# Mitigación de Ataques Bizantinos usando Modelos Históricos en Aprendizaje Federado Descentralizado

Enrique Tomás Martínez Beltrán[1], Pedro Miguel Sánchez Sánchez[1], Gérôme Bovet[2], Gregorio Martínez Pérez[1] and Alberto Huertas Celdrán[3]
[1]Departamento de Ingeniería de la Información y las Comunicaciones, Universidad de Murcia, 30100, España
Email: {enriquetomas,pedromiguel.sanchez,gregorio}@um.es
[2]Cyber-Defence Campus, Armasuisse Science and Technology, 3602 Thun, Switzerland
Email: gerome.bovet@armasuisse.ch
[3]Communication Systems Group, Department of Informatics (IFI), University of Zurich, 8050 Zürich, Switzerland
Email: huertas@ifi.uzh.ch

***Resumen*—El Aprendizaje Federado Descentralizado emerge como una solución prometedora para entrenar modelos de inteligencia artificial de manera colaborativa, sin compartir directamente los datos y sin la necesidad de un servidor central. Sin embargo, esta arquitectura enfrenta desafíos significativos en términos de seguridad donde nodos maliciosos podrían comprometer la integridad y eficacia de los modelos. Ante este escenario, se propone `DFLShield`, un mecanismo de mitigación que se apoya en el análisis de modelos históricos para la actualización segura. Esta solución contempla la recolección y evaluación crítica de modelos de nodos adyacentes, junto con el uso de un modelo promedio enriquecido con datos históricos del nodo local. Mediante la aplicación de técnicas de análisis de similitud coseno y métodos de clustering, se facilita la identificación precisa y la eliminación de modelos potencialmente perjudiciales, integrando en la agregación aquellos evaluados como fiables. Este mecanismo se valida en un caso de uso con diez nodos en una topología totalmente conectada y utilizando el conjunto de datos CIFAR10 junto con una CNN personalizada. Los resultados preliminares demuestran que no solo atenúa eficazmente los impactos de los ataques bizantinos, sino que también promueve una mejora sustancial en la robustez y la fiabilidad de los modelos.**

***Index Terms*—Aprendizaje Federado, Modelos Colaborativos, Descentralización, Mitigación, Seguridad, Privacidad**

**Tipo de contribución:** *Investigación en desarrollo*

## I. Introducción

En el panorama actual de la tecnología, el Aprendizaje Federado Descentralizado (DFL por sus siglas en inglés) emerge como un paradigma revolucionario que promete transformar el entrenamiento de modelos de Inteligencia Artificial (IA), haciendo énfasis en la colaboración sin comprometer la privacidad ni la seguridad de los datos [1]. El proceso se puede resumir de la siguiente manera: (1) cada participante de la federación entrena un modelo local utilizando sus propios datos, (2) se realiza un intercambio de parámetros del modelo entre los participantes, (3) cada nodo procede a la agregación local de los parámetros recibidos, y (4) se actualiza el modelo local con los parámetros agregados, refinando así su precisión y capacidad predictiva. Además, esta arquitectura descentralizada aborda varios problemas inherentes a los sistemas centralizados, como los cuellos de botella en el procesamiento de datos y la necesidad de confiar en una entidad central para la gestión y el almacenamiento de datos. Al mismo tiempo, se aumenta la escalabilidad mientras que se eliminan puntos únicos de fallo, mitigando el riesgo de ataques centrados en servidores los cuales comprometen toda la federación. Este enfoque permite que una amplia gama de dispositivos, desde teléfonos móviles hasta vehículos autónomos, drones y sensores industriales, intercambien directamente los parámetros de sus modelos locales con el objetivo de entrenar modelos federados [2]. DFL puede aplicarse en una multitud de campos, cada uno con sus propios desafíos y requisitos únicos. En el ámbito militar, la capacidad de procesar datos de inteligencia en el punto de recolección mejora la rapidez y la seguridad de las decisiones críticas sin dependencia con un servidor. En el sector móvil y de telecomunicaciones, mejora la personalización y eficiencia de los servicios sin centralizar el aprendizaje, mientras que en la industria, permite la optimización de procesos mediante el aprendizaje a partir de datos de múltiples fuentes sin revelar información sensible entre competidores [3].

La descentralización, a pesar de sus numerosas ventajas, abre la puerta a un conjunto único de problemas de seguridad y vulnerabilidades que no están presentes en los sistemas centralizados tradicionales. Al distribuir el entrenamiento y la agregación de modelos a través de una red de nodos autónomos, se aumenta la superficie de ataque. Además, la falta de una entidad central de confianza dificulta la implementación de políticas de seguridad cohesivas y la rápida identificación y mitigación de amenazas [4]. En este contexto surgen distintos ataques adversariales como el envenenamiento de datos, donde los atacantes insertan datos corruptos o engañosos durante el entrenamiento para manipular los resultados del modelo, y ataques directos al modelo, en los que se intenta extraer información confidencial o inducir errores en las predicciones [5]. Dentro de este espectro de vulnerabilidades, los ataques bizantinos emergen como una de las amenazas más complejas y dañinas. Estos ataques se caracterizan por la actuación de nodos que, ya sea intencionalmente por ser maliciosos o accidentalmente por fallos, proporcionan información falsa o contradictoria. Entre los posibles ataques destaca el Ataque de Gradiente Signado Rápido (FGSM, por sus siglas en inglés) por su eficacia y prevalencia [6]. FGSM manipula los datos de entrada mediante la adición de perturbaciones imperceptibles diseñadas para engañar al modelo y provocar una clasificación errónea, lo que representa una amenaza insidiosa para la integridad del modelo federado. En el contexto del DFL, esto significa que un nodo comprometido podría enviar actualiza-

ciones maliciosas del modelo que, si no se detectan, podrían sesgar o degradar la precisión de los modelos del resto de nodos. Además de FGSM, otros métodos como los ataques de envenenamiento de datos y los ataques de modelo se suman al arsenal de tácticas que los actores malintencionados pueden desplegar en este escenario.

