Agregación Federada de Mecanismos de Atención Mediante Similitud para Predicción de Series Temporales

Beatriz Bello-García[1 [0000−0002−3943−5736]], Ignacio Aguilera-Martos[1 [0000−0001−9620−5235]], Nuria Rodríguez-Barroso[1 [0000−0001−7172−9059]], Julián Luengo[1 [0000−0003−3952−3629]], and Francisco Herrera[1 [0000−0002−7283−312X]]

Departamento de Ciencias de la Computación e Inteligencia Artificial, Instituto Andaluz Interuniversitario en Ciencia de Datos e Inteligencia Computacional (DaSCI), Universidad de Granada, Granada, España.
{bbgarcia,nacheteam,rbnuria}@ugr.es, {julianlm,herrera}@decsai.ugr.es

Abstract. La predicción de series temporales es una tarea crítica en dominios como la salud, la energía, las finanzas y el control de procesos industriales. Las arquitecturas basadas en Transformers han demostrado un rendimiento destacado en este ámbito gracias a su capacidad para capturar dependencias a largo plazo mediante mecanismos de auto-atención. Sin embargo, en escenarios reales los datos suelen estar distribuidos entre múltiples fuentes heterogéneas y sensibles, lo que hace inviable el entrenamiento centralizado. El Aprendizaje Federado surge como solución natural a este reto, permitiendo el entrenamiento colaborativo sin comprometer la privacidad de los datos. Este trabajo propone AFAS (Agregación Federada de la Atención mediante Similitud), un nuevo método de agregación federada diseñado específicamente para modelos Transformer en entornos de predicción distribuida basado en la similitud coseno entre cabezas de atención. AFAS se evalúa sobre varios conjuntos de datos reales de series temporales, y los resultados muestran que el método propuesto mejora el rendimiento predictivo respecto a FedAvg en diversos escenarios, preservando al mismo tiempo la privacidad de los datos.

Keywords: Agregación · Aprendizaje Federado · Atención de múltiples cabezas · Predicción · Series temporales

1 Introducción

La predicción de series temporales es esencial en diversos campos como el análisis del consumo energético [8], la predicción meteorológica [23] o el control de procesos industriales [21]. Una serie temporal es una secuencia cronológica de observaciones en la que cada valor depende, en mayor o menor medida, de los anteriores. Esta dependencia plantea retos específicos para los modelos de predicción, que deben identificar patrones a corto y largo plazo, incluyendo tenden-

cias, estacionalidades, ciclos irregulares y posibles anomalías. En este contexto, la predicción adquiere un papel central al permitir anticipar el comportamiento futuro del sistema a partir de datos históricos, lo que resulta crucial para optimizar recursos, prevenir fallos, mejorar la eficiencia operativa y facilitar una toma de decisiones más informada y estratégica.

Los modelos basados en Transformers han demostrado un alto rendimiento en el análisis de series temporales gracias a su capacidad para procesar secuencias largas y capturar dependencias temporales a largo plazo mediante su arquitectura de auto-atención [2]. Su componente clave es el mecanismo de atención multi-cabeza, que pondera la importancia de cada elemento de la secuencia respecto a los demás, permitiendo una representación rica y contextualizada [3]. Modelos como Informer [24] o Local Attention Mechanism (LAM) [1] han sido desarrollados para mejorar la eficiencia y la precisión en la predicción, optimizando las relaciones a largo plazo y reduciendo la complejidad computacional de la auto-atención. Sin embargo, estos modelos asumen típicamente la disponibilidad de datos centralizados, una condición que no siempre puede garantizarse.

El Aprendizaje Federado (AF) permite entrenar modelos de forma colaborativa sin necesidad de compartir los datos locales [11, 22], lo que lo convierte en una opción natural y necesaria para la predicción en entornos distribuidos con restricciones de privacidad. Diversos estudios han demostrado que este enfoque puede alcanzar un rendimiento comparable al del aprendizaje centralizado preservando la confidencialidad de los datos [16]. No obstante, en contextos con alta heterogeneidad entre clientes o patrones temporales complejos [5], el rendimiento del modelo federado puede verse comprometido por la dificultad de capturar patrones generales a partir de información local dispersa. Este problema se agrava en arquitecturas Transformer cuando el método de agregación FedAvg promedia indiscriminadamente los pesos de las cabezas de atención, destruyendo la especialización que cada cliente ha desarrollado localmente.

