UNIVERSIDAD DE MURCIA

ESCUELA INTERNACIONAL DE DOCTORADO

TESIS DOCTORAL

Intrusion detection based on federated learning for Internet of Things scenarios.

Detección de intrusiones basada en aprendizaje federado para escenarios del Internet de las Cosas.

D. **Enrique Mármol Campos**

2024

UNIVERSIDAD DE MURCIA

ESCUELA INTERNACIONAL DE DOCTORADO

TESIS DOCTORAL

Intrusion detection based on federated learning for Internet of Things scenarios.

Detección de intrusiones basada en aprendizaje federado para escenarios del Internet de las Cosas.

Autor: **D.** Enrique Mármol Campos

Director/es: D. Antonio Fernando Skarmeta Gómez
D. José Luis Hernández Ramos y
D.ª Aurora González Vidal

UNIVERSIDAD DE MURCIA

DECLARACIÓN DE AUTORÍA Y ORIGINALIDAD DE LA TESIS PRESENTADA EN MODALIDAD DE COMPENDIO O ARTÍCULOS PARA OBTENER EL TÍTULO DE DOCTOR

Aprobado por la Comisión General de Doctorado el 19-10-2022

D./Dña. Enrique Mármol Campos

doctorando del Programa de Doctorado en

Informatica

de la Escuela Internacional de Doctorado de la Universidad Murcia, como autor/a de la tesis presentada para la obtención del título de Doctor y titulada:

***Intrusion detection based on federated learning for Internet of Things scenarios./* Detección de intrusiones basada en aprendizaje federado para escenarios del Internet de las Cosas.**.

y dirigida por,

D./Dña. Antonio Fernando Skarmeta Gómez

D./Dña. José Luis Hernández Ramos

D./Dña. Aurora González Vidal

DECLARO QUE:

La tesis es una obra original que no infringe los derechos de propiedad intelectual ni los derechos de propiedad industrial u otros, de acuerdo con el ordenamiento jurídico vigente, en particular, la Ley de Propiedad Intelectual (R.D. legislativo 1/1996, de 12 de abril, por el que se aprueba el texto refundido de la Ley de Propiedad Intelectual, modificado por la Ley 2/2019, de 1 de marzo, regularizando, aclarando y armonizando las disposiciones legales vigentes sobre la materia), en particular, las disposiciones referidas al derecho de cita, cuando se han utilizado sus resultados o publicaciones.

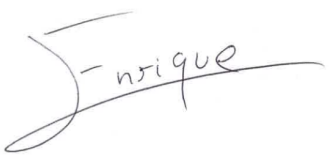
Además, al haber sido autorizada como compendio de publicaciones o, tal y como prevé el artículo 29.8 del reglamento, cuenta con:

- *La aceptación por escrito de los coautores de las publicaciones de que el doctorando las presente como parte de la tesis.*
- *En su caso, la renuncia por escrito de los coautores no doctores de dichos trabajos a presentarlos como parte de otras tesis doctorales en la Universidad de Murcia o en cualquier otra universidad.*

Del mismo modo, asumo ante la Universidad cualquier responsabilidad que pudiera derivarse de la autoría o falta de originalidad del contenido de la tesis presentada, en caso de plagio, de conformidad con el ordenamiento jurídico vigente.

En Murcia, a 26 de junio de 2024

Fdo.: Enrique Mármol Campos



Esta DECLARACIÓN DE AUTORÍA Y ORIGINALIDAD debe ser insertada en la primera página de la tesis presentada para la obtención del título de Doctor.

Información básica sobre protección de sus datos personales aportados	
Responsable:	Universidad de Murcia. Avenida teniente Flomesta, 5. Edificio de la Convalecencia. 30003; Murcia. Delegado de Protección de Datos: dpd@um.es
Legitimación:	La Universidad de Murcia se encuentra legitimada para el tratamiento de sus datos por ser necesario para el cumplimiento de una obligación legal aplicable al responsable del tratamiento. art. 6.1.c) del Reglamento General de Protección de Datos
Finalidad:	Gestionar su declaración de autoría y originalidad
Destinatarios:	No se prevén comunicaciones de datos
Derechos:	Los interesados pueden ejercer sus derechos de acceso, rectificación, cancelación, oposición, limitación del tratamiento, olvido y portabilidad a través del procedimiento establecido a tal efecto en el Registro Electrónico o mediante la presentación de la correspondiente solicitud en las Oficinas de Asistencia en Materia de Registro de la Universidad de Murcia

Agradecimientos

Aunque gran parte del trabajo de los artículos y de esta tesis ha sido gracias a mi esfuerzo, dedicación y resiliencia, todo esto hubiera sido imposible sin la ayuda y el apoyo de mis amigos, familiares, coautores, y mis codirectores. Todos ellos forman un sistema en serie, en que sin la presencia de alguno de ellos, nada hubiera salido adelante.

Primero de todo, quiero agradecer a mi director de tesis Antonio por darme la oportunidad de empezar este doctorado, por confiar en mí en un tema nuevo como es el federated learning sin apenas tener experiencia y ser un desconocido para él. Además, agradecer a mis codirectores José Luis y Aurora por guiarme y ayudarme en todo lo que han podido y más. Pero sobre todo, por su paciencia y comprensión por mis errores cometidos.

Agradecer a mis compañeros de la universidad, los comisionistas, Juan, Enrique, Belén, Fran, Toni, José Luis, Andrea, Ade, Alejandro, Adrián, Pablo, Alfredo, y Dani, por los buenos ratos dentro y fuera de la universidad. Por esas buenas comidas para desconectar del trabajo y coger fuerzas.

Gracias a mis compañeras de despacho, Valentina y Alicia. Sinceramente, las mejores compañeras que uno puede tener. ¿Qué habría hecho sin ellas? Valentina fue como una madre para mí durante mi estancia en Grecia, y Alicia ha sido con quien más tiempo he pasado en la universidad, ha sido mi mayor apoyo allí. A ambas les estoy agradecido por los ratos juntos, y por aguantarme en el despacho.

Quiero dar las gracias a mi familia por siempre estar ahí y darme su apoyo en todo momento. Aunque no supiesen qué hacía exactamente.

Por último, quiero agradecer a mis amigos Luis, Arturo, Juan Antonio, Arturo, Pablo, Alejandro, Joaquín, Velázquez, Darío, Chuche, Natalia, Mauro, Marta, Leo, Salva, Manu, Marisa, Eike, David, Ana, Lorena, Eva, y Sara. En este mundo aterrador, todo lo que tenemos son las conexiones que hacemos... Gracias a todos vosotros por vuestra compañía, aunque en algunos casos la distancia no lo permita, por vuestras bromas, por escucharme, animarme, etc. Básicamente, por todos los recuerdos de estos años, vuestro apoyo significa mucho para mí.

Summary

Motivation

The growing interconnectivity of devices through the Internet, especially via the Internet of Things (IoT), has not only increased the amount of data generated and processed but also the risks associated with cyberattacks. In this context, cybersecurity becomes crucial for protecting data, systems, and networks from unauthorized access or modifications, as defined by the National Institute of Standards and Technology (NIST). One of the most widely used approaches to address these risks is intrusion detection, which involves monitoring and analyzing events in systems or networks to identify potential security incidents. Intrusion Detection Systems (IDS) have automated this process, starting with signature-based approaches where events are compared to pre-existing patterns. However, with the expansion of connected devices and the sophistication of attacks, anomaly-based IDS have emerged by using Machine Learning (ML) techniques to identify unusual behaviors, allowing for the detection of unknown attacks and reducing false alarms.

Despite their benefits, many ML-based IDS still rely on centralized configurations, where data from multiple devices is collected on a single server to train the model. This raises concerns about data privacy and can cause delays due to centralized processing. To address these issues, Federated Learning (FL), a collaborative and decentralized approach, was introduced in 2016. FL allows devices to train ML models locally, without the need to share raw data with a central server or other devices. Instead of exchanging data, the devices send the trained model parameters (called weights) to a server, which aggregates them using an aggregation function and redistributes the aggregated result to continue the training process, thus preserving privacy.

The main objective of this thesis is to develop an FL setting for cyberattack detection in IoT scenarios. However, existing FL-based solutions have several limitations related to the heterogeneity of the clients' data, privacy and security. Many existing FL systems assume uniform data distribution among participants, which is unrealistic in real-world environments where data is neither independent nor identically distributed (non-iid). This mismatch can lead to a decrease in model effectiveness when applied to heterogeneous environments. During this thesis, the proposed scenarios are non-iid. Therefore, to address these problems related to non-iid scenarios, a combination of different rebalancing techniques, the implementation of alternative aggregation functions, and the optimization of model hyperparameters are used in order to reduce the impact of this data heterogeneity on model convergence.

While FL is designed as a privacy-preserving solution, it still faces significant privacy challenges. Model weights transmitted between clients and the server can be intercepted, and there is a risk that they could be used to infer private information. To protect against this, techniques such as Differential Privacy (DP) have been proposed, which add noise to model weights to ensure that private information cannot be inferred by attackers. However, the impact of these techniques on model convergence and the effectiveness of aggregation functions used in

FL remains to be explored in depth. This thesis proposes an exhaustive study of different DP methods and their impact on different aggregation functions.

In addition to privacy concerns, FL is vulnerable to poisoning attacks, where some clients may act maliciously and alter model weights or manipulate their datasets to affect the overall system's performance. These types of attacks aim to slow down model convergence or induce classification errors. To counter these attacks, this thesis proposes new aggregation strategies based on the Fast Fourier Transform (FFT), as well as methods for identifying and excluding malicious clients from the training process. These techniques aim to complement existing defense approaches, providing more robust protection against sophisticated attacks.

In summary, FL has emerged as a promising tool for improving cybersecurity, especially in the context of IDS in IoT environments. However, its implementation still faces significant challenges in terms of privacy, security, and the management of heterogeneous data. This field continues to evolve, with ongoing research focused on improving the robustness of FL systems and developing strategies to mitigate their vulnerabilities against external attacks.

Objectives

FL has become a notable approach for training ML models in the context of cyberattack detection, driving the development of the next generation of IDS. However, as analyzed in this thesis, much of the existing literature is based on unrealistic FL scenarios, where datasets are artificially divided among a fixed number of clients without considering the additional challenges involved in real-world implementations. Therefore, this thesis focuses on developing an anomaly-based IDS that leverages FL as a collaborative method for training ML models.

In this regard, several key challenges related to FL are addressed to improve the approaches proposed in the literature, such as data heterogeneity and the impact of applying different aggregation approaches. Privacy issues are also considered, which are mitigated through the application of various DP techniques. Additionally, a robust aggregation function has been developed, which is evaluated and compared with existing approaches to withstand various types of poisoning attacks, with the goal of improving the security and effectiveness of models trained using FL.

This thesis addresses the following objectives to achieve its main research goal:

- **O1:** To identify the current challenges and future trends in FL-enabled IDS.
- **O2:** To address data heterogeneity of common FL settings through the analysis of different aggregation functions and resampling techniques.
- **O3:** To protect the privacy of clients' datasets against inference attacks during the federated training.
- **O4:** To mitigate the impact of poisoning attacks in FL settings
- **O5:** To analyze the use of unsupervised learning techniques in the context of FL for cyberattack detection
- **O6:** To identify potential malicious clients during the federated training.

- **O7:** To alleviate the communication overhead in FL settings by reducing the number of clients.

Methodology and Results

The methodology of this thesis is oriented towards achieving the proposed objectives by implementing solutions at all levels that constitute FL. This includes addressing fundamental aspects such as data heterogeneity, privacy, and security in FL environments, ensuring these aspects are properly managed. To this end, work will be carried out at each stage of the FL process, from local data modification to model aggregation functions, ensuring that the proposed solutions offer significant improvements in accuracy, robustness, and privacy, aligning with the overall thesis objectives.

Mitigating Data Heterogeneity

One of the main challenges addressed in this thesis is data heterogeneity in FL environments, where data generated by IoT devices does not follow a uniform distribution, affecting the accuracy of the collaborative model. In this thesis, the proposed scenarios involve non-iid data distributions, which have a significant negative impact on weight convergence if the most common aggregation function, FedAvg, is used. To mitigate this issue, various data resampling techniques, aggregation functions, and model optimization tools were employed.

Among the resampling techniques, this thesis highlights the use of SMOTE-Tomek (Synthetic Minority Over-sampling Technique-Tomek), a technique that rebalances the classes in datasets, reducing inequality in data distribution among clients. This inequality is measured using Shannon entropy. SMOTE-Tomek helps improve model accuracy in scenarios where data follows a non-iid distribution.

At the server level, the results were evaluated by applying the FedAvg and Fed+ aggregation functions. While FedAvg is one of the most widely used functions in FL, its performance in non-iid scenarios is negatively affected. In contrast, Fed+ proved to be more effective in heterogeneous scenarios, as it allows for model personalization for each client, thereby improving performance. The thesis also includes an analysis of how model hyperparameter optimization can enhance its ability to handle data heterogeneity.

Protecting Privacy with Differential Privacy Techniques

FL is partly used for its ability to preserve data privacy, as it does not require direct sharing of information between devices. However, privacy risks persist, as attackers could infer sensitive details from the model parameters shared during training.

To address this concern, the thesis implemented DP techniques. These techniques add noise to model parameters before sharing them, making it harder for an attacker to extract useful information. In addition, we thoroughly tested the DP mechanisms to analyze their impact on final accuracy across several aggregation functions. Finally, the level of privacy added was compared to the accuracy obtained, in order to choose the best configuration for

each mechanism.

Improving Security in FL Environments

Finally, the thesis addresses poisoning attacks in FL, where malicious clients send false or manipulated data to corrupt the global model. These attacks can be devastating, as they affect all participants in the collaborative system.

To mitigate these attacks, a new aggregation function called FedRDF, based on FFT, was developed. This technique minimizes the impact of the weights sent by malicious clients during aggregation, thereby preserving model convergence. Additionally, FedRDF can detect the presence of malicious clients, allowing for the use of either FedAvg or FFT, depending on the situation, to maximize accuracy.