A pesar de existir soluciones que intentan mitigar estos ataques, estas a menudo incurren en compromisos indeseables, como la reintroducción de ciertos grados de centralización o la imposición de requisitos de confianza previa entre los nodos, lo que contradice los principios de descentralización y anonimato en los que se basa el DFL. Además, muchas de estas soluciones están diseñadas para entornos específicos y pueden no ser aplicables universalmente a todas las configuraciones de DFL, lo que limita su utilidad en un panorama diverso y en constante evolución. Ante este escenario, el presente trabajo propone un enfoque novedoso para abordar la seguridad en el DFL, centrado en el uso de modelos históricos para validar la integridad de las contribuciones de los nodos. Este enfoque no solo busca detectar y mitigar los ataques bizantinos sino también fortalecer la resiliencia general del sistema frente a una amplia gama de amenazas de seguridad. Al hacerlo, se espera no solo superar las limitaciones de las soluciones existentes sino también avanzar hacia un paradigma de DFL verdaderamente seguro y robusto, capaz de soportar las demandas de las aplicaciones críticas de hoy y del futuro.

Las contribuciones clave de este trabajo se resumen en:

- Diseño e implementación de `DFLShield`, un mecanismo de mitigación basado en el análisis de modelos históricos y clustering, diseñado para fortalecer la resistencia de DFL contra ataques bizantinos durante las fases de actualización de modelos. Además de ser aplicable en una amplia gama de configuraciones de DFL sin comprometer los principios de descentralización.
- Diseño de un modelo de amenaza bizantino específicamente adaptado al contexto del DFL, utilizando el método de FGSM con dos niveles de intensidad, $\alpha$ de 0.003 y 0.01 para simular impactos variados durante la federación.
- Validación de la solución en un caso de uso con diez nodos en una topología totalmente conectada, utilizando el conjunto de datos CIFAR10 y una Red Neuronal Convolucional (CNN) de aproximadamente 2.5 millones de parámetros, garantizando así la relevancia y aplicabilidad de los resultados obtenidos.
- Evaluación de mecanismo de mitigación propuesto, sometiéndolo a condiciones donde el 10 %, 30 % y 50 % de los nodos en la federación actúan maliciosamente. La eficacia del mecanismo se demuestra no solo en la capacidad de detectar y aislar a los nodos maliciosos sino también en su adaptabilidad para integrarse con algoritmos de agregación como FedAvg [7] y Krum [8], manteniendo un $F_1 score$ de 68 % y 75 % en las condiciones más adversas. Esta mejora en seguridad implica un incremento del 20 % en el consumo de CPU y +57 MB en el uso de RAM, mientras que el consumo de red permanece estable durante la federación.

## II. Trabajo Relacionado

La investigación en FL tradicional ha sido profusamente marcada por la preocupación hacia la seguridad, en particular por la amenaza que representan los ataques bizantinos. Esto es algo que ha sido heredado por el enfoque DFL, incrementando la preocupación en entornos descentralizados. Estos ataques, en los que nodos maliciosos comprometen la integridad de los modelos de aprendizaje, han servido como punto de partida para numerosos estudios que buscan fortalecer la resiliencia de los sistemas federados. Tabla I resume los trabajos identificados en distintas perspectivas: arquitectura de FL, tipo de ataque, técnica empleada para su mitigación y el objetivo de la solución.

Un enfoque innovador en este ámbito lo presenta el trabajo de Kamhoua et al. [9], identificando el impacto posible de las manipulaciones intencionadas durante el proceso de aprendizaje. Los autores proponen un algoritmo resiliente y verificable basado en un esquema de reputación diseñado para mitigar la influencia de partes no confiables, destacando la crucial importancia de validar la integridad de las actualizaciones del modelo en un entorno colaborativo. Esta solución no solo aborda el desafío de mantener la confiabilidad en sistemas tanto centralizados como descentralizados, sino que también subraya las complejidades asociadas con la gestión de la confianza entre una amplia gama de participantes. La complejidad de los ataques bizantinos y la diversidad de sus manifestaciones motivaron estudios más específicos. Li et al. [10] profundizaron en la evaluación de esquemas de agregación robustos ante ataques bizantinos utilizando técnicas de clustering encargadas de filtrar automáticamente las actualizaciones. Los autores demostraron un buen rendimiento bajo determinadas condiciones, pero flaquea ante contextos más adversos, especialmente en presencia de datos no independientes e idénticamente distribuidos (Non-IID). La investigación revela una brecha significativa en la efectividad de los esquemas de agregación frente a estrategias de ataque en constante evolución, poniendo de manifiesto la necesidad de soluciones más dinámicas y adaptables.

La búsqueda de mecanismos más resilientes condujo a la exploración de nuevos paradigmas, como lo ilustran Xu et al. [11]. Los autores abordan el desafío de detectar clientes bizantinos en el aprendizaje federado vertical, un contexto marcado por la heterogeneidad de las características. Este estudio propone un marco basado en la detección para identificar clientes maliciosos mediante la codificación de características y la validación cruzada. En un esfuerzo por asegurar el DFL contra ataques de envenenamiento, Feng et al. [12] presentan una estrategia de defensa que aprovecha la accesibilidad de los datos locales y define un protocolo de agregación en tres pasos para proteger contra actualizaciones maliciosas del modelo: filtrado de similitudes, validación *bootstrap* y normalización de los datos. En un esfuerzo por abordar las vulnerabilidades de DFL desde una perspectiva más holística, Miao et al. [13] fusionaron las capacidades de la tecnología *blockchain* con DFL para ofrecer una solución robusta a los ataques bizantinos. Al aprovechar la transparencia y la inmutabilidad de la *blockchain*, este enfoque busca establecer un marco de trabajo más seguro para el intercambio de actualizaciones de modelos. Sin embargo, la complejidad y la sobrecarga

