Framework de Seguridad Reforzado por Blockchain para Federated Learning en Entornos MEC-IoT

Luis Miguel García-Sáez
Universidad de Castilla-La Mancha
luism.garcia@uclm.es

Sergio Ruiz-Villafranca
Universidad de Castilla-La Mancha
sergio.rvillafranca@uclm.es

Javier Carrillo-Mondéjar
Universidad de Zaragoza
jcarrillo@unizar.es

José Roldán-Gómez
Universidad de Oviedo
roldangjose@uniovi.es

José Luis Martínez-Martínez
Universidad de Castilla-La Mancha
joseluis.martinez@uclm.es

***Resumen*—El auge de las arquitecturas distribuidas inherentes a los entornos *Internet of Things (IoT)* es notable en los últimos años. Dicha descentralización ha provocado que proliferen soluciones distribuidas basadas en tecnologías, como *Multi-access Edge Computing (MEC)* y *Federated Learning (FL)*. Estos paradigmas representan un avance significativo en el procesamiento de datos y la inteligencia artificial distribuida. Sin embargo, esta convergencia plantea ciertos desafíos y problemas a nivel de seguridad, incluidos el envenenamiento de datos, la manipulación de modelos y la inserción de nodos falsos, comprometiendo la integridad del aprendizaje federado. Frente a estos retos, la tecnología de *blockchain* se presenta como una posible solución estratégica, proporcionando un marco descentralizado, transparente e inmutable que garantice la autenticidad y la verificación de datos en la red. Este trabajo propone una arquitectura basada en *blockchain* para entornos *FL* basados en *MEC-IoT*, destacando su potencial para mitigar ataques, incrementar la seguridad de los datos y promover un ecosistema de aprendizaje colaborativo seguro.**

***Index Terms*—Ciberseguridad, *Internet of Things*, *Multi-access Edge Computing*, *Blockchain*, *Smart Contract*, *Federated Learning*, envenenamiento**

Tipo de contribución: *Investigación original*

I. Introducción

Durante los últimos años, la cantidad de dispositivos *Internet of Things (IoT)* se ha multiplicado por cinco, pasando de unos 3.6 mil millones de dispositivos en el año 2015, a más de 18 mil millones de dispositivos conectados en la actualidad [1]. Además, el cambio de paradigma originado tras la aparición de las redes *Multi-access Edge Computing (MEC)*, supuso numerosas ventajas para los dispositivos *IoT*, acercando las capacidades de cómputo al usuario final y reduciendo así las latencias de las aplicaciones y servicios que se implementen. Estas características han facilitado la adopción de técnicas de *Federated Learning (FL)* [2], que permiten entrenar un modelo global sin necesidad de que los dispositivos compartan los datos utilizados por cada uno y cuyo impacto está siendo notable en áreas como la salud [3].

Esta confluencia de tecnologías representa un paradigma emergente en el desarrollo de la infraestructura de red y de los modelos distribuidos de *Machine Learning (ML)* y *FL* [4]. Este entorno de interconexión supone avances significativos en la manera en que se procesan y analizan los datos, al trasladar su procesamiento y la toma de decisiones a la zona *MEC* de la red. Al mismo tiempo, se plantean desafíos únicos en términos de seguridad y privacidad de los datos. En este contexto, los dispositivos *IoT*, que son a menudo la fuente de los datos utilizados en el aprendizaje federado [5], pueden ser vulnerables a ataques que comprometan la integridad del proceso de aprendizaje. Además, la arquitectura *MEC* está diseñada para procesar datos cerca de su fuente y reducir la latencia de la comunicación con el servidor *Edge*, e introduce otra capa de complejidad en la gestión de la seguridad [6].

En este entorno, *FL* surge como una solución innovadora, permitiendo a los nodos *IoT* colaborar en el entrenamiento de modelos de Inteligencia Artificial (IA) sin necesidad de centralizar los datos. Esto permite que cada uno de los nodos utilice sus datos locales para entrenar su propia versión del modelo, salvaguardando así las necesidades relativas a la privacidad de la información. Sin embargo, a pesar de que los datos generados permanecen en sus dispositivos [7], incrementa el número de vectores de ataque por parte de los atacantes [8].

Por otro lado, a medida que la adopción de *MEC* e *IoT* se expande, también lo hacen los desafíos de seguridad asociados, y es que, los nodos *IoT* al estar frecuentemente dispersos y operando en entornos potencialmente no seguros, se vuelven objetivos fácilmente identificables para los atacantes. Los mensajes intercambiados por estos dispositivos viajan sin ningún tipo de protección o cifrado a través de la red, lo que introduce debilidades que pueden ser aprovechadas por los atacantes para comprometer el proceso de aprendizaje federado.

Es por ello que la comunidad investigadora ha empezado a centrar sus esfuerzos en el análisis de estas deficiencias y sus efectos en los procesos de *FL*. Sin embargo, muy pocos estudios están centrados en proporcionar soluciones que ayuden a prevenir los nuevos ataques introducidos en este ámbito [9].

Los ataques, especialmente aquellos dirigidos a comprometer el modelo de *FL* a través del envenenamiento de los datos y parámetros del modelo, representan una amenaza cada vez más significativa dentro de este entorno [10]. El envenenamiento de datos implica la inserción deliberada de información falsa o engañosa en el conjunto de datos de entrenamiento, con el objetivo de alterar o degradar la precisión y eficacia del modelo de aprendizaje federado [11]. La naturaleza distribuida de los procesos de *FL*, si bien aporta beneficios en términos de privacidad y eficiencia, también puede hacer más difícil la detección y mitigación de estos

ataques, ya que no hay un punto central de verificación de la integridad de los datos. De manera contigua a los ataques de envenenamiento, existe la posibilidad de alterar el modelo a través de la inclusión de nodos falsos en la red que alteren los datos o resultados finales [12]. De esta manera, los ataques pueden ir desde la inyección de datos falsos, hasta la manipulación sutil de las actualizaciones del modelo, representando una amenaza crítica para la integridad y confiabilidad de los sistemas de *FL* en entornos *MEC-IoT* [13].

Frente a estos desafíos de seguridad, la tecnología *blockchain* se presenta como una solución capaz de abordar las vulnerabilidades y ataques en el proceso de *FL*. La utilización de una red *blockchain* nos permite aprovechar un *ledger* inmutable y transparente, donde a través de un mecanismo de consenso descentralizado, podemos mejorar la integridad y seguridad de las transacciones de datos [14]. La integración de un proceso de *FL* junto a una *blockchain* externa, no solo fortalece la resistencia contra ataques maliciosos, sino que también fomenta un ecosistema de confianza entre los nodos, asegurando que las actualizaciones de modelos y los datos compartidos sean verificables y no repudiables.