Furthermore, a framework called FLAegis was developed, which, unlike FedRDF, helps identify specific malicious clients during the training process. FLAegis monitors client behavior and detects suspicious patterns, allowing for the exclusion of these clients from aggregation in that round. This significantly improves the system's robustness against poisoning attacks.

Results

The results of this thesis demonstrate that FL is an effective solution for intrusion detection in IoT environments, significantly improving the accuracy of models trained in scenarios with heterogeneous data through techniques such as SMOTE-Tomek and Fed+. Additionally, the system showed high resistance to poisoning attacks thanks to the FLAegis framework and the FedRDF function, both of which protect the global model. Experiments demonstrated that our approaches are more effective than those proposed in the literature for resisting poisoning attacks. Furthermore, our study of different DP mechanisms revealed which methods offered the best balance between protection and accuracy. Although the use of DP introduced slight degradation in model performance, this loss was minimal compared to the privacy benefits provided.

Conclusions and Future Work

In summary, the thesis focuses on the application of FL for cyberattack detection, particularly IDS, in the context of IoT scenarios. After an exhaustive study of the methods proposed in the current literature, this thesis establishes that these methods are based on unrealistic assumptions, and that the privacy and security methods proposed for federated environments do not evaluate their impact on final accuracy or are based on assumptions that are difficult to achieve in real-world scenarios. To improve the performance of the proposed approaches, various techniques have been applied and analyzed to create more robust and privacy-preserving federated settings capable of managing non-iid data distributions.

Implementing these solutions represents a significant advancement in FL-enabled IDS and cyberattack detection in general. Additionally, the obtained results pave the way for future developments to improve the robustness of FL environments and extend their application to

a broader range of cybersecurity problems beyond cyberattack detection. The methods and techniques developed during the thesis are available as open-source code, facilitating replication and the development of new techniques. As for future work, new directions are proposed, such as improving client selection techniques, eliminating reliance on a central server, and extending the FedRDF and FLAegis algorithms for greater robustness and accuracy. The implementation of more complex unsupervised models and the integration of FL into LLM training is also considered.

Resumen

Motivación

La creciente interconectividad de dispositivos en Internet, especialmente a través del Internet de las Cosas (IoT), ha incrementado no solo la cantidad de datos generados y procesados, sino también los riesgos asociados a ciberataques. En este contexto, la ciberseguridad se vuelve crucial para proteger datos, sistemas y redes de accesos no autorizados o modificaciones, tal como lo define el Instituto Nacional de Estándares y Tecnología (NIST). Uno de los enfoques más utilizados para abordar estos riesgos es la detección de intrusiones, que implica monitorear y analizar eventos en sistemas o redes para identificar posibles incidentes de seguridad. Los Intrusion Detection Systems (IDS) han automatizado este proceso, comenzando con enfoques basados en firmas, donde los eventos se comparan con patrones preexistentes. Sin embargo, con la expansión de los dispositivos conectados y la sofisticación de los ataques, han surgido IDS basados en anomalías, que utilizan técnicas de Machine Learning (ML) para identificar comportamientos inusuales, permitiendo detectar ataques desconocidos y reduciendo las falsas alarmas.

A pesar de sus beneficios, muchos IDS basados en ML aún dependen de una configuración centralizada, donde los datos de múltiples dispositivos se recopilan en un único servidor para entrenar el modelo. Esto genera preocupaciones sobre la privacidad de los datos y puede causar retrasos debido al procesamiento centralizado. Para solucionar estos problemas, en 2016 se introdujo el concepto de Federated Learning (FL), un enfoque colaborativo y descentralizado que permite a los dispositivos entrenar modelos de ML localmente, sin necesidad de compartir los datos con un servidor central o con otros dispositivos. En lugar de intercambiar datos, los dispositivos envían los pesos o parámetros del modelo entrenado a un servidor que los agrega mediante una función de agregación y los redistribuye para continuar el entrenamiento, preservando así la privacidad.

El objetivo principal de esta tesis es desarrollar un entorno de FL para la detección de ciberataques en escenarios IoT. Sin embargo, las soluciones existentes basadas en el FL tienen varias limitaciones relacionadas con la heterogeneidad de los datos de los clientes, la privacidad y la seguridad. Muchos sistemas de FL existentes asumen una distribución uniforme de los datos entre los participantes, lo cual es poco realista en entornos reales donde los datos no son ni independientes ni idénticamente distribuidos (no-iid). Este desajuste puede provocar una disminución de la eficacia del modelo cuando se aplica a entornos heterogéneos. En esta tesis, los escenarios propuestos presentan una distribución de datos no-iid. Por lo tanto, para abordar estos problemas relacionados con los escenarios no-iid, se utiliza una combinación de diferentes técnicas de reequilibrio, la implementación de funciones de agregación alternativas y la optimización de los hiperparámetros del modelo con el fin de reducir el impacto de esta heterogeneidad de datos en la convergencia del modelo.

A pesar de que el FL se presenta como una solución que preserva la privacidad, aún se

enfrenta a importantes problemas de privacidad. Los pesos de los modelos, que se transmiten entre los clientes y el servidor, pueden ser interceptados, y existe el riesgo de que se utilicen para inferir información privada. Para protegerse contra esto, se han propuesto técnicas como el Differential Privacy (DP), que añade ruido a los pesos del modelo para garantizar que la información privada no pueda ser deducida por atacantes. Sin embargo, aún queda por explorar en profundidad el impacto que estas técnicas tienen en la convergencia de los modelos y en la efectividad de las funciones de agregación utilizadas en FL. En esta tesis se propone un estudio exhaustivo de diferentes métodos de DP y su impacto en diferentes funciones de agregación.

Además de las preocupaciones de privacidad, el FL es vulnerable a los ataques de envenenamiento, donde algunos clientes pueden actuar de manera maliciosa y alterar los pesos del modelo o manipular sus conjuntos de datos para afectar el rendimiento global del sistema. Este tipo de ataques buscan desacelerar la convergencia del modelo o inducir errores de clasificación. Para contrarrestar estos ataques, esta tesis propone nuevas estrategias de agregación basadas en la Transformada Rápida de Fourier (FFT), así como métodos para identificar y excluir clientes maliciosos del proceso de entrenamiento. Estas técnicas pretenden complementar los enfoques existentes de defensa, ofreciendo una protección más robusta contra ataques sofisticados.

En resumen, el FL ha emergido como una herramienta prometedora para mejorar la ciberseguridad, especialmente en el contexto de IDSs en entornos IoT. Sin embargo, su implementación aún enfrenta retos significativos en términos de privacidad, seguridad y la gestión de datos heterogéneos. Este campo continúa evolucionando, con investigaciones enfocadas en mejorar la robustez de los sistemas de FL y en desarrollar estrategias que mitiguen sus vulnerabilidades frente a ataques externos.

Objetivos

El FL ha surgido como un enfoque prometedor para el entrenamiento de modelos ML en el contexto de la detección de ciberataques, impulsando el desarrollo de la próxima generación de IDSs. Sin embargo, como se analiza en esta tesis, gran parte de la literatura existente se basa en escenarios poco realistas de FL, donde los conjuntos de datos se dividen de manera artificial entre un número fijo de clientes, sin tener en cuenta los desafíos adicionales que conlleva su implementación en situaciones reales. Por ello, esta tesis se centra en desarrollar un IDS basado en anomalías que aproveche el FL como método colaborativo para el entrenamiento de modelos de ML.

En este sentido, se abordan varios retos clave relacionados con el FL para mejorar los enfoques propuestos en la literatura, tales como la heterogeneidad de los datos y el impacto de la aplicación de distintos enfoques de agregación. También se consideran las cuestiones relacionadas con la privacidad, que se mitigan mediante la aplicación de diversas técnicas de DP. Además, se ha desarrollado una función de agregación robusta, la cual es evaluada y comparada con otros enfoques existentes para resistir distintos tipos de ataques de envenenamiento, con el fin de mejorar la seguridad y eficacia de los modelos entrenados mediante FL.

Esta tesis aborda los siguientes objetivos para alcanzar el objetivo principal de esta investigación:

- **O1**: Identificar los retos actuales y las tendencias futuras de los IDS basados en FL.

- **O2**: Abordar la heterogeneidad de los datos de las configuraciones del FL comunes mediante el análisis de diferentes funciones de agregación y técnicas de remuestreo.
- **O3**: Proteger la privacidad de los conjuntos de datos de los clientes contra ataques de inferencia durante el entrenamiento federado.
- **O4**: Mitigar el impacto de los ataques de envenenamiento en entornos de FL.
- **O5**: Analizar el uso de técnicas de aprendizaje no supervisado en el contexto de FL para la detección de ciberataques.
- **O6**: Identificar potenciales clientes maliciosos durante el entrenamiento federado.
- **O7**: Aliviar la sobrecarga de comunicación en entornos de FL reduciendo el número de clientes.

Metodología y resultados

La metodología de esta tesis está orientada a alcanzar los objetivos propuestos, implementando soluciones en todos los niveles que conforman el FL. Esto incluye abordar aspectos fundamentales como la heterogeneidad de los datos, la privacidad y la seguridad en entornos de FL, garantizando que se gestionen de manera adecuada. Para ello, se trabajará en cada etapa del proceso de FL, desde la modificación de datos locales hasta la función de agregación de modelos, asegurando que las soluciones propuestas ofrezcan mejoras significativas en precisión, robustez y privacidad, alineándose con los objetivos generales del proyecto.

Mitigación de la heterogeneidad de los datos

Uno de los principales retos que aborda la tesis es la heterogeneidad de los datos en entornos de FL, donde los datos generados por los dispositivos IoT no siguen una distribución uniforme, lo que afecta la precisión del modelo colaborativo. En esta tesis, los escenarios propuestos son en los que la distribución de los datos son no-iid, lo que supone un gran impacto negativo en la convergencia de los pesos si se usa la función de agregación más común FedAvg. Para mitigar este problema, se utilizaron diversas técnicas de remuestreo de datos, funciones de agregación, y se optimizaron los modelos usados.

Entre las técnicas de remuestreo, la tesis destaca el uso de SMOTE-Tomek (Synthetic Minority Over-sampling Technique-Tomek), una técnica que rebalancea las clases en los conjuntos de datos, permitiendo reducir la desigualdad en la distribución de los datos entre los clientes, cuya desigualdad se mide usando la entropía de Shannon. Esta técnica ayuda a mejorar la precisión del modelo en escenarios donde los datos siguen una distribución no-iid.

A nivel del servidor, se evaluaron los resultados al aplicar las funciones de agregación FedAvg y Fed+. Mientras que FedAvg es una de las funciones más utilizadas en el FL, en escenarios no-iid su precisión obtenida se ve afectada negativamente. En cambio, Fed+ demostró ser más efectivo en escenarios heterogéneos, ya que permite personalizar el modelo para cada cliente, mejorando su rendimiento. La tesis también incluye un análisis de cómo la optimización de los hiperparámetros del modelo puede mejorar su capacidad para manejar la heterogeneidad de los datos.

Protección de la privacidad mediante técnicas de privacidad diferencial

El FL se utiliza, en parte, por su capacidad para preservar la privacidad de los datos, ya que no requiere compartir directamente la información entre los dispositivos. Sin embargo, persisten riesgos de privacidad, ya que los atacantes podrían inferir detalles sensibles a partir de los parámetros del modelo que se comparten durante el entrenamiento.

Para abordar esta preocupación, en la tesis se aplicaron técnicas de DP. Estas técnicas añaden ruido a los parámetros del modelo antes de compartirlos, lo que dificulta a un atacante la extracción de información útil. Además, probamos exhaustivamente los mecanismos de DP para analizar su repercusión en la precisión final a través de varias funciones de agregación. Finalmente, se comparó el nivel de privacidad añadida con la precisión obtenida, con el fin de elegir la mejor configuración para cada mecanismo.

Mejora de la seguridad en entornos de FL

Finalmente, la tesis aborda los ataques de envenenamiento en el FL, donde los clientes maliciosos envían datos falsos o manipulados para corromper el modelo global. Este tipo de ataques puede ser devastador, ya que afecta a todos los participantes del sistema colaborativo.

Para mitigar estos ataques, se desarrolló una nueva función de agregación llamada FedRDF, basada en la FFT. Esta técnica permite minimizar el impacto de los pesos enviados por clientes maliciosos en la agregación, preservando en este caso la convergencia del modelo. Además, FedRDF permite también comprobar la presencia de clientes maliciosos, para dependiendo del caso en el que se encuentre, utilizar FedAvg o la FFT, para así maximizar la precisión obtenida.

Además, se desarrolló un framework llamado FLAegis, que, a diferencia de FedRDF, ayuda a identificar a los clientes maliciosos en específico durante el proceso de entrenamiento. FLAegis monitoriza el comportamiento de los clientes y detecta patrones sospechosos, permitiendo la exclusión de estos clientes en la agregación de esa ronda. Esto mejora significativamente la robustez del sistema ante ataques de envenenamiento.

Resultados

Los resultados de la tesis demuestran que el FL es una solución eficaz para la detección de intrusiones en entornos IoT, mejorando significativamente la precisión de los modelos entrenados en escenarios con datos heterogéneos mediante técnicas como SMOTE-Tomek y Fed+. Además, el sistema mostró alta resistencia a ataques de envenenamiento gracias al framework FLAegis y la función FedRDF, que protegen el modelo global. Los experimentos demostraron que nuestros métodos son más eficaces que los enfoques propuestos en la literatura para resistir los ataques de envenenamiento. Además, nuestro estudio de los diferentes mecanismos de DP reveló qué método ofrecía la mejor relación entre protección y precisión. Aunque el uso de DP introdujo una ligera degradación en el rendimiento del modelo, se observó que esta pérdida era mínima en comparación con los beneficios de privacidad que proporciona.