Tabla I: Comparación de soluciones para la detección e identificación de ataques en FL

| Referencia | Arquitectura FL | Ataque | Técnica | Objetivo |
|---|---|---|---|---|
| [9] | Centralizado | Bizantino | Esquema de reputación y confianza | *DET / ID* |
| [10] | Centralizado | Bizantino | Clustering filtrando actualizaciones maliciosas al final de ronda | *DET* |
| [11] | Centralizado | Bizantino | Detección basada en características de los datos | *ID* |
| [12] | Decentralizado | Envenenamiento | Agregación basada en similitud de modelos | *DET / ID* |
| [13] | Decentralizado | Bizantino | Inmutabilidad de contribuciones usando *blockchain* | *DET* |
| [14] | Centralizado | Envenenamiento | Clustering de actualizaciones centralizadas por ronda de federación | *DET / ID* |
| [15] | Centralizado | Envenenamiento | Estimación de actualizaciones basada en el historial de servidor | *ID* |
| Este trabajo | Decentralizado | Bizantino | Clustering de similitud de modelos basados en historial de contribuciones previas | *DET / ID* |

*DET:* Detección, *ID:* Identificación.

adicional que implica la integración de la *blockchain* señalan las dificultades prácticas de implementar tales soluciones en entornos heterogéneos de DFL a gran escala.

Frente a estos desafíos, la literatura reciente ha comenzado a vislumbrar el potencial de los modelos históricos como una estrategia prometedora para reforzar la seguridad. Los modelos históricos se refieren a los parámetros que representan estados anteriores del modelo local al hacer la agregación con los modelos recibidos. Al conservar estos estados previos, se puede obtener una visión retrospectiva que permite tomar decisiones informadas basadas en las interacciones y comportamientos observados en fases anteriores de la federación. Esta capacidad de referenciar la historia del modelo ayuda a detectar anomalías o cambios inusuales en los datos o parámetros compartidos, facilitando así la identificación temprana de potenciales amenazas o ataques adversariales. Zhang et al. [14] identifican a los clientes maliciosos mediante el análisis y clustering de la consistencia de sus actualizaciones de modelo a lo largo del tiempo. El trabajo se basa en la premisa de que las actualizaciones de un cliente malintencionado presentarán inconsistencias significativas en múltiples iteraciones, en comparación con las de los participantes legítimos. El servidor de la federación predice la actualización de modelo de un cliente en cada iteración utilizando información histórica y señala como maliciosas aquellas actualizaciones que divergen de manera consistente de las predicciones. De igual manera, Cao et al. [15] ofrece una solución para la recuperación del modelo global tras la detección de ataques de envenenamiento. El servidor estima las actualizaciones de modelo de los clientes con un historial previo lo que permite la detección de anomalías y garantizar la resiliencia y recuperación del sistema frente a ataques. Este enfoque basado en modelos históricos no solo ofrece un método para discernir entre actualizaciones legítimas y malintencionadas sino que también presenta una oportunidad para desarrollar sistemas de DFL más resilientes y autónomos. Al incorporar la perspectiva temporal en la evaluación de la confiabilidad de los nodos, se abre la puerta a mecanismos de defensa que son inherentemente adaptativos y capaces de evolucionar en respuesta a las tácticas cambiantes de los adversarios.

## III. Diseño e Implementación de DFLShield

La incesante evolución de DFL y la persistente amenaza de ataques bizantinos exigen soluciones que no solo sean innovadoras sino también exhaustivas en su capacidad para proteger la integridad del aprendizaje colaborativo. En este contexto, la solución propuesta, DFLShield, representa un avance significativo en la mitigación de riesgos asociados a actores malintencionados en la federación. Figura 1 muestra la arquitectura general de la solución donde cada paso del mecanismo se indica mediante círculos numerados, facilitando así la comprensión de la secuencia lógica y el flujo operativo del enfoque de mitigación propuesto.

El mecanismo inicia con una recolección meticulosa de modelos de nodos vecinos. Este proceso no es un simple intercambio de información sino que representa el primer paso de una secuencia de verificación y validación. Cada nodo, al recibir los modelos de sus vecinos, no solo acumula datos, sino que también prepara el terreno para la construcción de un bastión contra la acción maliciosa del aprendizaje conjunto ①. Los modelos históricos de cada nodo funcionan como una memoria colectiva de la federación, un compendio de lecciones aprendidas que se consolidan en un modelo promedio ②. Este promedio permite una síntesis de la trayectoria de aprendizaje del nodo, que se emplea como un estándar para medir la validez de los nuevos modelos recibidos. Para este estudio preliminar se opta por considerar cinco modelos históricos en el cálculo para aliviar la demanda sobre los nodos de la federación, que frecuentemente son dispositivos con recursos limitados. Esta configuración puede ser modificada para adaptarse a las capacidades específicas y los requerimientos del escenario. En este punto, se realiza una comparación mediante similitud coseno entre el modelo promedio y los modelos recibidos, actuando como un espejo que refleja las inconsistencias y las anomalías de las contribuciones actuales ③. En concreto, el algoritmo de similitud $\frac{M_{avg,n}^{(r)} \cdot M_i^{(r)}}{\|M_{avg,n}^{(r)}\| \|M_i^{(r)}\|}$ de PyTorch calcula el coseno del ángulo entre el modelo promedio y cada uno de los modelos entrantes. Al hacerlo, se obtiene una métrica que cuantifica la cercanía entre los dos, donde un valor cercano a uno indica una gran similitud y, por ende, una alta probabilidad de que el modelo entrante sea coherente con el comportamiento histórico. Por el contrario, un valor significativamente menor reflejaría una discrepancia potencialmente alarmante que podría señalar una manipulación maliciosa.

