la explotación de los productos del proyecto para resolver problemas que estas enfrentan en el mercado. En los casos de uso, diferentes nodos de borde colaborarán para entrenar un modelo común mediante técnicas de aprendizaje federado, preservando la privacidad de los datos empleados para entrenar el modelo. De esta manera, los casos de uso servirán para probar los componentes de software desarrollados y depurar la "usabilidad" de los mismos en entorno pre-comercial, tanto para los casos de agregador centralizado en la nube como de agregador/cliente federado en el borde de la nube (FLaaS P2P).

Los casos de uso de MLEDGE incluirán:

1) **Caso de uso industrial relacionado con la economía tradicional**

Existen multitud de problemas en la economía tradicional en los que la tecnología desarrollada por MLEDGE puede ayudar a mejorar la eficiencia de los procesos o la toma de decisiones de diferentes actores de la cadena de valor. Por ejemplo, en el caso del sector sanitario, diferentes hospitales se podrían beneficiar de datos de pacientes de otros hospitales a la hora de entrenar modelos para la toma de decisiones sobre los tratamientos a aplicar o para entrenar modelos de diagnóstico o de predicción de la evolución de los pacientes. Sin embargo, por motivos regulatorios o de confidencialidad de los datos, los hospitales no pueden compartirlos, pero sí podrían emplear estos datos para entrenar un modelo "común" de forma colaborativa sin revelarlos. Otro ejemplo, en la industria a menudo es necesario usar modelos de estimación de costes o presupuestación, pero los diferentes agentes que intervienen en la cadena de valor no desean revelar sus "costes" o "precios" de los servicios pero sí participarían en el entrenamiento federado de modelos que permitan disponer de una herramienta de presupuestación de aplicación para toda la industria.

El objetivo de este caso de uso es demostrar la aplicación de las tecnologías de MLEDGE para resolver un problema de aplicación en alguna industria de la economía tradicional (construcción, finanzas, salud, etc.). El caso de uso será desarrollado por una empresa y empleará los componentes desarrollados en MLEDGE para mejorar sus procesos, o bien para mejorar los procesos de toma de decisiones basadas en datos o modelos provenientes de FL. Por tanto, el caso de uso deberá demostrar los beneficios del proyecto para entrenar un modelo de ML sin exponer los datos de las partes implicadas, e idealmente se usará para la toma de decisiones en tiempo real.

2) **Caso de uso relacionado con la economía digital**

De igual forma existen diversos casos de uso de empresas en la economía digital en los que diversas partes desean contribuir datos para el entrenamiento de modelos de aprendizaje automático. Por ejemplo, dentro del terreno de la salud digital se podría pensar en explotar la información que proporcionan dispositivos móviles o tecnologías vestibles (como pulseras de actividad, en inglés *wearables*) para vigilar la evolución y adaptar el tratamiento o las necesidades de atención médica a pacientes. Otro caso de uso es el empleo de datos de los usuarios para entrenar modelos de publicidad digital mediante técnicas de FL. El objetivo de este caso de uso es demostrar la aplicación de las

tecnologías desarrolladas en el seno de MLEDGE para resolver problemas de negocio en empresas relacionadas con la economía digital, los cuales emplearán el FL para entrenar un modelo de ML sin exponer los datos que se empleen para ello.

3) Caso de uso de optimización de infraestructuras CloudEdge

Un tercer caso de uso estará relacionado con el uso de aprendizaje federado para algoritmos de optimización de la propia infraestructura de CloudEdge en la que correrán los servicios. Dicha optimización empleará algoritmos de ML que se entrenarán de forma federada por los diferentes nodos que participen en la provisión de servicios FLaaS. De esta forma, esta funcionalidad es per se una funcionalidad clave y, a la vez, un caso de uso de los servicios proporcionados por MLEDGE.

2.4. Objetivos científicos

En base al modelo conceptual definido en el apartado anterior, se han identificado una serie de objetivos científicos en el proyecto que serán objeto de investigación y desarrollo durante el mismo. Los componentes desarrollados para avanzar en el estado del arte de estos desafíos se probarán en los demostradores y casos de uso del proyecto. Los principales objetivos científicos de MLEDGE son los siguientes:

1. OBJETIVO 1: DevOps y desarrollo continuo para servicios de aprendizaje automático (FLaaS) que se ejecutan en borde de la nube
2. OBJETIVO 2: Uso eficiente de FL en nubes híbridas y protección contra ataques
3. OBJETIVO 3: Protección de datos sensibles o confidenciales que sean intercambiados entre dominios administrativos en la nube y el borde de la nube
4. OBJETIVO 4: Equidad en términos de distribución de costos y ganancias cuando la computación en el borde se usa para entrenar de forma colaborativa modelos de ML
5. OBJETIVO 5: Gestión de los desafíos de portabilidad de datos en el borde
6. OBJETIVO 6: Gestión, optimización y monitorización de infraestructura cloud

En la siguiente sección se presenta el estado del arte al inicio del proyecto de cada uno de los objetivos científicos identificados.

3. Estado del Arte

Gracias al desarrollo de la inteligencia artificial (IA) y a la adopción masiva de algoritmos de aprendizaje automático (en inglés, ML), los datos se han convertido en un factor económico clave, comparable en importancia al capital, la tierra o la mano de obra. Sin embargo, debido a las peculiares características de este bien económico (replicable a coste nulo, reutilizable, no rival, cuyo valor es combinatorio y depende en gran medida del propósito y contexto de uso [Coyle20]) buena parte de las compañías que controlan los mismos son a menudo renuentes a compartirlos y la mayor parte de la información reside actualmente en silos corporativos.

Ya existía históricamente un mercado de datos en Internet y múltiples compañías que ofrecen servicios para diferentes verticales y datos de diferentes categorías en Internet. A estos proveedores de datos o servicios se les están sumando mercados de datos cuya misión es mediar entre proveedores y potenciales compradores de datos y dinamizar el mercado. Los desafíos de estos mercados de datos son diversos e incluyen [Andres22]:

- La estandarización del intercambio de datos para poder lidiar con la enorme fragmentación existente en el mercado.
- Los procesos de fijación de precios de los datos y la falta de transparencia del mercado
- La dificultad de establecer qué datos o fuentes de datos son óptimas para cada comprador y caso de uso
- Lidiar con los problemas de la propiedad de los datos y la posibilidad de que estos puedan ser replicados o revendidos
- Compensar adecuadamente a los proveedores e individuos que contribuyen información en una transacción

En este contexto, existe una clara tendencia hacia la federación o distribución de estas plataformas de intercambio de datos [Giaretta22], las cuales pueden además aprovechar las crecientes capacidades de computación en el borde de la nube. Además, en el ámbito de los datos para entrenar modelos de AI/ML existe un debate sobre si es más conveniente vender datos o si las plataformas deben vender “modelos entrenados” en su lugar (lo que implica añadir una capacidad de cómputo al puro dato y evita su intercambio “en crudo”). En esta dirección, de plataformas centralizadas que toman el control de los datos se está caminando a mercados distribuidos que simplemente gestionan el intercambio directo entre las partes (Ocean Protocol, Settlemint) a menudo apoyándose en cadenas de bloques y criptomonedas para las transacciones, y en esquemas de aprendizaje federado [McMahan17] para procesar los datos en la ubicación en que residen (Nokia DM, Acuratio).

En paralelo, la computación en el borde de la red se observa como una oportunidad de negocio por parte de los operadores de telecomunicaciones. Disponer de capacidades de computación a diferentes niveles permite adaptar los servicios a las necesidades de los casos de uso en materia de latencia, precisión y carga ofrecida a la red. La granularidad de las redes de los principales operadores les permite aportar infraestructuras de computación en diferentes niveles de la red más cerca o más lejos del cliente en función de las necesidades de la aplicación específicas en términos de latencia / capacidad de cómputo. Las principales compañías de Internet ya disponen de stack y servidores para computación en el borde y existen ya muchos acuerdos a nivel mundial [STLP23] con operadores de telecomunicaciones y de infraestructuras para un mercado que alcanzará los 245 miles de millones de € en el año 2027 [LightR23].

En las siguientes secciones, se repasa el estado del arte y la situación de la computación en el borde de la red, base que cimenta los desarrollos tecnológicos del presente proyecto.

3.1. Cloud, Edge and Fog Computing

Ante la necesidad de tratar y analizar la información que generamos en una economía crecientemente digital surgen tres tipos de procesamiento de esa información: ***cloud computing***, ***fog computing*** y ***edge computing***.

El *cloud computing* o la **computación en la nube** es un conjunto de tecnologías que permiten el acceso remoto, generalmente por medio de Internet, a software o servicios, a recursos de almacenamiento y procesamiento de datos. Lo podemos visualizar como un **servidor remoto** al que estamos mandando los datos que generan nuestros procesos, y esos datos pueden ser tratados en esos servidores mediante *software* o algoritmos de inteligencia artificial.

Fog computing o **computación en la niebla** se puede definir como una tecnología *cloud* por la cual los datos que generan los dispositivos y procesos no se suben directamente a la nube, sino que se preparan primero en **centros de datos descentralizados** más pequeños y cercanos. Hablamos de *fog nodes*, nodos de procesamiento previos a la nube que actúan de instancia mediadora entre la nube y los distintos dispositivos de IoT, sensores, máquinas y robots. Podríamos considerarla como un red local con menor latencia que una red *cloud*, pensada para actuar de forma inmediata y rápida.

Edge computing o **computación en el borde** es el caso extremo de computación o procesamiento local. El sensor, dispositivo o aparato que está recogiendo los datos tiene capacidad para procesarlos y almacenarlos en tiempo real y poner en marcha una respuesta si fuese necesario. En *edge computing* **los datos se procesan en el dispositivo o sensor en sí** sin ser transferidos a ninguna parte.

La mayoría de las aplicaciones de Internet se sirven ahora desde servidores ubicados en centros de datos. Según algunos estudios, el procesamiento en el borde puede ahorrar al menos entre un 20 y un 30% de energía en comparación con el procesamiento centralizado en grandes centros de datos [Valancius09]. Esto se debe a que los grandes centros de datos son propensos a: i) sobredimensionar recursos para mantener la demanda en las horas punta, ii) a requerir una elevada inversión en refrigeración para disipar adecuadamente el calor, y iii) a necesitar más equipos de red para conectarse a los usuarios finales distantes y, por tanto, un mayor consumo de energía [Valancius09]. Para reducir el impacto del sobreaprovisionamiento, los centros de datos pueden programar las cargas de trabajo para ocuparse de las aplicaciones que no requieren respuesta en tiempo real, como las copias de seguridad, la propagación de actualizaciones y la migración de datos sólo cuando los recursos están infrautilizados [Laoutaris11].

Con el incremento de la demanda de aplicaciones y servicios con menor latencia, la tendencia es a usar **recursos de computación en el borde de la nube** (**Cloud Edge computing**). Se trata de pequeños nodos de procesamiento en el extremo de la red que evitan que el tráfico tenga que subir a nodos más alejados del usuario, y que por otro lado proporcionan mayor capacidad de proceso que la computación en el extremo del usuario.

En paralelo a la capacidad de computación, los modelos de aprendizaje automático también se están adaptando a los cambios en las infraestructuras que los soportan. A continuación, presentamos un resumen del estado del arte en aprendizaje automático federado.

3.2. Aprendizaje Federado

Los métodos de aprendizaje federado buscan el entrenamiento de un modelo de ML de forma colaborativa entre un conjunto de clientes que disponen de los datos necesarios para entrenarlo, evitando el intercambio de los datos locales entre las partes y comunicando en su lugar los cambios en el modelo que han tenido lugar tras realizar un entrenamiento local del modelo. Como resultado, cada una de las partes obtiene un modelo de ML, bien el mismo o diferente. En general, se cumple que el rendimiento del modelo conjunto medido en base a una métrica concreta (ej. la exactitud de sus predicciones medidas contra un conjunto de datos de prueba), es mayor que el rendimiento de los modelos locales de cada una de las partes.

Los datos “crudos” nunca abandonan el cliente que los controla, y lo que se intercambian son los cambios en los modelos resultados de los entrenamientos locales. Este enfoque se contrapone al más tradicional entrenamiento centralizado en el que los datos de las diferentes fuentes se suelen llevar a un único punto de procesamiento en el que se utilizan para entrenar este modelo centralizado.

3.2.1. Un sencillo ejemplo

Vamos a explicarlo con un ejemplo simple. Imaginemos un simple modelo que calcula (o aprende) la edad media de la población de un país. En un enfoque tradicional, toda la información de la fecha de nacimiento de los individuos en una base de datos común (por ejemplo, el Registro Civil) y se realiza una consulta para calcular la edad media. Para ello, todos los registros de todos los puntos del país han compartido previamente la información de cada uno de los individuos con la base de datos central sobre la que se realiza este cálculo.

¿Y si no queremos revelar la información de la edad de los individuos a este gestor central? Un posible enfoque de aprendizaje federado sería que cada distrito del país almacenara la información de los nacidos en ese distrito y actuaran como clientes, y uno de ellos actuara como gestor central. Este gestor central compartiría la información del modelo global (paso 1), que en este caso básicamente sería el número de individuos que controla y su cálculo local de la edad media, con todos los individuos cuyos datos controla. En respuesta, los nodos locales responden con el número de nodos que controlan y su media (paso 2). El nodo central consolida esta información y calcula el resultado con todos los nodos (paso 3), además puede comunicar de nuevo el resultado (o el nuevo modelo global) a todos los nodos para que dispongan del mismo. En ningún caso los datos sensibles (la edad de los individuos) abandonan los clientes que poseen esta información.

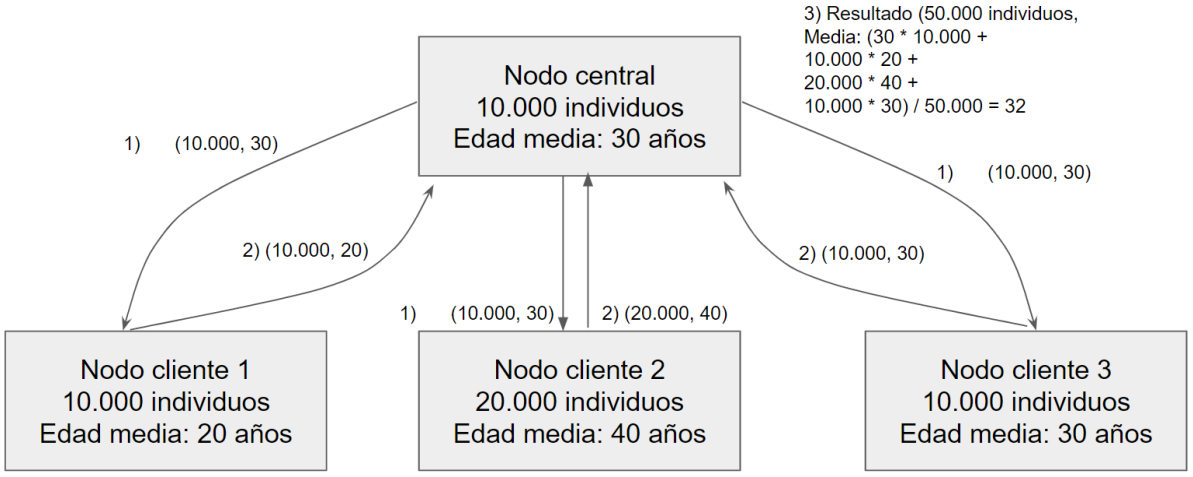

Figura 1. Ejemplo simple de aprendizaje federado

De nuevo, el ejemplo es simple y no pretende capturar toda la complejidad del aprendizaje federado, sino dar a entender el concepto detrás de este término. Los modelos de aprendizaje federado a menudo son redes neuronales con miles de parámetros, los insumos cambian con el tiempo, los modelos pueden requerir varias rondas de entrenamiento para alcanzar una eficiencia óptima, etc.

3.2.2. Estado del arte

La primera referencia al aprendizaje federado de redes neuronales profundas (Deep Neural Networks o DNN) data del año 2016-17 [McMahan17], aunque el número de trabajos posteriores se ha incrementado significativamente. Existen diversos artículos que resumen el estado del arte en materia de aprendizaje federado desde diferentes perspectivas [Kairouz21, Li21_2, Li19].

Desde un punto de vista de sistemas, los componentes de un sistema de aprendizaje federado son básicamente tres [Li21_2]:

- **Clientes o partes** que son dueños de los datos y beneficiarios del aprendizaje federado, para los cuales hay que definir su número, su capacidad de hardware y comunicación, su estabilidad y la distribución de datos entre los mismos.
- **Gestor**, que también puede actuar como cliente, y que generalmente es un servidor estable y confiable que realiza el entrenamiento del modelo global y gestiona las comunicaciones entre las partes.
- **Marco de computación y comunicación** que permitirá i) entrenar el modelo localmente por parte de cada una de los clientes y agregar o entrenar el modelo global por parte del gestor y ii) comunicar los resultados (pesos de los modelos) de los entrenamientos locales y el modelo global entre las partes y el gestor.