En este trabajo, se propone una arquitectura de seguridad mejorada, basada en la integración de *blockchain* y la utilización de *Smart Contracts (SC)*, con el objetivo de prevenir ataques basados en la inclusión de nodos maliciosos a la red *MEC* o ataques de envenenamiento que degraden las prestaciones del modelo. Se ha diseñado un mecanismo de registro y autenticación de nodos basado en firma digital e identificador único, con el objetivo de garantizar la legitimidad de los nodos. Además, nuestro sistema resguarda los parámetros que serán empleados durante el entrenamiento, así como los *hashes* de los datos que serán utilizados por cada uno de los nodos, evitando que sean modificados por terceras personas.

El resto del trabajo se estructura de la siguiente manera: en la *Sección II* se presentan las tecnologías clave implicadas en la propuesta, tales como *MEC*, *FL* y *blockchain*. En la *Sección III* se analizan los trabajos relacionados, destacando algunas de las limitaciones de las soluciones actuales que nuestra propuesta aborda. La *Sección IV* detalla el entorno desplegado junto a nuestra propuesta, cuya evaluación de resultados se realiza en la *Sección V*. Por último, la *Sección VI* presenta las conclusiones más importantes de nuestro trabajo, así como las posibles líneas de trabajo futuro.

II. Fundamentos Técnicos

En esta sección se presenta una revisión de las tecnologías utilizadas en nuestra propuesta. Inicialmente se introducen las características principales de un entorno *MEC* junto a *IoT*. Seguidamente, se revisan los principios fundamentales que atañen a los procesos de *FL*, junto a los principales ataques de envenenamiento. Finalmente, se analizan las características principales de la tecnología *blockchain* así como su justificación como solución viable para aumentar la seguridad y prevenir ciertos ataques durante el aprendizaje federado en entornos *MEC-IoT*.

II-A. Entorno MEC con dispositivos IoT

Inicialmente, el término *MEC* hacía referencia a *Mobile Edge Computing*, un concepto que comenzó a ganar popularidad a partir de la década de 2010. *MEC* se popularizó inicialmente para abordar las necesidades específicas de las aplicaciones móviles, con el propósito de extender los servicios de computación en la nube hacia el *Edge* de estas redes [15]. La idea inicial consistía en aprovechar las estaciones base de telefonía y otros elementos *Edge* de la red, para ofrecer capacidades de procesamiento y almacenamiento cercanas a los usuarios finales, mejorando así la eficiencia, la latencia y el consumo de energía de las aplicaciones móviles e *IoT*, con respecto a los modelos de computación en la nube tradicionales. Estas ventajas y características reflejan el papel clave de *MEC* en el soporte para tecnologías como el *5G* [16] o *Internet of Vehicles (IoV)* [17].

Con el tiempo, el concepto evolucionó hacia *Multi-access Edge Computing*, un término adoptado por la *European Telecommunications Standards Institute (ETSI)* para reflejar una visión más amplia e inclusiva. Esta evolución reconoce que el procesamiento *Edge* puede beneficiar no solo a los dispositivos móviles y sus usuarios, sino también a todo tipo de servicios y aplicaciones en tiempo real [18]. Es así que esta combinación de las capacidades *MEC* junto a dispositivos *IoT*, cuenta con una amplia variedad de aplicaciones en varios sectores, como ciudades inteligentes, industria 4.0 o *IoV*.

II-B. Proceso de Federated Learning

El *FL* es un campo que está evolucionando rápidamente durante estos últimos años, con importantes implicaciones a nivel de privacidad en los procesos de aprendizaje automático. *FL* es un enfoque de aprendizaje automático en el que múltiples clientes entrenan de manera colaborativa un modelo bajo la coordinación de un servidor central, pero sin compartir los datos en crudo [19]. A pesar de la existencia de un modelo global, cada uno de los clientes entrena su propio modelo local utilizando sus propios datos. Este entrenamiento se realiza de manera independiente y se centra en los datos específicos que posee cada cliente. El objetivo es aprender de los datos disponibles localmente, sin compartir esos datos con el servidor o con el resto de dispositivos.

Es un error común pensar en la ausencia de un servidor central en los procesos de *FL*, ya que, aunque el entrenamiento de los modelos se realiza de manera local en cada uno de los nodos o dispositivos, se requiere la existencia de un servidor central que recibe los modelos o actualizaciones de los modelos entrenados localmente desde los dispositivos, los agrega para mejorar el modelo global, y luego distribuye este modelo actualizado de vuelta a los dispositivos [20]. Una vez los clientes reciben el modelo global actualizado del servidor, pueden optar por reemplazar completamente su modelo local por el global, o utilizar el modelo global para ajustar y mejorar sus modelos locales. Esta estrategia dependerá del enfoque específico de *FL* y de los requisitos que se quieran satisfacer.

El principal problema de este ecosistema es la falta de seguridad y robustez en los dispositivos *IoT*. Esto provoca que durante el proceso de *FL* que llevan a cabo los nodos, puedan realizarse ciertos ataques que pueden afectar al rendimiento global del modelo a través de la modificación de los datos

de entrenamiento, parámetros del modelo o la inclusión de nodos con modelos locales alterados. De ahí la aparición de numerosos estudios que analizan en profundidad las amenazas que pueden afectar a este tipo de entornos y que los atacantes podrían emplear para romper la seguridad [13].

II-C. Ataques de envenenamiento en FL

A través de los ataques de envenenamiento, los atacantes buscan corromper los datos en los dispositivos *IoT* durante las actualizaciones locales o corromper las actualizaciones del modelo [21]. Se analizan los diferentes tipos de ataque más usuales llevados a cabo por los atacantes, tales como:

- **Ataque de Cambio de Etiqueta (*Label Flipping Attack*)**: El atacante altera las etiquetas de un subconjunto de datos de entrenamiento. El objetivo es confundir al modelo durante el entrenamiento para degradar su precisión, haciendo que clasifique incorrectamente las entradas. Este tipo de ataque puede ser relativamente fácil de perpetrar y solo requiere de acceso al conjunto de datos.
- **Ataque de Eliminación Dirigida (*Targeted Dropping Attack*)**: En este caso, el ataque se dirige directamente a los datos en lugar de a las etiquetas de estos. Se pretende eliminar selectivamente ciertos datos importantes o críticos del conjunto de entrenamiento para reducir la efectividad general del modelo.
- **Ataque de Etiqueta Limpia (*Clean-Label Attack*)**: Se introducen perturbaciones mínimas pero efectivas en los datos de entrada, sin cambiar las etiquetas originales. A diferencia del primer caso, en los ataques de etiqueta limpia, los datos envenenados parecen ser instancias normales del conjunto de datos. Estos ataques son especialmente difíciles de detectar ya que los datos modificados no presentan diferencias obvias en sus etiquetas en comparación con su contenido.
- **Ataque de manipulación de parámetros (*Parameter Tampering Attack*)**: Este ataque se centra en la manipulación de los parámetros enviados por el servidor a los clientes, con el objetivo de comprometer el proceso de aprendizaje desde el lado del servidor. Si un atacante logra manipular los parámetros que el servidor envía a los clientes, puede influir de manera significativa en el comportamiento del modelo para sesgar el aprendizaje o reducir su precisión.