Para abordar estos desafíos, este trabajo propone AFAS (Agregación Federada de la Atención mediante Similitud), un método de agregación federada diseñado específicamente para modelos Transformer aplicados a la predicción de series temporales multivariantes. AFAS introduce una estrategia de selección de cabezas de atención basada en la similitud con el modelo global y el consenso entre clientes, ofreciendo una alternativa al promediado clásico que preserva la especialización aprendida localmente. El objetivo es proporcionar una solución viable frente a los esquemas centralizados, especialmente en escenarios con restricciones de privacidad y dinámicas de datos complejas. AFAS se evalúa sobre varios conjuntos de datos reales con patrones temporales variados, considerando distintas configuraciones de número de clientes y horizontes de predicción.

Este trabajo se estructura como sigue. La Sección 2 revisa los estudios relacionados e introduce los conceptos fundamentales que sustentan esta investigación. La Sección 3 describe el método propuesto y sus componentes principales. La Sección 4 detalla los conjuntos de datos, las configuraciones de parámetros y el

diseño experimental. La Sección 5 presenta los resultados y ofrece un análisis detallado de los mismos. Finalmente, la Sección 6 resume las principales contribuciones y conclusiones del estudio.

2 Marco teórico

En esta sección se presentan los conceptos fundamentales que sustentan este trabajo, abordando los aspectos esenciales de la predicción de series temporales, el uso de Transformers en este contexto y los avances recientes en la aplicación del AF a series temporales.

2.1 Transformers

La arquitectura Transformer [19] ha revolucionado el procesamiento de secuencias gracias al mecanismo de autoatención, el cual permite procesar datos en paralelo y capturar dependencias complejas sin importar la distancia entre los elementos. Sin embargo, en la formulación original, la atención completa calcula las interacciones entre todos los pares de una secuencia de longitud L mediante $\text{Attn}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_q}}\right)V$, donde Q, K y V representan las matrices de consulta (*query*), clave (*key*) y valor (*value*), respectivamente, y d_q es la dimensión de las consultas. Este cálculo genera una matriz de coste cuadrático $\mathcal{O}(L^2)$ en tiempo y memoria, lo que limita severamente su aplicación en series temporales extensas.

Para mitigar esta limitación, el modelo Informer [24] introduce el mecanismo *ProbSparse Self-Attention*. Este enfoque selecciona únicamente las consultas Q_i estadísticamente más significativas aquellas cuya distribución de atención se aleja más de una distribución uniforme [12], reduciendo la complejidad computacional de $\mathcal{O}(L^2)$ a $\mathcal{O}(L \log L)$. El mecanismo se define como $\text{ProbAttn}(Q, K, V) = \text{softmax}\left(\frac{\text{ProbSelect}(Q,K)K^T}{\sqrt{d_q}}\right)V$, donde $\text{ProbSelect}(Q, K)$ denota el subconjunto de las $u = c \cdot \ln L$ consultas seleccionadas, mientras que las restantes se reemplazan por un promedio simple para minimizar el cómputo. El uso de mecanismos como *ProbSparse* permite a los nodos procesar series temporales de largo horizonte localmente de forma eficiente, reduciendo los tiempos de entrenamiento y la carga computacional por nodo, lo que abre nuevas oportunidades para su aplicación en dominios críticos como la salud o la monitorización industrial. Por estas razones hemos seleccionado este modelo para la experimentación de este trabajo.

2.2 Aprendizaje Federado

Trabajos recientes han explorado la integración de modelos basados en Transformers en entornos federados para tareas de series temporales y datos secuenciales [6,14,15,17]. Estos enfoques demuestran la efectividad de las arquitecturas

basadas en atención para capturar patrones temporales complejos en escenarios con datos distribuidos y heterogéneos. A pesar de estos avances, la mayoría de los enfoques existentes en AF con Transformers se basan en esquemas de agregación estándar, principalmente derivados de FedAvg, donde los parámetros de los modelos locales se promedian de manera uniforme para construir el modelo global. Este tipo de estrategias trata todos los parámetros de la red de forma homogénea, sin considerar su función dentro del modelo.

Asimismo, en la literatura se han propuesto diversas estrategias orientadas a explorar técnicas de agregación y personalización en diferentes contextos de AF. Muchas de estas metodologías han sido recopiladas y sistematizadas en el estudio exhaustivo de Qi et al. [13], abarcando enfoques de optimización y alineación de parámetros como FedProx [10], SCAFFOLD [7] y FedMA [20], así como propuestas de personalización entre las que destacan FedTP [9] y FedPerfix [18]. Sin embargo, todos estos métodos continúan operando de manera agnóstica a la arquitectura o a nivel de parámetros completos. Como resultado, los procesos de agregación actuales ignoran la granularidad de los componentes específicos de los Transformers, omitiendo por completo los patrones aprendidos por las distintas cabezas de atención durante la consolidación global del sistema. No obstante, a pesar de explotar el concepto de atención para guiar la combinación de modelos, estas estrategias siguen reduciéndose a un esquema de promediado ponderado por capa.