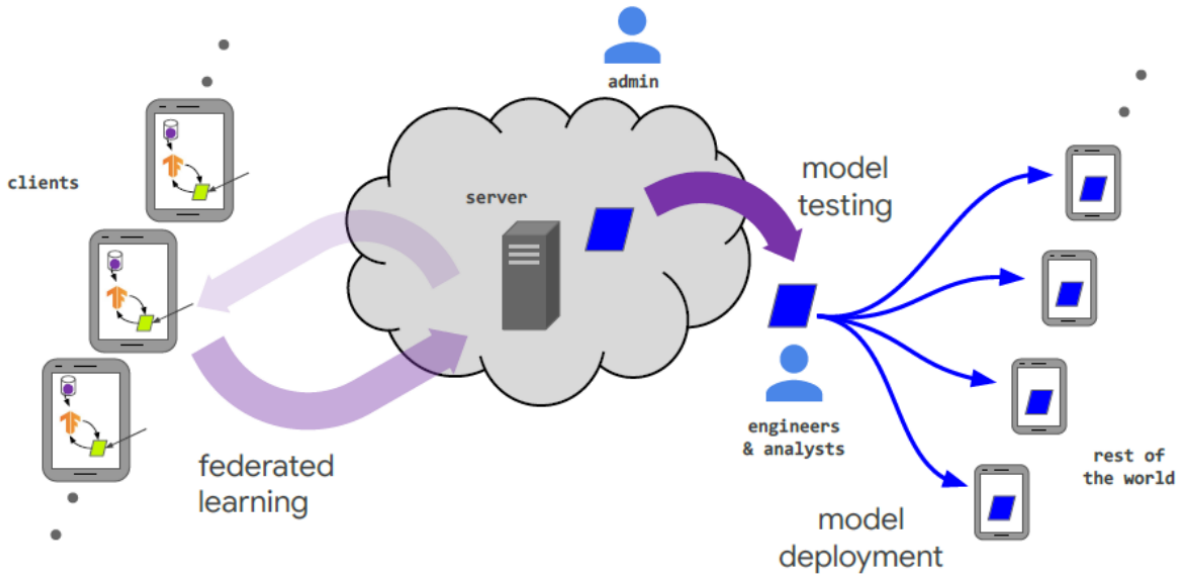

Figura 1. Ejemplo de esquema de aprendizaje federado [Li21_2]

El aprendizaje federado está relacionado con el **aprendizaje distribuido** empleado en centros de datos para distribuir el trabajo de entrenamiento entre diferentes máquinas. Sin embargo existen muchas diferencias entre ambos, ya que el aprendizaje distribuido está orientado a entrenar un solo modelo en diferentes nodos de cómputo dentro de un centro de datos. Por ello el cuello de botella del aprendizaje distribuido está más bien en la computación y no en la comunicación. El gestor distribuye entre estos nodos tanto el modelo como los datos y por tanto, los datos de entrenamiento son repartidos arbitrariamente, e incluso volver a distribuirlos de forma diferente durante el proceso de entrenamiento [Kairouz21].

El aprendizaje en el borde de la red se encuentra de alguna manera a mitad de camino entre el aprendizaje federado y el aprendizaje distribuido. El número de clientes no es tan alto como los modelos federados pero generalmente será mayor que en el caso del aprendizaje distribuido. Igualmente, aunque la capacidad de comunicación de los nodos en el borde de la red es mayor que la de los dispositivos móviles del aprendizaje federado, no es comparable a la hiperconectividad existente en los servidores de los centros de datos. A diferencia del aprendizaje distribuido, en el aprendizaje federado en el borde de la red los datos no abandonan los clientes para proteger la privacidad de los clientes.

Desde un punto de vista del uso, FL ha demostrado su valor en una gran cantidad de aplicaciones del mundo real, que van desde la informática o computación móvil [Feng20, Hard18, Yu20] hasta aplicaciones médicas del campo de la salud [Brisimi18, Huang19, Lee18].

3.2.3. Clasificación de modelos de aprendizaje federado

Se puede realizar una clasificación de los modelos y métodos de aprendizaje en base a diversos criterios[Li21_2]:

- **En base a la partición de datos** se habla de *aprendizaje federado horizontal, vertical e híbrido*. En el primero, los clientes comparten un mismo espacio de datos

pero cada uno aporta diferentes muestras. En el aprendizaje federado vertical, los clientes tienen diferentes espacios de datos (diferentes campos de datos), pero comparten en su totalidad o en parte los individuos de muestra, los cuales se emplean para entrenar el modelo empleando tecnologías de encriptación. Los modelos híbridos combinan partición de datos horizontales y verticales.

- **En base al modelo objetivo**, el aprendizaje federado puede estar orientado a entrenar una D*NN, árboles de decisión* (Gradient Boosting)*, modelos lineales* (como los support vector machines SVM) o *modelos de ensamblado*.
- **En base a la arquitectura de comunicación**, el aprendizaje federado puede ser *centralizado o distribuido*. La arquitectura centralizada opera con un solo nodo "gestor" que centraliza y agrega los resultados del entrenamiento de los clientes, Las arquitecturas descentralizadas operan con varios gestores y son menos utilizadas pero más resilientes. Su diseño es complicado y se han utilizado técnicas de P2P, grafos de clientes y blockchain.

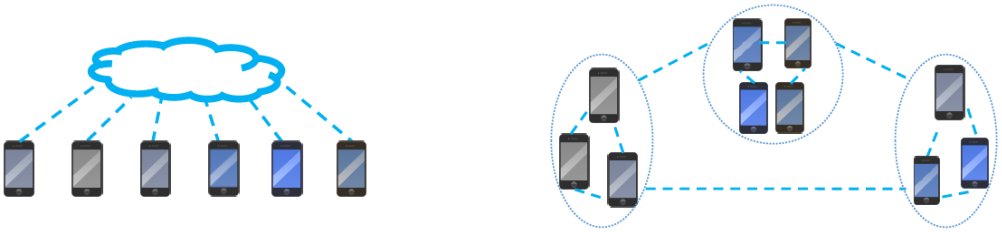

Figura 2. Aprendizaje federado centralizado (izq.) y distribuido (der.) [Li21_2]

- **En base a la escala de federación** se habla de *aprendizaje federado entre dispositivos o entre silos* de datos. En el primero hay un gran número de clientes que aportan un conjunto limitado de datos y tienen limitaciones en cuanto a la capacidad de proceso. En el aprendizaje federado entre silos, los clientes son pocos centros de datos u organizaciones que disponen de gran cantidad de datos y capacidad de cómputo.
- **En base a la motivación**, se puede hablar de *aprendizaje federado movido por la regulación* (necesidad de entrenar un modelo pero imposibilidad de compartir datos, por ejemplo dentro de una organización) y de aprendizaje federado movidos por incentivos, bien sea obtener un mejor modelo o bien sea incentivos económicos.

En el problema particular que nos ocupa, los clientes son operadores de redes o de infraestructuras que colaboran para obtener una mejor visión de la situación y el estado de la red. Por tanto, es de esperar que haya un cierto solapamiento de los espacios de datos y se puede asumir que existe buena capacidad de comunicación entre los clientes del aprendizaje distribuido y que en este caso serían unos pocos clientes los que contribuirían a entrenar el modelo. En relación a la motivación, en este caso el incentivo a participar es mejorar la gestión de la red y los servicios que se ofrecen entre las partes, por lo que se entiende que están interesados en colaborar en un modelo global común. No obstante, asumiremos que los participantes tienen interés en mantener la confidencialidad de sus datos y de los datos personales de sus clientes a la hora de entrenar los modelos, por lo que el esquema de aprendizaje federado les permite no compartir estos sino las actualizaciones del modelo global como una de las medidas para preservar la privacidad de la información.

3.2.4. Desafíos en modelos de aprendizaje federado

Diferentes autores han señalado los diferentes desafíos del aprendizaje federado [Li19]. Estos incluyen la comunicación, la heterogeneidad de los sistemas, la heterogeneidad de los datos, el adecuado diseño de los incentivos para formar parte de la federación, y aspectos relativos a la privacidad. En esta sección desarrollamos mínimamente los tres primeros, e introducimos los problemas de privacidad que plantea el aprendizaje federado, los cuales desarrollaremos en mayor profundidad en el siguiente apartado dado que es el principal objetivo del proyecto.

A diferencia de los sistemas centralizados, el aprendizaje federado asume que tanto los datos como la capacidad de computación están distribuidos por diferentes clientes. Si bien este no es el mayor desafío cuando los clientes residen en servidores de centros de datos hipercomunicados, **optimizar la capacidad requerida de comunicación** y adaptarla al problema que se trata de resolver es un aspecto habitual en esquemas de aprendizaje federado. Para ello, se han propuesto soluciones que trabajan en diferentes direcciones:

- Balancear el compromiso entre comunicación y computación en los nodos de cliente, realizando varias actualizaciones del modelo antes de comunicar los pesos al gestor [Reddi16, Zhang15]
- Reducir el tamaño de la información intercambiada mediante diferentes técnicas de compresión, muestreo y cuantización de los cambios [Wang18, Zhang17]
- Usando topologías descentralizadas de aprendizaje federado (ver figura 5) para aliviar las necesidades de comunicación del gestor [He18]. Algunos trabajos señalan también que las topologías descentralizadas pueden mejorar a las centralizadas en el caso de aprendizaje federado o distribuido en centros de datos [Lian17]

Figura 3. Topologías centralizadas (izq.) y descentralizadas (der.) de aprendizaje federado [Li19]

En algunos escenarios, los modelos de aprendizaje federado emplean **clientes heterogéneos**, es decir, clientes con diversas capacidades de cómputo y comunicación, lo que complica la tarea del gestor. Por ejemplo, algunos clientes pueden no responder o responder tarde y retrasar la actualización del modelo. Algunas de las soluciones que se han propuesto incluyen la comunicación asíncrona para evitar depender de clientes rezagados [Dai15, Ho13, Zinkevich10], el muestreo de los clientes [He18] o mecanismos de tolerancia a los fallos de los clientes [Bonawitz19, Li18]. Por el contrario, otros enfoques buscan favorecer clientes aparentemente “adversarios” [Mohri19].

Relacionado con lo anterior, los modelos de aprendizaje federado pueden sufrir de problemas de **heterogeneidad estadística de los datos**, con clientes que aporten visiones e información muy diferentes, lo cual puede afectar a la convergencia de los modelos.. Este problema ha sido ampliamente estudiado en el aprendizaje automático, y algunas de estas

técnicas se han extendido al aprendizaje federado. Por ejemplo, se plantean esquemas de aprendizaje de diferentes modelos separados pero relacionados (aprendizaje multitarea) [Smith17], en lugar de aprender una modelo centralizado. Muchos trabajos realizan pruebas de rendimiento y convergencia en entornos de datos en ausencia de la hipótesis de datos independientes e idénticamente distribuidos. FedProx extiende el enfoque de FedAvg para estos entornos heterogéneos, [Li18] y asegura la convergencia en entornos heterogéneos, y otros autores han propuesto diferentes heurísticas que usan evitar el sesgo natural del aprendizaje automático a clientes of grupos de clientes homogéneos que aportan mayor información. Otro ejemplo es q-Fair Federated Learning (q-FFL) que modifica FedAvg para introducir un sesgo hacia el uso de clientes que aportan datos de mayor "valor" [Li20], al contrario de otros enfoques que buscan favorecer clientes aparentemente "adversarios" para lograr optimizar diferentes distribuciones en los clientes y evitar los sesgos hacia determinados clientes que ofrecen los métodos tradicionales [Mohri19].

El aprendizaje federado se realiza predominantemente con una topología en estrella. Sin embargo existen otras alternativas que se explorarán en este proyecto y que tienen que ver con el aprendizaje distribuido de diferentes modelos, como se muestra en la siguiente figura:

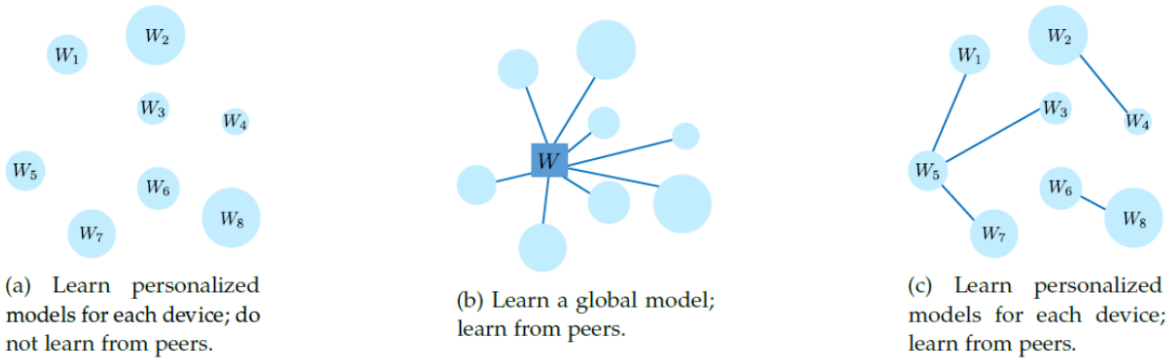

Figura 4. Alternativas al aprendizaje centralizado de un único modelo global [Li19]

Usualmente todos los nodos contribuyen al aprendizaje de un modelo global W que en agregador comunica a los clientes como se representa en la subfigura central. Esto contrasta con el aprendizaje automático tradicional donde cada nodo aprende su propio modelo (subfigura izquierda). Algunos investigadores están experimentando con topologías distribuidas donde los nodos entrenan modelos locales personalizados que responden a un único esquema común, pero colaboran con algunos nodos vecinos en su entrenamiento (subfigura derecha). Este tipo de topologías descentralizadas se ha empleado con éxito en aprendizaje distribuido en centros de datos [Ho13, Lian17], y teóricamente podría ser más rápido que el entrenamiento centralizado en condiciones de bajo ancho de banda y alta latencia. Otros autores han experimentado con topologías jerárquicas con el objetivo de aliviar la carga del servidor central, con dos niveles de servidores (Edge y cloud) que puede ser útil para el tipo de redes del proyecto MLEDGE [Lin18, Liu19]. Estas topologías centralizadas enfrentan desafíos adicionales como por ejemplo el diseño de la topología óptima.

Si bien uno de los principales objetivos del aprendizaje federado es mantener los datos en los clientes a nivel local, se ha demostrado que compartir las actualizaciones del modelo con el gestor puede revelar información sensible [Bhowmick18, Carlini18, Melis19]. Por tanto, **la privacidad de los clientes** resulta también un desafío relevante en modelos de

aprendizaje federado. Algunos autores han alertado de que este tipo de ataques también pueden ser aplicados en esquemas de aprendizaje federado descentralizado [Pasquini23]. Es por ello que las tecnologías de preservación de la privacidad como la privacidad diferencial o la anonimización, son importantes para la transmisión de los resultados del modelo.

3.2.5. Aprendizaje federado en el borde de la nube

De alguna forma, el aprendizaje federado en el borde de la nube se encuentra en un punto intermedio entre el aprendizaje federado que involucra a los dispositivos de los usuarios finales y el aprendizaje centralizado en el cloud. Por tanto, hereda muchos de los desafíos que tiene el aprendizaje federado, aunque se puede asumir que los nodos en el borde de la red son menos numerosos y disponen de mucha mayor y más homogénea capacidad de cómputo que los dispositivos de los clientes. De alguna forma, este modelo de entrenamiento federador está jerarquizado en dos niveles: usuario final - borde de la red, borde de la red - modelo global [Lin18, Liu19]. Los nodos del borde de la red pueden agregar la información de los usuarios finales que dependen de los mismos, y usarla para mejorar el modelo global o bien para mejorar modelos locales en cooperación con otros nodos.

3.2.6. Conclusión

El aprendizaje federado tiene ventajas y plantea significativos desafíos de diseño. Como principales ventajas cabe señalar i) que permite proteger la privacidad de los datos y reducir el riesgo de brechas de seguridad, ii) que permite reducir el tráfico en la red al reducir el trasiego de datos y iii) que distribuye el conocimiento entre diferentes nodos. Es por ello que se está pensando en utilizar en sectores y casos de uso donde la privacidad de los datos es prioritaria, como en ciencias de la salud. Además, ya se está utilizando en aplicaciones prácticas que usamos en el día a día como en el texto predictivo de los móviles o en aplicaciones de reconocimiento facial.

Como principales desafíos, el aprendizaje federado puede adolecer de ineficiencias en la comunicación, involucra diferentes tipos de dispositivos, tiene limitaciones a la hora de limpiar la información de entrenamiento y filtrar clientes “deshonestor”, y debe garantizar los incentivos de las diferentes partes implicadas en el entrenamiento, especialmente cuando estas no están especialmente interesadas en el modelo final y tan solo desean aportar datos. Adicionalmente, diversos trabajos han apuntado a potenciales brechas de privacidad si el gestor es capaz “inferir” la información de los clientes. En el ejemplo anterior, si los clientes tienen información de un solo individuo entonces en gestor es capaz de inferir esta edad con los datos que le entrega como resultado del entrenamiento del modelo.

Todos estos desafíos se pueden gestionar con un adecuado diseño. Los diferentes objetivos científicos del proyecto MLEDGE tienen como objetivo contribuir a superar estos desafíos y mejorar el estado del arte en cada uno de ellos. Una vez introducido el estado del arte de algunos de los ingredientes básicos de MLEDGE, en las siguientes secciones profundizamos sobre el estado del arte de los objetivos científicos particulares del proyecto.