La estrategia de clustering seleccionada para reforzar el mecanismo de seguridad es el algoritmo Density-Based Spatial Clustering of Applications with Noise (DBSCAN), conocido por su capacidad para identificar clusters basados en la densidad de los puntos en el espacio de características [16]. Este algoritmo juega un papel vital en discernir entre comportamientos normales y atípicos dentro de la red federada ④. La aplicación de DBSCAN al conjunto de similitud de coseno se lleva a cabo utilizando un valor de $\epsilon$ de 3, seleccionado

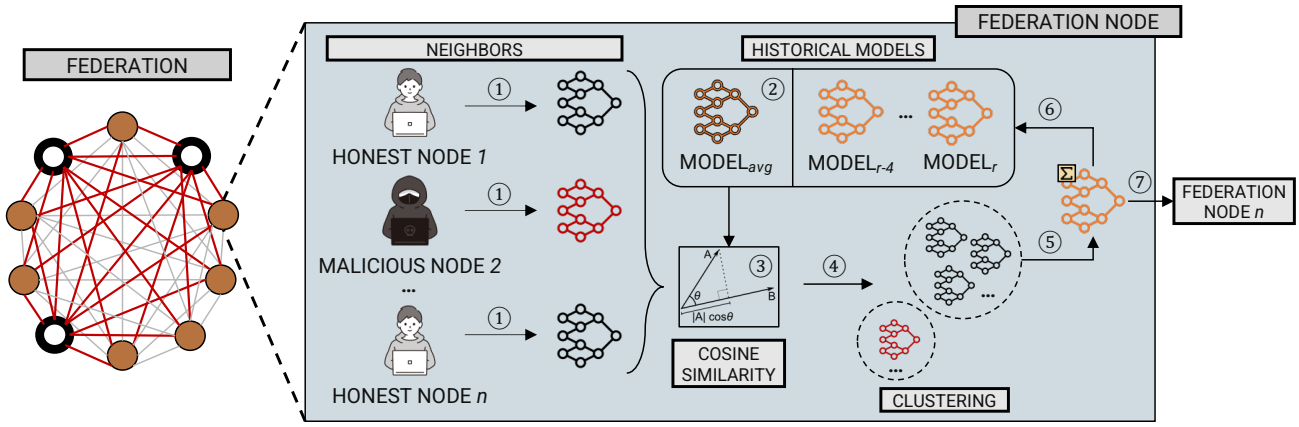



Figura 1: Arquitectura general de DFLShield, destacando la secuencia de operaciones a través de círculos numerados.

cuidadosamente tras un proceso de búsqueda de hiperparámetros. Este valor crítico determina la proximidad necesaria entre puntos para ser agrupados en el mismo clúster. De esta manera, el algoritmo permite identificar de manera efectiva los grupos de modelos que mantienen coherencia con el historial de aprendizaje etiquetándolos como honestos, mientras que aquellos que presentan discrepancias significativas son catalogados como maliciosos. Esta demarcación facilita una segregación precisa y promueve la consolidación de un marco de confianza dentro del aprendizaje federado, asegurando que solo las contribuciones verificadas influyan en la agregación de los modelos ⑤. Este algoritmo de agregación destaca por su versatilidad, permitiendo una adaptación sencilla para integrar diversas técnicas según los requerimientos únicos de cada red federada. Seguidamente, el modelo enriquecido con la parámetros de los modelos honestos de la federación es añadido como nuevo modelo histórico para futuras rondas de federación ⑥. Este modelo no solo contiene el conocimiento actualizado sino que también lleva consigo el legado de la resiliencia aprendida. Finalmente, el modelo agregado es incorporado localmente en el nodo y transmitido a los nodos adyacentes en la ronda siguiente ⑦.

El Algoritmo 1 encapsula este proceso en una serie de pasos lógicos y acciones ejecutables. Primero, inicia con la configuración de las condiciones iniciales y el establecimiento de las bases de recolección y promediación. Seguidamente, promedia los modelos históricos locales para establecer una referencia de integridad y aplica DBSCAN para identificar modelos maliciosos. La exclusión de modelos potencialmente perjudiciales del proceso de agregación, realizado con FedAvg [7] por su simplicidad y eficiencia, y adaptable para incorporar otros algoritmos como Krum [8] en escenarios que demanden mayor resistencia a ataques, culmina en una ronda de aprendizaje federado. DFLShield está implementado en Python, lo que facilita su adaptabilidad y extensibilidad, asegurando compatibilidad con librerías de entrenamiento de modelos ampliamente utilizadas como PyTorch o TensorFlow.

**Algoritmo 1** Algoritmo de DFLShield.

**Input:** Local node $n$ with historical models $M_n^{(r)}$ ... $M_n^{(r-4)}$
**Output:** Updated local model $M_n^{(r)}$
**procedure** DFLShield Update
  Initialize set $R_n$ to store received neighbor models
  Collect models $\{M_i^{(r)} | i \in \text{neighbors}(n)\}$ into $R_n$ ①
  Initialize similarity list $S_n$
  $M_{avg,n}^{(r)} \leftarrow \frac{\sum_{i=0}^{4} M_n^{(r-i)}}{5}$ ② ▷ Average of historical models
  **for** each model $M_i^{(r)}$ in $R_n$ **do**
    $S_{ni} \leftarrow \frac{M_{avg,n}^{(r)} \cdot M_i^{(r)}}{\|M_{avg,n}^{(r)}\| \|M_i^{(r)}\|}$ ③ ▷ Cosine similarity
    Add $S_{ni}$ to $S_n$
  **end for**
  $C_{honest,n}, C_{malicious,n} \leftarrow$ DBSCAN$(S_n, \epsilon = 3)$ ④
  Exclude $C_{malicious,n}$ from aggregation ▷ Mitigation
  $M_n^{(r)} \leftarrow \frac{1}{|C_{honest,n}|} \sum_{M_h \in C_{honest,n}} M_h$ ⑤
  Update historical models with $M_n^{(r)}$ ⑥
  **return** $M_n^{(r)}$ ⑦ ▷ Model parameters to be transmitted
**end procedure**

## IV. Modelo de Amenazas

Entre los diversos tipos de ataques posibles, los ataques bizantinos se erigen como una de las amenazas más insidiosas, capaces de comprometer la integridad y eficacia de los modelos de inteligencia artificial entrenados colaborativamente en un entorno federado. En este contexto, los ataques FGSM se identifican como técnicas particularmente efectivas y representativas de las capacidades ofensivas de un adversario. Los ataques FGSM manipulan sutilmente las actualizaciones de gradientes, añadiendo una perturbación deliberadamente calculada para inducir errores durante el entrenamiento del modelo local y que posteriormente transmitirá a los nodos adyacentes.