3 AFAS: Agregación federada de la atención mediante similitud

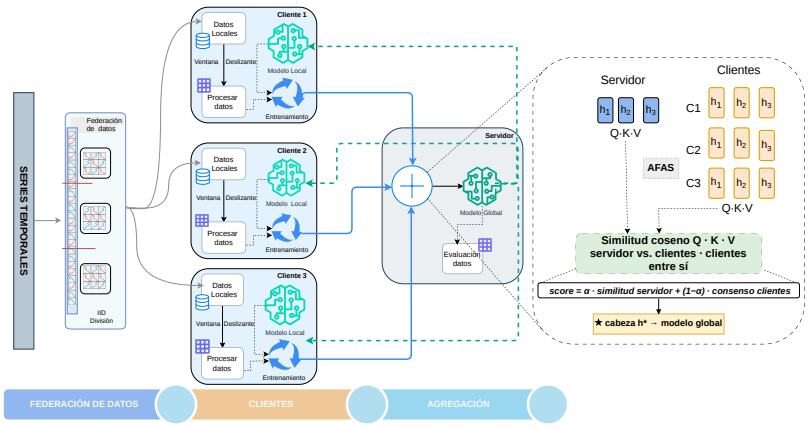


Fig. 1: Metodología para la agregación federada de la atención mediante similitud para predicción de series temporales: AFAS, ejemplo con tres clientes y un servidor central. El panel «Agregación» ofrece una vista detallada y ampliada de la propuesta de agregación.

En esta sección se describe AFAS, un nuevo método de agregación federada diseñado específicamente para modelos Transformer en entornos de predicción distribuida. A diferencia de FedAvg, que promedia indiscriminadamente los pesos de todas las capas del modelo, AFAS introduce una estrategia de selección específica para las capas de atención multi-cabeza: en lugar de promediar, elige para cada posición de cabeza del modelo global la cabeza cliente más adecuada, combinando dos criterios complementarios el consenso entre clientes y la coherencia con el modelo global. De este modo se preserva la especialización que cada cliente ha desarrollado localmente, evitando la pérdida de información que el promedio introduce en estas capas. El resto de capas del modelo se agrega mediante FedAvg estándar.

AFAS aborda la agregación de las capas de atención mediante un proceso de selección guiada que, en cada ronda de entrenamiento, evalúa las cabezas de atención de todos los clientes como candidatas para cada posición del modelo global. Para ello, un conjunto de datos de referencia pequeño y controlado, independiente de los datos locales de cada cliente y gestionado exclusivamente por el servidor, es procesado por los modelos de todos los clientes y del servidor, obteniendo los vectores de activación normalizados de las proyecciones Q, K y V de cada cabeza. Dichos vectores permiten cuantificar, mediante similitud coseno, tanto el alineamiento de cada cabeza del servidor con cada cabeza disponible entre todos los clientes, como la representatividad de cada cabeza cliente respecto al consenso de los demás clientes. La cabeza seleccionada para cada posición es aquella que mejor combina ambas señales, y sus pesos se copian directamente al modelo agregado sin ser promediado.

Notación

Para el resto del artículo, C denota el número de clientes y N el número de cabezas de atención por capa, por lo que $M = C \times N$ es el número total de cabezas-cliente disponibles. Las activaciones se obtienen pasando un conjunto de referencia $\mathcal{D}_{\text{ref}}$ por el modelo y normalizando ($\|\cdot\| = 1$) las salidas de Q, K y V para cada cabeza. Así, $\mathbf{q}_h^{(c)}$, $\mathbf{k}_h^{(c)}$ y $\mathbf{v}_h^{(c)}$ representan las activaciones normalizadas del cliente c en la cabeza h, mientras que $\hat{\mathbf{q}}_i^{(s)}$, $\hat{\mathbf{k}}_i^{(s)}$ y $\hat{\mathbf{v}}_i^{(s)}$ representan las activaciones equivalentes del servidor en la cabeza i. El parámetro $\alpha \in [0, 1]$ pondera la similitud con el servidor, y κ limita la reutilización máxima de cada cabeza-cliente.