3.3. Estado del arte de los objetivos científicos

3.3.1. DevOps y desarrollo continuo para servicios de aprendizaje automático (FLaaS) en el borde de la nube

El FLaaS surge como una evolución de los servicios de aprendizaje automático como servicio (MLaaS) que ofrecen diversas plataformas cloud como AWS, Google Cloud o Azure. En todas ellas, los proveedores de datos deben subir su información al proveedor cloud para que se entrenen y se exploten los modelos. FLaaS trata de extender el concepto del "as a service" al aprendizaje federado (FL).

Algunos diseños previos de la comunidad científica han identificado y tratado de solucionar algunos desafíos que comporta el FLaaS desde el punto de vista del operador de infraestructuras de telecomunicaciones [Kourtellis20], en concreto:

- Habilitar el modelado colaborativo entre aplicaciones de terceras partes y nubes híbridas de diferentes proveedores
- Realizar una gestión efectiva de los permisos y la privacidad de los datos
- Aprovechar las ventajas de las redes de telecomunicaciones distribuidas

Si bien existen algunas implementaciones "comerciales" de FLaaS y entrenamiento distribuido como servicio, se trata de implementaciones propietarias que no permiten la interconexión entre las mismas. En el seno del proyecto europeo PIMCITY, se realizó una primera prueba de concepto de FLaaS [Katevas22] que se enfocaba en entrenamiento federado por parte de aplicaciones móviles (FLaaS-enabled), y permitía configurar y gestionar tareas de aprendizaje federado, así como monitorizar la salud del sistema y del modelo.

3.3.2. Portabilidad de datos

Europa está apostando por estándares como los International Data Spaces [IDSA22] (Espacios Internacionales de Datos o IDS), que define espacios seguros y confiables de intercambio de datos, asegurando la soberanía de sus propietarios y proveedores, y ontologías para la descripción de los mismos, o el proyecto Gaia-X [GaiaX], que agrega una serie de servicios federados y un ecosistema de federación de servicios de infraestructuras. Dichas iniciativas, buscan flexibilizar el intercambio de datos y la provisión de en aras de habilitar casos de uso que requerirán la colaboración de diferentes agentes.

Compartir recursos de datos entre distintas organizaciones, investigadores, gobiernos y ciudadanos requiere el suministro de metadatos. Esto es independiente de que los datos sean abiertos o no. En esta dirección, se ha venido trabajando en la ontología Data Catalog Vocabulary (DCAT) que es un vocabulario para publicar catálogos de datos en la Web, desarrollado originalmente en el contexto de catálogos de datos gubernamentales como data.gov y data.gov.uk, pero también es aplicable y se ha utilizado en otros contextos.

DCAT proporciona clases y propiedades RDF que permiten describir conjuntos de datos y servicios de datos e incluirlos en un catálogo. El uso de un modelo y un vocabulario estándar facilita el consumo y la agregación de metadatos de múltiples catálogos, lo que puede aumentar las posibilidades de encontrar conjuntos y servicios de datos, y permitir la búsqueda federada de conjuntos de datos en catálogos de varios sitios.

Los datos descritos en un catálogo pueden presentarse en muchos formatos, desde hojas de cálculo hasta formatos especializados, pasando por XML y RDF. DCAT no hace ninguna

suposición sobre estos formatos de serialización de los conjuntos de datos, pero sí distingue entre el conjunto de datos abstracto y sus diferentes manifestaciones o distribuciones.

A menudo, los datos se proporcionan a través de un servicio que admite la selección de un extracto, subconjunto o combinación de datos existentes, o de nuevos datos generados por alguna función de procesamiento de datos. DCAT permite también especificar y describir estos servicios de acceso a datos.

DCAT está relacionado con el Modelo de Información de IDS [IDSA21]. Este define una "lingua franca" agnóstica de los espacios de datos. Constituye un lenguaje común compartido por sus participantes y componentes, que facilita la compatibilidad, la interoperabilidad y la búsqueda y localización de activos y servicios de datos en los "intermediarios de datos" (o "data brokers" según el modelo de referencia de IDSA [IDSA22]).

El Modelo de Información tiene como objetivo principal la descripción, publicación y descubrimiento de productos de datos y software de procesamiento de datos dentro del estándar IDS. El almacenamiento estructurado de estos metadatos y los modelos semánticos asociados garantizan la localización de los activos más relevantes, adecuados para la tarea de un cliente. Una vez identificados, el Modelo de Información permite el consumo automatizado de esas ofertas de datos o servicios mediante definiciones de interfaz de servicio y vinculación de protocolos. Además de estos activos básicos, el Modelo de Información describe las propiedades esenciales de las entidades de los Espacios Internacionales de Datos, sus participantes, componentes de infraestructura y procesos.

Este movimiento de estandarización en el intercambio y portabilidad de los datos está relacionado con la visión que la comunidad científica tiene sobre la evolución del mercado de computación en la nube [Chasins22]. Esta visión fomenta la compatibilidad de los sistemas de computación en la nube, y la flexibilidad en la planificación de las cargas de trabajos en nubes híbridas. Para ello, se sirve de agentes de intermediación que permiten la conexión de recursos en diferentes nubes particulares. El proyecto Gaia-X propone un ecosistema de infraestructuras que busca de igual forma estandarizar la forma en que se provee IaaS y comunicación involucrando diferentes agentes e infraestructuras de nube.

3.3.3. Intercambio de datos seguro (FedWM)

Desgraciadamente, como ocurre con todos los activos digitales, el hecho de poder de copiar/almacenar/transmitir conjuntos de datos con un coste casi nulo facilita la creación de copias ilegales. Y lo que es peor, a diferencia de los contenidos y el software, la cuestión de la propiedad es menos obvia cuando se trata de conjuntos de datos. cuando se trata de conjuntos de datos. Cualquier película, canción, libro electrónico o programa informático suele poder atribuirse a un director/productor, músico, autor o empresa, respectivamente, pero es difícil hacer lo mismo con grandes conjuntos de datos. Igualmente puede ocurrir con los datos intercambiados entre clientes y agregadores en modelos de aprendizaje automático.

Pensemos en un conjunto de datos de movilidad anonimizados que registran los desplazamientos de las personas en una ciudad. Este conjunto de datos puede obtenerse mediante la recopilación de lecturas GPS de los teléfonos inteligentes de las personas que utilizan una aplicación de mapas, o puede deducirse analizando las huellas de teléfonos móviles [Lutu20] o los registros de descripción de llamadas (CDR) que mantienen los operadores de telefonía móvil.

En los modelos de aprendizaje federado se comparte información entre las partes que puede contener datos sensibles de individuos u organizaciones. Por esta razón, cobra importancia que se empleen técnicas de preservación de la privacidad en estos intercambios de datos. En particular, se estudiará el uso de técnicas de anonimización y de privacidad diferencial

La anonimización se refiere al proceso de modificar los datos personales de tal manera que los individuos no puedan ser identificados de nuevo y no se pueda aprender información sobre ellos. Esta técnica se aplica en varios casos de análisis de datos.

La mayoría de los modelos de anonimización de privacidad se dividen en dos categorías distintas. La primera incluye la k-anonimidad [Sweeney02] y sus extensiones [Truta06] [Machanavajjhala07] [Li07], mientras que la segunda consiste en la noción de privacidad diferencial [Dwork06], junto con algunas variaciones [Gehrke12] [Machanavajjhala15]. Mientras que la k-anonimidad es un mecanismo que se aplica a los datos para hacerlos un poco más borrosos y evitar revelar a los individuos, la privacidad diferencial se aplica a las respuestas de las consultas en el conjunto de datos para revelar información privada.

La privacidad diferencial es la segunda categoría de modelos de privacidad, que tiene como objetivo anonimizar las respuestas a las consultas interactivas enviadas a una base de datos. Los conjuntos de datos con privacidad diferencial se pueden generar mediante la creación de un conjunto de datos simulado a partir de un conjunto de datos con privacidad diferencial o agregando ruido a los atributos del conjunto de datos inicial. El principal inconveniente de la privacidad diferencial es que las garantías de privacidad se deterioran con el uso repetido y diversas técnicas.

El despliegue de tecnologías avanzadas de mejora de la privacidad (PET), como la computación entre múltiples partes [Evans18], el cifrado (totalmente) homomórfico (fully homomorphic encryption of FHE en inglés) [Gentry09], el cifrado funcional [Boneh11], y los entornos de ejecución entornos de ejecución fiables (Trusted execution environments TEE, en inglés) [Sabt15] pueden evitar que los datos se filtren sin autorización, y permitir cálculos (previamente acordados) sobre los mismos sin obstaculizar la economía basada en datos. Ejemplos prácticos de esto son el cálculo de conjuntos privados [Li21], bases de datos cifradas [Poddar19], computación segura [Anciaux19] agregación segura de datos [Ren22] y bases de datos verificables [Zhou21]. Sin embargo, la mayoría de estos enfoques se enfrentan a graves problemas de escalabilidad que dificultan su aplicación en casos reales.

Una alternativa al despliegue de soluciones de PET consiste en recurrir a herramientas puramente jurídicas y a términos y condiciones para proteger la propiedad de los datos en casos reales y condiciones para proteger la propiedad de los datos en el contexto de la nueva economía de los datos [Jougleux22] . De hecho, la mayoría de los DM hacen exactamente eso: intercambiar versiones en texto plano de los datos [Andres22_2], suponiendo que las distintas partes se atendrán a los términos y condiciones acordados previamente. Con garantías de propiedad garantías de propiedad débiles o inexistentes por medios técnicos, es difícil imaginar que la economía de los datos llegue a florecer y alcanzar su potencial previsto [Banterie20]. De hecho, cualquier copia vendida de un conjunto de datos puede ser "pirateada" por un comprador convertido en vendedor que, a continuación, puede revender el mismo conjunto de datos en un DM a un precio potencialmente inferior al propietario legítimo y haciendo inútil su inversión.

La marca de agua es una técnica muy conocida para proteger los derechos de propiedad frente a la copia y la distribución no autorizada. Inicialmente se propuso como medio para proteger la propiedad intelectual en soportes digitales [Asikuzzaman18, Begum20] y software [Ma19]. Más recientemente, también se han propuesto técnicas de marca de agua para conjuntos de datos [Aggrawal02, Agrawal09, Zhou05] y modelos de aprendizaje automático [Wang21]. En general, la marca de agua consta de dos algoritmos principales: generación (o incrustación) y detección. La generación permite a un propietario incrustar una marca de agua invisible (o visible) en sus datos utilizando un secreto de alta entropía (marca de agua) y produce una versión con marca de agua de los datos introduciendo una distorsión tolerable sin degradar la utilidad de los datos. Durante el algoritmo de detección, el propietario demuestra su propiedad sobre los datos sospechosos (aunque estén modificados) utilizando el mismo secreto de marca de agua generado durante la generación de la marca de agua. Si el resultado de la detección es 1 (o acepta), el propietario puede utilizarlo para demostrar su propiedad sobre los datos (sospechosos) con marca de agua. Se supone que un esquema de marca de agua es seguro frente al ataque de suposición (en el que un atacante intenta exponer el secreto de la marca de agua) y robusto frente a alteraciones/modificaciones (no) intencionadas (es decir, una marca de agua debería seguir siendo detectable incluso bajo una diversidad de ataques potencialmente lanzables contra el proceso [Agarwal19_2, Aggrawal02, Aggrawal03, Cohen18, Ji21, Quiring18, Sapana22]).

Limitaciones de las marcas de agua en datos. Las técnicas de marca de agua, dependiendo de la naturaleza de su aplicación, pueden tener objetivos muy diferentes, por ejemplo la marca de agua de bases de datos numéricas que controla la distorsión de la media y la desviación estándar [Aggrawal03], la marca de agua reversible que permite a los propietarios reconstruir los datos originales [Tang21], la marca de agua de conjuntos de datos de texto que preserva el significado de un texto [Li22] y/o las frecuencias de las palabras [Perez21], la marca de agua categórica que preserva las categorías (predefinidas) (por ejemplo, sexo, talla de la ropa) de un conjunto de datos [Lin21]. Todas estas soluciones se centran en un tipo de datos en un dominio específico [Kumar21, Sapana22]. Otra limitación de las técnicas existentes se refiere al nivel de control que ofrecen al usuario en cuanto al control de la distorsión introducida en los los datos originales debido a la marca de agua. Existen, por ejemplo técnicas que mantienen la media y la desviación estándar de un campo numérico [Aggrawal02, Mehdi21, Shehab08] pero que pueden conducir distorsión arbitraria entre el original y los datos con marca de agua cuando se cuando se considera toda la distribución de valores que va más allá de la media y la desviación estándar.

3.3.4. Seguridad en el aprendizaje federado (FedSecure)

En la introducción al aprendizaje federado, se señaló la seguridad como uno de los principales desafíos a los que se enfrentan estos sistemas. En efecto, existen diversidad de ataques que se pueden lanzar contra el esquema de aprendizaje distribuido y que pueden resultar en efectos tan diversos como un mal rendimiento del proceso de aprendizaje o incluso en filtraciones de datos privados de los clientes. En esta sección revisamos el estado del arte de la seguridad en el aprendizaje federado, uno de los objetivos específicos del proyecto MLEDGE.

Ataques

Se han documentado y demostrado una serie de ataques sobre los modelos de aprendizaje federado:

- Ataques de puerta trasera (*backdoor attack*) o de envenenamiento (*poisoning attack*) que permite malograr el modelo global mediante la inserción intencionada de

actualizaciones maliciosas [Bagdasaryan20, Bhajogi18] o simplemente cambiando las etiquetas de las muestras (label flipping) [Fu21]. Algunos autores han determinado que el éxito de estos ataques está determinado por la complejidad de la tarea y por el porcentaje de clientes implicados en el ataque [Sun19]. Otros autores han propuesto ataques de puerta trasera distribuidos más complejos y que desafían las defensas que se han propuesto para estos [Xie19, Wang20].

- Ataques bizantinos (*byzantine attacks*), en los que los adversarios toman el control de clientes y se comportan de manera arbitraria con el objetivo de corromper el proceso. Estos ataques se documentaron ya en aprendizaje distribuido [Blanchard17, Mhamdi18], pero se han estudiado también en el caso de aprendizaje federado. Además, se ha demostrado que extender el ataque en el tiempo puede romper estos esquemas de defensa [Mahdi18]. [Karimireddy21] recientemente demostró que incluso después de infinitas épocas de entrenamiento, cualquier mecanismo de agregación que ignore el pasado no convergerá a una solución eficiente.
- Ataques de inferencia (*inference attacks*), los cuales tratan de reconstruir el modelo global [Fredrikson15] o los datos de un usuario a partir de los pesos que comunica al gestor. Algunos autores han demostrado que es posible reconstruir imágenes de los clientes a partir de los gradientes informados al gestor [Geiping20].

Estos ataques aplican también a arquitecturas descentralizadas que incluyen nodos en el borde del cloud. Estos nodos podrían ser deshonestos y tratar de envenenar el modelo global, en vez de colaborar con el mismo,

Técnicas para mitigar o responder a estos ataques

Los **métodos de aprendizaje automático con preservación de la privacidad** (PPML) tratan de proteger los sistemas de aprendizaje federado contra los ataques de inferencia. Se basan en técnicas de minería de datos, seguridad y sistemas existentes como SMC, HE, TEEs, u otras como las políticas de control de acceso basadas en roles, la privacidad diferencial (DP), etc. El aprendizaje federado (FL) se ha introducido para permitir un ML descentralizado y que preserve la privacidad, manteniendo los datos en la fuente de generación. FL se ha acoplado naturalmente con DP en diferentes escenarios, localmente en los clientes de FL [Robin17], o centralmente en el servidor FL. Para ello, se introduce ruido gaussiano en la suma de la actualización de los gradientes para proteger el conjunto de datos del cliente en vez de datos específicos. Otros autores introducen DP local para proteger los parámetros del modelo de FL [Bhowmick18].

Existen diversos casos de implementación de SMC en esquemas de aprendizaje federado. Por ejemplo, una posibilidad es emplear esta técnica para proteger los parámetros locales en el proceso de calcular los promedios (FedAvg) [Bonawitz19]. Para ello, se presenta un protocolo para computar de forma segura la suma de los vectores de los clientes basado en compartir un secreto con los mismos [Shamir19]. Otros trabajos combinan SMC y DP. Por un lado se inyecta ruido en las actualizaciones locales de los clientes, y además se encripta esta información antes de ser enviada al gestor [Truex19].

Para lidiar contra los ataques de envenenamiento, típicamente se opta por algoritmos que permiten seleccionar los clientes cuyos datos y actualizaciones tienen un “mayor valor” para el modelo global. Algunos autores han propuesto modificaciones a FedAvg para introducir

un sesgo hacia el uso de clientes que aportan datos de mayor “valor [Fu21, Fung18, Yin18, Fung20], como por ejemplo Por ejemplo, q-Fair Federated Learning (q-FFL) [Li20].