La Ec. (1) presenta una versión adaptada del método FGSM, especificamente diseñada para entornos DFL. En este sentido, $g$ representa el gradiente original de la función de pérdida respecto a los parámetros del modelo y $\alpha$ denota la magnitud de la perturbación complementada con un término adicional $(1 + \rho \cdot C_n)$ que ajusta la intensidad de esta perturbación en función del número de conexiones $C_n$ de un nodo. Este ajuste busca reflejar la influencia potencialmente mayor de los nodos con más conexiones dentro de la red DFL, siendo modulado por el coeficiente $\rho$. La función de pérdida $J(\theta, g, y)$ se utiliza para orientar la dirección de la

perturbación, asegurando que esté alineada con los objetivos del ataque y maximice su efecto disruptivo en el aprendizaje del modelo.

$$g_{\text{adv}} = g + \alpha \cdot (1 + \rho \cdot C_n) \cdot \text{sign}(\nabla_g J(\theta, g, y)) \quad (1)$$

La selección del FGSM como componente central del modelo de amenaza no es arbitraria, sino que se justifica por su relevancia y aplicabilidad dentro del contexto de FL y por sus posibles implicaciones en escenarios descentralizados. Esta técnica, al ser sutil y adaptable, encarna el tipo de estrategias que un adversario sofisticado podría emplear para evadir la detección mientras compromete la integridad del modelo de aprendizaje colaborativo. Además, la capacidad del FGSM para generar perturbaciones efectivas con un conocimiento limitado del modelo subyacente lo convierte en una herramienta valiosa para evaluar la robustez del sistema frente a ataques realistas y bien fundamentados.

## V. Caso de Uso

El caso de uso seleccionado para demostrar la viabilidad del mecanismo de defensa se desarrolla en un escenario DFL con diez nodos activos, donde se emplean diez rondas de federación que consisten en entrenamiento local, intercambio de los parámetros del modelo entre los nodos, y validación. Cada nodo en esta red participa plenamente, compartiendo y recibiendo información que contribuye a la formación de un modelo de inteligencia artificial colectivo, como se ilustra en la Figura 2. Este ambiente simula un sistema de aprendizaje colaborativo en el que todos los participantes están conectados, permitiendo un flujo constante de datos e información. Además, se opta por limitar los modelos históricos a cinco con el objetivo de hacer más eficiente la federación.

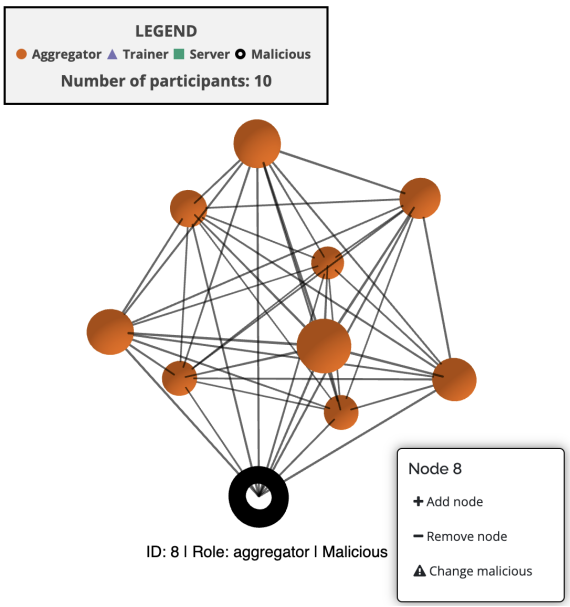



Figura 2: Despliege de una topología de nodos totalmente conectados usando una plataforma DFL interactiva.

Se han seleccionado dos algoritmos de agregación: FedAvg, por su aplicación general en la generación de modelos a partir de la media de las contribuciones de los nodos, y Krum, debido a su reconocida robustez ante ataques bizantinos. Krum ha sido elegido específicamente para evaluar su desempeño intrínseco frente a tales amenazas y para observar cómo se complementa con la solución propuesta, que tiene como objetivo reforzar la seguridad de la red al detectar y mitigar los efectos de los nodos malintencionados.

El escenario de evaluación se basa en el conjunto de datos CIFAR10, que pone a prueba la capacidad de una CNN adaptada para identificar y clasificar una variedad de imágenes. La arquitectura de la CNN, que se muestra en la Figura 3, ha sido cuidadosamente optimizada para este conjunto de datos en particular, garantizando que el modelo esté bien equipado para la tarea en cuestión. La eficacia de los modelos se mide mediante el $F_1 score$, una métrica confiable que equilibra la precisión y la exhaustividad en la clasificación, proporcionando así una evaluación integral de la capacidad del modelo para identificar correctamente las categorías de imágenes en el conjunto de datos.

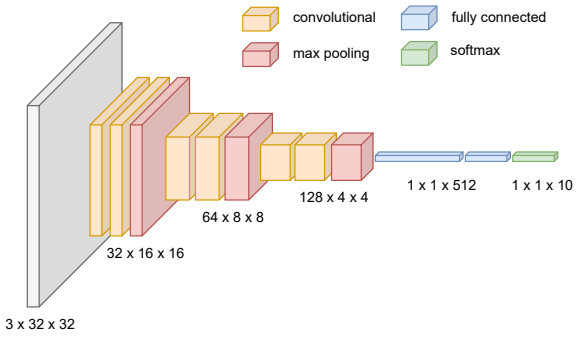



Figura 3: Arquitectura de la CNN implementada en cada nodo de la federación.