II-D. Red blockchain

Una red *blockchain* es una red *Peer-to-Peer (P2P)* formada por pares de nodos que pueden comunicarse directamente. A su vez, *blockchain* es una forma de *Distributed Ledger Technology (DLT)* [22], al disponer de una base de datos distribuida entre todos y cada uno de los nodos que conforman la red, a la que se denomina *ledger*. Este *ledger* es lo que comúnmente se conoce como *cadena de bloques*, y en él se registran todas las operaciones o transacciones que se realicen a través de la red, que serán previamente validadas entre todos los nodos mediante mecanismos de consenso.

Este *ledger* está formado por una cadena consecutiva de bloques. Dichos bloques contienen información relativa a las transacciones validadas en la red y otros parámetros que permitan garantizar la integridad del *ledger*. La estructura a la que responde la cadena de bloques se conoce como *Merkle tree* [23]. En cada bloque contamos esencialmente con:

- ***Hash* del bloque**: Es un solo *hash* que representa todas las transacciones dentro del bloque y que permite su rápida verificación sin necesidad de revisar cada una individualmente.
- ***Hash* bloque anterior**: Genera una dependencia de manera que si un bloque es alterado, todos los bloques siguientes tendrán *hashes* inválidos, revelando así la alteración.
- **Sello temporal**: Este sello temporal garantiza que los datos se han registrado en una secuencia de tiempo específica.

Los *hashes* de las transacciones se combinan dos a dos para formar los *hashes* del siguiente nivel, hasta llegar al *hash root*. Así, se asegura que cualquier cambio en una sola transacción cambiará el *hash root*, lo que se identifica de inmediato al compararlo con el *hash root* acordado por la red.

De esta manera, el *Merkle tree* resulta clave para garantizar la integridad de los datos en la cadena de bloques, al hacer extremadamente difícil alterar las transacciones sin ser detectado. Este diseño es crucial para la escalabilidad y seguridad en *blockchain*. Por otro lado, los *SC* son otros elementos fundamentales que forman parte de la gran mayoría de redes *blockchain* actuales [24]. Estos se incorporan a la tecnología con el nacimiento de la red de *Ethereum* [25] y a día de hoy son claves en el desarrollo de aplicaciones y programas dentro de las redes *blockchain*. Un *SC* es un programa informático que se introduce en la *blockchain* y que se ejecuta automáticamente cuando se cumplen ciertas condiciones o requisitos, o del que podemos hacer uso al satisfacer ciertas necesidades. Permite que procesos como la verificación, ejecución y/o cumplimiento de los términos de dicho *SC* se realicen de forma automática. La ejecución se realiza dentro de la cadena de bloques, lo que garantiza que el *SC* sea inmutable y distribuido, proporcionando así un alto nivel de seguridad.

III. Estado del Arte

En esta sección se examinan algunos estudios y propuestas ya existentes que emplean *blockchain* para mejorar la seguridad en entornos de *FL* [26].

Existen propuestas novedosas basadas en la utilización de una red *blockchain* como un sistema de detección y monitorización del proceso completo de *FL*. Sin embargo, en muchas de ellas se emplea la *blockchain* para almacenar y analizar las actualizaciones locales de los nodos de la red, con el objetivo de detectar dinámicamente cuándo se podría estar siendo víctima de un ataque de envenenamiento. Es el caso del trabajo planteado por Preuveneers [27]. El principal inconveniente de la propuesta es que la latencia total, fruto de la comunicación con la *blockchain*, se incrementa drásticamente. Además, propuestas como la de Al Mallah [28], no indican los patrones empleados para filtrar los *miners* legítimos y tampoco aportan métricas de rendimiento que indiquen el impacto real que ha tenido la integración de la *blockchain* con respecto a la latencia o tiempo total de ejecución.

Por otro lado, existen otro tipo de propuestas basadas en un sistema de puntuaciones o reputación de los nodos

[29]. Lei Feng [30] apuesta por la utilización de un sistema *blockchain* junto a las puntuaciones de los nodos, que son calculadas en base a un valor de entropía. Esta entropía se emplea para evaluar y ponderar la importancia y confiabilidad de los modelos locales entrenados por los nodos durante el aprendizaje. La red *blockchain* se encargará de registrar estos pesos junto a las actualizaciones locales de los modelos y al modelo global. Gracias a este mecanismo, se pueden identificar las actualizaciones maliciosas y mitigarlas, detectando posibles nodos maliciosos y expulsándolos de la red. Si bien es una propuesta robusta y efectiva, introduce una capa de complejidad basada en la entropía, cuyo cálculo debe ser cuidadosamente calculado para equilibrar las contribuciones de los nodos y la calidad del modelo global. Nuestra propuesta no requiere de esta lógica adicional de cálculo de la entropía, pues incluye un sistema de autenticación específico [31] que trata de asegurar la legitimidad de los nodos. Por otro lado, en lugar de almacenar las actualizaciones locales de los modelos para detectar los posibles ataques de envenenamiento, almacenamos los parámetros y datos empleados para entrenar el modelo, cuya validez es comprobada antes de las etapas críticas de entrenamiento y evaluación, garantizando así la validez de los modelos locales de los nodos.

Por último, existen trabajos que persiguen una mayor descentralización, mediante la sustitución del servidor central que coordina el aprendizaje federado por una red *blockchain* [32]. Esta *blockchain* se emplea para el almacenamiento del modelo global y el envío de las actualizaciones locales de los nodos. Muchos de estos trabajos incorporan el diseño de algoritmos de consenso específicos para los procesos de *FL* [33]. El enfoque de estos estudios se basa en premiar más frecuentemente a los dispositivos honestos, es decir aquellos que mandan actualizaciones correctas de su modelo local, con el objetivo de que solo las actualizaciones legítimas se empleen para actualizar el modelo global. De esta manera, se intenta minimizar el riesgo de que un nodo malicioso pueda insertar actualizaciones falsas de su modelo en la *blockchain*. Sin embargo, esta no es una solución directa frente a los ataques de envenenamiento y no evita que si un atacante tiene control completo sobre un nodo, continúe enviando actualizaciones incorrectas de su modelo local.