El parecido entre dos cabezas cualesquiera se mide mediante la similitud coseno promediada sobre las tres proyecciones:

$$\text{sim}(a, b) \;=\; \frac{1}{3}\big(\mathbf{q}_a^\top \mathbf{q}_b + \mathbf{k}_a^\top \mathbf{k}_b + \mathbf{v}_a^\top \mathbf{v}_b\big) \tag{1}$$

Ronda 1: selección por centralidad

En la primera ronda el servidor aún no dispone de pesos entrenados, por lo que la única señal disponible es el acuerdo entre clientes. Para cada capa de atención y cada posición de cabeza h, AFAS calcula la *centralidad par a par* de cada cliente su similitud media con el resto del grupo y selecciona la cabeza más representativa del consenso:

$$\gamma_h^{(c)} \;=\; \frac{1}{C-1}\sum_{c'\neq c}\mathrm{sim}((c,h),\,(c',h))\,, \qquad c_h^{*} = \arg\max_{c}\;\gamma_h^{(c)} \tag{2}$$

Rondas $\geq$ 2: puntuación combinada

A partir de la segunda ronda el servidor ya tiene pesos entrenados y sus activaciones sirven como referencia adicional. Cada cabeza-cliente se indexa como $j = c \cdot N + h$, con $c(j) = \lfloor j/N \rfloor$ y $h(j) = j \bmod N$. Para cada par (cabeza de servidor i, cabeza-cliente j) se calcula:

$$S[i,j] \;=\; \mathrm{sim}\Big(\hat{\mathbf{q}}_i^{(s)}, \hat{\mathbf{k}}_i^{(s)}, \hat{\mathbf{v}}_i^{(s)};\; \mathbf{q}_{h(j)}^{(c(j))}, \mathbf{k}_{h(j)}^{(c(j))}, \mathbf{v}_{h(j)}^{(c(j))}\Big) \tag{3}$$

$$\gamma[j] \;=\; \frac{1}{C-1}\sum_{c'\neq c(j)}\mathrm{sim}((c(j),h(j)),\,(c',h(j))) \tag{4}$$

Ambos términos se normalizan a $[0,1]$ mediante min-max ($\widetilde{S}$, $\widetilde{\gamma}$) para que α actúe como un peso verdaderamente interpretable. La puntuación combinada final es:

$$\mathrm{score}[i,j] \;=\; \alpha\,\widetilde{S}[i,j] \;+\; (1-\alpha)\,\widetilde{\gamma}[j] \tag{5}$$

La asignación $[N]\to[M]$ se resuelve con un algoritmo greedy con restricción de capacidad $\kappa = \max(1, \lfloor N/2 \rfloor)$, que evita la reutilización excesiva de un mismo cliente y garantiza diversidad en el modelo agregado.

El Algoritmo 1 resume el procedimiento completo. Las tres fases recolección de activaciones, tratamiento de capas de atención y FedAvg para el resto se ejecutan en cada ronda de entrenamiento federado.

4 Diseño Experimental

La evaluación metodológica se realiza sobre 5 conjuntos de datos reales de series temporales multivariantes que varían en frecuencia de muestreo, dimensionalidad y dominio de aplicación. Este diseño permite analizar el comportamiento de la propuesta en diversos escenarios temporales y evaluar su rendimiento en entornos federados.

Algorithm 1 AFAS ($\{\theta^{(c)}\}_{c=1}^{C}$, $\theta^{(s)}$, $\mathcal{D}_{\text{ref}}$, α, κ, t)

Entrada: pesos de clientes $\{\theta^{(c)}\}$; pesos del servidor $\theta^{(s)}$; datos de referencia $\mathcal{D}_{\text{ref}}$; parámetros α, κ, ronda t. **Salida:** θ^{*}.