Contra los ataques bizantinos, algunos autores han propuesto métodos de agregación estocástica robustos en escenarios de datos no independientes no idénticamente distribuidos [Li19_2]. Algunos de ellos han utilizado técnicas de detección de atípicos basadas en métodos estadísticos [Blanchard17, Cong19, Yin18],

Otros autores han desafiado estos mediante la manipulación de modelos locales de forma que el modelo global se desvíe en la dirección contraria a la correcta [Fang20]. Algunos autores optan por optimizar el modelo global [Karimireddy21] mientras que otros estudios recientes emplean análisis espectral [Shejwalkar21], privacidad diferencial [Naseri22] y inspección profunda de modelos [Rieger22] para proteger sobre ataques de envenenamiento, pero no usan información histórica para evaluar la confiabilidad de los clientes.

Contra la extensión en el tiempo de estos ataques, los estudios recientes incorporan mecanismos que tienen en cuenta el rendimiento a largo plazo de los clientes para juzgar su honestidad y que a menudo requieren utilizar datos auxiliares etiquetados que recolecta el agregador para [Cao21, Guo21]. Por ejemplo, se puede asignar una puntuación de “confiabilidad” a las actualizaciones de los clientes de acuerdo con la similaridad del coseno de las actualizaciones del servidor, calculada con la información raíz del servidor, y del cliente [Cao21]. Sin embargo, estos métodos requieren transmitir más información entre clientes y servidor, lo que ha demostrado comprometer su eficacia mediante ataques de inferencia [Melis19] y de reconstrucción [Ligeng19, Geiping20] de la información por parte de un agregador “honesto pero curioso”.

3.3.5. Incentivos y ‘fairness’ en aprendizaje federado (FLaaS Manager)

En escenarios donde la puesta a disposición del modelo global no es suficiente incentivo para participar en el entrenamiento del modelo, el **adecuado diseño de los incentivos** es un factor clave del éxito de un modelo de aprendizaje distribuido. Un punto de partida habitual en los modelos de aprendizaje federado es que clientes y servidor tienen incentivos suficientes con disponer de un modelo global efectivo y eficiente. Sin embargo, esta hipótesis no aplica en todos los casos, y en la economía de los datos aparecen diversos casos de uso donde una parte dispone de la información que se requiere para entrenar y mejorar modelos en los que no está particularmente interesada.

Por el contrario, el diseño de mecanismos de incentivos es un tema central en las propuestas de mercados de datos orientados a ML por parte de la comunidad [Agarwal19, Chen19]. En general, la mayor parte de la literatura ha recurrido al valor de Shapley, un concepto bien conocido en la teoría de juegos cooperativos con una serie de propiedades destacadas (simetría, jugador nulo, linealidad, etc.) [Rozemberczki22]. Sin embargo, el cálculo del valor de Shapley conlleva importantes retos computacionales, y se han desarrollado algunos algoritmos de aproximación específicos del contexto para hacer su cálculo más eficiente [Andres22, Ghorbani19, Jia19]. Esto es especialmente relevante para las implementaciones de MLEDGE que se acercan a los usuarios finales y donde el número de nodos de borde que participan en el entrenamiento de un modelo es elevado (incluso individuos en un caso extremo).

El aprendizaje federado se ha ocupado de la asignación de beneficios entre los nodos de borde que contribuyen al entrenamiento de un modelo de ML [Song19], y habitualmente se

monitoriza la calidad de las actualizaciones que envían los clientes al servidor agregador con el objetivo de evitar ataques de envenenamiento [Li20]. Algunos autores han definido un mecanismo de incentivos por el cual los clientes que aportan datos de mayor calidad basándose en una métrica acordada reciben mayores “recompensas” por participar [Kang19]. También algunos trabajos han aplicado nociones de imparcialidad y justicia en la aplicación de incentivos y reparto del beneficio obtenido en el modelo entre las diferentes partes que contribuyen al mismo [Ohrimenko19], o nociones de teoría cooperativa de juegos basadas en el valor de Shapley [Wang19, Liu21]. En particular, se ha definido el concepto de “federated Shapley Value” como [Wang20_2] que mantiene las propiedades del valor de Shapley sin incurrir en costes extra de comunicación entre las partes y capturando el efecto del orden de participación en el valor de los datos. Otros trabajos se han preocupado de cómo seleccionar las fuentes de datos más beneficiosas para un comprador en mercados federados en mercados de datos orientados al aprendizaje automático [Andres22_4, Galhotra23]. Otros trabajos han ampliado estas propuestas para distribuir los ingresos y un presupuesto utilizando algoritmos de aprendizaje automático que favorece la selección de las fuentes que mejor sirven al modelo del consumidor / comprador de datos[Zhao23].

Los anteriores estudios estudian el valor relativo de los datos en el entrenamiento de un modelo de aprendizaje federado. Sin embargo, pocos de estos estudios hablan de cómo establecer el precio absoluto de los datos o calcular su valor económico.

Desde una perspectiva macroeconómica, la OCDE publicó un interesante estudio en el que se resumen distintos enfoques para determinar el valor monetario de los datos personales [OECD13] , entre los que se incluyen métodos muy heterogéneos como el examen de la capitalización de mercado, los ingresos o los beneficios netos de las empresas de datos por individuo, el análisis de los ingresos o los beneficios netos por registro/usuario, o la evaluación del coste de las violaciones de datos, que a su vez asume los datos personales como un pasivo. Otra metodología habitual para abordar este problema es mediante experimentos económicos y encuestas sobre la disposición a pagar de los usuarios para proteger sus datos [Carrascal13]. Un estudio bibliográfico más reciente añade a los métodos de la OCDE la valoración basada en el impacto, que también tiene en cuenta los resultados sociales y económicos de los casos de uso de datos [Coyle20].

Un aspecto ortogonal y muy relevante y desafiante para la comunidad científica es el establecimiento de precios de los datos [Pei20]. Además, esta disciplina ha reunido investigadores en diferentes ramas técnicas y económicas. Diferentes escuelas con investigadores de diferentes disciplinas basan sus métodos en subastas [Goldberg01, Goldberg03, Goldber06, Aggarwal08], criterios de calidad de los datos [Heckman15, Yu17], cuantificar la pérdida de la privacidad [Ghosh11, Li15, Niu18] o determinar el precio de diferentes vistas de una base de datos [Balazinska11, Koutris12, Lin14, Koutris15, Chawla19.

Un reciente estudio ha reunido y analizado información sobre más de 200.000 productos ofrecidos en 43 mercados y proveedores de datos comerciales, qué categorías son más populares y valoradas, qué características definen los productos más caros en el mercado [Andres22_3] y cuáles de éstas son empleadas para poner precio a los productos. Basándose en esta información, modelos de ML permiten comparar productos entre mercados y aprenden la relación entre sus características y los precios del mercado como primer paso para predecir los mismos e incrementar la transparencia [Andres23].

4. Soluciones comerciales y herramientas de código abierto existentes

4.1. Aprendizaje Federado

Como ya hemos discutido anteriormente, el Aprendizaje Federado es una buena solución para el uso de datos compartidos por parte de distintos usuarios. Para llevar a cabo este uso, existen diversas librerías y herramientas de código abierto para entrenar modelos de aprendizaje automático de forma federada:

- Flower: A Friendly Federated Learning Framework (un marco amigable de trabajo de aprendizaje federado)
- OpenMined PySyft: La biblioteca syft.js permite el entrenamiento e inferencia de modelos de aprendizaje automático dentro de un navegador web. Los desarrolladores pueden escribir el modelo y el plan de entrenamiento en PyTorch y PySyft normal, y syft.js se encarga del resto. (https://github.com/OpenMined/syft.js/)
- OpenFL, inicialmente impulsado por Intel, es una biblioteca Python* 3 para el aprendizaje federado que permite a las organizaciones entrenar de forma colaborativa un modelo sin compartir información confidencial. OpenFL es independiente del marco de aprendizaje automático profundo flexible y compatible con cualquier marco de aprendizaje profundo, como TensorFlow* o PyTorch*, a través de un mecanismo de plugins.
- Fate es un proyecto de código abierto iniciado por el Departamento de Inteligencia Artificial de Webank para proporcionar un marco informático seguro que respalde el ecosistema de inteligencia artificial federada. Implementa múltiples protocolos de computación segura para permitir la colaboración de big data con el cumplimiento de la normativa de protección de datos. Con una canalización de modelado modular escalable, una interfaz visual clara y un sistema de programación flexible, FATE asegura ofrecer una facilidad de uso inmediata y un rendimiento operativo excelente.
- Tensorflow Federated (TFF) es un framework de código abierto de aprendizaje automático y otros cálculos con datos descentralizados. TFF se desarrolló para facilitar la investigación y experimentación abierta con aprendizaje federado (FL), un enfoque del aprendizaje automático en el que varios clientes participantes que conservan sus datos de entrenamiento de forma local entrenan un modelo general compartido.

  Federated Core (FC) es la capa central de TFF y como entorno se puede usar para desarrollar lógica de programa que combina código de TensorFlow con Federated Averaging, calculando sumas, promedios y otros tipos de agregaciones distribuidas sobre un conjunto de dispositivos cliente.

  Federated Core de TFF proporciona a investigadores y profesionales control explícito sobre los patrones específicos de comunicación distribuida al tiempo que ofrece un lenguaje flexible y extensible para expresar algoritmos de flujo de datos distribuidos, en lugar de un conjunto concreto de capacidades de entrenamiento distribuido implementadas.

- NVIDIA Flare (Federated Learning Application Runtime Environment) es un SDK de aprendizaje federado independiente del dominio, de código abierto y ampliable. Permite a los investigadores y científicos de datos adaptar el flujo de trabajo de ML/DL existente a un paradigma federado y permite a los desarrolladores de plataformas crear una oferta segura y que preserva la privacidad para una colaboración multipartita distribuida.

- IBM Federated Learning - IBM Federated Learning es un marco de Python para el aprendizaje federado en un entorno empresarial que proporciona un tejido básico para el aprendizaje federado, al que se pueden agregar funciones avanzadas. No depende de ningún marco específico de aprendizaje automático y admite diferentes topologías de aprendizaje, por ejemplo, un agregador compartido y protocolos.

  Los puntos clave de diseño de IBM Federated Learning son la facilidad de uso para el profesional del aprendizaje automático, la configurabilidad para diferentes entornos computacionales, desde centros de datos hasta dispositivos edge, y la extensibilidad. IBM Federated Learning viene con una gran biblioteca de algoritmos de fusión para enfoques de aprendizaje profundo y de ML clásicos, que consisten en implementaciones de algoritmos de fusión comunes y publicados, así como nuevos.

En general, los analistas refieren problemas de inmadurez, falta de transparencia en la comunidad de usuarios, diferentes enfoques y alcances o altas barreras de entrada (ver detalle en el siguiente informe) como los principales escollos a sortear para utilizar estas librerías de aprendizaje federado.

4.2. Aprendizaje federado como servicio (FLaaS)

Además, existen plataformas más complejas que se ejecutan como intermediarias para conectar todos las fuentes de datos en un solo lugar y aplicar distintos modelos de aprendizaje federado. A este respecto cabe destacar Acuratio (acuratio.com), la cual posee una plataforma con una interfaz de bajo nivel (y también accesos a código de más alto nivel para tunear los modelos) capaz de conectar de manera segura distintas fuentes de datos sin almacenaje de la información por su parte y con el uso de herramientas de anonimización. Esta plataforma permite a diversas compañías compartir su conocimiento sin necesidad de compartir datos, así como ampliar sus modelos combinando fuentes muy amplias de datos. Otra plataforma similar es FedML (fedml.ai), en su etapa actual, FedML está desarrollando y manteniendo una plataforma de aprendizaje automático que permite el aprendizaje federado y el análisis sin código, liviana, multiplataforma y demostrablemente segura. Permite el aprendizaje automático a partir de datos descentralizados en varios nodos de usuario/silo/edge sin requerir la centralización de datos en la nube, proporcionando así máxima privacidad y eficiencia. Consiste en un SDK de Edge AI liviano y multiplataforma que se puede implementar en GPUs de Edge, smartphones y dispositivos IoT. Además, también proporciona una plataforma MLOps fácil de usar para simplificar el aprendizaje automático descentralizado y la implementación en el mundo real. FedML admite soluciones verticales en una amplia gama de industrias (salud, finanzas, seguros, ciudades inteligentes, IoT, etc.) y aplicaciones (visión por computadora, procesamiento de lenguaje natural, minería de datos y pronóstico de series temporales) [FedML23]. También existen otras plataformas de este estilo como Devfi (devfi.com) o Sherpa (sherpa.ai), esta última dice utilizar además encriptación homomórfica para algunos problemas específicos.

En el ámbito más cercano del Edge, OctaiPipe es una plataforma de inteligencia artificial de extremo a extremo multiplataforma optimizada para crear, desplegar y gestionar soluciones de aprendizaje automático en entornos IoT industriales. Los despliegues en OctaiPipe son más asequibles, privados, escalables y resistentes para la inteligencia en el dispositivo gracias a sus avanzadas capacidades de aprendizaje automático federado en el dispositivo, así como a la innovadora tecnología edge MLOps.

4.3. Librerías que implementan técnicas de anonimización y privacidad diferencial

La anonimización de datos es una técnica importante para garantizar la privacidad y la seguridad de la información en diferentes aplicaciones y sistemas informáticos. Hay varias herramientas y técnicas disponibles para anonimizar datos, como la perturbación de datos, la agregación de datos y la encriptación.

En este sentido, hay diferentes librerías de código abierto que permiten integrar estas técnicas de anonimización en distintas aplicaciones y sistemas:

ARX - ARX es un software de código abierto completo para anonimizar datos personales sensibles. Soporta una amplia variedad de modelos de privacidad y riesgo, métodos para transformar datos y métodos para analizar la utilidad de los datos resultantes. (arx.deidentifier.org)

ARX es capaz de manejar grandes conjuntos de datos en hardware común y cuenta con una interfaz gráfica de usuario multiplataforma.

Pynonymizer - pynonymizer reemplaza los datos de identificación personal en su base de datos con datos seudorandom realistas, provenientes de la biblioteca Faker u otras funciones. Existe una amplia variedad de tipos de datos disponibles que deberían adaptarse a la columna en cuestión.

El mecanismo principal de reemplazo de datos de Pynonymizer, fake_update, es una selección aleatoria de un pequeño conjunto de datos (--seed-rows controla los datos de Faker disponibles). Este proceso se elige por su compatibilidad y velocidad de operación, pero no garantiza la unicidad. (pypi.org/project/pynonymizer/)

Anonympy - Anonympy es una biblioteca general de anonimización de datos para imágenes, PDF y datos tabulares. Sus características principales incluyen la eficiencia en datos tabulares así como los distintos métodos de anonimización incluidos para datos numéricos, categóricos, datetime y también para imágenes y PDFs. (https://pypi.org/project/anonympy/)

La privacidad diferencial permite equilibrar el nivel de privacidad y precisión con un valor positivo denominado ε (épsilon). Si ε es pequeño, se preserva más la privacidad pero empeora la precisión de los datos intercambiados. Si ε es grande, la privacidad será peor pero se conservará la precisión de los datos. Tenga en cuenta que ε va de 0 a infinito. Las bibliotecas de privacidad diferencial implementan varias técnicas que toman un parámetro épsilon como entrada y añaden ruido aleatorio a los valores del conjunto de datos original, proporcionalmente al parámetro ε dado. Así, cuanto menor sea el valor épsilon, más ruido se añadirá a los valores.

Algunas bibliotecas toman más parámetros que ε y permiten controlar cómo se añade el ruido aleatorio a los valores del conjunto de datos original, como la distribución de probabilidad de la que se extraen los números aleatorios (Laplace, Normal, etc.).

Algunas de estas bibliotecas también implementan el concepto de presupuesto de privacidad, en el que cada llamada a una función de la biblioteca utilizará una cantidad definida por el usuario del presupuesto de privacidad asignado originalmente. La teoría detrás de esto es que cada vez que se libera nueva información, la probabilidad de que un atacante recupere información sobre un individuo en el conjunto de datos aumenta. Una vez agotado el presupuesto de privacidad, la biblioteca puede emitir un error en lugar de devolver un valor.

En particular, las siguientes librerías permiten aplicar estas técnicas a los datos:

IBM Diffprivlib. IBM/differential-privacy-library (Python) es una librería de uso general para experimentar, investigar y desarrollar aplicaciones que incorporen privacidad diferencial. Soporta Python 3.8 a 3.11 a la hora de realizar este informe.

Google Differential Privacy library. google/differential-privacy (C++, Go, Java) sirve para generar estadísticas ε- y (ε, δ)-diferencialmente privadas sobre conjuntos de datos. Contiene las siguientes herramientas:

- Privacy on Beam es un marco de privacidad diferencial de extremo a extremo construido sobre Apache Beam. Se pretende que sea fácil de usar, incluso para los no expertos.
- Tres bibliotecas "DP building block", en C++, Go y Java. Estas bibliotecas implementan primitivas básicas de adición de ruido y agregaciones diferencialmente privadas. La privacidad en Beam se implementa utilizando estas bibliotecas.
- Un comprobador estocástico, utilizado para ayudar a detectar regresiones que podrían hacer que la propiedad de privacidad diferencial dejara de cumplirse.
- Una biblioteca de contabilidad de privacidad diferencial, utilizada para el seguimiento del presupuesto de privacidad.
- Una interfaz de línea de comandos para ejecutar consultas SQL diferencialmente privadas con ZetaSQL.