Conclusiones y trabajo futuro

En resumen, la tesis se centra en la aplicación del FL para la detección de ciberataques y, en particular, en los sistemas IDS, en el contexto de escenarios de IoT. Tras un exhaustivo estudio de los métodos propuestos por la literatura actual, esta tesis establece que estos métodos están basados en suposiciones irreales, en los que además los métodos proporcionados para mejorar la privacidad y seguridad del entorno federado no evalúan su impacto en la precisión final o también están basados en suposiciones que en un caso real es difícil de alcanzar. Para mejorar el rendimiento de los enfoques propuestos, se han aplicado y analizado diversas técnicas para crear entornos federados más robustos y que preserven mejor la privacidad, capaces de gestionar distribuciones de datos no-iid.

La implementación de estas soluciones representa un avance significativo en los IDS habilitados para el FL y en la detección de ciberataques en general. Además, los resultados obtenidos abren el camino para futuros desarrollos dirigidos a mejorar la robustez del entorno de FL y a extender su aplicación a una gama más amplia de problemas de ciberseguridad más allá de la detección de ciberataques. Los métodos y técnicas desarrollados durante la tesis están disponibles en código abierto, lo que facilita su replicación y el desarrollo de nuevas técnicas. En cuanto al trabajo futuro, se proponen nuevas direcciones, como mejorar la selección de clientes, eliminar la dependencia de un servidor central, y extender los algoritmos FedRDF y FLAegis para mayor robustez y precisión. También se plantea la implementación de modelos no supervisados más complejos y la integración del FL en el entrenamiento de LLMs.

Composing papers of the thesis

Principal works

This PhD thesis is a compilation of the following published articles, all co-authored by the PhD student.

- [1] **Campos, E. M.**, Saura, P. F., González-Vidal, A., Hernández-Ramos, J. L., Bernabe, J. B., Baldini, G., & Skarmeta, A. (2022). Evaluating Federated Learning for intrusion detection in Internet of Things: Review and challenges. Computer Networks, 203, 108661.

  The application of Machine Learning (ML) techniques to well-known Intrusion Detection Systems (IDS) is key to cope with increasingly sophisticated cybersecurity attacks through an effective and efficient detection process. In the context of the Internet of Things (IoT), most ML-enabled IDS approaches use centralized approaches where IoT devices share their data with data centers for further analysis. To mitigate privacy concerns associated with centralized approaches, in recent years the use of Federated Learning (FL) has attracted significant interest in different sectors, including healthcare and transport systems. However, the development of FL-enabled IDS for IoT is in its infancy, and still requires research efforts from various areas, in order to identify the main challenges for the deployment in real-world scenarios. In this direction, our work evaluates an FL-enabled IDS approach based on a multiclass classifier considering different data distributions for the detection of different attacks in an IoT scenario. In particular, we use three different settings that are obtained by partitioning the recent ToN_IoT dataset according to IoT devices' IP addresses and types of attacks. Furthermore, we evaluate the impact of different aggregation functions according to such settings by using the recent IBMFL framework as FL implementation. Additionally, we identify a set of challenges and future directions based on the existing literature and the analysis of our evaluation results.

- [2] Ruzafa-Alcázar, P., Fernández-Saura, P., **Mármol-Campos, E.**, González-Vidal, A., Hernández-Ramos, J. L., Bernal-Bernabe, J., & Skarmeta, A. F. (2021). Intrusion detection based on privacy-preserving federated learning for the industrial IoT. IEEE Transactions on Industrial Informatics, 19(2), 1145-1154.

  FL has attracted significant interest given its prominent advantages and applicability in many scenarios. However, it has been demonstrated that sharing updated gradients and weights during the training process can lead to privacy concerns, especially in the context of IoT devices that monitor environments involving people, potentially revealing their behaviors and therefore personal information. Our work provides a comprehensive evaluation of Differential Privacy (DP) techniques, which are applied during the training of an FL-enabled IDS for Industrial IoT (IIoT). Unlike previous approaches, we deal with non-iid data over the recent ToN_IoT dataset and compare the accuracy obtained considering different privacy requirements and aggregation functions, namely FedAvg and

the recently proposed Fed+. According to our evaluation, the use of Fed+ in our setting provides similar results even when noise is included in the federated training process.

- [3] Matheu, S. N., **Mármol, E.**, Hernández-Ramos, J. L., Skarmeta, A., & Baldini, G. (2022). Federated Cyberattack Detection for Internet of Things-Enabled Smart Cities. Computer, 55(12), 65-73.

  With the increasing digitization of our surrounding environment, the effective and efficient detection of cyberattacks is key to realizing trustworthy smart cities. In this context, the use of Artificial Intelligence (AI) has aroused a significant interest in dealing with increasingly sophisticated cyberattacks. However, the detection of such threats is typically based on analyzing large amounts of network traffic data, which can lead to privacy issues for citizens. Addressing this issue, this work proposes an FL approach to the identification of cyberattacks in the context of IoT-enabled smart cities. Our work integrates the Manufacturer Usage Description standard as a prevention/mitigation approach based on network rules with FL component for the identification of several cyberattacks. We demonstrate the feasibility of our approach under different FL settings using a dataset derived from network traffic of real IoT devices with an accuracy value of around 90%.

- [4] **Campos, E. M.**, Hernandez-Ramos, J. L., Vidal, A. G., Baldini, G., & Skarmeta, A. (2024). Misbehaviour detection in intelligent transportation systems based on federated learning. Internet of Things, 101127.

  Misbehavior detection represents a key security approach in vehicular scenarios to identify attacks that cannot be detected by traditional cryptographic mechanisms. In this context, the application of ML techniques has been widely considered to identify increasingly sophisticated misbehavior attacks. However, most of the proposed approaches are based on centralized settings, which could pose privacy issues, as well as an increased latency leading to severe consequences in the vehicular environment where real-time and scalability requirements are challenging. To address this issue, we propose a collaborative learning approach based on FL for vehicles' misbehavior detection. We use the reference misbehavior dataset VeReMi, which is re-balanced by applying the SMOTE-Tomek technique. We carry out a thorough evaluation considering different balancing settings and the number of nodes. The evaluation results overcome recent state-of-the-art approaches, with an overall accuracy of 93% using an optimized multilayer perceptron (MLP) for multiclass classification.

Secondary papers

In addition to the previous papers, the PhD student also collaborated in the elaboration of the following papers:

- [5] **Campos, E. M.**, Vidal, A. G., Ramos, J. L. H., & Skarmeta, A. (2023, May). Federated Transfer Learning for Energy Efficiency in Smart Buildings. In IEEE INFOCOM 2023-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS) (pp. 1-6). IEEE. **Current status:** Published.

  In nodaways systems, the generation of AI-based energy consumption models in buildings need IoT deployments to gather data that are well-maintained in time. Indeed, most existing buildings lack the appropriate equipment to obtain all the data required to create such models. Furthermore, energy consumption data can be correlated with additional information about buildings' users that could raise privacy concerns. Based on such aspects, we propose a Federated Transfer Learning (FTL) framework to handle these buildings' data without compromising any private information in which a set of buildings are clustered according to certain characteristics. On the one hand, our works leverage the properties of FL to train an energy forecasting model using a small portion of the available buildings respecting their privacy. On the other hand, we transfer the model to the rest of the buildings by using Transfer Learning (TL) for fine-tuning with the specific data of each building, getting a higher accuracy. We extensively evaluate our approach, and demonstrate it improves the results of alternative scenarios where FL and TL are used separately.

- [6] Hernandez-Ramos, J. L., Karopoulos, G., Chatzoglou, E., Kouliaridis, V., **Marmol, E.**, Gonzalez-Vidal, A., & Kambourakis, G. (2023). Intrusion Detection based on Federated Learning: a systematic review. arXiv preprint arXiv:2308.09522. **Current status:** *Under revision.*

  The evolution of cybersecurity is undoubtedly associated and intertwined with the development and improvement of AI. As a key tool for realizing more cyber-secure ecosystems, IDS have evolved tremendously in recent years by integrating ML techniques for the detection of increasingly sophisticated cybersecurity attacks hidden in big data. However, these approaches have traditionally been based on centralized learning architectures, in which data from end nodes are shared with data centers for analysis. Recently, the application of FL in this context has attracted great interest in coming up with collaborative intrusion detection approaches where data does not need to be shared. Due to the recent rise of this field, this work presents a complete, contemporary taxonomy for FL-enabled IDS approaches that stem from a comprehensive survey of the literature in the time span from 2018 to 2022. Precisely, our discussion includes an analysis of the main ML models, datasets, aggregation functions, as well as implementation libraries, which are employed by the proposed FL-enabled IDS approaches. On top of everything else, we provide a critical view of the current state of the research around this topic and describe the main challenges and future directions based on the analysis of the literature and our own experience in this area.

- [7] **Campos, E. M.**, Hernandez-Ramos, J. L., Vidal, A. G., Baldini, G., Skarmeta, A. (2024). Misbehavior detection in intelligent transportation systems based on federated learning. Internet of Things, 101127. **Current status:** *Under revision.*

FL has become an attractive approach to collaboratively train Machine Learning (ML) models while data sources' privacy is still preserved. However, most of the existing FL approaches are based on supervised techniques, which could require resource-intensive activities and human intervention to obtain labelled datasets. Furthermore, in the scope of cyberattack detection, such techniques are not able to identify previously unknown threats. In this direction, this work proposes a novel unsupervised FL approach for the identification of potential misbehavior in vehicular environments. We leverage the computing capabilities of public cloud services for model aggregation purposes, and also as a central repository of misbehavior events, enabling cross-vehicle learning and collective defense strategies. Our solution integrates the use of Gaussian Mixture Models (GMM) and Variational Autoencoders (VAE) on the VeReMi dataset in a federated environment, where each vehicle is intended to train only with its own data. Furthermore, we use Restricted Boltzmann Machines (RBM) for pre-training purposes, and Fed+ as an aggregation function to enhance the model's convergence. Our approach provides better performance (more than 80%) compared to recent proposals, which are usually based on supervised techniques and artificial divisions of the VeReMi dataset. it would entail.

- [8] **Campos, E. M.**, Vidal, A. G., Ramos, J. L. H., & Skarmeta, A. (2024). FedRDF: A Robust and Dynamic Aggregation Function against Poisoning Attacks in Federated Learning. arXiv preprint arXiv:2402.10082. **Current status:** *Under revision.*

  FL represents a promising approach to typical privacy concerns associated with centralized ML deployments. Despite its well-known advantages, FL is vulnerable to security attacks such as Byzantine behaviors and poisoning attacks, which can significantly degrade model performance and hinder convergence. The effectiveness of existing approaches to mitigate complex attacks, such as median, trimmed mean, or Krum aggregation functions, has been only partially demonstrated in the case of specific attacks. Our study introduces a novel robust aggregation mechanism utilizing the Fourier Transform (FT), which is able to effectively handle sophisticated attacks without prior knowledge of the number of attackers. Employing this data technique, weights generated by FL clients are projected into the frequency domain to ascertain their density function, selecting the one exhibiting the highest frequency. Consequently, malicious clients' weights are excluded. Our proposed approach was tested against various model poisoning attacks, demonstrating superior performance over state-of-the-art aggregation methods.

Contents

1 Introduction and motivation **1**

2 Objectives **7**

3 State of the Art **11**

3.1 Background . . . 11

3.1.1 ML models . . . 11

3.1.2 Aggregation functions . . . 13

3.1.3 Datasets . . . 16

3.2 Related work . . . 18

3.2.1 FL-enabled cyberattack detection . . . 18

3.2.2 Data heterogeneity . . . 19

3.2.3 Privacy . . . 21

3.2.4 Security . . . 22

4 Methodology **25**

4.1 Mitigating the impact of data heterogeneity . . . 26

4.1.1 Dataset resampling . . . 26

4.1.2 Analysis of alternative aggregation functions for data heterogeneity . . . 27

4.1.3 Optimization of the Model . . . 28

4.1.4 Results . . . 29

4.2 Addressing privacy concerns through DP techniques . . . 31

4.2.1 Analysis of the impact of different aggregation functions . . . 32

4.2.2 Study on the trade-off obfuscation-accuracy . . . 32

4.2.3 Results . . . 33

4.3 Enhancing FL settings' security . . . 34

4.3.1 FedRDF: a robust aggregation function for FL . . . 35

4.3.2 Identification of byzantine clients . . . 36

4.3.3 Results . . . 38

5 Conclusions and Future work **43**

1 Introducción y motivación **47**

2 Objetivos **53**

3 Estado del arte **57**

3.1 Trasfondo . . . 57

3.1.1 Modelos de ML . . . 57

3.1.2 Funciones de agregación . . . 60

3.1.3 Bases de datos . . . 61

3.2 Trabajos relacionados . . . 64

3.2.1 FL para la deteccion de ciberataques . . . 65

3.2.2 Heterogeneidad de los datos . . . 66

3.2.3 Privacidad . . . 68

3.2.4 Seguridad . . . 69

4 Metodología **71**

4.1 Mitigando el impacto de la heterogeneidad de los datos . . . 72

4.1.1 Remuestreo de las bases de datos . . . 72

4.1.2 Análisis de funciones de agregación alternativas para la heterogeneidad de los datos . . . 73

4.1.3 Optimizacion del Modelo . . . 74

4.1.4 Resultados . . . 75

4.2 Problemas de privacidad mediante DP . . . 78

4.2.1 Analisis del impacto en diferentes funciones de agregación . . . 78

4.2.2 Estudio sobre el equilibrio ofuscación-precisión . . . 79

4.2.3 Resultados . . . 80

4.3 Mejorando los ajustes de la seguridad en FL . . . 80

4.3.1 FedRDF: una función de agregación robusta para el FL . . . 81

4.3.2 Identificación de clientes bizantinos . . . 83

4.3.3 Resultados . . . 85

5 Conclusiones y trabajo futuro **91**

6 Publications composing PhD Thesis **111**

6.1 Evaluating Federated Learning for intrusion detection in Internet of Things: Review and challenges . . . 111

6.2 Intrusion Detection Based on Privacy-Preserving Federated Learning for the Industrial IoT . . . 114

6.3 Federated Cyberattack Detection for Internet of Things-Enabled Smart Cities . 116

6.4 Misbehavior detection in intelligent transportation systems based on federated learning . . . 118

List of Figures

1.1 Pictorial description of the FL training process. 3

4.1 Visual description of the different areas addressed through this thesis to reach the described objectives related to data heterogeneity, security, and privacy. . . . 25