Identificar $\mathcal{L}_{\text{attn}}$ y $\mathcal{L}_{\text{other}}$ a partir de la arquitectura del servidor.
for $c = 1, \ldots, C$ **do** ▷ Activaciones de clientes
    Pasar $\mathcal{D}_{\text{ref}}$ por $\theta^{(c)}$; extraer y normalizar $\mathbf{q}_{\ell,h}^{(c)}, \mathbf{k}_{\ell,h}^{(c)}, \mathbf{v}_{\ell,h}^{(c)}$ $\forall \ell \in \mathcal{L}_{\text{attn}}$, $h \in [N]$.
end for
if $t \geq 2$ **then** ▷ Activaciones del servidor
    Pasar $\mathcal{D}_{\text{ref}}$ por $\theta^{(s)}$; extraer y normalizar $\hat{\mathbf{q}}_{\ell,i}^{(s)}, \hat{\mathbf{k}}_{\ell,i}^{(s)}, \hat{\mathbf{v}}_{\ell,i}^{(s)}$ $\forall \ell \in \mathcal{L}_{\text{attn}}$, $i \in [N]$.
end if
for $\ell \in \mathcal{L}_{\text{attn}}$ **do** ▷ Agregación de atención
    if $t = 1$ **then**
        for $h \in [N]$ **do**
            Calcular $\gamma_h^{(c)}$ según ec. (2) para todo c.
            $c_h^{*} \leftarrow \arg\max_c \gamma_h^{(c)}$; copiar W_h^Q, W_h^K, W_h^V de cliente c_h^{*} a θ_ℓ^{*}.
        end for
    else
        Calcular $S[i,j]$ (3) y $\gamma[j]$ (4); normalizar min-max; construir score (5).
        $\texttt{asgn} \leftarrow \textsc{GreedyAssign}(\text{score}, \kappa)$.
        for $i \in [N]$ **do**
            Copiar $W_{h^*}^Q, W_{h^*}^K, W_{h^*}^V$ del cliente $c^{*} = c(\texttt{asgn}[i])$, cabeza $h^{*} = h(\texttt{asgn}[i])$, a la posición i de θ_ℓ^{*}.
        end for
    end if
end for
for $\ell \in \mathcal{L}_{\text{other}}$ **do** ▷ FedAvg para el resto
    $\theta_\ell^{*} \leftarrow \frac{1}{C} \sum_{c=1}^{C} \theta_\ell^{(c)}$
end for
return θ^{*}

Para esta experimentación se han seleccionado cinco conjuntos de datos ampliamente usados en la literatura como son ETTh1, ETT2, ECL, WTH y NYTaxi, cuyos detalles pueden ser consultados en el repositorio de GitHub. Los datos se normalizaron y estructuraron mediante ventanas móviles. Se evaluaron longitudes de secuencia de entrada (`seq_len`) y horizontes de predicción (`pred_len`) simétricos con valores de 24, 96 y 120 pasos temporales.

La simulación del entorno federado se implementó utilizando la librería Flexible [4]. Los datos se particionaron de forma estricta y sin solapamiento bajo distribuciones no-IID para reflejar la heterogeneidad del mundo real. Para cada conjunto de datos se evaluaron escenarios descentralizados configurando entornos de 5, 10, 20 y 30 clientes independientes. Para cuantificar la precisión del modelo en el subconjunto de prueba, se adoptan las métricas estándar de error absoluto medio (MAE) y error cuadrático medio (MSE).

El método propuesto se implementó en Python utilizando PyTorch, en el mecanismo de agregación AFAS se estableció $\alpha = 0.75$, los parámetros de entrenamiento y detalles de la infraestructura se detallan en el repositorio de la organización ARI-DaSCI[1].

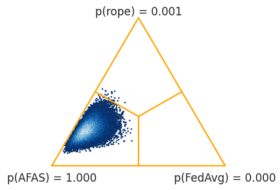
5 Resultados Experimentales

La Tabla 1 recoge los resultados comparativos de AFAS frente a FedAvg sobre los cinco conjuntos de datos evaluados, considerando cuatro configuraciones de número de clientes (5, 10, 20 y 30) y tres horizontes de predicción (24, 96 y 120). Los valores en negrita indican el mejor resultado en cada configuración. En términos globales, AFAS obtiene el mejor MSE en 53 de los 60 escenarios evaluados, lo que pone de manifiesto la superioridad consistente del método propuesto frente al promediado clásico.

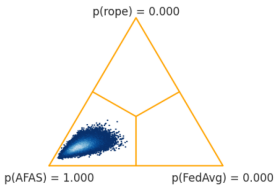
El rendimiento de AFAS es especialmente destacado en los conjuntos ECL, WTH y NYTaxi, donde la selección basada en similitud y consenso permite construir un modelo global que captura de forma más fiel las dependencias temporales distribuidas entre los clientes. En el caso de ETTh1 y ETTh2, AFAS también supera a FedAvg en la mayoría de las configuraciones, aunque se observan escenarios puntuales principalmente con 10 y 20 clientes en los que FedAvg resulta superior. La Figura 3 ilustra mejor esta dinámica al mostrar el MSE promediado sobre las longitudes de secuencia 24, 96 y 120 y donde se nota claramente la reducción consistente del error por parte de AFAS. En cuanto al efecto del número de clientes, AFAS muestra su mayor ventaja en los extremos del espectro: con pocos clientes (5), la selección de cabezas representativas permite aprovechar al máximo la información disponible sin diluirla; con muchos clientes (30), la centralidad par a par actúa como filtro eficaz frente a la mayor heterogeneidad del grupo, penalizando las cabezas atípicas y favoreciendo las más consensuadas. En los rangos intermedios (10 y 20 clientes) la ventaja se mantiene, aunque con mayor variabilidad entre datasets y horizontes de predicción.