Paquete diffpriv. El paquete brubinstein/diffpriv (R) diffpriv facilita la ciencia de datos con privacidad en R. diffpriv implementa el marco formal de privacidad diferencial: los mecanismos que implementa permiten liberar de forma segura a terceras partes no fiables: estadísticas calculadas, modelos ajustados o estructuras arbitrarias derivadas de datos sensibles a la privacidad. Debido a la naturaleza del marco en el peor de los casos, el desarrollo de mecanismos suele requerir un análisis teórico. Diffpriv ofrece un enfoque llave en mano a la privacidad diferencial mediante la automatización de este proceso con el muestreo de sensibilidad en lugar de análisis de sensibilidad teórica.

pyCANON - pyCANON es una biblioteca de Python y una interfaz de línea de comandos (CLI) para comprobar y evaluar el nivel de anonimato de un conjunto de datos mediante algunas de las técnicas de anonimización más comunes: k-anonimato, (α, k)-anonimato, ℓ-diversidad, entropía ℓ-diversidad, (c, ℓ)-diversidad recursiva, t-cercanía, β-similitud básica, β-similitud mejorada y δ-revelación de privacidad. La principal fortaleza de esta biblioteca es obtener un informe completo de los parámetros que se cumplen para cada una de las técnicas mencionadas anteriormente, con el único requisito del conjunto de cuasi-identificadores y atributos sensibles. (https://pypi.org/project/pycanon/)

5. Arquitectura y descripción de los componentes y casos de uso de MLEDGE

En esta sección, se presenta la arquitectura del proyecto MLEDGE junto con una descripción de los diferentes componentes de la misma. El proyecto busca:

i) Avanzar en el estado del arte de los componentes de acuerdo con los objetivos científicos

ii) Demostrar estos avances sobre plataformas y casos de uso comerciales de forma que se facilite la explotación posterior de los resultados del proyecto en la economía real.

El diagrama mostrado en la figura siguiente resume la arquitectura de MLEDGE y los bloques del proyecto que se presentó como propuesta del proyecto.

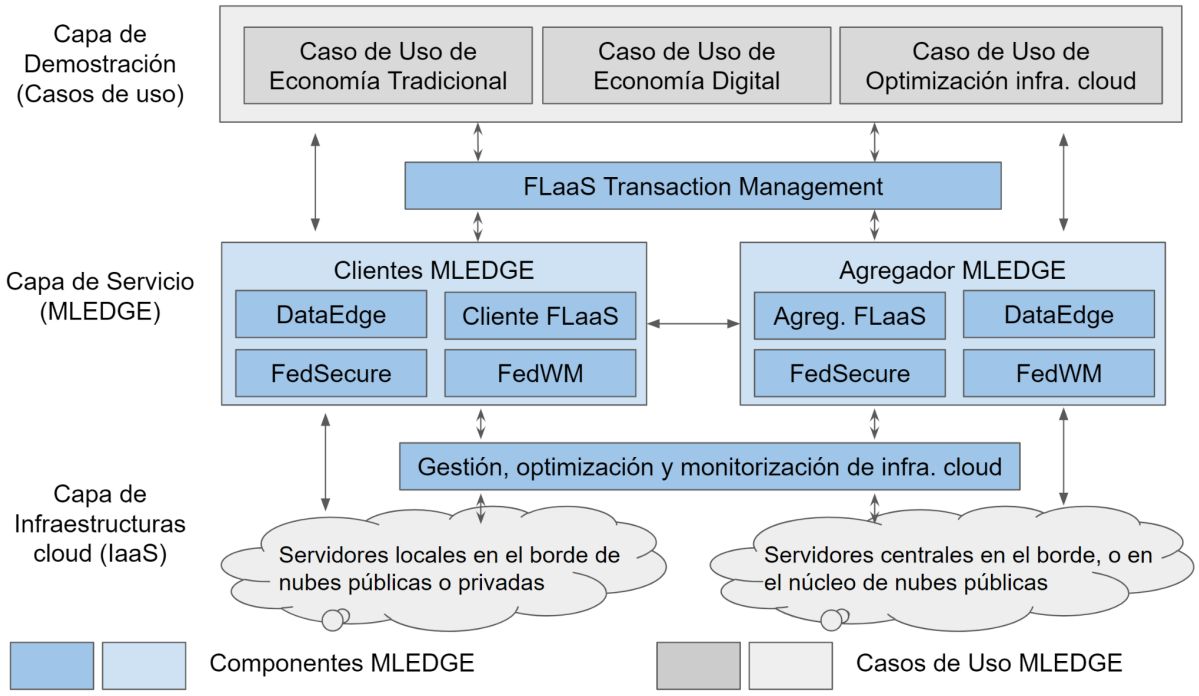

Figura 1. Diagrama de bloques de MLEDGE

La arquitectura de MLEDGE se articula en torno a tres capas:

- La **capa de infraestructuras** dispone los recursos de computación y comunicación necesarios para la ejecución del proyecto. Dicha capa de infraestructuras puede incluir recursos a diferentes niveles de la red, incluyendo clouds públicas o privadas "centralizadas" en el núcleo de la nube, nodos de computación en el borde de la nube, e incluso infraestructuras en casa o terminales de los usuarios.
- La **capa de servicios MLEDGE** busca integrar una serie de componentes que habiliten servicios FLaaS en el borde de la nube, y que puedan integrar componentes innovadores que provengan del desarrollo de los objetivos científicos del proyecto.

- La **capa de demostración** incluye casos de uso reales de empresas de la economía tradicional y digital, así como un caso de uso específico de optimización de infraestructuras cloud. Los casos de uso buscarán demostrar el uso del aprendizaje federado en el borde de la nube y de los componentes de MLEDGE en escenarios reales y con datos reales. Para ello, durante los primeros seis meses de proyecto se ha realizado un screening de empresa y se han elaborado pliegos para realizar licitaciones de estos tres casos de uso a empresas españolas externas a IMDEA.

En las secciones siguientese se describen los componentes de la capa de servicios en los que se está trabajando durante el proyecto, su relación con los objetivos científicos, y los casos de uso que se buscará incorporar de la industria mediante estas licitaciones.

5.1. Descripción de los componentes de la capa de servicios

La capa de servicios de MLEDGE trabajará sobre la base de alguna de las soluciones comerciales de FLaaS que se describieron en la sección 4.2. Adicionalmente, se incluye una serie de componentes perfectamente alineados con los objetivos científicos del proyecto que buscan extender el estado del arte de algunos componentes del aprendizaje federado en el borde de la nube. Además, su integración con casos de uso específicos de la industria permitirá probar estos componentes directamente sobre casos reales y acelerar el tiempo hasta la explotación de los mismos por parte de la industria.

La siguiente tabla relaciona los componentes de la capa de servicios de MLEDGE con los objetivos científicos del proyecto:

Tabla 1. Relación entre objetivos científicos y componentes de la capa de servicios de MLEDGE

Objetivo científico	**Componente MLEDGE**
1. DevOps y desarrollo continuo para servicios de aprendizaje automático (FLaaS) que se ejecutan en borde de la nube	FLaaS
2. Uso eficiente de FL en nubes híbridas y protección contra ataques	FedSecure
3. Protección de datos sensibles o confidenciales que sean intercambiados entre dominios administrativos en la nube y el borde de la nube	FedWM
4. Equidad en términos de distribución de costos y ganancias cuando la computación en el borde se usa para entrenar de forma colaborativa modelos de ML	FLaaS Manager
5. Gestión de los desafíos de portabilidad de datos en el borde	DataEdge

A continuación describiremos cada uno de estos componentes y los trabajos de investigación y desarrollo que se van a realizar para ir más allá del estado del arte de los objetivos científicos que atacan.

5.1.1. FLaaS

Se empleará durante el proyecto una plataforma de aprendizaje federado como servicio FLaaS comercial o de código abierto. Dicha plataforma debe cumplir los requisitos especificados en la siguiente tabla:

Tabla 2. Requisitos de la plataforma FLaaS base para ser utilizada en MLEDGE

#	Requisito	Descripción
FLaaS1	Multimodelo	Debe permitir el entrenamiento federado de diferentes tipos de modelos a ser especificados por los clientes finales
FLaaS2	Versatilidad	Debe permitir diferentes tipos de aprendizaje federado para servir a los casos de uso, incluyendo aprendizaje federado vertical y horizontal
FLaaS3	Multicloud	Debe ser una plataforma multicloud, que trabaje al menos con las principales plataformas de cloud en Internet (AWS, Azure, Google Cloud) y permitir su despliegue también en infraestructura en el domicilio del cliente
FLaaS4	Fácil de usar	Debe ser sencilla de usar y tener espíritu no code para facilitar a las empresas que la usen su integración con un coste reducido
FLaaS5	Comunicación segura	La herramienta debe incorporar mecanismos de agregación segura de información entre los nodos basados en el estado del arte
FLaaS6	Privacidad	La herramienta debería permitir el uso e incluir herramientas para proteger la privacidad de los usuarios, como por ejemplo privacidad diferencial, anonimización, etc.
FLaaS7	Autenticación y Roles	La herramienta debe contemplar diferentes roles de los usuarios finales e incorporar mecanismos seguros de autenticación e identificación de los usuarios.
FLaaS8	Administración y gestión	La plataforma debe proporcionar herramientas administrativas y permisos especiales para la gestión de los modelos a ser entrenados.

Esta herramienta FLaaS formará parte de una de las licitaciones del proyecto y se usará como base para probar los componentes de MLEDGE.

Durante el proyecto, se explorará la posibilidad y la mejor manera de realizar un aprendizaje federado distribuido, de forma que se diluyan las fronteras entre los nodos clientes y el nodo agregador. De la misma manera que en las redes P2P como BitTorrent los nodos actúan a la vez como transmisores y receptores de datos hacia sus vecinos, el objetivo es diseñar un esquema de aprendizaje federado en el que los clientes sean a la vez agregadores y entrenen sus propios modelos locales aprendiendo de los gradientes e informaciones que les envían sus vecinos.

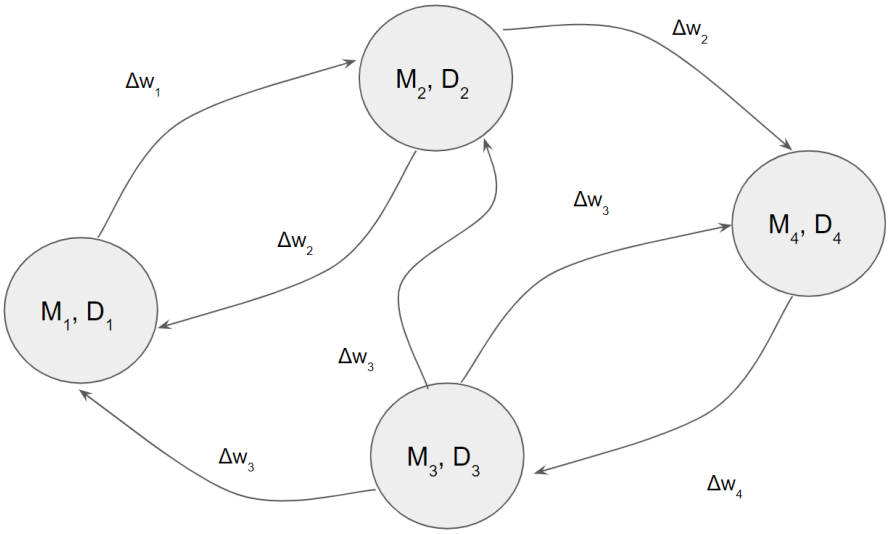

Figura 5. Aprendizaje federado descentralizado en una época t

Por tanto, buscaremos cambiar significativamente el esquema del aprendizaje federado hacia una arquitectura más distribuida en las que un conjunto de clientes c_i colaboran para entrenar sus propios modelos M_i locales, todos respondiendo a un esquema o diseño común, y basándose en los gradientes Δw_i que les pasan otros clientes c_j fruto del entrenamiento del modelo con sus conjuntos de datos D_j. Al igual que en el aprendizaje federado conjunto de un modelo local, se evita el intercambio de datos crudos entre clientes. Además, se evita la centralización de los gradientes en el nodo gestor o servidor, de forma que

- el nodo servidor tiene más requerimientos de procesamiento y de ancho de banda
- el servidor dispone de más información y le convierte en un punto de entrada para ataques, por ejemplo de inferencia

Además de estos temas de privacidad. Los defensores de arquitecturas descentralizadas argumentan que esta descentralización reduce el consumo de ancho de banda y da a los usuarios el control de quién se comunica con quién.

Por el contrario, estas arquitecturas son más complejas y enfrentan desafíos adicionales. Por ejemplo, será necesario definir la topología de los clientes la cual puede ser estática o dinámica. La utilización de topologías estáticas permitiría adecuar la misma a la topología de la red en dos etapas jerárquicas de agregación:

i) de los usuarios finales a los servidores en el borde del cloud, y

ii) de los servidores en el borde del cloud a un servidor central.

Por otro lado, se probarán *gossip protocol* como los empleados por las redes P2P como BitTorrent para construir de forma dinámica la topología de la red descentralizada de aprendizaje federado. En esta actividad, es fundamental identificar los factores clave que hagan que un cliente prefiera a otro cliente en la red (afinidad de la información, facilidad de comunicación, etc.) para intercambiar información para entrenar sus modelos de forma conjunta.

La comunicación entre los nodos se realizará mediante técnicas para proporcionar seguridad a las comunicaciones entre clientes acordes con el estado del arte (SSL o similar). Se estudiarán diferentes algoritmos para la creación de redes entre clientes, con cada nodo decidiendo de forma autónoma con qué otros clientes de la red interacciona para enviar sus actualizaciones el modelo. Para ello tendrá que tener en cuenta los beneficios que estos otros clientes traen a su modelo local.

Dado que tanto las topologías centralizadas como las distribuidas son susceptibles de sufrir diversos tipos de ataques [Pasquini23], será necesario investigar los potenciales ataques a estos esquemas de aprendizaje federado por parte de los nodos.

5.1.2. FedSecure

El módulo FedSecure trabajará en mejorar la seguridad del aprendizaje federado en el borde de la nube. Este trabajo se realizará en dos direcciones: 1) el desarrollo de modelos de reputación de los clientes para detectar ataques de envenenamiento, 2) la mejora de los criterios de agregación del modelo global.

En el primer caso, el objetivo es recolectar y almacenar información sobre la actividad de los clientes para construir un modelo de reputación de los mismos. En la actualidad se ha realizado una primera publicación en este sentido [Chu23], que se plantea extender para trabajar en un modelo descentralizado donde cada cliente almacena su propia percepción sobre la reputación de los diferentes nodos que colaboran en su modelo local.

En segundo lugar, IMDEA está actualmente desarrollando una solución con un nuevo método de votación cuadrática federada (FedQV: Federated Quadratic Voting) que agrega modelos globales basados en los votos de un mecanismo verídico [Chu22]. La eficiencia del método ha sido analizada exhaustivamente desde un punto de vista tanto teórico como experimental, demostrando que FedQV logra un desempeño superior en la defensa contra varios ataques de envenenamiento. Además, FedQV es un módulo reutilizable ya que se puede usar junto con los modelos de reputación de MLEDGE para asignar presupuestos de votación desiguales y otras técnicas robustas bizantinas y mecanismos de preservación de la privacidad para brindar resistencia tanto al ataque de envenenamiento como a los ataques a la privacidad.

Siguiendo esta línea, proponemos avanzar el estado del arte en esta área extendiendo estos esquemas de votación cuadrática basados en la reputación para filtrar los nodos del borde de la nube maliciosos. Adicionalmente, se estudiarán técnicas de ataques en los casos de uso del proyecto y se propondrán soluciones a las mismas que se buscará integrar dentro de la plataforma de FLaaS del proyecto, enriqueciéndola y facilitando la explotación comercial de los avances científicos en este campo.

5.1.3. DataEdge

Usaremos formatos estándar de representación de datos e interfaces como XML, RDF, REST API, adaptados a las recomendaciones de IDSA y GAIA-X para asegurar mecanismos estándar y controlados de intercambio de datos entre las partes que intervienen en el aprendizaje federado. En este sentido, no se pretende que la portabilidad de datos en el borde de la nube sea objeto de desarrollo científico relevante en el proyecto, sino que se mantendrá un contacto con las iniciativas que se están llevando a cabo en este campo en materia de estandarización, con las cuales se trabajará y se integrará su uso en la plataforma de FLaaS del proyecto.

5.1.4. FedWM

Las técnicas de propiedad de datos para medios y software comparten una propiedad muy fundamental: un archivo se considera inútil (corrupto) si se modifica, incluso si se divide en partes. Los conjuntos de datos pueden seguir siendo útiles incluso si están severamente alterados, por ejemplo, particionados. Esta propiedad exige un tratamiento diferente y garantiza la efectividad de nuestra propuesta. Desarrollaremos una nueva generación de técnicas de propiedad de datos basadas en frecuencias (Frequency Data Ownership) [İşler22] para datos estructurados y semiestructurados, manipulando ligeramente la frecuencia de aparición de diferentes tokens. La información de propiedad codificada en el conjunto de datos se puede usar para detectar quién hizo copias ilegales o filtró el conjunto de datos. Estas técnicas harán que la detección por parte de un adversario sea computacionalmente imposible. También garantizarán que la destrucción de la firma digital mediante inyección de ruido conduzca directamente a la destrucción de los datos que la portan. En comparación con los métodos de propiedad para bases de datos, nuestras técnicas también cubrirán conjuntos de datos no numéricos. El objetivo es crear un esquema de marca de agua o similar para proteger de la propiedad de los datos cuando se requiera la redistribución de los datos o metadatos que se necesiten intercambiar entre servidores en el borde en el marco del FLaaS.