4.2 Comparison of the different scenarios with various data distributions on the ToN_IoT dataset 30

4.3 Comparison of various scenarios using the original VeReMi dataset versus the balanced VeReMi dataset obtained through SMOTE-Tomek 31

4.4 Fed+ accuracy results for all perturbation mechanisms. 34

4.5 Visual description to calculate the FFT in FedRDF. 36

4.6 Visual description of the FLAegis process. 38

4.7 Results of different robust aggregation functions against the min-max attack. . . 40

4.8 Comparison of FedRDF with FedAvg and FFT. 40

4.9 Comparison of our framework with SignGuard and FoolsGold against different attacks. 41

4.10 Comparison of our framework with SignGuard and FoolsGold using FFT against different attacks. 42

1.1 Descripción gráfica del proceso de entrenamiendo del FL. 49

4.1 Descripción visual de las diferentes áreas abordadas a través de esta tesis para alcanzar los objetivos descritos relacionados con la heterogeneidad, la seguridad y la privacidad de los datos. 71

4.2 Comparación de los distintos escenarios con varias distribuciones de datos en el conjunto de datos ToN_IoT 76

4.3 Comparación de varios escenarios utilizando el conjunto de datos VeReMi original frente al conjunto de datos VeReMi equilibrado obtenido mediante SMOTE-Tomek. 77

4.4 Resultados de precisión de Fed+ para todos los mecanismos de perturbación. . . 81

4.5 Descripción visual para calcular la FFT en FedRDF. 83

4.6 Descripción visual del proceso de FLAegis. 85

4.7 Resultados de diferentes funciones de agregación robustas contra el ataque min-max. 87

4.8 Comparación de FedRDF con FedAvg y FFT. 87

4.9 Comparación de nuestro framework con SignGuard y FoolsGold frente a distintos ataques. 88

4.10 Comparación de nuestro framework con SignGuard y FoolsGold utilizando FFT contra diferentes ataques. 89

List of Tables

2.1 Description of the objectives of this thesis. 9

3.1 Description of several aggregation functions often considered in FL. 15

3.2 Description of the characteristics of the cyberattack-oriented datasets used in this thesis. 17

4.1 Table of the PCC of the different DP mechanisms for different values. 35

2.1 Descripción de los objetivos de esta tesis. 55

3.1 Descripción de varias funciones de agregación frecuentemente consideradas en FL. 62

3.2 Descripción de las características de los conjuntos de datos orientados a ciberataques utilizados en esta tesis. 64

4.1 Tabla del PCC de los diferentes mecanismos de DP para diferentes valores. . . . 82

Chapter 1

Introduction and motivation

The current Internet connects a vast amount of computer systems and network infrastructure. This increasing online connectivity is being realized through the integration of different types of everyday devices, including sensors and actuators composing the Internet of Things (IoT) [9] fostering a plethora of data-driven services, such as environmental monitoring, smart homes or advanced healthcare applications. For the realization of such services, a huge amount of data, measured in terabytes per second, is generated, processed, exchanged, and stored by different services on the Internet. Such hyperconnectivity has significantly expanded the attack surface to be exploited by potential cyberattackers [10]. In this direction, cybersecurity emerges as the backbone for corporations, governments, and individuals, enabling them to secure data, expand their businesses, and maintain privacy [11]. According to the National Institute of Standards and Technology (NIST), cybersecurity is defined as *the process of implementing protective measures and policies to safeguard data, programs, servers, and network infrastructures from unauthorized access or modification*[1].

One of the most well-known approaches in cybersecurity is represented by intrusion detection, which is usually referred as "*the process of monitoring the events occurring in a computer system or network and analyzing them for signs of possible incidents*" [12]. **Intrusion Detection Systems** (IDSs) are deployed to automate such process [13]. Initial IDSs were usually based on *signature-based* approaches in which such monitored events (e.g., related to network traffic) were compared with previously stored information. However, with the increasing amount of interconnected devices, as well as the emergence of more sophisticated attacks, Machine Learning (ML) became as a key component of *anomaly-based* IDSs to detect potential unusual behaviors or deviations from normal patterns [14, 15]. ML is a form of Artificial Intelligence (AI) that can automatically extract valuable insights from large datasets [16]. For this reason, the application of ML techniques for cyberattack detection has garnered significant interest in recent years across various fields, including the IoT [17].

The development of anomaly-based (or ML-based) IDSs is intended to improve the effectiveness of previous approaches by enabling the detection of unknown cyberattacks. Indeed,

[1] `https://csrc.nist.gov/glossary/term/cybersecurity`

they have been proved to reduce false positive rates, and adapt to evolving attack patterns [14]. Despite their well-known advantages, the deployment of anomaly-based IDSs is usually based on centralized settings, where a single entity collects network traffic data from various systems to train a specific ML model. Consequently, this entity has access to the entire systems' network traffic and local data used in the training process, potentially leading to privacy issues concerning the enforcement of existing data protection legal instruments, such as the well-known General Data Protection Regulation (GDPR) [18]. Such centralized scenarios could also present several issues around the delay associated with the centralized reasoning process (typically conducted in cloud data centers). In this direction, recent works [19–21] highlight the importance of protecting clients' personal information and the necessity of developing distributed ML approaches. These works also discuss the limitations of centralized systems, such as limited communication bandwidth, intermittent network connectivity, and strict delay constraints [20].

To tackle the issues associated with traditional centralized ML approaches, **Federated Learning** (FL) was introduced in 2016 by [22] as a decentralized and collaborative learning approach. FL is composed of various data sources (referred to as *clients* or *parties*) and a central entity known as the *server* or *aggregator*. The clients are responsible for collecting and storing their own dataset, which is unique and inaccessible to the other clients. These clients train a ML model with their own dataset. During training, the model adjusts its internal variables (known as parameters) to fit the dataset provided for accurate classification. In this thesis, the models used are Logistic Regression [23] (LR) and several Neural Networks [24] (NN). For LR, these parameters are the coefficients that multiply the input variables in the logistic equation, which can be referred to as weights. For NN, the variables are the weights that connect the different neurons between the model's layers. Therefore, for the sake of simplicity, we refer to these variables as *weights* throughout the reminder of this document. The local training results in a set of weights, which are sent to server. Then, this entity *aggregates* the weights sent by the different clients. Once the weights are aggregated, the server sends the result of such aggregation to the clients to continue their training with the aggregated weights.

The overall FL training process is reflected in the four main steps depicted in Fig. 1.1. During step (1), the clients begin the training of the ML model using their own data. Next, after several training iterations called *epochs*, in step (2), the clients send the weights produced during the training to the server. Then, the server, in step (3), once it has received all the weights, aggregates them using what is called an *aggregation function* F. Then, when the aggregation has been completed, the server sends the resulting weights back to the clients to continue their training step (4). All this process is called a *round*. Finally, the process continues for a predefined number of rounds or until the weights converge. During this process, the clients' local datasets are not shared; hence, FL provides a privacy-preserving a decentralised approach to training ML models.

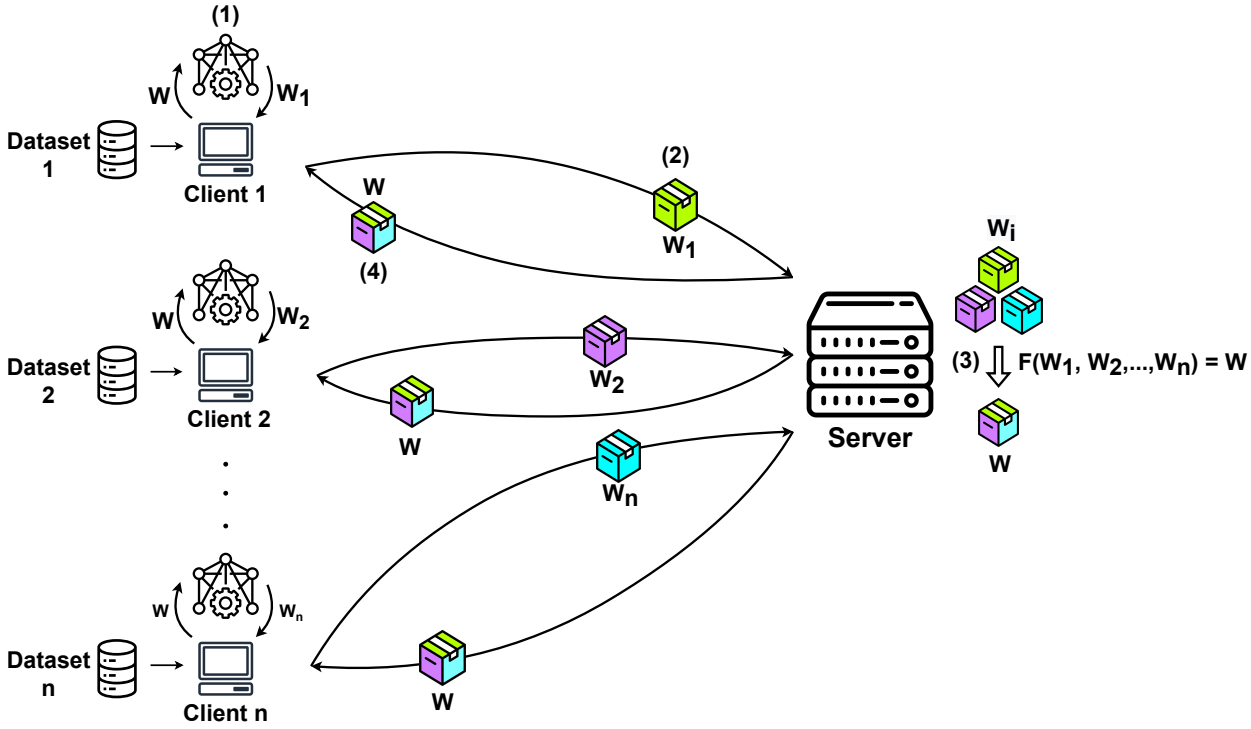


Figure 1.1. Pictorial description of the FL training process.

In the context of anomaly-based IDSs, FL provides several advantages, including the reduction of computational burden on the central processing server, safeguarding data privacy and optimizing bandwidth usage, [25, 26]. Indeed, as we analyzed during this thesis [6], FL has become a popular approach for the next generation of IDSs [25, 27, 28]. In this context, this thesis specifically focuses on FL-enabled IDS in IoT scenarios due to the widespread adoption of IoT in areas such as the Internet of Vehicles, smart cities, and Cyber-Physical Systems over the past few years [28]. Additionally, during the development of the thesis, we also delved into *misbehavior detection*, which can be seen as a specific type of intrusion detection in the vehicular context. In particular, it refers to the detection of vehicles transmitting false information that cannot be detected by typical cryptographic mechanisms [29, 30]. Therefore, we call Misbehavior Detection Systems (MDS) to the systems automating misbehavior detection approaches. In such scenario, FL mitigates the privacy concerns related to the continuous sharing of vehicles' position data, while it still provides an approach to collaboratively train a model for detecting misbehaving vehicles [31].

While the primary objective of this thesis was initially around the application of FL for cyberattack detection (and, in particular, in IoT scenarios), our exhaustive analysis carried out during the course of this PhD [1, 6] identified several challenges associated with the deployment of FL settings in terms of data/device heterogeneity, security and privacy concerns, among others. Furthermore, our analysis revealed that most of the existing works around FL-enabled IDSs are based on supervised learning techniques [32–34] that require labelled datasets for training. The labelling process often requires human intervention, making it a resource-

intensive and time-consuming task to obtain labelled examples necessary for achieving proper generalization [35, 36]. In addition, in the context of FL-enabled IDSs, many of the proposed approaches suffer from unrealistic data distributions among participants (as exposed in [6]), obsolete datasets, or rely on binary classification approaches, where traffic data is merely classified as either attack or benign [37]. These artificial distributions led to unrealistic results since real scenarios are characterised by non-independent and non-identically distributed (non-iid) data. One of the main approaches to deal with such data heterogeneity is the application of alternative aggregation functions. Indeed, the common aggregation approach is FedAvg [22], which consists of applying the average to the weights received in a certain training round. As demonstrated by several works [38], the use of FedAvg in non-iid data leads to a decrease in the model convergence. The performance degradation can primarily be attributed to the weight divergence of local models, which occurs because the aggregated weights do not fit well when applied to the clients' data. To cope with this aspect, alternative aggregation functions such as Fed+ [39] and FedProx [40] are intended to mitigate the impact of non-iid data distributions. In this thesis, we quantify the impact of using different aggregation functions and apply resampling techniques (such as SMOTE-Tomek [41]) to reduce the statistical heterogeneity among the clients, in order to address such concern in FL-enabled IDSs.

Moreover, despite FL is mainly proposed as a privacy-preserving approach for training a ML model without compromising the privacy of clients' datasets, it still faces significant privacy concerns. One major issue is that clients' weights can be intercepted during communication between the server and the clients. Indeed, in a typical FL setting, the server is usually able to access the weights uploaded by the clients throughout the rounds. Consequently, these weights could be used to launch various attacks aimed at inferring private information from the training data [42]. Therefore, the implementation of robust techniques to protect clients' privacy is crucial. In centralized ML, Differential Privacy (DP) [43] can be employed to address these privacy concerns. DP obfuscates model updates with the addition of noise, thus providing statistical privacy guarantees against adversaries. However, the existing literature lacks an in-depth analysis of the impact of existing DP methods on the convergence of FL and the performance of different aggregation functions. During this thesis, we conducted a comprehensive analysis of contemporary DP techniques, and their impact on the model performance by considering different aggregation functions. Additionally, we also examine the trade-off between the amount of noise introduced by DP, and the resulting loss in accuracy. By evaluating this trade-off, we aim to select the optimal DP mechanism that ensures the highest level of privacy for clients' datasets while maintaining acceptable accuracy levels.