Para evaluar la significancia estadística de las diferencias de rendimiento entre las variantes analizadas, se incorporan pruebas bayesianas de comparación de modelos. En concreto, se realizan dos pruebas, una centrada en el MSE y otra en el MAE, utilizando un ROPE de 0,01. Los resultados, que respaldan los datos presentados en la tabla 1 y se ilustran en la figura 2.

[1] https://github.com/ari-dasci/S-Federated-TimeSeries



(a) Prueba Bayesiana (MSE)



(b) Prueba Bayesiana (MAE)

Fig. 2: Prueba Bayesiana: AFAS vs FedAvg.

Table 1: Resultados comparativos de AFAS frente a FedAvg en cinco conjuntos de datos reales,con distintas configuraciones de número de clientes y horizonte de predicción. Los valores en **negrita** indican el mejor resultado para cada configuración.

Dataset	Clientes	Seq Len	MSE_{AFAS}	MAE_{AFAS}	MSE_{FedAvg}	MAE_{FedAvg}
ECL	5	24	**0.0153**	**0.0699**	0.0158	0.0736
		96	**0.0229**	**0.0993**	0.0269	0.1094
		120	**0.0222**	**0.1005**	0.0268	0.1114
	10	24	**0.0190**	**0.0802**	0.0198	0.0809
		96	**0.0360**	**0.1290**	0.0390	0.1309
		120	**0.0353**	**0.1273**	0.0943	0.2327
	20	24	**0.0270**	**0.1083**	0.0839	0.2116
		96	0.1398	0.2832	**0.0501**	**0.1513**
		120	**0.1467**	**0.2913**	0.1479	0.2948
	30	24	**0.0271**	**0.1012**	0.0308	0.1130
		96	**0.0459**	**0.1478**	0.0495	0.1539
		120	**0.1716**	**0.3235**	0.2109	0.3538
ETTh1	5	24	**0.0196**	**0.1027**	0.1591	0.2669
		96	**0.0400**	**0.1484**	0.2802	0.3761
		120	**0.0555**	**0.1770**	0.2443	0.3729
	10	24	**0.0095**	**0.0674**	0.0242	0.1119
		96	0.0408	0.1540	**0.0280**	**0.1271**
		120	**0.0307**	**0.1343**	0.0485	0.1685
	20	24	**0.0160**	**0.0908**	0.0305	0.1317
		96	0.0743	0.2080	**0.0411**	**0.1500**
		120	**0.0646**	**0.1936**	0.0721	0.2002
	30	24	**0.0393**	**0.1483**	0.0752	0.2095
		96	**0.0570**	**0.1751**	0.1481	0.2968
		120	**0.0954**	**0.2293**	0.2440	0.3705
WTH	5	24	**0.0233**	**0.0958**	0.0367	0.1320
		96	**0.0215**	**0.0993**	0.0401	0.1414
		120	**0.0201**	**0.0977**	0.0291	0.1104
	10	24	**0.0116**	**0.0636**	0.0164	0.0795
		96	**0.0483**	**0.1653**	0.0541	0.1439
		120	**0.0567**	**0.1547**	0.0580	0.1597
	20	24	0.0244	0.0958	**0.0157**	**0.0708**
		96	**0.0364**	**0.1300**	0.0453	0.1399
		120	**0.0462**	**0.1430**	0.0487	0.1645
	30	24	**0.0188**	**0.0798**	0.0420	0.1342
		96	**0.0571**	**0.1538**	0.0846	0.1983
		120	**0.0375**	**0.1328**	0.0715	0.1966
ETTh2	5	24	**0.0309**	**0.1165**	0.0555	0.1764
		96	**0.1071**	**0.2459**	0.3227	0.3824
		120	**0.1564**	**0.3156**	0.2570	0.3319
	10	24	0.0601	0.1890	**0.0550**	**0.1773**
		96	0.1451	0.2908	**0.0842**	**0.2082**
		120	0.1653	0.3178	**0.1050**	**0.2493**
	20	24	**0.0234**	**0.1133**	0.0521	0.1665
		96	**0.0906**	**0.2237**	0.1161	0.2441
		120	**0.1118**	**0.2510**	0.1145	0.2541
	30	24	**0.0459**	**0.1549**	0.1207	0.2715
		96	**0.0892**	**0.2167**	0.0899	0.2170
		120	**0.1509**	**0.2927**	0.1788	0.3371
NYTaxi	5	24	**0.0216**	**0.0927**	0.0495	0.1371
		96	**0.0229**	**0.0811**	0.1408	0.3000
		120	**0.0202**	**0.0972**	0.1233	0.2815
	10	24	**0.0492**	**0.0268**	0.0533	0.1464
		96	**0.0412**	**0.1335**	0.0624	0.1982
		120	**0.0160**	**0.0495**	0.0929	0.2445
	20	24	**0.0505**	**0.0698**	0.0552	0.1463
		96	**0.0898**	**0.2233**	0.0939	0.2792
		120	**0.0182**	**0.0731**	0.0845	0.2219
	30	24	**0.0515**	**0.1392**	0.0800	0.1606
		96	**0.0309**	**0.1182**	0.0635	0.1665
		120	**0.0218**	**0.0979**	0.0990	0.2458
#			**53**	**53**	7	7