Finalmente, para probar la resistencia del sistema, se implementa FGSM (ver Sección IV), un ataque deliberado que ajusta la magnitud de las perturbaciones a través de los valores $\alpha = \{0.003, 0.01\}$ y un coeficiente $\rho$ fijo de 0.01, asegurando que el efecto de la conectividad sobre la perturbación sea significativo pero no desproporcionado. Este ataque introduce un porcentaje variable de nodos maliciosos en la red, que oscila entre el 0 % y el 50 %, desafiando tanto a los algoritmos de agregación como a la capacidad general de la red para mantener la integridad del modelo.

El caso de uso se despliega a través de una plataforma de DFL actualmente en desarrollo, que se caracteriza por su capacidad de creación dinámica de topologías de red y el establecimiento de conexiones rápidas y eficientes basadas en el protocolo de transporte UDP. Además, esta plataforma permite la asignación precisa de ataques para simular entornos adversos, así como herramientas de monitorización, lo que proporciona una visibilidad detallada del uso de recursos y la eficacia de las estrategias de mitigación en tiempo real. La plataforma está implementada en Python, facilitando que DFLShield también esté implementado en este lenguaje para aprovechar su adaptabilidad y extensibilidad, permitiendo futuras actualizaciones de la solución.

## VI. Resultados Preliminares

Los resultados preliminares obtenidos en este estudio revelan patrones interesantes y conclusiones significativas sobre la eficacia de las estrategias de agregación y las técnicas de mitigación propuestas frente a ataques maliciosos. Figura 4 representa el rendimiento medio del $F_1score$ de los diez nodos en la federación utilizando un modelo CNN con el conjunto de datos CIFAR10, a lo largo de diez rondas de federación sin el despliegue de `DFLShield` ni la presencia de ataques. Los resultados obtenidos mediante el uso de FedAvg y Krum como algoritmos de agregación establecen una línea base sólida, contra la cual se pueden comparar los efectos de los ataques subsiguientes y la eficacia de la estrategia de mitigación propuesta. Ambos algoritmos exhiben un crecimiento sostenido en el rendimiento, alcanzando un pico aproximado de 0.8 en $F_1score$ a los 35 minutos. Después de este punto, el rendimiento se estabiliza y fluctúa alrededor de este máximo hasta completar las diez rondas de federación a los 60 minutos. Resulta notable que Krum muestre una variabilidad (o desviación estándar) un 15 % menor que FedAvg una vez alcanzado este pico, indicando una estabilidad superior en los resultados obtenidos con este método de agregación. Adicionalmente, Tabla II muestra el rendimiento medio de los modelos y el uso de recursos en un escenario sin ataques, comparando el despliegue de `DFLShield` con situaciones donde no se utiliza ningún mecanismo de protección durante la federación. Los resultados indican que la introducción de la solución propuesta mantiene un buen uso de recursos sin comprometer el rendimiento de los modelos federados.

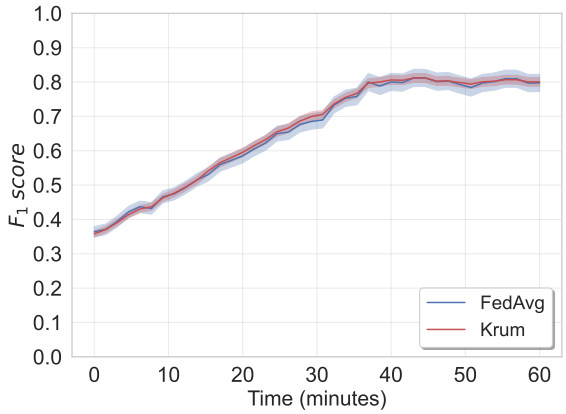



Figura 4: Rendimiento medio del modelo CNN usando CIFAR10 en una federación de diez nodos honestos.

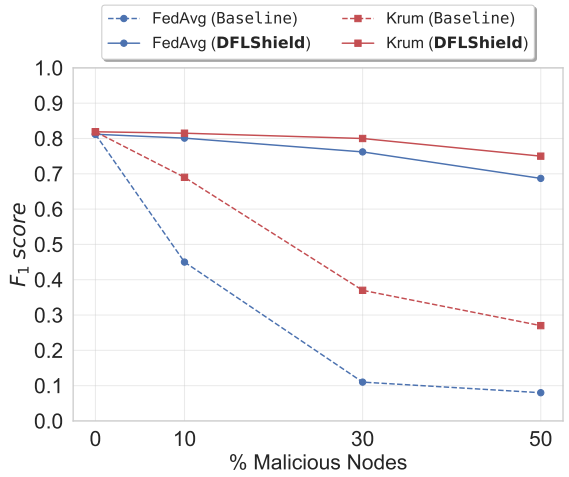
La Figura 5 profundiza en el análisis al introducir el ataque FGSM con diferentes niveles de impacto ($\alpha = \{0.003, 0.01\}$) y variando el porcentaje de nodos maliciosos en la red (0, 10, 30, 50 %). Ambas gráficas contrastan el rendimiento de los algoritmos de agregación FedAvg y Krum tanto en su forma original como en su versión mejorada con la implementación de `DFLShield`.

En la Figura 5a, la cual muestra el impacto del FGSM con un $\alpha$ de 0.003, se observa que mientras el rendimiento de FedAvg y Krum sin `DFLShield` disminuye drásticamente con el aumento del porcentaje de nodos maliciosos, el despliegue de la solución permite mantener un $F_1score$ significativamente más alto. En concreto, FedAvg logra mantener un $F_1score$ por encima de 0.68 incluso con un 50 % de nodos maliciosos, y Krum presenta un rendimiento aún más robusto, con un $F_1score$ mínimo de 0.75 bajo las mismas condiciones. La situación se torna más crítica en la Figura 5b, con un ataque FGSM de mayor intensidad ($\alpha$ de 0.01). Aquí, las versiones base de FedAvg y Krum sufren caídas importantes