Las diferentes propuestas analizadas ofrecen diversas formas de afrontar las debilidades en los entornos de *FL*, sin embargo, no implementan mecanismos específicos de autenticación para los nodos de la red, lo que supone una primera barrera frente a los atacantes a la hora de introducir nodos maliciosos en la red. Además, nuestra propuesta no se basa en el diseño de un algoritmo de consenso en específico o en establecer un sistema de nodos basado en la reputación para prevenir el envenenamiento, sino que la funcionalidad se logra gracias al uso de *SC*, cuya flexibilidad permite adaptar nuestra solución a gran variedad de situaciones y entornos, resultando en una solución mucho más generalista. Asimismo, en nuestra propuesta aportamos métricas reales sobre el impacto de la *blockchain* en la latencia, y buscamos minimizarla gracias a la utilización de la *blockchain* como un elemento pasivo, que será utilizado por los nodos del proceso de *FL* en las etapas críticas, en lugar de como un elemento activo que introduce latencia continuamente.

IV. Integración de Blockchain para mejorar la Seguridad en FL

Para proteger la red de los ataques mencionados y mejorar la seguridad durante el proceso de *FL*, se ha optado por la integración de la tecnología *blockchain*, la cual nos proporciona una serie de características y propiedades únicas que nos permiten mitigar de manera efectiva las deficiencias que atañen al proceso de aprendizaje y garantizar la validez de los datos. Entre ellas destacan, la inmutabilidad de los datos almacenados en el *ledger*, la naturaleza pública del *ledger*, que aporta transparencia y trazabilidad a las transacciones o la capacidad de ejecutar *SC*.

Los *SC* son el medio principal empleado para combatir los ataques de envenenamiento y tratar de garantizar la legitimidad de los nodos que participan en el proceso de *FL*. Permiten definir una serie de reglas personalizadas y específicas que determinan su funcionalidad, las condiciones de ejecución del contrato y las políticas de acceso a su funcionalidad. Esto permite controlar el acceso por parte de los diferentes nodos de la red a los *SC* y a sus respectivas funcionalidades internas.

Gracias a estas características y a su capacidad de automatización, creamos una arquitectura combinada de comunicación, que nos permite realizar de manera automática gestiones dentro de la *blockchain* y consultas a ésta mientras se lleva a cabo el proceso de *FL*. Esto permite mitigar el efecto de los ataques descritos previamente, en tiempo real.

Para el establecimiento inicial del entorno *MEC-IoT*, se ha empleado el emulador *MECInOT* [34]. Este emulador nos permite realizar el despliegue de topologías *MEC-IoT* para la experimentación en un contexto de ciberseguridad. Entre muchas de sus características, el emulador nos ofrece un entorno realista en el que poder realizar experimentos, y obtener datos de manera similar a como lo haríamos con una topología real, sin necesidad de utilizar dispositivos físicos. Ligado a estos experimentos, *MECInOT* ofrece gran flexibilidad a la hora de decidir la cantidad de dispositivos *IoT* a desplegar en la red gracias al uso de la virtualización y permite también la inclusión de dispositivos reales de red.

IV-A. Arquitectura blockchain y diseño de Smart Contracts

Nuestra propuesta se centra en evitar tanto los ataques de tipo *Sybil*, como los ataques de envenenamiento de datos y de los parámetros del modelo. Para ello, hemos diseñado una arquitectura *blockchain* basada en el uso de *SC*, con el objetivo de proporcionar una solución adaptable que permita mitigar el efecto de todos estos ataques sobre el proceso de *FL*.

En la *Figura 1*, observamos una imagen global del entorno que combina tanto la red *blockchain* como la red *MEC-IoT*. En la parte inferior podemos identificar la red *MEC-IoT* con los nodos que estarán ejecutando el algoritmo de *FL*. La red *MEC-IoT* se comunicará con la red *blockchain*, que podemos identificar en la parte superior de la figura. Dicha red estará formada por una serie de nodos que se limitarán a los procesos de validación de transacciones a través del algoritmo de consenso implementado y a codificar dichas transacciones en los bloques que posteriormente serán incluidos en el *ledger*. A su vez, se ha realizado el diseño de una serie de *SC*, que permitirán a los diferentes nodos de la red *MEC-IoT* interactuar con la *blockchain* y acceder a las

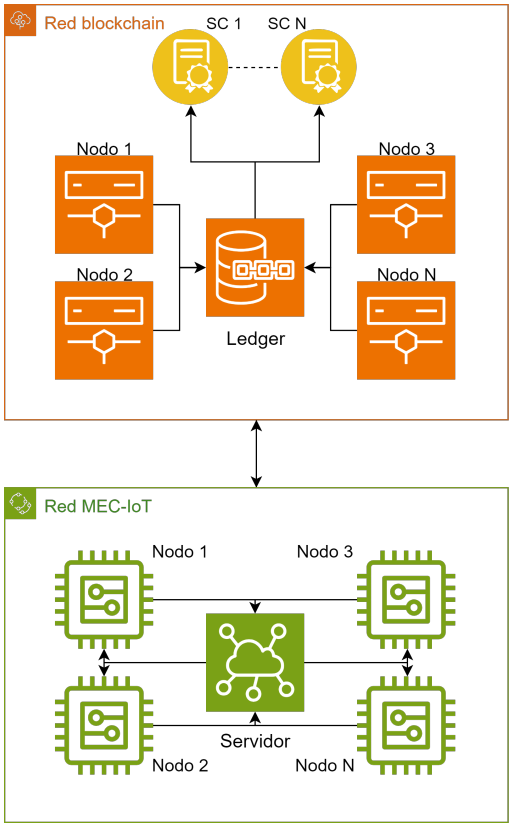

Figura 1. Estructura del entorno de pruebas

diferentes funcionalidades de estos *SC*, al tiempo que se lleva a cabo el proceso de *FL*. Estos *SC*, una vez introducidos en el *ledger* de la red, son inmutables e inalterables, por lo que no pueden ser objeto de los atacantes.

El primero de los *SC* se ha diseñado específicamente para garantizar la legitimidad de todos los nodos de la red y prevenir posibles ataques de tipo *Sybil*. La Figura 2 refleja el comportamiento y flujo de interacción con este *SC*. Previamente a la interacción con este contrato, se generará un par de claves pública-privada para cada uno de estos nodos, de forma que cada uno será responsable de la seguridad de su clave privada. A través de este *SC*, los nodos de la red deberán darse de alta en la plataforma *blockchain*, por lo que la red será consciente de la existencia de dicho nodo y se almacenará su clave pública. Asimismo, una vez registrado el nodo, se le asignará y proporcionará un *Identificador (ID) de nodo* que será único y que servirá para tener un sistema de doble autenticación basado en el par clave pública-privada y el ID del nodo.