In the context of IDS and MDS, the main purpose of FL is to collaboratively train a model to detect cyberattacks. However, FL itself is susceptible to poisoning attacks [44] since some parties can act as byzantine clients, i.e., clients which act maliciously and try to harm the model convergence. The purpose of these byzantine clients is to produce a delay in the convergence of the model by sending fake weights to the server (model poisoning), or altering

the clients' dataset directly (data poisoning). Then, the benign clients retrain with corrupted aggregated weights that could lead to a misclassification of the samples after the federated training. Indeed, previous studies confirm that FL is susceptible to poisoning attacks, especially emphasizing the impact when FedAvg is used as an aggregation approach [45]. In this context, one of the objectives of this thesis is to protect the federated environment against such attacks. In particular, two complementary strategies are designed and implemented. Firstly, we design and implement a novel aggregation approach based on the Fast Fourier Transform (FFT) [46], as an alternative to well-known robust aggregation techniques, such as the median, trimmed mean [47] or Krum [48]. Our approach is intended to address some of the weaknesses of these methods, which usually assume the number of attackers is known, and do not provide protection against sophisticated model poisoning attacks where several clients collude [49]. Secondly, we explore new methods for identifying such malicious clients, so these compromised nodes can be discarded for future training rounds. Indeed, while robust aggregation functions could mitigate the impact of poisoning attacks, the identification of specific compromised devices could help in the development of potential mitigation techniques and defense strategies [50].

In summary, this thesis aims to develop an FL setting for cyberattack detection in IoT scenarios. In addition to the analysis of different ML models, datasets and aggregation functions in this context, we also address pressing challenges related to data heterogeneity, privacy, and security to enhance the overall robustness and privacy of FL systems. This work is organised as a compendium of four high-impact research papers, which form the main body of the thesis. The information presented is available in both English and Spanish. Finally, the structure of this thesis is described as follows: this first section introduces the context and motivation of the topic of this thesis. The second section describes the main objectives of the proposed research and their relationship with the four publications composing the thesis. The third section exposes the baseline models, techniques, and related work around the thesis' topic. The fourth section describes the methodology followed to address the challenges previously described and achieved the proposed objectives proposed. The fifth section concludes this thesis and details the future topics of our work.

Chapter 2

Objectives

As already described in the previous section, FL has emerged as a promising approach for training ML models in the context of cyberattack detection to build the next generation of IDSs. Nonetheless, as already analyzed during this thesis [1, 6], most of the existing literature is based on unrealistic FL settings where datasets are artificially divided among a certain number of clients without considering additional challenges associated with the deployment of FL scenarios. Therefore, this thesis aims to develop a realistic anomaly-based IDS by leveraging federated learning as a collaborative method for training various ML models. In that sense, we address several significant challenges around FL itself, by considering data heterogeneity and the impact of applying different aggregation approaches, and privacy concerns, which are mitigated through the application of different DP techniques. In addition, we develop a robust aggregation function, which is evaluated and compared with existing approaches against different poisoning attacks.

To achieve these goals, the following objectives are defined:

- **O1:** To identify the current challenges and future trends in FL-enabled IDS.
- **O2:** To address data heterogeneity of common FL settings through the analysis of different aggregation functions and resampling techniques.
- **O3:** To protect the privacy of clients' datasets against inference attacks during the federated training.
- **O4:** To mitigate the impact of poisoning attacks in FL settings
- **O5:** To analyze the use of unsupervised learning techniques in the context of FL for cyberattack detection
- **O6:** To identify potential malicious clients during the federated training.
- **O7:** To alleviate the communication overhead in FL settings by reducing the number of clients.

Table 2.1 presents the correlation between the objectives and the papers composing this thesis. This table gives a brief description of the approach to achieve each objective.

Objective	Publications	Approach
O1	**[1]**, [6]	A comprehensive review of the current literature on the challenges associated with the FL in IDS systems is conducted. Some of the most critical identified include privacy issues (e.g., related to inference attacks), security concerns (including poisoning attacks), and data heterogeneity among clients.
O2	**[1–4]**, [5, 7, 8]	This objective is common in several papers. To address non-iid scenarios where clients' data is heterogeneously distributed, several methods are applied in these works, including local resampling techniques, alternative aggregation functions, and customization of ML models.
O3	**[2]**	An FL setting could suffer from inference attacks, where information regarding clients' data can be inferred from the exchange of weights throughout the training rounds. To prevent this issue, our work presents an analysis of different DP techniques to obfuscate such weights, thereby preventing information leakage. The work also evaluates the trade-off between different privacy levels and the model accuracy to identify the optimal DP method technique.
O4	[8]	A novel aggregation function based on the Fast Fourier Transform (FFT) is proposed to mitigate the impact of poisoning attacks in FL settings with the presence of byzantine clients.
O5	[7]	A novel technique based on Gaussian Mixture Models (GMMs) and Variational Autoencoders (VAEs) for unsupervised FL training of a misbehavior detection model.
O6	**[3]**	We analyze the use of a recent standard for discarding misbehaving devices during the FL training process. The exclusion of these clients is intended to protect the FL setting during the training.

Objective	Publications	Approach
O7	**[4]**, [5]	We propose a client selection approach to reduce the required computation and communication overhead in FL settings. The proposed selection approach is based on the length of the clients' dataset, and the trade-off between accuracy and time required for the training process.

Table 2.1

Description of the objectives of this thesis.

Chapter 3

State of the Art

This section presents the state-of-the-art related to the core of this thesis that is divided into two parts: Background and Related Work. Section 3.1 provides a brief description of the fundamental components of FL, including specific models, aggregation functions, and datasets employed throughout this thesis. Then, Section 3.2 provides a succinct overview of current research on FL for cyberattack detection. Furthermore, it presents an analysis of related studies addressing some of the main FL areas explored in this thesis: data heterogeneity, security, and privacy concerns in such settings. Such analysis is intended to provide a clear overview of the relationship between existing works with the objectives and results of this thesis. It should be noted that an extended description about these concepts can be found at [6].

3.1 — Background

The main components of an FL setting are represented by the ML model for training, the datasets employed by the clients, and the aggregation function implemented by the server. This section provides a detailed analysis of these key elements and the description of the specific alternatives employed in this research.

3.1.1. ML models

The main goal of using FL for cyberattack detection is to collaboratively train an ML model to distinguish between normal or intended behavior, and specific attacks. Depending on the nature of the training data, ML models can be mainly classified into two types: supervised and unsupervised.

Supervised learning

Supervised learning techniques depend on labelled data for model training. The training dataset includes inputs paired with their correct outputs, which help the model learn by fine-tuning its parameters. This dataset instructs the model on the relationship between inputs and outputs, allowing it to predict accurately when presented with new data. Some of the most commonly used supervised learning techniques are Logistic Regression (LR) [23], Decision Trees (DT) [51], Random Forest (RF) [52], Neural Networks (NN) [24], and Support Vector Machines (SVM) [53]. During the development of this thesis, the models used are LR and several types of NN: Multilayer Perceptron (MLP) [54], Long Short Term Memory (LSTM) [55], and Convolutional Neural Network (CNN) [56]. The main rationale behind the use of such models is that the focus of this thesis is not on ML models employed, but rather on specific aspects of FL deployments, such as the impact of using different aggregation functions and data heterogeneity. Consequently, we use simple and well-known models to prioritize these critical aspects in FL. Specifically, LR was chosen in our initial works [1, 2] due to its straightforward definition and the fact that its weights can be aggregated in FL directly unlike the hyperplanes of SVM, or the trees structures of the RF, and DT. Regarding NNs, besides their parameter structure is less complex than other models (similar to LR), in the current literature, as described in [6], most related works use NN models. Hence, this allows for more precise comparisons between their results and ours.

Based on previous aspects, we provide a succinct description of the models employed throughout the thesis. **LR** is a statistical method used for classification tasks in ML. It models the probability of an input belonging to a class using the logistic function, producing outputs between 0 and 1. LR is effective for linear relationships between features and outcomes. On the other hand, MLPs, LSTMs, and CNNs are key architectures in the field of NN. **MLPs**, the simplest form of NN, consists of multiple layers of neurons, each fully connected to the next, primarily used for tasks requiring feature extraction and classification. **LSTMs**, a type of recurrent neural network (RNN), are designed to process sequences of data by maintaining long-term dependencies, effectively addressing the vanishing gradient problem in traditional RNNs, making them suitable for time-series prediction and natural language processing. **CNNs**, specialized for processing grid-like data such as images, employ convolutional layers to automatically and adaptively learn spatial hierarchies of features, excelling in image recognition and classification tasks. Together, these architectures form the core of ML models, each uniquely suited to different types of data and problem domains.

Unsupervised learning

Unsupervised learning corresponds to ML algorithms that are able to classify samples without the need for a labelled dataset. In particular, unsupervised learning is employed to uncover the

structure and hierarchy within the data by using data samples without requiring ground truth labels. The knowledge representation derived from this process can serve as a foundation for a deep model [35]. Due to its more complex nature, the number of works implementing unsupervised learning in FL scenarios is scarce. Indeed, according to our exhaustive analysis on the use of ML models in FL-enabled IDS, only 16% of related works employ unsupervised learning techniques. The unsupervised methods used during this thesis are: two types of unsupervised NN, autoencoders (AEs) [57], and variational AE (VAEs) [58]; and several clustering techniques, specifically, Gaussian Mixture Models (GMM) [59], K-means [60], and spectral clustering [61].

An **AE** is a type of MLP where the input and output dimensions are the same, and its structure is symmetric. AEs aim to replicate the original data closely without needing labels. They consist of an encoder that compresses the input data into a lower-dimensional code (latent space) and a decoder that reconstructs this code back to the original data. The difference between input and output is called reconstruction error (RE), and is often measured using root mean square error. In cybersecurity, to detect anomalies, the AEs train only with benign data, and then, based on the RE of the samples, if this value overcomes a certain threshold, it is considered an anomaly. A **VAE** is a type of AE with a regularized encoding distribution, usually approaching the latent space to a standard normal distribution during training. This architecture enables VAEs to generate new, plausible data samples by sampling from the latent space. The key innovation of VAEs is their ability to learn a smooth, continuous latent space that allows for meaningful interpolations between data points.

Clustering groups data objects using a similarity measure, grouping elements with high intra-cluster similarity and low inter-cluster similarity. **K-means**, a simple partitional clustering algorithm, finds K non-overlapping clusters represented by their centroids. The process involves selecting K initial centroids, assigning data points to the nearest centroid, and updating centroids iteratively until convergence. K-means assigns each point to a single cluster, which can cause ambiguities with overlapping clusters. In contrast, **GMMs** use a probabilistic approach, allowing data points to belong to multiple clusters. Specifically, a GMM is a probabilistic model that assumes all the data points are generated from a mixture of a finite number of Gaussian distributions with unknown parameters. **Spectral clustering** uses a similarity matrix to create a graph, then, by calculating its eigenvectors, it calculates the final clusters. Traditional methods like K-means and GMM work well in convex scenarios, while spectral clustering excels in complex, nonlinear shapes and does not require a predefined number of clusters.

3.1.2. Aggregation functions

The aggregation function is a central element of FL. It aggregates the weights from all clients into new ones which are distributed back to the clients to resume their local training. In

Table 3.1, we show some of the main aggregation functions considered in the literature. In this table, we explain the main characteristic of these functions, including their advantages and disadvantages. The most used aggregation function is called **FedAvg** [22], which involves averaging the weights. This straightforward approach is widely used by existing works on FL. From the papers analysed in our work [6], the majority of approaches (87/104 or 83.7%) use FedAvg or they are mostly based on it as the aggregation function. However, in scenarios with non-iid data, several studies demonstrated the limitations of this approach [1], where FedAvg's performance is degraded, as well as in terms of security and efficiency [38, 62].

To address these limitations, advanced aggregation techniques have been developed. For data heterogeneity, **FedProx** [40] introduces a proximal term in the loss function of the clients to mitigate the effects of heterogeneous local updates. While it effectively handles non-iid settings, it requires higher computational resources and imposes additional performance demands on the devices involved. Building on FedProx's principles, **Fed+** [39] (or Fedplus) modifies the loss function of FedAvg by adding a penalty function on each client to remove the restriction that all clients' weights must converge to the same point. This approach results in a set of algorithms designed to handle data heterogeneity. Aggregation occurs in two phases: first, the server aggregates the weights, and then the clients average this value of the server with their own weights. This results in each client having its own model, which is known as personalization techniques [63]. This flexibility allows for a more tailored approach to different aggregation scenarios, but it increases complexity. Moreover, the effectiveness of Fed+ and FedProx depends heavily on the appropriate selection of the penalty constant. **FedMA** [64], focuses on the neurons' permutation invariance through a matching process of the clients' NN. This technique adapts to global model size and data heterogeneity, improving convergence while requiring fewer training rounds. Nonetheless, computing the permutation matrix may increase overall computation time. Additionally, FedMA is considered specifically for CNN and LSTM. For reducing computational cost, **FedPAQ** [65] improves the aggregation process by introducing periodic averaging, partial device participation, and quantized message-passing. These modifications significantly reduce communication and computation overhead, making the process more efficient. However, implementing FedPAQ requires additional functionality on both the aggregator and client sides, and it still struggles with non-iid data distributions.