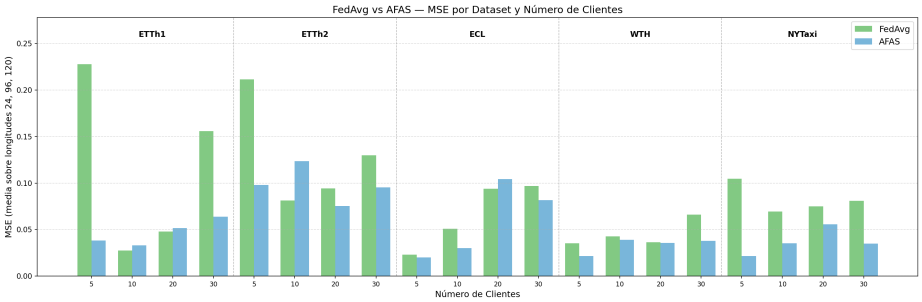


Fig. 3: Comparación FedAvg y AFAS atendiendo al MSE medio.

6 Conclusiones y trabajos futuros

Se propuso AFAS (Agregación Federada de la Atención mediante Similitud), un nuevo método de agregación federada diseñado para modelos Transformer aplicados a la predicción de series temporales en entornos distribuidos. AFAS selecciona para cada cabeza de atención del modelo global la cabeza cliente más representativa, combinando la similitud con el modelo global y el consenso entre clientes como criterios de selección.

Como resultado de la evaluación del agregador propuesto se demuestra su adaptabilidad a escenarios heterogéneos y su capacidad para generalizar entre dominios de aplicación diversos. Los resultados muestran que AFAS supera consistentemente a FedAvg, obteniendo el mejor MSE y MAE en 53 de las configuraciones evaluadas, lo que evidencia que la selección guiada de cabezas de atención preserva información relevante que el promediado destruye. Estos resultados confirman que una estrategia de agregación específica para las capas de atención constituye una dirección prometedora para mejorar el rendimiento de modelos Transformer en entornos federados con datos heterogéneos.

Como trabajo futuro, se prevé extender la evaluación de AFAS a un conjunto más amplio de dominios y frecuencias de muestreo, así como la prueba con otras propuestas del estado del arte. Además de realizar un estudio de ablación y análisis de sensibilidad sobre el parámetro α , con el fin de caracterizar su influencia en el rendimiento del agregador bajo distintas condiciones de heterogeneidad entre clientes y determinar estrategias para su selección automática.

Acknowledgments. Esta investigación es parte del Proyecto TSI-100927-2023-1, financiado por Plan de Recuperación, Transformación y Resiliencia Europeo, de la Unión Europea-Next Generation EU a través del Ministerio para la Transformación Digital y de la Función Pública y está parcialmente financiada por el Ministerio de Ciencia y Tecnología de España en el marco del proyecto PID2023-150070NB-I00, financiado por MICIU/AEI/10.13039/501100011033 y por FEDER, UE.

Disclosure of Interests. Los autores declaran que no tienen conflicto de intereses con respecto a la publicación de este artículo.