De esta manera, para realizar la autenticación del nodo, este deberá firmar digitalmente un contenido utilizando su clave privada. La *blockchain* se encargará de, empleando el *hash* de la firma y datos, obtener la clave pública del nodo y comprobar que está registrado en el sistema. Por último, se comprobará que esta clave pública está vinculada al ID del nodo que solicita la autenticación y no a otro nodo distinto.

Este primer *SC* resulta esencial para el correcto funcionamiento de la red, ya que la autenticación de los nodos determinará su capacidad de acceso o no a las funciones del

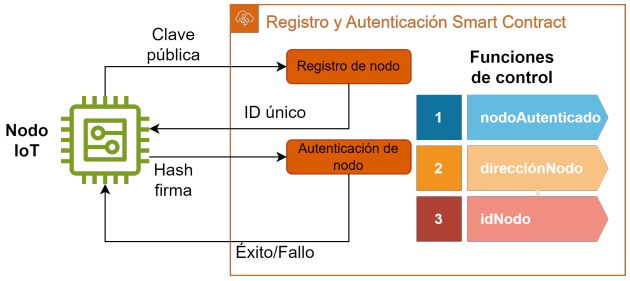

Figura 2. *Smart Contract* para registro y autenticación de nodos

segundo *SC*.

El segundo de los *SC* ha sido diseñado específicamente para contrarrestar los ataques de envenenamiento de los datos y parámetros del modelo de *FL*:

- **Envenenamiento de datos**: Para prevenir el envenenamiento de los datos utilizados por cada uno de los clientes, tanto para los procesos de entrenamiento como de evaluación, este *SC* se encargará de almacenar cierta información. Inicialmente serán almacenados 4 *hashes* identificativos de los datos y etiquetas del *dataset* global que será utilizado. Esto permitirá que si un atacante modifica algún dato o etiqueta, podamos identificarlo rápidamente y detectar el tipo de ataque de envenenamiento (*Label Flipping*, *Targeted Dropping*, entre otros). Además, cada uno de los nodos genera 10 particiones sobre el conjunto inicial de datos y selecciona aleatoriamente una para realizar el entrenamiento. Para ello, este contrato se encargará también de almacenar un *hash* conjunto para los datos y etiquetas de cada uno de los subconjuntos de datos seleccionados por los nodos. Es así que se consigue mantener la trazabilidad de los datos desde el inicio hasta el final.
- **Envenenamiento de parámetros**: Como medida que contrarreste estos ataques, el contrato está diseñado para comunicarse con el servidor del proceso de *FL*, de forma que los parámetros que este envía a todos y cada uno de los clientes que forman parte del proceso, serán almacenados en la *blockchain*. Así pues, cada uno de los clientes, previamente a comenzar con las etapas de entrenamiento y evaluación, comprobará que los parámetros que reciba del servidor coinciden con aquellos almacenados en la *blockchain*, evitando así la posibilidad de que un atacante intercepte la comunicación cliente-servidor y altere maliciosamente los parámetros del modelo.

IV-B. Mitigación de los ataques

Gracias al proceso de autenticación doble basado en firma digital e ID de nodo, tratamos de asegurar la legitimidad de todos los nodos miembros de la red. Un atacante no tendrá la posibilidad de introducir un nodo de manera subrepticia sin ser detectado por la red. Las claves públicas de los nodos legítimos se encuentran registradas por la *blockchain* por lo que en caso de identificar una clave pública desconocida, sería detectada de inmediato. Además, cada nodo deberá pasar el proceso de autenticación del *SC* correspondiente para poder acceder a las funcionalidades propias del proceso de *FL*. El doble factor de autenticación de esta función del *SC* garantiza

que, a pesar de que un atacante consiga acceso a la clave privada de un nodo, deberá obtener también el ID exclusivo de ese nodo que le fue otorgado por la *blockchain*.

Por otro lado, para mitigar los ataques de envenenamiento una vez ha comenzado el proceso de *FL*, se ha diseñado el esquema de comunicación de la *Figura 3*. En primer lugar, el servidor que coordina el aprendizaje almacena en la *blockchain* los parámetros del modelo que serán proporcionados a cada uno de los clientes. Una vez estos parámetros han sido registrados en el *SC* correspondiente, el servidor procede a enviarlos a los clientes. De esta manera, se consigue que, en el supuesto de que un atacante consiga interceptar esta comunicación cliente-servidor y alterar los valores finales que llegan a los clientes, se producirá una discrepancia con respecto a los valores registrados en la *blockchain* y se detendrá el proceso de aprendizaje. El tercer paso consiste en almacenar los *hashes* relativos al conjunto de datos que serán utilizados para las fases de entrenamiento y evaluación. Esta tarea la realiza el servidor y almacena hasta 4 *hashes*, para los datos de entrenamiento y evaluación, y para las etiquetas respectivas de dichos datos.

En este punto, los clientes obtienen los datos igual que lo hizo el servidor y comprueban que los *hashes* almacenados en la *blockchain* coinciden con los *hashes* locales calculados por cada uno de ellos. De esta manera, aseguramos que todos han obtenido acceso al mismo *dataset*. A continuación y como tarea previa al entrenamiento, cada uno de los clientes particiona el conjunto de datos en un número concreto de particiones y selecciona una de ellas. Una vez la partición final sobre la que se entrenará ha sido seleccionada, se construye un *hash* conjunto para los datos y etiquetas de dicha partición, los cuales son almacenados por cada nodo en la *blockchain*. Así, se mantiene un registro de *hashes* de las particiones finales con las que entrenarán cada uno de los nodos.

Una vez comenzado el entrenamiento, los clientes comprueban los parámetros del modelo recibidos por parte del servidor. Cada cliente accede al *SC* almacenado en el *ledger* y obtiene los valores iniciales para dichos parámetros que el servidor almacenó. Si tanto los parámetros recibidos del servidor, como los almacenados en la *blockchain* son equivalentes, se prosigue con normalidad con el proceso de entrenamiento. En caso contrario, significa que un atacante ha conseguido alterar los parámetros del modelo y el entrenamiento se detiene. Una vez verificada la validez de los parámetros, se realiza el paso 7, donde cada nodo obtiene el *hash* de su subconjunto de datos de entrenamiento. Recalcula los *hashes* locales del subconjunto y comprueba que coinciden con los devueltos por el *SC*, garantizando su integridad y procediendo, en caso de éxito, con el entrenamiento del modelo.

Posteriormente, se realizan los pasos de forma análoga para la fase de evaluación, comprobando la validez de los parámetros del modelo y datos de evaluación con la *blockchain*. Una vez finalizada la evaluación, se retorna el modelo local al servidor.