Additionally, there is a set of aggregation functions focused on security and increasing the robustness of the FL setting and they could be named as *robust aggregation functions*. The main goal of such approaches is to mitigate the impact of poisoning attacks launched by malicious or byzantine clients. The most well-known robust aggregation functions are represented by the median [47], trimmed mean [47], and Krum [48]. The **median**, as its name suggests, uses the median (instead of the average) to aggregate the weights. The **trimmed mean** is an alternative method for calculating the mean. It involves choosing a value n smaller than half of the total number of clients, and then removing the lowest n values and the highest n values in a coordinate-wise manner and computing the mean from the remaining values. The **Krum**

aggregation function is derived from the Krum function and works by choosing the number f which represents the number of malicious clients, and then selecting the client's weight with the lowest sum among their $K - f - 2$ nearest neighbours, where K is the number of clients. The primary limitation of these aggregation functions is their vulnerability to sophisticated attacks. Indeed, while they can mitigate the impact of simple attacks with a low percentage of malicious nodes [50], their performance significantly degrades under more advanced attacks involving collusion among multiple nodes. Additionally, in the absence of malicious attacks, the accuracy of these functions is inferior to that achieved with FedAvg [8]. Furthermore, when faced with a high percentage of malicious clients, these functions exhibit substantial performance decline. Additionally, trimmed mean and Krum require determining the number of malicious clients, which in real-case scenarios could be unfeasible.

These advanced aggregation strategies highlight ongoing efforts to overcome the challenges inherent in FL, particularly those related to data heterogeneity and resistance against poisoning attacks. Each method offers unique advantages and potential drawbacks, underscoring the need for careful consideration when selecting an aggregation technique to suit specific FL applications.

Technique	**Core**	**Advantages**	**Disadvantages**
FedAvg	Based on the weighted average of the updated weights provided by clients	It is widely used due to its low level of complexity	It poses convergence issues in FL settings with non-IID data distributions
FedProx	It adds a proximal term to limit the impact of the different local updates	It addresses both data heterogeneity considering non-IID settings and device heterogeneity	It increases computation requirements and its performance largely depends on the appropriate selection of the proximal term
Fed+	Similar to the previous proximal term, a penalty constant is also added but different aggregation functions can be employed (beyond average)	It adds an increasing level of flexibility by considering different aggregation functions beyond average	A higher flexibility comes with a cost in complexity, and the performance also depends on the right choice of the penalty constant
FedMA	It is based on the neurons' permutation invariance through a matching process of the clients' NNs to perform layer-wise averaging	It adapts to global model size and data heterogeneity and improves convergence, while requiring fewer training rounds	The computation of the permutation matrix may increase the computation time, and it is specific for CNNs and LSTM
FedPAQ	It is based on averaging model updates but provides a higher degree of efficiency through periodic averaging, partial device participation and quantized message-passing	It reduces communication and computation overhead	It requires additional functionality on both aggregator and clients, and still poses issues with non-IID settings
Median	In a coordinate-wise manner, it calculates the median of the received weights	It protects the system against outlier values and certain attackers	Easily manipulable by attackers, and Worse performance than FedAvg in case of no attackers
Trimmed mean	Compute the average of the weights removing the higher and lower n values	It resists the presence of the malicious clients better compared with FedAvg	It relays on the choice of n, and inherits the issues of FedAvg in the case of non-iid data
Krum	Select the weights that minimize the distance with it $K-f-2$ neighbours	It can overcome certain attacks compared with FedAvg	Worse performance than FedAvg in scenarios without attacks, it assumes the number of attackers is known, and weak against certain sophisticated attacks.

Table 3.1

Description of several aggregation functions often considered in FL.

3.1.3. Datasets

During the course of this PhD, several datasets related to cyberattack detection (both IDS and MDS) were considered. In [5], the dataset considered is related to the energy consumption, so it is not described. Furthermore, we use four datasets: three related to cyberattack detection, and one serving as a baseline model for testing security-related aspects in FL. In Table 3.2, we provide a brief description of the cyberattack detection datasets, including the attacks considered, how it is divided among the different clients, or the attack/benign traffic ratio.

The first dataset is **ToN_IoT** [66], which is built using an IoT/IIoT (Industrial IoT) testbed that includes edge/fog nodes and cloud components to replicate an IoT/IIoT production environment. ToN_IoT is designed to gather and analyze mixed data sources from both IoT and IIoT environments. It contains heterogeneous data collected from various sources, including telemetry data from connected devices, Windows and Linux system logs, and system network traffic. This approach enables the detection of additional attacks beyond the network level in such environments. Specifically, we utilize the CIC-ToN-IoT dataset [67], which was generated using the CICFlowMeter tool [68] from the original pcap files of the ToN_IoT dataset. This tool was used to extract 83 features, which were then reduced by removing those with non-numeric values (e.g., flow ID) into 79. Subsequently, we separate the samples of the entire dataset according to the destination IP address and select the 10 IP addresses with the most samples. This subset represents the 82,29% of the dataset. The attacks contained in this dataset are: DoS, Scanning, DDoS, Backdoor, MITM, Password, Injection, Ransomware, and XSS.

The second dataset utilized is the **VeReMi** [69] (Vehicular Reference Misbehavior) dataset. VeReMi is a labelled dataset that records the message logs of both compromised and benign vehicles. These logs include reception timestamps, claimed transmission times, claimed senders, unique message IDs, GPS positions (x, y, z), RSSI values, position noise, and speed noise vectors for each receiving vehicle in various scenarios. The dataset was generated through 225 simulations using the VEINS simulator [70], considering 5 types of position forging attacks, 3 levels of vehicle density (low, medium, and high), and 3 levels of attacker density (10%, 20%, and 30%). Each parameter set was repeated 5 times to ensure randomization. Each simulation within the dataset is based on the Luxembourg SUMO Traffic (LuST) [71] scenario, which ensures a comprehensive representation of urban traffic conditions. VEINS, a vehicular network simulation framework, is used to simulate vehicle behavior, incorporating realistic models of signal interference, fading, and shadowing. The attacks simulated in the VeReMi dataset focus on position falsification, a common threat in VANETs (Vehicular Ad-hoc Networks). The five types of attacks implemented are: the constant attacker, the constant offset attacker, the random attacker, the random offset attacker, and the eventual stop attacker.

The third dataset is the **UNSW-SOSR2019**, which was created by the authors in [72]. It comprises network traffic data linked to 10 real devices. The dataset was specifically crafted

to align with the Manufacturer Usage Description (MUD) [73]. To construct the dataset for each client, the MUD profile of each device is translated into flow-table rules for monitoring the anticipated device traffic. Subsequently, these flow rules are utilized to extract features and assign ground truth labels. Considering potential packet variations within a protocol, the authors analyze the total, mean, and standard deviation of packets/bytes over time windows of 2, 3, and 4 minutes. Consequently, there are 20 features per flow rule at any given time, with the number of features varying according to the number of flow rules for each device. For evaluation purposes, a reduced dataset is generated based on a common set of features (12) across all devices. This dataset encompasses network traffic associated with 2 types of attacks: 4 direct attacks (Fraggle (UDP flooding), ARP spoofing, TCP SYN flooding, and Ping of Death) and 4 reflection attacks (TCP SYN, SSDP, SNMP, and Smurf).

It should be noted that, although there are popular IDS-related datasets such as CIC-IDS2017 [74], NLS-KDD [75] and N-BaIoT [76], these works lack modern attacks, and more importantly, they cannot be divided in a realistic and proper FL distribution, where each client could be the IP attack address or device type since this information is not provided. In this direction, we use, as previously mentioned, ToN_IoT, VeReMi, and UNSW-SOSR2019 datasets since they contain modern attacks and can also be divided following real divisions. In the case of VeReMi, it is the only dataset with these conditions in the context of vehicular misbehaviour.

Finally, the last dataset used is the **FEMNIST** dataset, a federated version of EMNIST dataset [77], created by LEAF [78] that is publicly available[1]. Although FEMNIST is not an IDS dataset, it is widely used in the FL literature for testing methods against byzantine clients launching poisoning attacks. In particular, the EMNIST dataset, in turn, is derived from the widely-used MNIST dataset, it consists of 62 classes of handwritten characters, the 52 letters in upper and lower case, and the numbers from 0 to 9. In this direction, what sets FEMNIST apart is its federated structure, emphasizing the privacy and security aspects inherent in real-world distributed systems. FEMNIST divided this dataset into 3550 non-iid clients.

Dataset	Year	# Features	# Samples (packets/flows)	Attack/benign traffic ratio	Attacks	Division
ToN_IoT	2021	79	≈5.3M p	0.88:1	Backdoor, DoS, DDoS, Injection, MITM, Password, Ransomware, Scanning, XSS	IP address
VeReMi	2018	17	≈2.2M p	0.35:1	Misbehavior (position forging) attacks	Vehicle
UNSW-SOSR2019	2019	12	≈24.6m p	0.1:1	Fraggle, ARP spoofing, TCP SYN flooding, Ping of Death, TCP SYN, SSDP, SNMP, and Smurf	Device

Table 3.2

Description of the characteristics of the cyberattack-oriented datasets used in this thesis.

[1] https://github.com/TalwalkarLab/leaf

3.2 — Related work

This section reviews the related work about FL in IDS and MDS in general, and the solutions proposed by these works in the three mentioned areas: data heterogeneity, privacy, and security. Initially, we analyze several works in the context of FL-enabled IDS/MDS to provide a general overview of the current landscape in this area. It should be noted that a more comprehensive analysis of these works can be found in [4, 6]. Next, we analyse how such works address some of our designed objectives in terms of data heterogeneity, privacy, and security. The main goal is to describe the current state-of-the-art in these areas and to identify gaps that will be addressed through our proposed methodology in the next section.

3.2.1. FL-enabled cyberattack detection

The application of FL in cybersecurity has attracted considerable interest recently due to its potential in various IoT scenarios [79]. Indeed, FL has been employed to develop IDS in IoT systems [80], serving as an alternative to traditional centralized approaches. The rising popularity of IoT applications and services has provoked an increasing attack surface with potential attacks affecting critical infrastructures. A key aspect of the IDS's design is the ability to detect such attacks effectively and efficiently while user data is not shared. FL addresses this challenge by dispersing ML models to local devices or systems, which are in charge of training ML models with their local data. FL fosters a collaborative approach for such training to build a model inteneded to identify cyberattacks [81, 82]. Indeed, it also enables large-scale scenarios where organizations can share threat information without sharing actual data, providing the foundations for privacy-preserving Cyber Threat Information (CTI) sharing [83]. Early works, such as [84–86], used Gated Recursive Units (GRUs) with FedAvg in scenarios with artificial client division. In our work [6] we show that most works are based on artificial divisions of the datasets NSL-KDD [75] and CIC-IDS2017 [74], with FedAvg as the aggregation function. Therefore, these works are based on obsolete datasets that do not represent the current landscape of network protocol. Indeed, a significant observation is that, according to our analysis in [6], most of the recent datasets can be realistically divided (e.g., using the IP address) to be used in an FL scenario. However, our analysis reveals that most of existing works are still based on iid data distributions and, in some cases, they do not specify the number of clients used, such as in [87, 88].

Moreover, based on our analysis [6], the implemented approaches heavily rely on supervised models. While many of these studies exhibit high performance in detecting various cyberattacks, they predominantly depend on labeled data, posing a significant challenge in real-world scenarios. Given the dynamic nature, scale, and heterogeneity of existing deployments, this assumption is often impractical, especially in environments like IoT, where numerous de-

vices need this information to identify potential threats. Additionally, supervised learning techniques are limited in their ability to detect novel attacks, representing another substantial drawback inherent in most FL-enabled IDS approaches. Another key observation is that, despite the use of IoT-related datasets, most studies implement models without considering the specific constraints of IoT devices and networks. Only one study [89] incorporates Binarized NNs, which could offer a promising solution for IoT environments. Regarding unsupervised techniques, AE is the most frequently utilized approach. However, although AE is well-known, most of the reviewed works employ simple or stacked AEs, with only one study using VAEs [90], recognized as a promising method for intrusion detection. Furthermore, the application of Generative Adversarial Networks (GANs) for generating attack data has been scarcely explored. With the recent advancements in generative models, it is highly probable that these techniques will gain interest in IDS development in the near future.

MDS-enabled FL is an emerging field, with recent studies focusing on adapting centralized approaches to FL environments. [91] provides an overview of FL for Vehicular IoT, highlighting benefits like low communication overhead and efficiency. [92] introduces the Federated Vehicular Network concept, using collaborative learning and blockchain to prevent malicious behavior. [93] discusses FL's advantages in traffic management and autonomous driving. Based on our analysis [6], we verified that most of the datasets are based on a single vehicle from which the data was obtained, therefore it is rather infeasible to create an FL scenario with a realistic division. Only [32] proposes an FL-based MDS using the VeReMi dataset, lacking details on the NN and data distribution. Regarding unsupervised techniques, [94] uses VAE for data reduction and GMM for clustering, while [95] presents a federated DAGMM, where clients share AE's updated weights. Other works, such as [80, 96, 97], use AEs in federated scenarios in various contexts, including wireless sensor networks.

Our analysis [6] reveals a lack of realistic client distribution, even in the most recent studies. The balanced distribution used in these works fails to address the non-iid aspects inherent to real-world scenarios. While the primary focus of these studies is on enhancing ML models, it is important to recognize that, in the context of FL, factors such as data distributions, and the implemented aggregation functions are equally important. Indeed, the work proposed in this thesis addresses such aspects through an integrated methodology, which is further described in the next section.

3.2.2. Data heterogeneity

As already mentioned, a main feature of FL settings is the presence of non-iid data distributions. This scenario frequently arises in real-world situations where different client devices may have unbalanced data. Such data/statistical heterogeneity [98] is characterized by devices having varying data sizes and class distributions, meaning that data from one device does not represent

the entire dataset [99]. While these challenges also affect centralized ML environments, they are more pronounced in FL settings due to the wide diversity of clients and their respective datasets. As highlighted in previous studies [100], statistical heterogeneity significantly impacts the convergence of the federated training process, especially when FedAvg is used. In the context of IDS development, non-iid data distributions occur when devices have a large number of samples of a particular type of attack. This is common in real-world scenarios where some devices are more vulnerable and act as entry points to the system. Notably, some studies (e.g.,[101–103]) evaluate their approaches under non-iid data configurations, although they do not propose specific solutions. As mentioned earlier, while the aggregation function is directly correlated to the convergence of the system in non-iid environments, according to Section 3.1.2, most proposed schemes use aggregation approaches based on FedAvg, which has convergence issues [100].