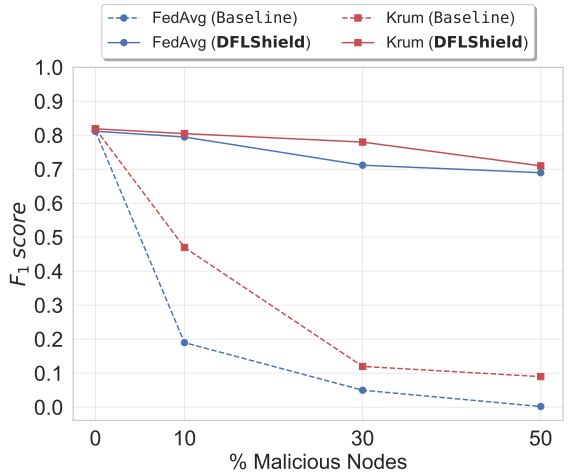
Tabla II: Comparación de rendimiento medio usando la solución propuesta sin ataques.

| Protección | Algoritmo Agregación | Modelo[1] ($F_1score$) | CPU (%) | RAM (MB) | Red (MB) | Tiempo[2] (s) |
|---|---|---|---|---|---|---|
| Ninguna | FedAvg | 0.803 ±0.012 | 61 ±9 | 67 ±7 | 132 ±10 | 60 ±2 |
| | Krum | 0.809 ±0.009 | 68 ±3 | 77 ±5 | 135 ±11 | 60 ±3 |
| `DFLShield` | FedAvg | 0.801 ±0.008 | 75 ±4 | 132 ±14 | 134 ±12 | 71 ±2 |
| | Krum | 0.808 ±0.005 | 82 ±5 | 136 ±11 | 139 ±15 | 72 ±6 |

[1] Máximo rendimiento obtenido durente la federación
[2] Tiempo en alcanzar diez rondas de federación




(a) Ataque FGSM ($\alpha = 0.003$)




(b) Ataque FGSM ($\alpha = 0.01$)

Figura 5: $F_1$ *score* medio en base a impacto y % de nodos maliciosos.

en el rendimiento, llegando a un $F_1 score$ prácticamente nulo con un 50 % de nodos maliciosos. No obstante, el despliegue de DFLShield demuestra una resiliencia notable para ambos algoritmos de agregación. En este sentido, FedAvg y Krum no solo resisten el aumento de la intensidad del ataque sino que también mantienen valores de $F_1 score$ de 0.69 y 0.71 respectivamente, incluso en el escenario más adverso.

Para complementar el análisis de eficacia de DFLShield, Tabla III muestra el uso de recursos del sistema, utilizando Krum como algoritmo de referencia para medir las diferencias en el consumo cuando se activa la mitigación, FGSM ($\alpha$ de 0.01) y 10 % de nodos maliciosos. Los resultados muestran que el aumento en la seguridad y la robustez del modelo conlleva un uso más intensivo de recursos. Específicamente, el uso de la CPU aumenta del 75 % al 91 %, reflejando la carga adicional derivada del cálculo de similitud, clustering y la evaluación de los modelos recibidos. La RAM también experimenta un incremento, de 79 MB a 136 MB, debido al almacenamiento de modelos históricos necesarios para la mitigación. El consumo de red se mantiene relativamente estable, mientras que el tiempo necesario para completar diez rondas de federación se incrementa de 60 segundos a 78 segundos, lo que refleja el procesamiento adicional y las verificaciones de seguridad incorporadas por DFLShield.

Tabla III: Comparación de rendimiento medio usando Krum, FGSM ($\alpha$=0.01) y 10 % de nodos maliciosos.

| Protección | Modelo[1] ($F_1 score$) | CPU (%) | RAM (MB) | Red (MB) | Tiempo[2] (s) |
|---|---|---|---|---|---|
| Ninguna | 0.476 ±0.008 | 75 ±9 | 79 ±7 | 142 ±21 | 60 ±2 |
| DFLShield | 0.785 ±0.012 | 91 ±4 | 136 ±19 | 145 ±17 | 78 ±5 |

[1] Máximo rendimiento obtenido durente la federación
[2] Tiempo en alcanzar diez rondas de federación

Estos resultados preliminares subrayan la eficacia de DFLShield, especialmente en colaboración con el algoritmo de agregación Krum, que por sí mismo ofrece una robustez inherente frente a ataques bizantinos. Esta colaboración logra mantener un rendimiento elevado incluso bajo condiciones de ataque intensas, demostrando la sinergia efectiva entre métodos de agregación y mecanismos de defensa. Sin embargo, es importante considerar el balance entre seguridad y uso de recursos. La implementación de DFLShield requiere un análisis cuidadoso del consumo de recursos, especialmente en términos de CPU y RAM, para garantizar que las mejoras en seguridad no comprometan la eficiencia operativa del sistema. Es relevante destacar que el despliegue de modelos más complejos, como las arquitecturas ResNet [17], o el aumento significativo de nodos pueden incrementar el consumo de recursos. Para mitigar este impacto, adoptar estrategias como la selección inteligente de modelos históricos o el almacenamiento parcial de parámetros, preservando solo aquellos críticos para la definición de la similitud entre modelos, podría ofrecer mejoras sustanciales en la gestión de recursos.

## VII. Conclusiones

Este trabajo ha proporcionado una solución para mitigar ataques bizantinos en DFL llamado DFLShield, siendo fundamental para combatir las amenazas para la seguridad y la integridad de los modelos colaborativos. Mediante un enfoque sistemático, se diseñó un modelo de amenaza utilizando FGSM con niveles de intensidad variables para simular una gama de impactos posibles. A continuación, se presentó un mecanismo de mitigación innovador, basado en el análisis exhaustivo de modelos históricos y el empleo de técnicas de clustering avanzadas. La efectividad de este mecanismo se validó en un escenario práctico utilizando una red de diez nodos y el conjunto de datos CIFAR10, junto con una CNN optimizada para este fin. Los resultados mostraron una notable capacidad para identificar y aislar nodos malintencionados, así como una integración fluida con algoritmos de agregación clave como FedAvg y Krum, alcanzando un $F_1 score$ de 68 % y 75 % respectivamente, incluso en condiciones donde la mitad de los nodos presentaban comportamientos maliciosos. A pesar de ello, la implementación de DFLShield implica un aumento moderado en el uso de recursos, con un incremento del 21 % en la utilización de la CPU y 57 MB en el consumo de RAM. Sin embargo, este incremento se considera justificado dada la protección sustancial que ofrece contra ataques disruptivos, manteniendo la fiabilidad y la integridad en DFL.