En este punto, ya ha finalizado la primera ronda de entrenamiento del proceso de *FL*. Seguidamente se realizarían el resto de rondas de entrenamiento y finalmente se obtendrían los valores finales del modelo. Gracias a la arquitectura implementada y al diseño del esquema de comunicación de

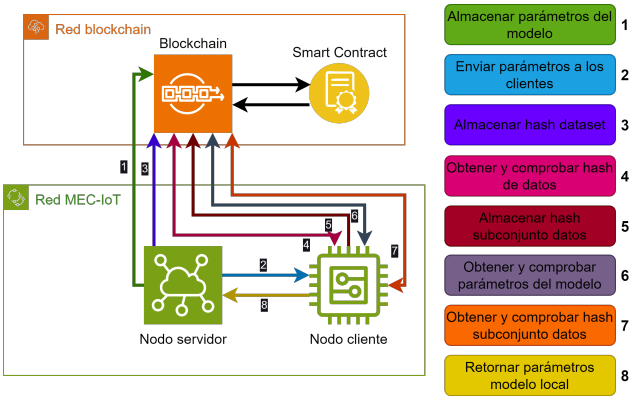

Figura 3. Esquema de comunicación con la *blockchain*

los nodos *IoT* con la *blockchain*, hemos conseguido evitar tanto la posibilidad de inclusión de nodos maliciosos que afecten al rendimiento de nuestro modelo, como la posibilidad de que a través de ataques *Man-in-the-Middle (MitM)*, un atacante pueda alterar los datos que serán utilizados para el entrenamiento y/o evaluación del modelo, así como los parámetros del modelo que el servidor, que coordina el aprendizaje federado, envía a los nodos clientes.

V. Evaluación del sistema

Los ataques presentados a lo largo del trabajo han sido implementados en nuestro entorno de pruebas, con el objetivo de medir el impacto sobre el rendimiento que podrían tener y cómo nuestra solución es capaz de mitigar sus efectos. Se ha utilizado *MNIST* como *dataset* junto a un algoritmo de regresión logística, aunque el sistema sería válido para cualquier algoritmo de aprendizaje. Las pruebas han sido realizadas con un total de 6 nodos, 1 servidor y 5 clientes.

En primer lugar, se ha realizado un ataque de tipo *Sybil*, donde se ha introducido un nodo malicioso en la red, cuya función es enviar valores erróneos y alejados de las predicciones del modelo, al servidor. En este caso, al tener control total sobre el nodo, el atacante pretende afectar al máximo a la precisión del modelo, por lo que este nodo realiza predicciones inversas a lo dictado por el modelo.

En segundo lugar, se ha realizado un ataque de envenenamiento de datos combinando *Label Flipping* junto a *Targeted Dropping*. Concretamente se han alterado los valores para los datos de entrenamiento y los valores para las etiquetas de los datos de evaluación de uno de los clientes. En él, si bien no conseguimos modificar directamente los valores de la precisión del modelo que los clientes devuelven al servidor, podemos alterar directamente los datos que luego generarán esa precisión. Hemos añadido una desviación aleatoria a los datos de entrenamiento del cliente seleccionado, e invertido las etiquetas para los datos de evaluación.

En tercer y último lugar, se ha llevado a cabo un ataque para envenenar los parámetros de entrenamiento de uno de los clientes. Este ataque consiste en interceptar los paquetes enviados por el servidor a cada uno de los clientes junto a los parámetros de entrenamiento justo antes de sus fases de entrenamiento y evaluación. En este caso se ha alterado el número máximo de iteraciones por ronda de entrenamiento, estableciéndolo en 1.

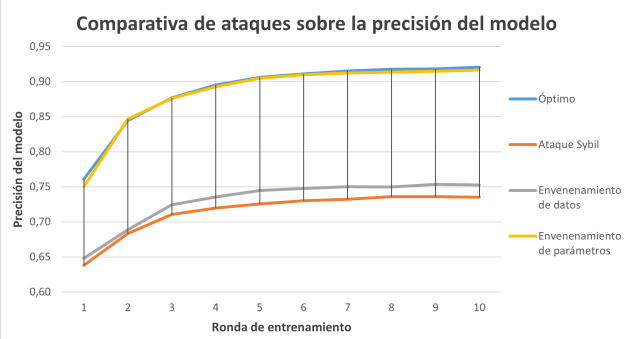

Figura 4. Efecto de los ataques sobre la precisión del modelo

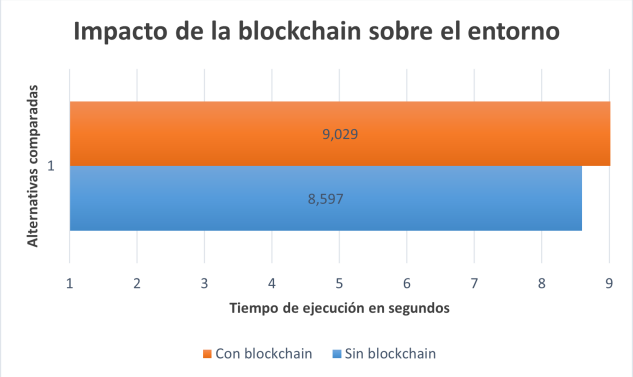

Figura 5. Impacto de la comunicación con la red *blockchain*

Si revisamos la *Figura 4*, podemos observar cómo han afectado cada uno de estos ataques sobre la precisión del modelo global con respecto a la ejecución en un entorno seguro sin ningún ataque. El ataque que tenido un mayor efecto sobre la precisión ha sido el *Sybil attack*, seguido del ataque de envenenamiento de datos. Esto tiene sentido ya que en el primero de ellos, independientemente de los datos utilizados, se altera la precisión del modelo de forma directa. En el segundo conseguimos modificar la precisión de forma indirecta a través de la alteración intencional y dirigida de los datos utilizados. Por último, resalta el bajo impacto que ha tenido el ataque de envenenamiento de los parámetros del modelo. Esto se debe a que en este ataque, sí se utilizan los datos legítimos y a pesar de que reduzcamos a 1 el número de iteraciones por ronda, existe una mejora continuada ronda tras ronda sobre el modelo local del cliente, es decir, se consigue afectar negativamente al modelo pero no lo incapacita por completo. De ahí el bajo impacto sobre el rendimiento con respecto a la ejecución original sin ataques. En caso de conseguir alterar los parámetros de entrenamiento para varios de los clientes, este ataque tendría un efecto mayor.

Sin embargo, hemos de considerar que con estos ataques se ha afectado únicamente a uno de los clientes, por lo que en caso de introducir un número mayor de clientes maliciosos en la red o de alterar las comunicaciones cliente-servidor de más nodos, los efectos causados pueden ser mucho mayores.