One way to avoid non-iid issues, as discussed in Section 3.1.2, is to use a different aggregation function. Nevertheless, only a few approaches employ alternative aggregation functions non-iid-oriented, such as Fed+ or FedProx, while other researchers [104] suggest alternative aggregation methods (FedBatch). Specifically, the authors of [105] propose a FedProx-based aggregation mechanism where nodes are grouped so that, in each training round, certain nodes are selected according to their descending gradient order. Additionally, in [106] and [107], although is more oriented to security, it is proposed a robust version of FedProx for arregating the weights. Regarding Fed+, in [108] the Fed+ approach is compared with other commonly used aggregation functions, such as FedAvg and the median. In more recent IDS, such as [109], the authors propose an FL environment where 10 clients, using different datasets, train an AE model using FedProx as the aggregation function. These solutions seek to offer a degree of personalization, enabling the global model to adapt to the unique characteristics of the clients' data. Another common approach to address convergence issues related to non-iid data distributions is employing oversampling and undersampling techniques to balance the data distributions. While some proposed methods rely on well-known techniques like SMOTE-Tomek [110] or SMOTE-ENN [111], a current trend involves generative approaches based on Generative Adversarial Networks (GANs) [112].

Despite the considerable attention given to data heterogeneity, some aspects still need further exploration. As noted, although many studies consider non-iid configurations, most approaches still depend on FedAvg for aggregation. Additionally, as mentioned in [99], evaluations need to include tuning of FL hyperparameters to ensure the convergence of developed systems, even in scenarios with a high degree of data heterogeneity.

3.2.3. Privacy

With the increasing concern for data security and personal information protection, privacy preservation has become a significant global issue, especially in big data applications and distributed learning systems. One of the primary benefits of FL is the possibility of training the data decentralised enabling the distinct parties not to share their dataset, protecting any type of information that can be extracted from these. However, FL still faces privacy challenges, as the global model updates provided by parties can be exploited to launch various attacks aimed at inferring the private information of the training data [42, 113]. Recent studies have shown that various inference attacks are still feasible during the federated training process by accessing the weights uploaded by the FL clients to the server [42], which in IDS systems this leak of information may lead to the revealing of important information about the security of the environment. In particular, in NNs, there are four primary types of privacy inference attacks [114]. Property inference attacks exploit the similarity in models trained on similar datasets to infer sensitive properties about the training data. Model extraction attacks aim to replicate the target model by exploiting black-box or gray-box access, thereby potentially revealing training data and bypassing security mechanisms. Model inversion attacks use information from the model to reconstruct input data or infer training data properties. Lastly, membership inference attacks attempt to ascertain whether specific data points were included in the model's training dataset by leveraging overfitting characteristics and prediction outputs.

To address these attacks, recent works have presented the application of various privacy-preserving techniques, such as Secure Multi-Party Computation (SMPC) and Differential Privacy (DP), to FL scenarios. In the context of IDS, most of the presented works rely on the application of cryptographic techniques for privacy preservation, such as DP, SMPC, and Homomorphic Encryption (HE). The works based on DP include [115] and [116], as well as [117], which also integrates HE. These cryptographic methods enable computations to be carried out on encrypted data. Moreover, SMPC refers to a cryptographic protocol where multiple parties can compute a function together without revealing their inputs. In this way, in the context of FL, the aggregator would not be able to obtain the updates generated by each client [118]. Specifically, the use of DP for FL is explored by the authors of [119], in which they examine the effect of applying only Gaussian noise as a DP method within an FL setting, without providing a comparison with more techniques. Also, their evaluation is limited to the widely-used MNIST dataset. In another study, [120], an IoT scenario with resource limitations is investigated, where the authors applied and evaluated a relaxed version of DP using various datasets. In [121], the authors utilize activity recognition data from smartphones to develop personalized models on each device. Likewise, [122] leverage DP in combination with blockchain technology, ensuring that the computation required for the consensus mechanism also contributes to the federated training process. Nonetheless, in this case, the DP technique implemented is unknown, making it hard to replicate the results for other researchers.

Based on the analysis of current approaches, we perceive that most works rely on well-known cryptographic techniques (especially those based on DP) to prevent potential attackers from accessing model updates in each training round. However, the conducted analyses are insufficient to demonstrate the applicability of these techniques in different contexts, including the development of FL-enabled IDS. As described in [99], additional analysis is required regarding the impact of different parameters in a federated environment, such as the aggregation function used. In fact, these aspects should be considered along with the type of data and the environment where the IDS is deployed, as they can have a crucial impact on finding trade-offs between privacy and system effectiveness in detecting attacks.

3.2.4. Security

The security aspects of FL settings have garnered significant interest in recent years [42, 123]. Similar to centralized approaches, FL scenarios are vulnerable to poisoning attacks, where weights or datasets of the clients are maliciously altered [124] at any round. These attacks involve clients introducing malicious data or weights to undermine model performance, affecting all participating models [44]. There are two main types: data poisoning, altering the dataset of the client; and local model poisoning, which modify the weights after the training. Data poisoning can be clean-label (modifying training samples without altering labels) or dirty-label (targeting both samples and labels), with label flipping being a common example. Local model poisoning alters training weights, and can be untargeted (causing widespread prediction errors) or targeted (misclassifying specific classes) [125]. These attacks can severely impair IDS performance, reducing its ability to detect cyberattacks, which can have serious implications depending on the deployment location of the IDS. Additionally, this could lead to false alarms resulting from misclassification during the training process.

To address this, some proposals suggest alternative aggregation functions to FedAvg to mitigate the effect of malicious clients. For instance, in [106] a robust approach based on FedProx to defend against various malicious devices is introduced. Additionally, [126] explores the effectiveness of different robust aggregation functions against data and model poisoning attacks. Specifically, researchers have examined well-known functions like the trimmed mean [47] to defend against label-flipping and gradient modification attacks. Additionally, GANs have been extensively considered to improve the robustness of FL-enabled IDS [127]. For example, [127] proposes the use of GANs to enhance system robustness by training with data related to previously unseen attacks. These tools can be complemented by trust and reputation approaches to evaluate the level of trustworthiness offered by FL clients, as proposed by [128]. However, GANs can be a double-edged sword, as they might also generate synthetic weights and gradients to identify potential malicious nodes. In contrast, in [129] it is used k-means to detect nodes sending false gradients during training. Additionally, in [106] and [107] is proposed a robust version of FedProx where nodes suspected of producing fake updates are excluded from

the process.

As described in [123], various mechanisms can improve the security of FL environments, which are able to be applied to FL-enabled IDS contexts. Although some works have considered more robust aggregation functions, there is a lack of comprehensive analysis considering additional well-known aggregation functions such as Krum to be deployed in FL-enabled IDS. Furthermore, most analyses overlook the complexity of these functions, which could significantly impact IDS deployment, given the need to detect potential attacks as soon as possible. Moreover, there is a lack of a comprehensive list of attacks to be considered in FL environments as well. This absence of consensus on defining attacks complicates the comparison of different aggregation techniques and their robustness evaluation. Additionally, most evaluations rely on questionable assumptions, such as attackers being isolated nodes with limited system knowledge. This issue is addressed by [49], which generates specific attacks for some of the previously mentioned aggregation functions. Beyond defining robust aggregation techniques, as mentioned by [123], using statistical approaches can be crucial to identify nodes sending forged updates.

Chapter 4

Methodology

During this PhD, we identified several challenges to create models for cyberattack detection. Therefore, the objectives outlined in Section 2 are categorized into three main areas: data heterogeneity, privacy, and security. By addressing these objectives, we aim to create a more secure and privacy-preserving federated environment where the model's performance is maximized and the impact of data heterogeneity is mitigated. Figure 4.1 provides a visual summary of the methods implemented to address the issues previously discussed. This figure builds upon the FL training process depicted in Figure 1.1, extending the different layers with our solutions. Specifically, we apply resampling techniques in the datasets layer to address data heterogeneity. Furthermore, before sending the weights to the server, we apply DP mechanisms to obfuscate the weights for privacy protection. Additionally, on the server, we implement different aggregation functions depending on the scenario to enhance security and handle data heterogeneity.

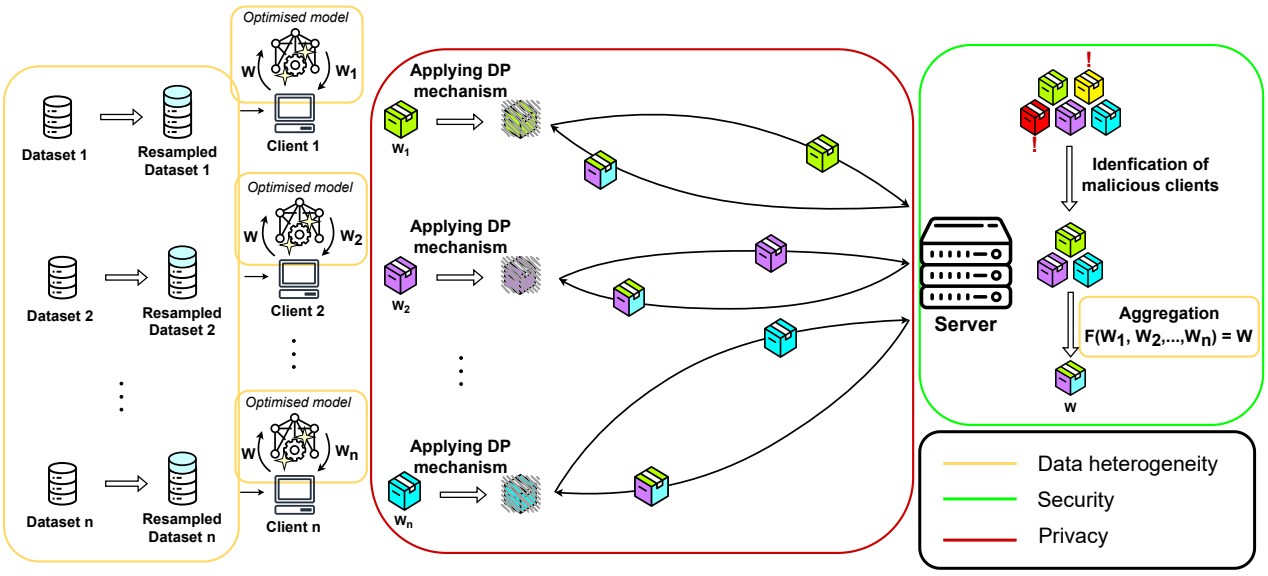


Figure 4.1. Visual description of the different areas addressed through this thesis to reach the described objectives related to data heterogeneity, security, and privacy.

4.1 — Mitigating the impact of data heterogeneity

Our primary assumption is that clients' datasets need to be linked to identifiers, such as IP addresses, to associate each client with its generated data, thereby reflecting a realistic scenario. This results in non-iid datasets, presenting certain challenges that need to be addressed. To mitigate the impact of these non-iid distributions, as shown in Figure 4.1, we preprocess the clients' datasets through resampling techniques, optimise their model, and implement alternative aggregation functions. These aspects are further elaborated below.

4.1.1. Dataset resampling

The characterization and performance of an FL scenario are intrinsically tied to the specific use case and the dataset employed. The partitioning of data among multiple clients introduces non-iid issues. To mitigate the impact on overall accuracy, it is beneficial to adjust the clients' datasets to balance class distribution. However, modifying the entire dataset collectively is infeasible, as clients only have access to their own data and lack knowledge of other clients' distributions. Consequently, we use resampling techniques to balance class distributions within each client's dataset, thereby reducing inter-client class unbalance. Resampling involves increasing or decreasing the number of samples in various classes to achieve a balanced distribution.

With the datasets of section 3.1.3, as previously mentioned, we apply resampling techniques to balance the local datasets of each client. For measuring the unbalance of a dataset we use the Shannon entropy [130], which is defined as $\frac{-\Sigma_{i=1}^{k} \frac{c_i}{n} \log \frac{c_i}{n}}{\log k}$, where n is the dataset length, k is the number of the different classes of the dataset, and c_i is the length of the corresponding class. The value of this function represents the balance of the dataset, valuing 0 if all classes are 0 except one, and 1 if all $c_i = \frac{n}{k}$, i.e., equally distributed. In the case of ToN_IoT, for balancing the dataset, we select among the 10 clients selected in Section 3.1.3 the ones which are better balanced in terms of Shannon entropy. In this case, we selected those with a Shannon entropy higher than 0.2 (i.e., 4 clients), and then randomly removed samples from the classes with more samples of the until reaching a value between 0.65 and 0.7 of entropy.

Concerning the VeReMi and UNSW-SOSR2019 datasets, as illustrated in Table 3.2, the limited sample sizes of these datasets make the application of a subsampling technique similar to that used for ToN_IoT impractical. In this direction, we employed a different approach known as SMOTE-Tomek [110], which combines oversampling and undersampling techniques. This combination addresses the potential issue of overfitting that can arise when using only oversampling methods, particularly in cases of significant class overlap [131]. SMOTE-Tomek integrates the SMOTE oversampling technique [41] with the Tomek-links undersampling method [132] The

process of balancing the dataset is as follows. First, SMOTE creates new samples in the minority classes by making linear combinations between samples, specifically, from a sample s and its k-nearest neighbours [133] s^i, the new samples s^i_{new} are calculated by $s^i_{new} = rs^i + (1-r)s$, where r is a value between 0 and 1 which produced intermediate points between s and s^i. The number of points and neighbours taken is defined depending on the number of points needed. Lastly, these new points can overlap with each other. In order to provide better-differentiated classes, Tomek-links removes similar points of different classes to make the difference among clients higher. Specifically, from a couple of neighbour points (x_i, x_j), one from the minority class and the other from the majority class, Tomek-links removes the point belonging to the majority class, leading to better-defined classes.