References

1. Aguilera-Martos, I., Herrera-Poyatos, A., Luengo, J., Herrera, F.: Local attention mechanism: Boosting the transformer architecture for long-sequence time series forecasting. arXiv preprint arXiv:2410.03805 (2024)
2. Ahmed, S., Nielsen, I.E., Tripathi, A., Siddiqui, S., Ramachandran, R.P., Rasool, G.: Transformers in time-series analysis: A tutorial. Circuits, Systems, and Signal Processing **42**(12), 7433–7466 (2023)
3. Ashish, V.: Attention is all you need. Advances in neural information processing systems **30**, I (2017)
4. Herrera, F., Jiménez-López, D., Argente-Garrido, A., Rodríguez-Barroso, N., Zuheros, C., Aguilera-Martos, I., Bello, B., García-Márquez, M., Luzón, M.V.: Flex: Flexible federated learning framework. Information Fusion **117**, 102792 (2025)
5. Kairouz, P., McMahan, H.B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A.N., Bonawitz, K., Charles, Z., Cormode, G., Cummings, R., et al.: Advances and open problems in federated learning. Foundations and trends® in machine learning **14**(1–2), 1–210 (2021)
6. Kamei, S., Taghipour, S.: A comparison study of centralized and decentralized federated learning approaches utilizing the transformer architecture for estimating remaining useful life. Reliability Engineering & System Safety **233**, 109130 (2023)
7. Karimireddy, S.P., Kale, S., Mohri, M., Reddi, S., Stich, S., Suresh, A.T.: Scaffold: Stochastic controlled averaging for federated learning. In: International conference on machine learning. pp. 5132–5143. PMLR (2020)
8. Kim, M.K., Cremers, B., Fu, N., Liu, J.: Predictive and correlational analysis of heating energy consumption in four residential apartments with sensitivity analysis using long short-term memory and generalized regression neural network models. Sustainable Energy Technologies and Assessments **71**, 103976 (2024)
9. Li, H., Cai, Z., Wang, J., Tang, J., Ding, W., Lin, C.T., Shi, Y.: Fedtp: Federated learning by transformer personalization. IEEE transactions on neural networks and learning systems (2023)
10. Li, T., Sahu, A.K., Zaheer, M., Sanjabi, M., Talwalkar, A., Smith, V.: Federated optimization in heterogeneous networks. Proceedings of Machine learning and systems **2**, 429–450 (2020)
11. McMahan, B., Moore, E., Ramage, D., Hampson, S., y Arcas, B.A.: Communication-efficient learning of deep networks from decentralized data. In: Artificial intelligence and statistics. pp. 1273–1282. PMLR (2017)
12. Pérez-Cruz, F.: Kullback-leibler divergence estimation of continuous distributions. In: 2008 IEEE international symposium on information theory. pp. 1666–1670. IEEE (2008)
13. Qi, P., Chiaro, D., Guzzo, A., Ianni, M., Fortino, G., Piccialli, F.: Model aggregation techniques in federated learning: A comprehensive survey. Future Generation Computer Systems **150**, 272–293 (2024)
14. Raza, A., Tran, K.P., Koehl, L., Li, S.: Anofed: Adaptive anomaly detection for digital health using transformer-based federated learning and support vector data description. Engineering Applications of Artificial Intelligence **121**, 106051 (2023)

15. Raza, A., Tran, K.P., Koehl, L., Li, S., Zeng, X., Benzaidi, K.: Lightweight transformer in federated setting for human activity recognition. arXiv preprint arXiv:2110.00244 (2021)
16. Rodríguez-Barroso, N., Jiménez-López, D., Luzón, M.V., Herrera, F., Martínez-Cámara, E.: Survey on federated learning threats: Concepts, taxonomy on attacks and defences, experimental study and challenges. Information Fusion **90**, 148–173 (2023)
17. Saghir, A., Beniwal, H., Tran, K., Raza, A., Koehl, L., Zeng, X., Tran, K.: Explainable transformer-based anomaly detection for internet of things security. In: International Conference on Safety and Security in IoT. pp. 83–109. Springer (2023)
18. Sun, G., Mendieta, M., Luo, J., Wu, S., Chen, C.: Fedperfix: Towards partial model personalization of vision transformers in federated learning. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 4988–4998 (2023)
19. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)
20. Wang, H., Yurochkin, M., Sun, Y., Papailiopoulos, D., Khazaeni, Y.: Federated learning with matched averaging. arXiv preprint arXiv:2002.06440 (2020)
21. Yan, P., Abdulkadir, A., Luley, P.P., Rosenthal, M., Schatte, G.A., Grewe, B.F., Stadelmann, T.: A comprehensive survey of deep transfer learning for anomaly detection in industrial time series: Methods, applications, and directions. IEEE Access **12**, 3768–3789 (2024)
22. Yu, H., Yang, Q., Kang, Y., Cheng, Y., Liu, Y., Chen, T.: Federated learning. Morgan & Claypool Publishers (2019)
23. Zhang, H., Liu, Y., Zhang, C., Li, N.: Machine learning methods for weather forecasting: A survey. Atmosphere **16**(1), 82 (2025)
24. Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., Zhang, W.: Informer: Beyond efficient transformer for long sequence time-series forecasting. In: Proceedings of the AAAI conference on artificial intelligence. vol. 35, pp. 11106–11115 (2021)