Finalmente, se ha realizado un análisis del impacto de la *blockchain* sobre el rendimiento de la red, utilizando como principal medida de rendimiento, el tiempo de ejecución (TE) de todo el proceso de *FL*, desde que es iniciado por el servidor, hasta que todos los clientes finalizan sus rondas de entrenamiento y se obtiene la versión final del modelo global entrenado. Este impacto sobre el rendimiento puede observarse en la *Figura 5*, donde se aprecia que la sobrecarga de nuestra solución sobre el sistema original es de tan solo el 5.03 % manteniéndose constante en las pruebas realizadas, con ejecuciones de hasta 5 nodos. En el caso observado en la figura (ejecución con 3 nodos), el TE se incrementa desde los 8.597 hasta los 9.029 segundos. Destacamos la baja latencia introducida por nuestra solución, frente a otras propuestas empleando *blockchain* con sobrecargas mucho mayores (superiores incluso al 1000 %)[35] u otras que no ofrecen ninguna métrica sobre la latencia introducida por la comunicación con la *blockchain* en el sistema [32].

VI. Conclusiones y Trabajos Futuros

El trabajo presentado ofrece una solución innovadora y estratégica para abordar los desafíos de seguridad en el ámbito de *FL*, aplicado a entornos *MEC* integrados con dispositivos *IoT*. A través de la implementación de una infraestructura *blockchain* basada en el uso de *SC*, esta arquitectura es capaz de enfrentar vulnerabilidades específicas como los ataques de envenenamiento de datos, la manipulación de los modelos locales de aprendizaje o la inserción en la red de nodos maliciosos, manteniendo un bajo impacto sobre el rendimiento global de la red, y garantizando la integridad del modelo global de *FL*.

Una de las principales fortalezas de la propuesta reside en su capacidad para proporcionar un marco de trabajo descentralizado, transparente e inmutable gracias a la integración con *blockchain*. Este enfoque garantiza la autenticidad y la verificación de los datos empleados por los nodos, lo que resulta crucial para garantizar la validez de los modelos locales generados y que posteriormente serán agregados por el servidor. La *blockchain* actúa como un elemento unificador que ofrece resistencia contra ataques, y gracias a la flexibilidad proporcionada por el uso de contratos inteligentes, permite una fácil escalabilidad para abordar otros ataques o posibles vulnerabilidades futuras. Además, al ofrecer una solución que no solo es segura sino también eficiente, se satisfacen los desafíos de latencia y sobrecarga en la red requeridos para el correcto funcionamiento de la red *MEC*.

Como trabajo futuro, pueden optimizarse los mecanismos de comunicación y validación para reducir aún más el tiempo de ejecución y la latencia introducida por la interacción con la *blockchain*. La arquitectura propuesta se centra en mitigar ataques específicos como los de envenenamiento de datos o los ataques de tipo *Sybil*, por lo que futuras investigaciones pueden expandirse para cubrir un rango más amplio de vulnerabilidades y ataques, como ataques de inferencia, ataques de reconstrucción de datos, u otros más sofisticados. Además, trabajos futuros podrían enfocarse en implementaciones prácticas y experimentación en otros entornos sin *MEC*.

Agradecimientos

Este trabajo contó con el apoyo de la Universidad de Castilla-La Mancha bajo el contrato predoctoral 2022-PRED-20677 y el proyecto 2023-GRIN-34056, ambos financiados

por el Fondo Social Europeo Plus (FSE+), y por la JCCM bajo el contrato de investigación 2023-CACT-12003 y el proyecto SBPLY/21/180501/000195. Además, este trabajo forma parte del proyecto de I+D PID2021-123627OB-C52, financiado por el MCIN y el Fondo Europeo de Desarrollo Regional: "una forma de hacer Europa". Este trabajo también fue apoyado en parte por TED2021-131115A-I00 y PID2022-142332OA-I00, financiada por MCIN/AEI/10.13039/501100011033, por los fondos del Plan de Recuperación, Transformación y Resiliencia, financiados por la Unión Europea (Next Generation), por el Instituto Nacional de Ciberseguridad de España (INCIBE) bajo *Proyectos Estratégicos de Ciberseguridad – CIBERSEGURIDAD EINA UNIZAR*, y por el Departamento de Universidad, Industria e Innovación del Gobierno de Aragón bajo *Programa de Proyectos Estratégicos de Grupos de Investigación* (grupo de investigación DisCo, ref. T21-23R).

Referencias

[1] "Internet de las cosas (IoT): dispositivos conectados en el mundo 2015-2027." [Online]. Available: https://t.ly/4xx1W

[2] M. Mahbub, M. S. Apu Gazi, S. A. Arabi Provat, and M. S. Islam, "Multi-Access Edge Computing-Aware Internet of Things: MEC-IoT," in *2020 Emerging Technology in Computing, Communication and Electronics (ETCCE)*. Bangladesh: IEEE, Dec. 2020, pp. 1–6.

[3] B. Yuan, S. Ge, and W. Xing, "A Federated Learning Framework for Healthcare IoT devices," 2020.

[4] D. C. Nguyen, M. Ding, P. N. Pathirana, A. Seneviratne, J. Li, and H. Vincent Poor, "Federated Learning for Internet of Things: A Comprehensive Survey," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 3, pp. 1622–1658, 2021.

[5] T. Zhang, L. Gao, C. He, M. Zhang, B. Krishnamachari, and A. S. Avestimehr, "Federated Learning for the Internet of Things: Applications, Challenges, and Opportunities," *IEEE Internet of Things Magazine*, vol. 5, no. 1, pp. 24–29, Mar. 2022, conference Name: IEEE Internet of Things Magazine.

[6] P. Ranaweera, A. D. Jurcut, and M. Liyanage, "Survey on Multi-Access Edge Computing Security and Privacy," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 2, pp. 1078–1124, 2021.

[7] Z. Li, V. Sharma, and S. P. Mohanty, "Preserving Data Privacy via Federated Learning: Challenges and Solutions," *IEEE Consumer Electronics Magazine*, vol. 9, no. 3, pp. 8–16, May 2020.

[8] S. K. Choudhary, A. K. Kar, and Y. K. Dwivedi, "How does Federated Learning Impact Decision-Making in Firms: A Systematic Literature Review."

[9] M. A. Ferrag, B. Kantarci, L. C. Cordeiro, M. Debbah, and K.-K. R. Choo, "Poisoning Attacks in Federated Edge Learning for Digital Twin 6G-enabled IoTs: An Anticipatory Study," Mar. 2023, arXiv:2303.11745 [cs].

[10] V. Shejwalkar, A. Houmansadr, P. Kairouz, and D. Ramage, "Back to the Drawing Board: A Critical Evaluation of Poisoning Attacks on Production Federated Learning," in *2022 IEEE Symposium on Security and Privacy (SP)*. San Francisco, CA, USA: IEEE, May 2022, pp. 1354–1371.