5.1.5. FLaaS Manager

De alguna manera, el aprendizaje federado asume que los diferentes nodos que participan en el entrenamiento del modelo de aprendizaje distribuido tienen incentivo suficiente en mejorar el modelo para participar activamente en el proceso. Esto ha sido así en los primeros modelos de aprendizaje federado, como los empleados por el teclado de Google. Sin embargo, según se desarrolle la tecnología y aumenten los casos de uso, puede haber ocasiones en que la entidad que dispone los datos para entrenar los modelos no necesariamente es la que está interesada en el desarrollo de estos modelos.

Utilizaremos las propuestas de los diseños de mercados de datos orientados a aprendizaje automático existentes y los trabajos recientes que han estudiado cómo funcionan los mismos en plataformas descentralizadas [Andres22_2, Ohrimenko19], para diseñar mecanismos que hagan que la plataforma MLEDGE sea atractiva (desde el punto de vista comercial) para todos sus usuarios (propietarios y proveedores de datos, proveedores de servicios y consumidores de datos).

También abordaremos el problema de cómo compartir los costes de procesamiento y asignar los ingresos de la plataforma entre los proveedores de datos que aportaron datos para el entrenamiento de un modelo de ML. Estos algoritmos deberán ser automáticos,

explicables y entendibles para las organizaciones que forman parte del proyecto. Para este último propósito, utilizaremos trabajos existentes [Kang19, Wang19, Liu21, Wang20_2] y nuestros anteriores [Andres22] para calcular el valor de los datos de una fuente en el entrenamiento de un modelo mediante mecanismos de aprendizaje federado. Dicho cálculo se particularizará para los casos de uso que se desarrollarán en el transcurso del proyecto.

Adicionalmente. prestaremos especial atención a la eficiencia del proceso, adaptando las técnicas existentes al caso general del uso de FLaaS en el borde de la nube y sus particularidades (múltiples modelos, limitada visibilidad de los diferentes nodos de la información empleada en el entrenamiento, limitaciones en la capacidad de procesamiento, etc). La clave para estas dos tareas es poder calcular de forma escalable el llamado valor de Shapley, para lo cual implementaremos una serie de algoritmos de aproximación diferentes que deben adaptarse a cada caso de uso. Adicionalmente, probaremos a sustituir el valor de Shapley por otras heurísticas o modelos más sencillos que permitan abaratar el coste de cálculo del valor de los datos sin perder precisión en el mismo. También probaremos algunos de estos métodos en los casos de uso del demostrador y se buscará activamente la publicación de artículos científicos en este sentido.

En un entorno de aprendizaje distribuido descentralizado. cada cliente debe seleccionar los clientes en los que confía para entrenar su modelo y que mejor sirven a su propósito particular. En este contexto, la valoración de los datos o de los gradientes aportados por esos otros clientes al modelo que está entrenando cada nodo participante es fundamental para resolver este problema. Para ello, nos inspiraremos en estrategias empleadas por redes P2P de transmisión de datos como choke/unchoke [Legout06, Legout07].

Los algoritmos que se desarrollarán e implementarán en este componente tratan problemas genéricos que los broker de datos de IDSA o entidades que puedan proveer servicios de FL en Gaia-X enfrentarán. Por tanto, deben contribuir a sentar las bases para que estos servicios se conviertan en realidad dentro de estas plataformas.

5.2. Casos de Uso

Una parte fundamental del proyecto MLEDGE es generar sinergias con la industria, identificar y desarrollar casos de uso de aprendizaje federado en el borde de la nube que respondan a necesidades de los negocios, y en este contexto probar los desarrollos de los módulos para enriquecer una plataforma de aprendizaje federado comercial. En esta sección, presentamos los casos de uso identificados en el proyecto y que estarán siendo objeto de contratación con empresas españolas de diferentes industrias. Cada caso de uso de los tres apartados siguientes se corresponde con cada uno de los tres lotes de que consta esta la misma.

5.2.1. Caso de uso industrial relacionado con la economía tradicional

Existen multitud de problemas en la economía tradicional en los que la tecnología desarrollada por MLEDGE puede ayudar a mejorar la eficiencia de los procesos o la toma de decisiones de diferentes actores de la cadena de valor. Por ejemplo, en el caso del sector sanitario, diferentes hospitales se podrían beneficiar de datos de pacientes de otros hospitales a la hora de entrenar modelos para la toma de decisiones sobre los tratamientos a aplicar o para entrenar modelos de diagnóstico o de predicción de la evolución de los pacientes. Sin embargo, por motivos regulatorios o de confidencialidad de los datos, los

hospitales no pueden compartirlos, pero sí podrían emplear estos datos para entrenar un modelo “común” de forma colaborativa sin revelarlos. Otro ejemplo, en la industria a menudo es necesario usar modelos de estimación de costes o presupuestación, pero los diferentes agentes que intervienen en la cadena de valor no desearían revelar sus “costes” o “precios” de los servicios pero sí participarían en el entrenamiento federado de modelos que permitan disponer de una herramienta de presupuestación de aplicación para toda la industria.

El objetivo de este caso de uso es demostrar la aplicación de las tecnologías de MLEDGE para resolver un problema de aplicación en alguna industria de la economía tradicional (construcción, finanzas, salud, etc.). El caso de uso será desarrollado por una empresa y empleará los componentes desarrollados en MLEDGE para mejorar sus procesos, o bien para mejorar los procesos de toma de decisiones basadas en datos o modelos provenientes de FL. Por tanto, el caso de uso deberá demostrar los beneficios del proyecto para entrenar un modelo de ML sin exponer los datos de las partes implicadas, e idealmente se usará para la toma de decisiones en tiempo real.

Por tanto, la entidad adjudicataria se encargará de:

- Justificar el caso de uso y estudiar el estado del arte.
- Aportar los datos necesarios para alimentar modelos que mejoren sus procesos de toma de decisiones.
- Diseñar los componentes para el caso de uso de economía tradicional, implementar dichos componentes, y probarlos.
- Colaborar en la demostración del caso de uso.
- Desarrollar estrategias de explotación del desarrollo realizado.

Para la correcta ejecución de este contrato, se requerirá por parte del contratista los siguientes medios humanos y materiales que deberán ser adscritos al contrato, incluyendo el equipo de trabajo, datos para alimentar el caso de uso, modelos, software y plataformas necesarias para los demostradores más allá de la plataforma FLaaS, así como infraestructura IT tal como almacenamiento cloud, servidores, dispositivos periféricos, redes, etc… para llevar a cabo la correcta demostración de las actividades.

El perfil idóneo para esta empresa es, aunque no de forma excluyente, el de una entidad con sólida experiencia en una industria tradicional (por ejemplo, construcción, transporte, salud, etc…) que esté desarrollando iniciativas de digitalización internas y que busque resolver algún problema real de negocio susceptible de ser resuelto mediante las tecnologías de aprendizaje federado de MLEDGE. También podrán concurrir entidades que ofrezcan servicios IT o de consultoría con experiencia en estas industrias.

5.2.2. Caso de uso relacionado con la economía digital

De igual forma existen diversos casos de uso de empresas en la economía digital en los que diversas partes desean contribuir datos para el entrenamiento de modelos de aprendizaje automático. Por ejemplo, dentro del terreno de la salud digital se podría pensar en explotar la información que proporcionan dispositivos móviles o tecnologías vestibles (como pulseras de actividad, en inglés *wearables*) para vigilar la evolución y adaptar el tratamiento o las necesidades de atención médica a pacientes. Otro caso de uso es el empleo de datos de los usuarios para entrenar modelos de publicidad digital mediante técnicas de FL. El objetivo de este caso de uso es demostrar la aplicación de las tecnologías desarrolladas en el seno de MLEDGE para resolver problemas de negocio en

empresas relacionadas con la economía digital, los cuales emplearán el FL para entrenar un modelo de ML sin exponer los datos que se empleen para ello.

Por tanto, la entidad adjudicataria se encargará de:

- Justificar el caso de uso y estudiar el estado del arte.
- Diseñar los componentes para el caso de uso de economía digital, implementar dichos componentes, y probarlos.
- Aportar los datos necesarios para alimentar modelos que mejoren sus procesos de toma de decisiones.
- Colaborar en la demostración del caso de uso.
- Desarrollar estrategias de explotación del desarrollo realizado.

Para la correcta ejecución de este contrato, se requerirá por parte del contratista los siguientes medios humanos y materiales que deberán ser adscritos al contrato, incluyendo el equipo de trabajo, datos para alimentar el caso de uso, modelos, software y plataformas necesarias para los demostradores más allá de la plataforma FLaaS, así como infraestructura IT tal como almacenamiento cloud, servidores, dispositivos periféricos, redes, etc… para llevar a cabo la correcta demostración de las actividades.

El perfil idóneo para esta empresa es, aunque no de forma excluyente, el de una entidad con sólida experiencia en una economía digital (FinTech, identidad digital, servicios web, etc…) y que busque resolver algún problema real de negocio susceptible de ser resuelto mediante las tecnologías de aprendizaje federado de MLEDGE. También podrán concurrir entidades que ofrezcan servicios IT a empresas de estas industrias.

5.2.3. Caso de uso de optimización de infraestructuras CloudEdge

Un tercer caso de uso estará relacionado con el uso de aprendizaje federado para algoritmos de optimización de la propia infraestructura de CloudEdge en la que correrán los servicios. Dicha optimización empleará algoritmos de ML que se entrenarán de forma federada por los diferentes nodos que participen en la provisión de servicios FLaaS. De esta forma, esta funcionalidad es per se una funcionalidad clave y, a la vez, un caso de uso de los servicios proporcionados por MLEDGE.

Por tanto, la entidad adjudicataria se encargará de:

- Estudiar el estado del arte, complementando este entregable
- Implementar la capa de servicio de FLaaS para ser utilizado con los adjudicatarios de los lotes 1 y 2
- Aportar las infraestructuras necesarias para la ejecución de modelos de aprendizaje federado y colaborar con los adjudicatarios de los lotes 1 y 2 en el entrenamiento y la explotación de los modelos que se utilizarán en los casos
- Trabajar con la Fundación para la prueba, integración y demostración de los módulos de FL seguro, adición de marcas de agua en las comunicaciones y de una capa o lógica económica o de negocio en la plataforma
- Trabajar con los adjudicatarios de los lotes 1 y 2 para la integración prueba y uso de los componentes en los casos de uso elaborados como parte del proyecto
- Diseñar los componentes para mejorar la eficiencia en la distribución de las cargas de procesamiento en los diferentes niveles de la nube, haciendo uso de modelos de aprendizaje federado.

- Colaborar en la demostración de los casos de uso del proyecto
- Desarrollar estrategias de explotación del desarrollo realizado

El perfil idóneo para esta empresa es, aunque no de forma excluyente, el de una entidad relacionada en el ámbito IT que proporcione servicios de cloud a empresas para que ponga en marcha las infraestructuras de nube y de computación en el borde necesarias para las pruebas de concepto. Se valorará experiencia de trabajo acreditada con modelos o plataformas de aprendizaje automático distribuido o federado. Esta entidad trabajará con el equipo de la Fundación y prestará servicios a las empresas adjudicatarias del resto de lotes de los casos de uso. Además, se encargará de implementar un caso de uso relativo a modelos de optimización de las cargas de trabajo de las infraestructuras de cloud y del reparto de los beneficios y los costes de las infraestructuras entre los diferentes componentes del servicio.

Para la correcta ejecución de este contrato, se requerirá por parte del contratista los siguientes medios humanos y materiales que deberán ser adscritos al contrato, incluyendo el equipo de trabajo, modelos, software y plataformas necesarias para los demostradores más allá de la plataforma FLaaS, así como infraestructura IT tal como almacenamiento cloud, servidores, dispositivos periféricos, redes, etc… para llevar a cabo la correcta demostración de las actividades.

5.2.4. Entregables y cronograma de las licitaciones

En los tres casos, se ha propuesto un calendario de entregables coordinado que comparte fechas de entrega y naturaleza de los entregables objetivo. La siguiente tabla resume la lista de entregables común de las tres licitaciones.

Tabla 3. Entregables de las licitaciones para el desarrollo de los casos de uso de MLEDGE

Nº	Descripción	Clasificación	Criterio de aceptación	Fecha
M1	Informe detallado sobre el diseño del caso de uso X a implementar y cómo utilizará los componentes de MLEDGE.	Documental	Informe de estado del arte y diseño del caso de uso y del software / plataforma necesario.	30/06/2024
M2	Componentes software, informe y documentación preliminares de los componentes y el caso de uso de optimización de infraestructuras cloud.	Módulos	Documentación completa del caso de uso, código funcionando con adecuada documentación. Especificación de las líneas de trabajo futuras. Infraestructuras cloud necesarias y capa FLaaS para probar los prototipos MLEDGE listas. Modelos de ejemplo cargados y probados (sólo aplicable al Caso de Uso 3)	31/12/2024

Nº	Descripción	Clasificación	Criterio de aceptación	Fecha
M3	Componentes software, informe y documentación finales de los componentes y el caso de uso de optimización de infraestructuras cloud.	Prototipos	Documentación completa del caso de uso, código funcionando con adecuadas prestaciones. Estrategias de comunicación y explotación detalladas.	30/06/2025

Estos entregables están plenamente en línea con la planificación del proyecto MLEGDE. La duración del proyecto está prevista que sea de 30 meses. El Diagrama de Gantt de la figura siguiente detalla los paquetes de trabajo y sus actividades, así como los entregables e hitos relacionados. La tabla 1 detalla la correspondencia de las principales tareas técnicas del proyecto, detalladas en las innovaciones tecnológicas y las actividades correspondientes en los paquetes de trabajo. La Tabla 2 que se muestra a continuación informa el esfuerzo específico planificado a nivel de actividad para cada paquete de trabajo, y muestra el aporte de cada uno de los socios.

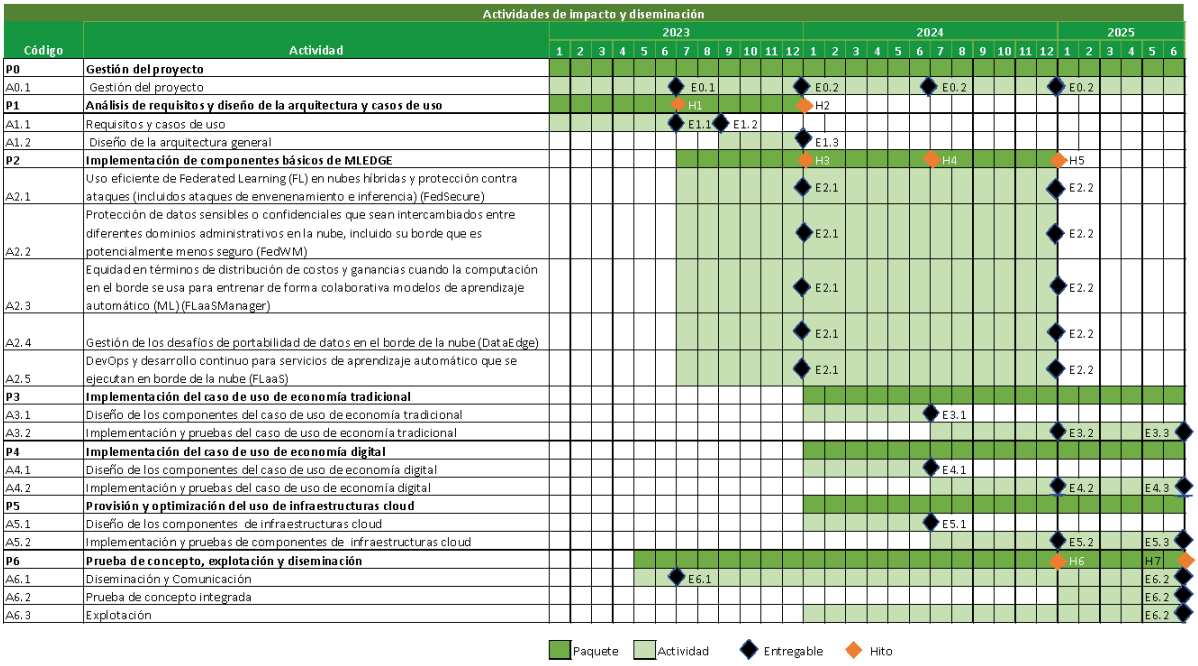

Figura 3. Gantt del proyecto MLEDGE

6. Conclusión

En el presente documento se ha presentado la arquitectura final del proyecto MLEDGE. Esta ha incluido la definición del problema que se pretende atacar, los principales objetivos científicos del proyecto, el estado del arte en cada uno de ellos y cómo el proyecto contribuirá a su desarrollo en sus dos años de duración.