Como líneas futuras de investigación, se destacan múltiples campos de interés, muchos de los cuales ya están siendo explorados activamente. Un enfoque clave sería plantear otros ataques con el objetivo de evaluar la solución propuesta, como los ataques de Carlini & Wagner, y presentar un análisis más robusto ejecutando pruebas un número elevado de veces en la federación. En cuanto a la optimización en el uso de los recursos, se podría plantear la conservación selectiva de prototipos de modelos en lugar de la totalidad de sus parámetros al crear los históricos de modelos, buscando una mayor eficiencia sin comprometer la protección. Otro posible avance sería el desarrollo de un sistema ponderado para adaptar mejor la mitigación a entornos non-IID y minimizar el riesgo de identificar falsos positivos en la detección de nodos maliciosos. Finalmente, examinar diferentes configuraciones de red podría arrojar luz sobre la flexibilidad y capacidad de escalado de las estrategias de mitigación en diversos escenarios de DFL.

## Agradecimientos

Este trabajo ha sido financiado parcialmente por *(a)* 21629/FPI/21, Fundación Séneca - Agencia de Ciencia y Tecnología de la Región de Murcia (España), *(b)* the strategic project CDL-TALENTUM and DEFENDER from the Spanish National Institute of Cybersecurity (INCIBE) by the Recovery, Transformation and Resilience Plan, Next Generation EU, *(c)* the Swiss Federal Office for Defense Procurement (armasuisse) with the DATRIS and CyberMind projects, y *(d)* the University of Zürich UZH.

## Referencias

[1] E. T. Martínez Beltrán, M. Quiles Pérez, P. M. Sánchez Sánchez, S. López Bernal, G. Bovet, M. Gil Pérez, G. Martínez Pérez, and A. Huertas Celdrán, "Decentralized Federated Learning: Fundamentals, State of the Art, Frameworks, Trends, and Challenges," *IEEE Communications Surveys & Tutorials*, vol. 25, no. 41, pp. 2983-3013, 2023.

[2] H. Xie, M. Xia, P. Wu, S. Wang and K. Huang, "Decentralized Federated Learning With Asynchronous Parameter Sharing for Large-Scale IoT Networks," *IEEE Internet of Things Journal*, 2024.

[3] A. L. Perales Gómez, E. T. Martínez Beltrán, P. M. Sánchez Sánchez, and A. Huertas Celdrán, "TemporalFED: Detecting Cyberattacks in Industrial Time-Series Data Using Decentralized Federated Learning," *arXiv preprint arXiv:2308.03554*, 2023.

[4] E. T. Martínez Beltrán, P. M. Sánchez Sánchez, S. López Bernal, G. Bovet, M. Gil Pérez, G. Martínez Pérez, and A. Huertas Celdrán, "Mitigating communications threats in decentralized federated learning through moving target defense," *Wireless Network*, 2024.

[5] N. Rodríguez Barroso, D. Jiménez López, M. Victoria Luzón, F. Herrera, and E. Martínez Cámara, "Survey on federated learning threats: Concepts, taxonomy on attacks and defences, experimental study and challenges," *Information Fusion*, 2023.

[6] A. Reisizadeh, F. Farnia, R. Pedarsani, and A. Jadbabaie, "Robust Federated Learning: The Case of Affine Distribution Shifts," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, pp. 21554-21565, 2020.

[7] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. Agüera y Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," *arXiv preprint arXiv:1602.05629*, 2016.

[8] P. Blanchard, E. M. El Mhamdi, R. Guerraoui, J. Stainer, "Machine Learning with Adversaries: Byzantine Tolerant Gradient Descent," *Advances in Neural Information Processing Systems*, 2017.

[9] G. Kamhoua, E. Bandara, P. Foytik, P. Aggarwal and S. Shetty, "Resilient and Verifiable Federated Learning against Byzantine Colluding Attacks," *Third IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA)*, 2021.

[10] S. Li, E. C. . -H. Ngai and T. Voigt, "An Experimental Study of Byzantine-Robust Aggregation Schemes in Federated Learning," *IEEE Transactions on Big Data*, 2023.

[11] J. Xu, Y. Jiang, H. Fan and Q. Wang, "SVFLDetector: a decentralized client detection method for Byzantine problem in vertical federated learning," *Computing*, 2024.

[12] C. Feng, A. Huertas Celdrán, J. Baltensperger, E. T. Martínez Beltrán, G. Bovet, and B. Stiller, "Sentinel: An Aggregation Function to Secure Decentralized Federated Learning," *arXiv preprint arXiv:2310.08097*, 2023.

[13] Y. Miao, Z. Liu, H. Li, K. -K. R. Choo and R. H. Deng, "Privacy-Preserving Byzantine-Robust Federated Learning via Blockchain Systems," *IEEE Transactions on Information Forensics and Security*, vol. 17, pp. 2848-2861, 2022.

[14] Z. Zhang, X. Cao, J. Jia, and N. Zhenqiang, "FLDetector: Defending Federated Learning Against Model Poisoning Attacks via Detecting Malicious Clients," *28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022.

[15] X. Cao, J. Jia, Z. Zhang and N. Z. Gong, "FedRecover: Recovering from Poisoning Attacks in Federated Learning using Historical Information," *IEEE Symposium on Security and Privacy*, 2023.

[16] M. Ester, H.P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," *Second International Conference on Knowledge Discovery and Data Mining (KDD'96)*, 1996.

[17] K. He, X. Zhang, S. Ren, J. Sun, "Deep Residual Learning for Image Recognition," *arXiv preprint arXiv:1512.03385*, 2015.