[11] S. Alharbi, Y. Guo, and W. Yu, "Collusive Backdoor Attacks in Federated Learning Frameworks for IoT Systems," *IEEE Internet of Things Journal*, pp. 1–1, 2024, conference Name: IEEE Internet of Things Journal.

[12] Y. Jiang, Y. Li, Y. Zhou, and X. Zheng, "Sybil Attacks and Defense on Differential Privacy based Federated Learning," in *2021 IEEE 20th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*. Shenyang, China: IEEE, Oct. 2021, pp. 355–362.

[13] W. Han, Y. Cho, and Y. Paek, "A Survey on Threats to Federated Learning," 2023.

[14] M. N. M. Bhutta, A. A. Khwaja, A. Nadeem, H. F. Ahmad, M. K. Khan, M. A. Hanif, H. Song, M. Alshamari, and Y. Cao, "A Survey on Blockchain Technology: Evolution, Architecture and Security," *IEEE Access*, vol. 9, pp. 61 048–61 073, 2021, conference Name: IEEE Access.

[15] N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie, "Mobile Edge Computing: A Survey," *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 450–465, Feb. 2018, conference Name: IEEE Internet of Things Journal.

[16] L. Tomaszewski, S. Kukliński, and R. Kołakowski, "A New Approach to 5G and MEC Integration," in *Artificial Intelligence Applications and Innovations. AIAI 2020 IFIP WG 12.5 International Workshops*, I. Maglogiannis, L. Iliadis, and E. Pimenidis, Eds. Cham: Springer International Publishing, 2020, pp. 15–24.

[17] X. Xu, H. Li, W. Xu, Z. Liu, L. Yao, and F. Dai, "Artificial intelligence for edge service optimization in Internet of Vehicles: A survey," *Tsinghua Science and Technology*, vol. 27, no. 2, pp. 270–287, Apr. 2022, conference Name: Tsinghua Science and Technology.

[18] A. Filali, A. Abouaomar, S. Cherkaoui, A. Kobbane, and M. Guizani, "Multi-Access Edge Computing: A Survey," *IEEE Access*, vol. 8, pp. 197 017–197 046, 2020, conference Name: IEEE Access.

[19] M. Aledhari, R. Razzak, R. M. Parizi, and F. Saeed, "Federated Learning: A Survey on Enabling Technologies, Protocols, and Applications," *IEEE Access*, vol. 8, pp. 140 699–140 725, 2020.

[20] C. Zhang, Y. Xie, H. Bai, B. Yu, W. Li, and Y. Gao, "A survey on federated learning," *Knowledge-Based Systems*, vol. 216, p. 106775, Mar. 2021.

[21] V. Tolpegin, S. Truex, M. E. Gursoy, and L. Liu, "Data Poisoning Attacks Against Federated Learning Systems," in *Computer Security – ESORICS 2020*, L. Chen, N. Li, K. Liang, and S. Schneider, Eds. Cham: Springer International Publishing, 2020, pp. 480–501.

[22] A. Panwar and V. Bhatnagar, "Distributed Ledger Technology (DLT): The Beginning of a Technological Revolution for Blockchain," in *2nd International Conference on Data, Engineering and Applications (IDEA)*. Bhopal, India: IEEE, Feb. 2020, pp. 1–5.

[23] M. Rouse, "Merkle Tree (Blockchain Hash Tree)," Sep. 2023. [Online]. Available: https://www.techopedia.com/definition/32919/merkle-tree

[24] M. Kõlvart, M. Poola, and A. Rull, "Smart Contracts," in *The Future of Law and eTechnologies*, T. Kerikmäe and A. Rull, Eds. Cham: Springer International Publishing, 2016, pp. 133–147.

[25] H. Arslanian, "Ethereum," in *The Book of Crypto: The Complete Guide to Understanding Bitcoin, Cryptocurrencies and Digital Assets*, H. Arslanian, Ed. Cham: Springer International Publishing, 2022, pp. 91–98.

[26] W. Issa, N. Moustafa, B. Turnbull, N. Sohrabi, and Z. Tari, "Blockchain-Based Federated Learning for Securing Internet of Things: A Comprehensive Survey," *ACM Computing Surveys*, vol. 55, no. 9, pp. 191:1–191:43, 2023.

[27] D. Preuveneers, V. Rimmer, I. Tsingenopoulos, J. Spooren, W. Joosen, and E. Ilie-Zudor, "Chained Anomaly Detection Models for Federated Learning: An Intrusion Detection Case Study," *Applied Sciences*, vol. 8, no. 12, p. 2663, Dec. 2018.

[28] R. Al Mallah and D. López, "Blockchain-based Monitoring for Poison Attack Detection in Decentralized Federated Learning," in *2022 International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME)*, Nov. 2022, pp. 1–6.

[29] J. Kang, Z. Xiong, D. Niyato, Y. Zou, Y. Zhang, and M. Guizani, "Reliable Federated Learning for Mobile Networks," *IEEE Wireless Communications*, vol. 27, no. 2, pp. 72–80, Apr. 2020, conference Name: IEEE Wireless Communications.

[30] L. Feng, Y. Zhao, S. Guo, X. Qiu, W. Li, and P. Yu, "BAFL: A Blockchain-Based Asynchronous Federated Learning Framework," *IEEE Transactions on Computers*, vol. 71, no. 5, pp. 1092–1103, May 2022, conference Name: IEEE Transactions on Computers.

[31] D. Li, W. Peng, W. Deng, and F. Gai, "A Blockchain-Based Authentication and Security Mechanism for IoT," in *2018 27th International Conference on Computer Communication and Networks (ICCCN)*. Hangzhou: IEEE, Jul. 2018, pp. 1–6.

[32] G. Li, X. Ren, J. Wu, W. Ji, H. Yu, J. Cao, and R. Wang, "Blockchain-based mobile edge computing system," *Information Sciences*, vol. 561, pp. 70–80, Jun. 2021.

[33] H. Chen, S. A. Asif, J. Park, C.-C. Shen, and M. Bennis, "Robust Blockchained Federated Learning with Model Validation and Proof-of-Stake Inspired Consensus," 2021.

[34] S. Ruiz-Villafranca, J. Carrillo-Mondéjar, J. M. Castelo Gómez, and J. Roldán-Gómez, "MECInOT: a multi-access edge computing and industrial internet of things emulator for the modelling and study of cybersecurity threats," *The Journal of Supercomputing*, vol. 79, no. 11, pp. 11 895–11 933, Jul. 2023.

[35] M. Shayan, C. Fung, C. J. M. Yoon, and I. Beschastnikh, "Biscotti: A Blockchain System for Private and Secure Federated Learning," *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 7, pp. 1513–1525, Jul. 2021, conference Name: IEEE Transactions on Parallel and Distributed Systems.