Uno de los grandes objetivos del proyecto y del PRTR es la explotación de los avances científicos generados por parte de la industria. En este sentido, se ha introducido el contenido de tres casos de uso que han sido objeto de sendas licitaciones en el marco del proyecto para contratar agentes de la industria que se encarguen de demostrar la viabilidad del aprendizaje federado en el borde de la nube, y que sirvan a la vez como prueba de concepto para los diferentes componentes científicos que se desarrollen durante el proyecto. Dichas licitaciones se han revisado durante el último trimestre de 2023, y previsiblemente se adjudicarán a principios del año 2024, comenzando inmediatamente la actividad relativa a los casos de uso.

Por tanto, los siguientes pasos en el proyecto son fundamentalmente tres:

- Incorporar al proyecto a las empresas que se encarguen de elaborar los casos de uso del proyecto
- Continuar con el trabajo de investigación y desarrollo de los componentes
- Estudiar la integración de estos componentes con la plataforma FLaaS que se emplee

Durante este proceso y según se disponga de mayor visibilidad respecto a los planes de las diferentes empresas adjudicatarias, se utilizará la información del presente entregable para alinearla con los planes de las empresas adjudicatarias. Esto con el objetivo de incorporar más detalles sobre los casos de uso, sus requisitos y el diseño de la plataforma sobre la que se desarrollarán estos en los documentos de análisis de requisitos y diseño de los casos de uso. Así mismo, se informará en estos entregables sobre qué módulos se van a probar en cada caso de uso y con qué demostradores prácticos se probará su funcionamiento.

7. Referencias

[Agarwal19] Agarwal, A., et al. A Marketplace for Data: An Algorithmic Solution. In Proc. of ACM EC (2019).

[Agarwal19_12] Namita Agarwal, Amit Kumar Singh, and Pradeep Kumar Singh. 2019. Survey of robust and imperceptible watermarking. Multim. Tools Appl. (2019). https://doi.org/10.1007/s11042-018-7128-5.

[Aggarwal05] G. Aggarwal, A. Fiat, A. V. Goldberg, J. D. Hartline, N. Immorlica, and M. Sudan. Derandomization of auctions. In Proc. of STOC. ACM, 2005.

[Agrawal02] Agrawal R., et al. Watermarking relational databases. VLDB (2002).

[Agrawal03] Rakesh Agrawal, Peter J. Haas, and Jerry Kiernan. 2003. A System for Watermarking Relational Databases. In ACM SIGMOD International Conference. https://doi.org/10.1145/872757.872865.

[Anciaux19] Nicolas Anciaux, Luc Bouganim, Philippe Pucheral, Iulian Sandu Popa, and Guillaume Scerri. 2019. Personal Database Security and Trusted Execution Environments: A Tutorial at the Crossroads. Proc. VLDB Endow. (2019). https://doi.org/10.14778/3352063.3352118.

[Andres22] Andrés S., Paraschiv M. and Laoutaris N.. Computing the Relative Value of Spatio-Temporal Data in Data Marketplaces. ACM SIGSPATIAL (2022)

[Andres22_2] Andrés S, and Laoutaris N., A Survey of Data Marketplaces and their Business Models ACM SIGMOD Record, 51(3), (Sep 2022).

[Andres22_3] Andrés S. and Iordanou C. and Laoutaris N.. Measuring the Price of Data in Commercial Data Marketplaces. 1st ACM Data Economy Workshop in CoNEXT'22 (2022).

[Andres22_4] Andrés S, and Laoutaris N.,Try before you buy: a practical data purchasing algorithm for real-world data marketplaces. 1st ACM Data Economy Workshop in CoNEXT'22 (2022).

[Andres23] Andrés S. and Iordanou C. and Laoutaris N. Understanding the Price of Data in Commercial Data Marketplaces. IEEE International Conference on Data Engineering ICDE 2023.

[Asikuzzaman18] Md. Asikuzzaman and Mark R. Pickering. 2018. An Overview of Digital Video Watermarking. IEEE Trans. Circuits Syst. Video Technol. (2018). https://doi.org/10.1109/TCSVT.2017.271216.

[Bagdasaryan20] Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. How to backdoor federated learning. In International Conference on Artificial Intelligence and Statistics, pages 2938–2948. PMLR, 2020.

[Balazinska11] . Balazinska, B. Howe, and D. Suciu. Data markets in the cloud: An opportunity for the database community. In Proc. VLDB, 2011.

[Banterie20] Francesco Banterle. 2020. Data ownership in the data economy: a European dilemma. EU Internet Law in the Digital Era: Regulation and Enforcement (2020), 199–225. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3277330.

[Begum20] Mahbuba Begum and Mohammad Shorif Uddin. 2020. Digital Image Watermarking Techniques: A Review. Inf. (2020). https://doi.org/10.3390/info11020110.

[Bhajogi18] Arjun Nitin Bhagoji, Supriyo Chakraborty, Prateek Mittal, and Seraphin Calo. Analyzing federated learning through an adversarial lens, 2018.

[Bhowmick18] A. Bhowmick, J. Duchi, J. Freudiger, G. Kapoor, and R. Rogers. Protection against reconstruction and its applications in private federated learning. arXiv preprint arXiv:1812.00984, 2018.

[Blanchard17] Peva Blanchard, Rachid Guerraoui, Julien Stainer, et al. Machine learning with adversaries: Byzantine tolerant gradient descent. In Advances in Neural Information Processing Systems, pages 119–129, 2017.

[Bonawitz19] K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, C. Kiddon, J. Konecny, S. Mazzocchi, H. B. McMahan, T. V. Overveldt, D. Petrou, D. Ramage, and J. Roselander. Towards federated learning at scale: system design. In Conference on Systems and Machine Learning, 2019.

[Boneh11] Boneh D., Sahai A., and Waters B.. 2011. Functional Encryption: Definitions and Challenges. In Theory of Cryptography Conference, TCC. Springer. https://doi.org/10.1007/978-3-642-19571-6_16

[Brisimi18] Brisimi, T., et al. FL of Predictive Models from Federated Electronic Health Records. J. Of Medical Informatics. 112, 59-67 (2018).

[Cao21] Cao, X., Fang, M., Liu, J. & Gong, N. FLTrust: Byzantine-robust Federated Learning via Trust Bootstrapping. Proceedings Of NDSS.(2021)

[Carlini18] N. Carlini, C. Liu, J. Kos, Ú. Erlingsson, and D. Song. The secret sharer: Measuring unintended neural network memorization & extracting secrets. arXiv preprint arXiv:1802.08232, 2018.

[Carrascal13] J.P. Carrascal, C. Riederer, V. Erramilli, M. Cherubini, and R. de Oliveira. Your browsing behavior for a big mac: Economics of personal information online. In Proc. of WWW, 2013.

[Chandan20] Chandan, K., et al. SPIHT-based multiple image watermarking in NSCT domain. Concurrency and Computation: Practice and Experience 32.1 (2020).

[Chasins22] Chasins, S., et al. The Sky Above The Clouds - A Berkeley View on the Future of Cloud Computing. ArXiv (2022).

[Chawla19] S. Chawla, S. Deep, P. Koutris, and Y. Teng. Revenue maximization for query pricing. Proc. of VLDB, 13, 2019.

[Chen19] Chen, L., et al. Towards Model-Based Pricing for ML in a Data Marketplace. ACM SIGMOD'19.

[Chu22] Chu, T., et al. FedQV: Leveraging Quadratic Voting in Federated Learning (2022).

[Chu23] Tianyue Chu, Alvaro Garcia-Recuero, Costas Iordanou, Georgios Smaragdakis, Nikolaos Laoutaris, *Securing Federated Sensitive Topic Classification against Poisoning Attacks*, Network and Distributed System Security (NDSS) Symposium 2023

[Cohen18] Aloni Cohen, Justin Holmgren, Ryo Nishimaki, Vinod Vaikuntanathan, and Daniel Wichs. 2018. Watermarking Cryptographic Capabilities. SIAM J. Comput. (2018). https://doi.org/10.1137/18M1164834.

[Cong19] Cong Xie, Sanmi Koyejo, and Indranil Gupta. Zeno: Distributed stochastic gradient descent with suspicion-based fault-tolerance. International Conference on Machine Learning, pages 6893–6901. PMLR, 2019.

[Coyle20] D. Coyle, S. Diepeveen, J. Wdowin, J. Tennison, and L. Kay. The value of data – policy implications. Bennett Institute for Public Policy, Cambridge, 2020

[Dai15] W. Dai, A. Kumar, J. Wei, Q. Ho, G. Gibson, and E. P. Xing. High-performance distributed ML at scale through parameter server consistency models. In AAAI Conference on Artificial Intelligence, 2015.

[Dwork06] Dwork, C. (2006). Differential Privacy. In: Bugliesi, M., Preneel, B., Sassone, V., Wegener, I. (eds) Automata, Languages and Programming. ICALP 2006. Lecture Notes in Computer Science, vol 4052. Springer, Berlin, Heidelberg. https://doi.org/10.1007/11787006_1

[EC20] European Commission. A European Strategy for Data (2020).

[Evans18] David Evans, Vladimir Kolesnikov, and Mike Rosulek. 2018. A Pragmatic Introduction to Secure Multi-Party Computation. Found. Trends Priv. Secur. (2018). https://doi.org/10.1561/3300000019.

[Fang20] Minghong Fang, Xiaoyu Cao, Jinyuan Jia, and Neil Gong. Local model poisoning attacks to byzantine-robust federated learning. In USENIX, 2020.

[Feng20] Feng, J., et al. PMF: A Privacy-preserving Human Mobility Prediction Framework via Federated Learning. Proc. of ACM IMWUT. 4, 1-21 (2020).

[Fredrikson15] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, pages 1322–1333. ACM, 2015.

[Fu21] Fu, S., Xie, C., Li, B. & Chen, Q. Attack-resistant Federated Learning with Residual-based Reweighting. AAAI Workshops: Towards Robust, Secure And Efficient Machine Learning. (2021)

[Fung20] Fung, C., Yoon, C. & Beschastnikh, I. The Limitations of Federated Learning in Sybil Settings. 23rd International Symposium On Research In Attacks, Intrusions

And Defenses (RAID 2020). pp. 301-316 (2020,10), https://www.usenix.org/conference/raid2020/presentation/fung.

[Fung18] Fung, C., Yoon, C. & Beschastnikh, I. Mitigating Sybils in Federated Learning Poisoning. ArXiv Preprint ArXiv:1808.04866. (2018)

[GaiaX] Gaia-X. Gaia-X Architecture Document, 2022.

[Galhotra23] Galhotra S., Gong Y., Castro Fernandez R.. Metam: Goal-Oriented Data Discovery. IEEE International Conference on Data Engineering ICDE 2023.

[Geiping20] Jonas Geiping, Hartmut Bauermeister, Hannah Dr¨oge, and Michael Moeller. Inverting gradients– how easy is it to break privacy in federated learning? arXiv preprint arXiv:2003.14053, 2020.

[Gentry09] Craig Gentry. 2009. A fully homomorphic encryption scheme. Ph.D. Dissertation. Stanford University, USA. https://searchworks.stanford.edu/view/8493082

[Gehrke12] Gehrke, J., Hay, M., Lui, E., & Pass, R. (2012). Crowd-Blending Privacy. 32nd Annual Cryptology Conference. Santa Barbara.

[Ghorbani19] Ghorbani, A., et al. Data Shapley: Equitable Valuation of Data for Machine Learning (2019).

[Giaretta22] L. Giaretta, T. Marchioro, E. Markatos, and ˇS. Girdzijauskas. Towards a decentralized infrastructure for data marketplaces: Narrowing the gap between academia and industry. In Proc. Workshop on the Data Economy, DE '22. ACM, 2022.

[Ghosh11] A. Ghosh and A. Roth. Selling privacy at auction. In Proc. of ACM EC. 2011.

[Goldberg01] A. V. Goldberg, J. D. Hartline, and A. Wright. Competitive auctions and digital goods. In Proc. of ACM-SIAM SODA, 2001.

[Goldberg03] A. V. Goldberg and J. D. Hartline. Competitiveness via consensus. In Proc. of ACM-SIAM SODA, 2003.

[Goldberg06] A. V. Goldberg, J. D. Hartline, A. R. Karlin, M. Saks, and A. Wright. Competitive auctions. Games and Economic Behavior, 55(2), 2006. Special Issue: Electronic Market Design.

[Guo21] Hanxi Guo, Hao Wang, Tao Song, Yang Hua, Zhangcheng Lv, Xiulang Jin, Zhengui Xue, Ruhui Ma, and Haibing Guan. Siren: Byzantine-robust federated learning via proactive alarming. In Proceedings of the ACM Symposium on Cloud Computing, pages 47–60, 2021

[Gwenael03] Gwenael, D., et al. A guide tour of video watermarking. Signal processing: Image comm. (2003).

[Hard18] Hard, A., et al. FL for Mobile Keyboard Prediction. ArXiv Preprint ArXiv:1811.03604. (2018).

[He18] L. He, A. Bian, and M. Jaggi. Cola: Decentralized linear learning. In Advances in Neural Information Processing Systems, 2018.

[Heckman15] . R. Heckman, E. Boehmer, E. H. Peters, M. Davaloo, and N. G Kurup. A pricing model for data markets. In Proc. of iConference, 2015

[Ho13] Q. Ho, J. Cipar, H. Cui, S. Lee, J. K. Kim, P. B. Gibbons, G. A. Gibson, G. Ganger, and E. P. Xing. More effective distributed ML via a stale synchronous parallel parameter server. In Advances in Neural Information Processing Systems, 2013.

[Huang19] Huang, L., et al. Patient Clustering Improves Efficiency of Federated ML to Predict Mortality and Hospital Stay Time using Distributed Electronic Medical Records. J. of Biomedical Informatics (2019).

[IDSA21] International Data Spaces Association. International Data Spaces Information Model: Ontology draft, 2021..

[IDSA_GaiaX] International Data Spaces Association. Gaia-X and International Data Spaces, 2021.

[IDSA22] International Data Spaces Association. International Data Spaces Reference Architecture Model v4.0, 2022.

[İşler22] İşler, Devriş; Cabana, Elisa; Laoutaris, Nikolaos. FreqyWM: Frequency WaterMarking for the New Data Economy. https://dspace.networks.imdea.org/handle/20.500.12761/1626

[Ji21] Tianxi Ji, Emre Yilmaz, Erman Ayday, and Pan Li. 2021. The Curse of Correlations for Robust Fingerprinting of Relational Databases. In RAID : International Symposium on Research in Attacks, Intrusions and Defenses. ACM. https://doi.org/10.1145/3471621.3471853.

[Jia19] Jia, R., et al. Efficient task-specific data valuation for nearest neighbor algorithms. VLDB'19.

[Jougleux22] Philippe Jougleux. 2022. Data Ownership (and Succession Law). In Facebook and the (EU) Law: How the Social Network Reshaped the Legal Framework. Springer, 129–143.

[Kairouz21] Peter Kairouz, H. Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, Rafael G. L. D'Oliveira, Hubert Eichner, Salim El Rouayheb, David Evans, Josh Gardner, Zachary Garrett, Adrià Gascón, Badih Ghazi, Phillip B. Gibbons, Marco Gruteser, Zaid Harchaoui, Chaoyang He, Lie He, Zhouyuan Huo, Ben Hutchinson, Justin Hsu, Martin Jaggi, Tara Javidi, Gauri Joshi, Mikhail Khodak, Jakub Konecný, Aleksandra Korolova, Farinaz Koushanfar, Sanmi Koyejo, Tancrède Lepoint, Yang Liu, Prateek Mittal, Mehryar Mohri, Richard Nock, Ayfer Özgür, Rasmus Pagh, Hang Qi, Daniel Ramage, Ramesh Raskar, Mariana Raykova, Dawn Song, Weikang Song, Sebastian U. Stich, Ziteng Sun, Ananda Theertha Suresh, Florian Tramèr, Praneeth Vepakomma, Jianyu Wang, Li Xiong, Zheng Xu, Qiang Yang, Felix X. Yu, Han Yu and Sen Zhao (2021), "Advances and Open Problems in Federated Learning", Foundations and Trends® in Machine Learning: Vol. 14: No. 1–2, pp 1-210. http://dx.doi.org/10.1561/2200000083.

[Kang19] J. Kang, Z. Xiong, D. Niyato, S. Xie and J. Zhang, "Incentive Mechanism for Reliable Federated Learning: A Joint Optimization Approach to Combining Reputation and Contract Theory" in IEEE Internet of Things Journal, vol. 6, no. 6, pp. 10700-10714, Dec. 2019, doi: 10.1109/JIOT.2019.2940820.

[Karimireddy21] Karimireddy, S., He, L. & Jaggi, M. Learning from History for Byzantine Robust Optimization. International Conference On Machine Learning. pp. 5311-5319 (2021)

[Katevas22] Katevas, K., et al. FLaaS - practical federated learning as a service for mobile applications. In Proc. HotMobile (2022).

[Konečny16] Konečny, J., et al. FL: Strategies for Improving Communication Efficiency. Proc. of NIPS (2016)

[Kourtellis20] Kourtellis N., Katevas K., and Perino D. 2020. FLaaS: Federated Learning as a Service. In Proceedings of the 1st Workshop on Distributed Machine Learning (DistributedML'20). https://doi.org/10.1145/3426745.3431337

[Koutris12] Koutris, P. Upadhyaya, M. Balazinska, B. Howe, and D. Suciu. Querymarket demonstration: Pricing for online data markets. Proc. of VLDB, 5, 2012.

[Koutris15] Koutris, P. Upadhyaya, M. Balazinska, B. Howe, and D. Suciu. Query-based data pricing. Journal of ACM, 62(5), November 2015.

[Koutsos20] . Koutsos, D. Papadopoulos, D. Chatzopoulos, S. Tarkoma, and P. Hui. Agora: A privacy-aware data marketplace. In IEEE ICDCS, 2020.

[Kumar20] Sanjay Kumar, Binod Kumar Singh, and Mohit Yadav. 2020. A Recent Survey on Multimedia and Database Watermarking. Multim. Tools Appl. 79, 27-28 (2020), 20149–20197. https://doi.org/10.1007/s11042-020-08881-y.

[Laoutaris11] Laoutaris, N., et al. Inter-datacenter bulk transfers with netstitcher. SIGCOMM. 74–85. (2011).

[Lee18] Lee, J., et al. Privacy-preserving Patient Similarity Learning in a Federated Environment: Development and Analysis. JMIR Medical Informatics. 6, e7744 (2018).

[Legout06] Arnaud Legout, G. Urvoy-Keller, and P. Michiardi. 2006. Rarest first and choke algorithms are enough. In Proceedings of the 6th ACM SIGCOMM conference on Internet measurement (IMC '06).

[Legout07] Arnaud Legout, Nikitas Liogkas, Eddie Kohler, and Lixia Zhang. 2007. Clustering and sharing incentives in BitTorrent systems. In Proceedings of the 2007 ACM SIGMETRICS international conference on Measurement and modeling of computer systems (SIGMETRICS '07)

[Li06] Yingjiu Li and Robert Huijie Deng. 2006. Publicly verifiable ownership protection for relational databases. In Proceedings of th ACM Symposium on Information, Computer and Communications Security, ASIACCS. ACM. https://doi.org/10.1145/1128817.1128832.

[Li15] C. Li, D. Y. Li, G. Miklau, and D. Suciu. A theory of pricing private data. ACM Trans.Database Syst., 2015.

[Li18] T. Li, A. K. Sahu, M. Sanjabi, M. Zaheer, A. Talwalkar, and V. Smith. Federated optimization for heterogeneous networks. arXiv preprint arXiv:1812.06127, 2018.

[Li19] Li, Tian & Sahu, Anit & Talwalkar, Ameet & Smith, Virginia. (2019). Federated Learning: Challenges, Methods, and Future Directions.

[Li20] Li, T., and Sanjabi M. and Beirami A. and Smith V. “Fair Resource Allocation in Federated Learning” 2020. ICLR.

[Li21] Yin Li, Dhrubajyoti Ghosh, Peeyush Gupta, Sharad Mehrotra, Nisha Panwar, and Shantanu Sharma. 2021. PRISM: Private Verifiable Set Computation over Multi-Owner Outsourced Databases. In SIGMOD: International Conference on Management of Data, Virtual. ACM. https://doi.org/10.1145/3448016.3452839.

[Li21_2] Q. Li et al., "A Survey on Federated Learning Systems: Vision, Hype and Reality for Data Privacy and Protection," in IEEE Transactions on Knowledge and Data Engineering, doi: 10.1109/TKDE.2021.3124599.

[Li22] Wenling Li, Ning Li, Jianen Yan, Zhaoxin Zhang, Ping Yu, and Gang Long. 2022. Secure and High-Quality Watermarking Algorithms for Relational Database Based on Semantic. IEEE Transactions on Knowledge and Data Engineering (2022), 1–14. https://doi.org/10.1109/TKDE.2022.3194191.

[Lian17] X. Lian, C. Zhang, H. Zhang, C.-J. Hsieh, W. Zhang, and J. Liu. Can decentralized algorithms outperform centralized algorithms? a case study for decentralized parallel stochastic gradient descent. In Advances in Neural Information Processing Systems, 2017.

[Liang20] Liang X., et al. Robust reversible audio watermarking based on high-order difference statistics. Signal Processing 173 (2020).

[Ligeng19] Ligeng Zhu, Zhijian Liu, and Song Han. Deep leakage from gradients. Advances in neural information processing systems, 32, 2019

[LightR23] Mike Dano. Telecom suits up for predicted edge-computing boom. Light Reading. https://www.lightreading.com/the-edge/telecom-suits-up-for-predicted-edge-computing-boom/d/d-id/783958, último acceso en junio 2023.

[Lin14] B. R. Lin and D. Kifer. On arbitrage-free pricing for general data queries. Proc. VLDB Endow., 7(9), 2014.

[Lin18] T. Lin, S. U. Stich, and M. Jaggi. Don’t use large mini-batches, use local sgd. arXiv preprint arXiv:1808.07217, 2018.

[Lin21] Chia-Chen Lin, Thai-Son Nguyen, and Chin-Chen Chang. 2021. LRW-CRDB: Lossless Robust Watermarking Scheme for Categorical Relational Databases. Symmetry (2021). https://doi.org/10.3390/sym13112191.

[Liu19] L. Liu, J. Zhang, S. Song, and K. B. Letaief. Edge-assisted hierarchical federated learning with non-iid data. arXiv preprint arXiv:1905.06641, 2019.

[Liu21] Liu Z., Chen Y., Yu H., Liu Y., Cui L. GTG-Shapley: Efficient and Accurate Participant Contribution Evaluation in Federated Learning. arXiv:2109.02053

[Lutu20] Andra Lutu, Diego Perino, Marcelo Bagnulo, Enrique Frias-Martinez, and Javad Khangosstar. 2020. A Characterization of the COVID-19 Pandemic Impact on a Mobile Network Operator Traffic. In Proceedings of the ACM Internet Measurement Conference (IMC '20). Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3419394.3423655

[Ma19] Haoyu Ma, Chunfu Jia, Shijia Li,Wantong Zheng, and DinghaoWu. 2019. Xmark: Dynamic Software Watermarking Using Collatz Conjecture. IEEE Trans. Inf. Forensics Secur. (2019). https://doi.org/10.1109/TIFS.2019.2908071.

[Machanavajjhala07] Machanavajjhala, A., Kifer, D., Gehrke, J., & Venkitasubramaniam, M. (2007). L-diversity: PrivacyBeyond K-anonymity. ACM Transactions on Knowledge Discovery from Data, vol. 1, no. 1.

[Machanavajjhala15] Machanavajjhala, A., & Kifer, D. (2015). Designing Statistical Privacy for Your Data Communications of the ACM, vol. 58, no. 3, (pp. 58-67)

[Mahdi18] El Mahdi, E. M., Guerraoui, R., Rouault, S. The Hidden Vulnerability of Distributed Learning in Byzantium. International Conference OnMachine Learning. pp. 3521-3530 (2018)

[McMahan17] McMahan, H., et al. Communication-Efficient Learning of Deep Networks from Decentralized Data. Conf. AI And Statistics, 1273-1282 (2017).

[Mehdi21] Mehdi Hassan Jony, Fatema Tuj Johora, and Jannatul Ferdous Katha. 2021. A Robust and Efficient Numeric Approach for Relational Database Watermarking. In IEEE International Conference on Sustainable Technologies for Industry 4.0 (STI). https://doi.org/10.1109/STI53101.2021.9732582.

[Melis19] L. Melis, C. Song, E. De Cristofaro, and V. Shmatikov. Exploiting unintended feature leakage in collaborative learning. In IEEE Symposium on Security & Privacy, 2019.

[Mhamdi18] El Mahdi El Mhamdi, Rachid Guerraoui, and S´ebastien Rouault. The hidden vulnerability of distributed learning in byzantium. arXiv preprint arXiv:1802.07927, 2018.

[Mohri19] Mehryar Mohri, Gary Sivek, and Ananda Theertha Suresh. Agnostic federated learning. In International Conference on Machine Learning, 2019.

[Naseri22] Naseri, M., Hayes, J. & De Cristofaro, E. Local and Central DifferentialPrivacy for Robustness and Privacy in Federated Learning. In Proceedings of NDSS. (2022)

[Niu19] Niu, Z. Zheng, F. Wu, X. Gao, and G. Chen. Achieving data truthfulness and privacy preservation in data markets. IEEE Trans. KDE, 31, 2019.

[OECD13] OECD. Exploring the economics of personal data: A survey of methodologies for measuring monetary value.OECD Digital Economy Papers 2013.

[Ohrimenko19] Ohrimenko, Olga, Shruti Tople and Sebastian Tschiatschek. “Collaborative Machine Learning Markets with Data-Replication-Robust Payments.” ArXiv abs/1911.09052 (2019)

[Pasquini23] Pasquini D., Raynal M. y Troncoso C. On the (In)security of Peer-to-Peer Decentralized Machine Learning. ArXiv:2205:08443v2.

[Pei20] J. Pei. Data pricing – from economics to data science. In Proc. of SIGKDD. ACM, 2020

[Perez21] Maikel Lázaro Pérez Gort, Martina Olliaro, Agostino Cortesi, and Claudia Feregrino Uribe. 2021. Semantic-driven watermarking of relational textual databases. Expert Syst. Appl. (2021). https://doi.org/10.1016/j.eswa.2020.114013.

[Poddar19]Rishabh Poddar, Tobias Boelter, and Raluca Ada Popa. 2019. Arx: An Encrypted Database using Semantically Secure Encryption. Proc. VLDB Endow. (2019). https://doi.org/10.14778/3342263.3342641.

[Quiring18] Erwin Quiring, Daniel Arp, and Konrad Rieck. 2018. Forgotten Siblings: Unifying Attacks on Machine Learning and Digital Watermarking. In IEEE European Symposium on Security and Privacy, EuroS&P. IEEE. https://doi.org/10.1109/EuroSP.2018.00041.

[Reddi16] S. J. Reddi, J. Koneˇcn` y, P. Richtárik, B. Póczós, and A. Smola. Aide: Fast and communication efficient distributed optimization. arXiv preprint arXiv:1608.06879, 2016.

[Ren22] Xuanle Ren, Le Su, Zhen Gu, Sheng Wang, Feifei Li, Yuan Xie, Song Bian, Chao Li, and Fan Zhang. 2022. HEDA: Multi-Attribute Unbounded Aggregation over Homomorphically Encrypted Database. Proc. VLDB Endow. (2022). https://www.vldb.org/pvldb/vol16/p601-gu.pdf.

[Rieger22] Rieger, P., Nguyen, T., Miettinen, M. & Sadeghi, A. DeepSight: Mitigating Backdoor Attacks in Federated Learning Through Deep Model Inspection. In Proceedings of NDSS. (2022)

[Robin17] Robin C Geyer, Tassilo Klein, and Moin Nabi. Differentially private federated learning: A client level perspective. arXiv preprint arXiv:1712.07557, 2017.

[Rozemberczki22] Rozemberczki, B., et al. The Shapley Value in Machine Learning (2022).

[Sabt15] Mohamed Sabt, Mohammed Achemlal, and Abdelmadjid Bouabdallah. 2015. Trusted Execution Environment: What It is, and What It is Not. In TrustCom/Big-DataSE/ISPA. IEEE. https://doi.org/10.1109/Trustcom.2015.357

[Sapana22] Sapana Rani and Raju Halder. 2022. Comparative Analysis of Relational Database Watermarking Techniques: An Empirical Study. IEEE Access 10 (2022), 27970–27989. https://doi.org/10.1109/ACCESS.2022.3157866.

[Shamir79] Shamir, A. 1979. “How to Share a Secret.” CACM 22 (11): 612–613.

[Shehab08] Mohamed Shehab, Elisa Bertino, and Arif Ghafoor. 2008. Watermarking Relational Databases Using Optimization-Based Techniques. IEEE Trans. Knowl. Data Eng. (2008). https://doi.org/10.1109/TKDE.2007.190668.

[Shejwalkar21] Shejwalkar, V. & Houmansadr, A. Manipulating the Byzantine: Optimizing Model Poisoning Attacks and Defenses for Federated Learning. NDSS. (2021)

[Smith17] V. Smith, C.-K. Chiang, M. Sanjabi, and A. Talwalkar. Federated multi-task learning. In Advances in Neural Information Processing Systems, 2017.

[Song19] Song, T., et al. Profit Allocation for Federated Learning. IEEE International Conference on Big Data (Big Data), pp. 2577-2586 (2019).

[STLP23] Ahmed Ali. Hyperscalers & telcos: Edge partnership opportunities https://stlpartners.com/articles/edge-computing/hyperscalers-telcos-edge-partnership-opportunities/, último acceso en junio 2023.

[Sun19] Ziteng Sun, Peter Kairouz, Ananda Theertha Suresh, and H Brendan McMahan. Can you really backdoor federated learning? arXiv preprint arXiv:1911.07963, 2019.

[Sweeney02] Sweeney, L. (2002). K-anonymity: A Model for Protecting Privacy. International Journal of Uncertainty, Fuzziness and Knowledge-Based, vol. 10, no. 5, (pp. 557-570).

[Tang21] Xiao Tang, Zhenfu Cao, Xiaolei Dong, and Jiachen Shen. 2021. PKMark: A Robust Zero-distortion Blind Reversible Scheme for Watermarking Relational Databases. In IEEE International Conference on Big Data Science and Engineering. https://doi.org/10.1109/BigDataSE53435.2021.00020.

[Truex19] Stacey Truex, Nathalie Baracaldo, Ali Anwar, Thomas Steinke, Heiko Ludwig, Rui Zhang, and Yi Zhou. A hybrid approach to privacy-preserving federated learning. In Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security, pages 1–11. ACM, 2019.

[Truta06] Truta, T. M., & Vina, B. (2006). Privacy Protection: p-Sensitive k-Anonymity Property. 22nd International Conference on Data Engineering Workshops. Atlanta.

[Valancius09] Valancius, V., et al. Greening the internet with nano data centers. Proc. of ACM CoNEXT (2009).

[Xie19] Chulin Xie, Keli Huang, Pin-Yu Chen, and Bo Li. Dba: Distributed backdoor attacks against federated learning. In International Conference on Learning Representations, 2019.

[Yin18] Yin, D., Chen, Y., Kannan, R. & Bartlett, P. Byzantine-robust Distributed Learning: Towards Optimal Statistical Rates. International Conference On Machine Learning. pp. 5650-5659 (2018)

[Yu17] . Yu and M. Zhang. Data pricing strategy based on data quality. Computers and Industrial Engineering, 112, 2017.

[Yu20] Yu, T., et al. Learning Context-aware Policies from Multiple Smart Homes via Federated Multi-task Learning. IEEE/ACM IoTDI, 104- 115 (2020).

[Wang18] H. Wang, S. Sievert, S. Liu, Z. Charles, D. Papailiopoulos, and S. Wright. Atomo: Communication-efficient learning via atomic sparsification. In Advances in Neural Information Processing Systems, 2018.

[Wang19] G. Wang. Interpret Federated Learning with Shapley Values. arXiv:1905.04519

[Wang20] Hongyi Wang, Kartik Sreenivasan, Shashank Rajput, Harit Vishwakarma, Saurabh Agarwal, Jyyong Sohn, Kangwook Lee, and Dimitris Papailiopoulos. Attack of the tails: Yes, you really can backdoor federated learning. Advances in Neural Information Processing Systems, 33, 2020.

[Wang20_2] Tianhao Wang, Johannes Rausch, Ce Zhang, Ruoxi Jia, Dawn Song. *A Principled Approach to Data Valuation for Federated Learning*. arXiv:2009.06192

[Wang21] Tianhao Wang and Florian Kerschbaum. 2021. RIGA: Covert and Robust White-Box Watermarking of Deep Neural Networks. In WWW: The Web Conference. https://doi.org/10.1145/3442381.3450000.

[Zhang15] S. Zhang, A. E. Choromanska, and Y. LeCun. Deep learning with elastic averaging sgd. In Advances in Neural Information Processing Systems, 2015.

[Zhang17] H. Zhang, J. Li, K. Kara, D. Alistarh, J. Liu, and C. Zhang. ZipML: Training linear models with end-to-end low precision, and a little bit of deep learning. In International Conference on Machine Learning, 2017.

[Zhao23] Zhao B., Lyu B., Castro Fernandez R., Kolar M. *Addressing Budget Allocation and Revenue Allocation in Data Market Environments Using an Adaptive Sampling Algorithm* . arXiv:2306.02543

[Zhou05] Xuan Zhou, HweeHwa Pang, Kian-Lee Tan, and Dhruv Mangla. 2005. WmXML: A System for Watermarking XML Data. In International Conference on Very Large Data Bases (VLDB). ACM. http://www.vldb.org/conf/2005/papers/p1318-zhou.pdf.

[Zhou21] Wenchao Zhou, Yifan Cai, Yanqing Peng, ShengWang, Ke Ma, and Feifei Li. 2021. VeriDB: An SGX-based Verifiable Database. In SIGMOD: International Conference on Management of Data. ACM. https://doi.org/10.1145/3448016.3457308.

[Zinkevich10] M. Zinkevich, M. Weimer, L. Li, and A. J. Smola. Parallelized stochastic gradient descent. In Advances in Neural Information Processing Systems, 2010.