Finally, the FEMNIST dataset was used to evaluate our approach for identifying and mitigating the impact of poisoning attacks. Given that each client in the FEMNIST dataset has 1,315 samples, 47 classes, and an average Shannon entropy of 0.93, we assumed the dataset to be balanced. In our experiments, we selected random subsets of 50 and 100 clients.

4.1.2. Analysis of alternative aggregation functions for data heterogeneity

The aggregation function plays a fundamental role in FL as it is in charge of aggregating client weights, with FedAvg being the most commonly used. However, in non-iid scenarios, the use of FedAvg may provoke convergence issues because of the discrepancies between the aggregated weights and individual client models. FedAvg is widely used; indeed, as described in [6], the 83.7% of the analysed works about FL-enabled IDSs use FedAvg. In scenarios with balanced datasets, FedAvg performs relatively well. However, in more complex scenarios characterized by non-iid data distributions, the performance of FedAvg significantly deteriorates due to statistical heterogeneity, which affects the convergence of the federated training process. In fact, several recent studies have demonstrated the limitations of this approach in scenarios with a high degree of data heterogeneity [1, 38]. Hence, the choice of the FL aggregation approach is crucial for the development of a robust and secure FL-enabled IDS.

In this direction, Fed+ was created to face data heterogeneity and to mitigate the impact derived from non-iid data distributions. Fed+ offers a significantly flexible approach since it adds a penalty function to the loss function of FedAvg on each client to remove the restriction that all clients' weights converge to the same point. In particular, the weights of each client k are uploaded following Equation 4.1, where r is the round, $\theta = \frac{1}{1+\nu\mu}$ a constant that controls the degree of regularization in which μ is a user-choice constant and ν the learning rate, A is an aggregation function (e.g., FedAvg), and $B(\cdot,\cdot)$ is a distance function that penalizes the deviation of a local model W^k from the output of $A(\cdot)$. In our cases, $B(W^k, A(W^1, \ldots, W^K)) = A(W^1, \ldots, W^K)$, and $A(W^1, \ldots, W^K)$ is the average of the weights.

$$W_{r+1}^k \longleftarrow \theta[W_r^k - \nu \nabla f_k(W_r^k)] + (1-\theta)B(W^k, A(W^1, \ldots, W^K)), \tag{4.1}$$

Although there are more aggregation functions (described in Section 3.1.2), such as FedPAQ, FedMA, or FedProx, we opted for implementing only Fed+ to solve non-iid issues due to two main reasons. First, with an appropriate tuning of parameters, other functions can be considered as special cases of it, for instance, FedProx or FedOpt [134]. Second, other functions focus on solving other aspects of FL or can be only used in the case of specific ML models. In particular, FedPAQ focuses on computational efficiency, FedMA is only applicable on LSTM and CNN models, Turbo-Aggregate [135] on data privacy, and SAFA [136] on asynchrony aspects. Hence, in our context of FL-enabled IDS, these reasons justify the use of Fed+ for improving the results achieved by FedAvg. In each work, apart from the implementation of FedAvg (since it is the most used one), we also implement Fed+, with its corresponding tuning of the parameters to maximize the accuracy, in order to compare the results and to analyze its behavior in non-iid settings.

4.1.3. Optimization of the Model

Proper parameter selection helps ensure that the model performs well across diverse client datasets, addressing heterogeneity by tailoring the model to fit various client needs. In the case of NNs, the choice of hyperparameters such as the number of layers, neurons, and learning rate is crucial since a lack of optimization can result in NN being stuck in local minima or suffering from slow convergence rates, leading to suboptimal performance in both training speed and accuracy [137], and consequently slowing the convergence in the rest of the clients in an FL context. For the federated training, these hyperparameters must be chosen to best fit the diverse needs of each client and its dataset, ensuring optimal performance. For that reason, we use the GridSearchCV module[1] since it achieves notable results in other works such as [138, 139]. Such module selects the optimal number of neurons, layers, optimizer, and other parameters, due to the fact that it employs cross-validation through a grid search to find the best combination of hyperparameters. This optimal choice helps the model avoid overfitting issues for the clients during the training [140, 141]. Consequently, preventing overfitting helps maintain robust convergence across all clients, ensuring that the global model remains effective despite data heterogeneity. Additionally, as the Fed+ parameters depend on the learning rate, in our works we performed a grid study to choose the best learning rate that maximizes global accuracy. In the case of unsupervised models, as we used AEs and VAEs, we initialised the weights of these unsupervised NN using Restricted Boltzmann Machines (RBM) [142]. RBMs are a type of NN that can be used as a pre-trained method to initialise the weights of AE and

[1] `https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html`

VAE, since an incorrect initialization may lead to convergence issues [143–145]. Specifically, RBMs are a form of stochastic NN distinguished by their unique two-layer structure. They are defined by symmetric connections between layers and the absence of self-feedback loops. Significantly, RBMs have full connectivity between the two layers but no connections within the same layer. Hence, in the context of FL, a delay in the convergence in the accuracy of any client affects the aggregation, and consequently, the rest of the clients. Therefore, this pre-training step assists AE and VAE in initializing its parameters in a manner that enhances the likelihood of effective convergence during the subsequent training phase.

4.1.4. Results

Before presenting the results, it is important to mention the FL implementations employed in this thesis: Flower [146] and IBMFL [147]. **Flower** is an open-source, Python-based FL framework that focuses on large-scale experiments involving heterogeneous devices. It offers several advantages, including stability, broad support for multiple programming languages, operating systems, and ML frameworks. Additionally, Flower supports scenarios with varying privacy requirements. Moreover, **IBM Federated Learning (IBMFL)** is an open-source Python library to facilitate the deployment of FL settings in productive environments. IBMFL is designed as an enterprise-level solution, providing a fundamental FL layer upon which more advanced features can be built. Our works [1] and [2] were implemented using IBMFL, whereas the rest employed Flower, which has received a significant support from industry and academia in recent years.

As discussed, addressing non-iid data distributions requires implementing alternative methods at both the client and server levels of FL, particularly focusing on datasets, models, and aggregation functions. These adaptations help mitigate the possible negative impacts of partitioning the total dataset among multiple entities. In this section, we show the main results obtained in our works. In particular, in Fig. 4.2 we provide the main results of our initial paper on this field [1]. In this work, we consider 3 different distributions of the ToN_IoT dataset: the basic, the balanced, and the mixed. The basic refers to using the dataset of the 10 clients directly without resampling techniques. The balanced refers to resampling equally the whole dataset into 10 clients. Finally, the mixed is the distribution described in Section 4.1.1, meaning a federated scenario involving the 4 clients with the highest Shannon entropy, where their datasets have been adjusted to enhance the distribution of their classes. We compare these 3 distributions under 3 cases, using FedAvg, using Fed+, and the distributed case, where the clients do not send the weights during the training to the server, i.e., without forming a federated environment. In the figure, we notice that the basic scenario under the FedAvg and Fed+ cases reaches a constant value without any evolution. Next, as expected, the balanced scenario reaches the best results in each case, but obtaining this scenario is almost impossible in a real case. However, looking at the mixed scenario, this resampling technique enables the

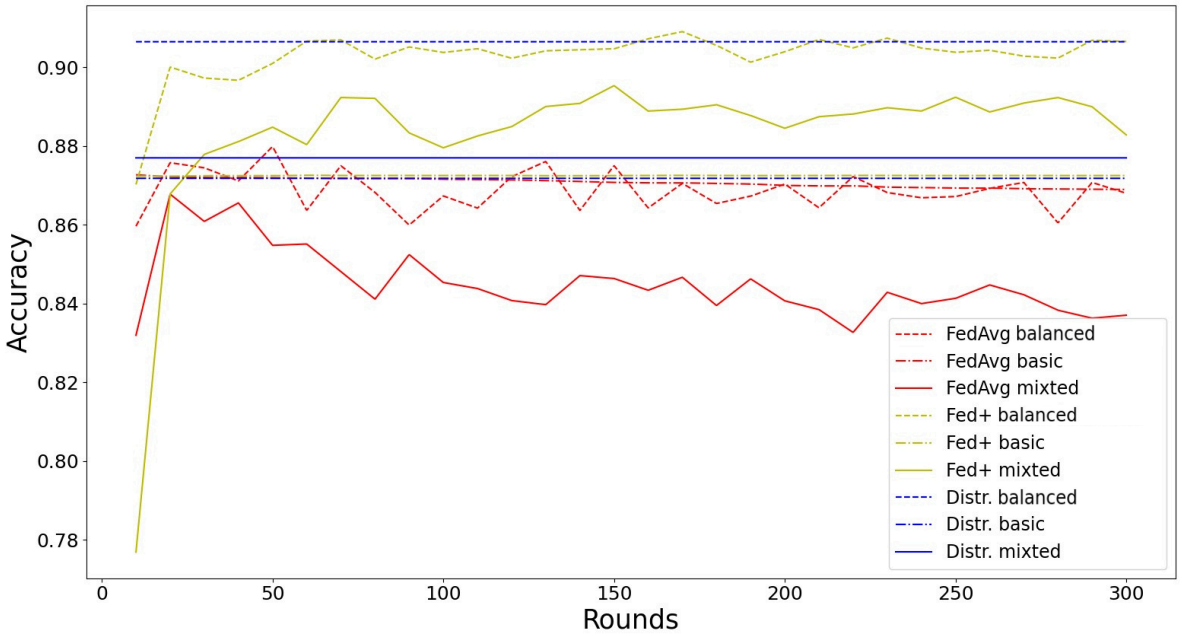
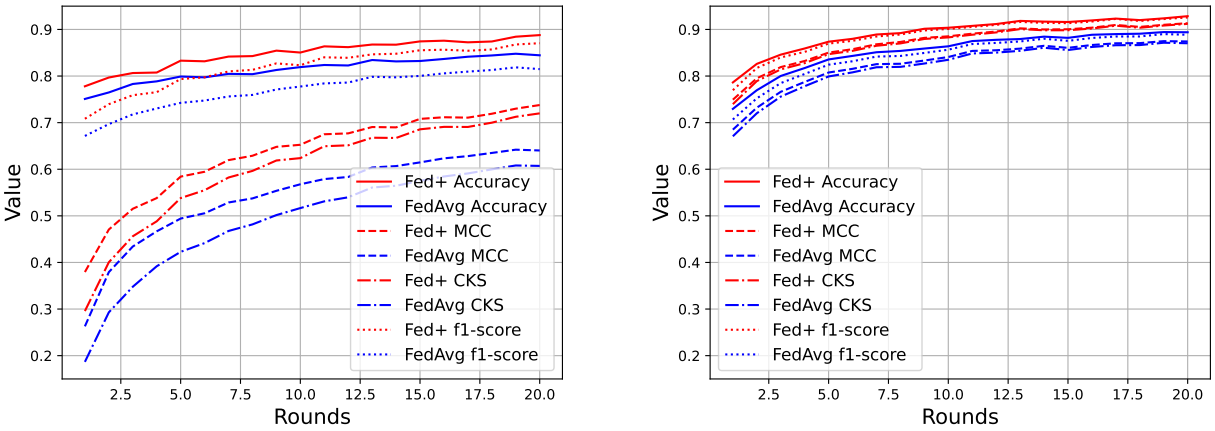


Figure 4.2. Comparison of the different scenarios with various data distributions on the ToN_IoT dataset

clients to improve from the accuracy evolution of the basic scenario to an evolution similar to the balanced scenario. Additionally, the implementation of Fed+ enables to achieve higher values than FedAvg, and even overcome the distributed case.

Based on the results provided in [4], Fig. 4.3 provides the evaluation results of training an MLP model, previously optimized, with the VeReMi dataset under different configurations: rebalancing or not with SMOTE-Tomek, and implementing FedAvg or Fed+. To measure the performance of each configuration on an equal footing, especially considering that unbalanced datasets can often exhibit high accuracy by predicting the majority class correctly while ignoring the minority class [148–150], we evaluate the performance using additional metrics such as the F1-score, the Matthews Correlation Coefficient (MCC) and the Cohen Kappa Score (CKS) [151]. These two latter have been created for measuring the reliability of the accuracy, as they treat all classes equally. In Fig. 4.3a and 4.3b, the Fed+ function obtains better results in each case, as we want to prove with the implementation of Fed+. Additionally, looking at both pictures, without applying SMOTE-Tomek we notice a significant gap between the accuracy and f1-score with the MCC and CKS. As we mentioned, this gap is linked to the unbalance of the dataset, being smaller when SMOTE-Tomek is applied, which means that the accuracy obtained in this case is a fair representation of the model's performance. Hence, in this context, the accuracy alone is not enough to measure the performance of a model. Finally, the combination of Fed+ and SMOTE-Tomek helps to achieve better results compared to most current works, as previously described in Section 3.



(a) Metrics using the original Veremi dataset with FedAvg and Fed+

(b) Metrics using the balanced dataset with FedAvg and Fed+

Figure 4.3. Comparison of various scenarios using the original VeReMi dataset versus the balanced VeReMi dataset obtained through SMOTE-Tomek

4.2 — Addressing privacy concerns through DP techniques

One of the main advantages of using FL is related to privacy since the data of the different sources do not need to be shared to train the global model. However, as extensively discussed in Section 3.2.3, the exchange of weights during the training process could raise serious privacy concerns, as information derived from the local dataset can still be inferred by malicious parties, including the server [42, 98, 152]. As described in [99], privacy threats can involve diverse types of actors, such as honest-but-curious aggregators, who attempt to infer information from each client's dataset using the shared updates in each training round. Likewise, even clients belonging to the federated environment may also be able to infer information from the weights of other clients using the information from the global models sent by the aggregator. To extract any information from the training data through the weights, these curious parties can perform reconstruction attacks using various techniques, including GANs. In fact, GANs can also be utilized to conduct membership inference attacks, enabling an attacker to determine if a certain party's local data was used in the training process [153]. In the context of IDS, since the dataset contains sensitive information about device security, any data breach could significantly impact the system's integrity. Additionally, other attacks may involve inferring the participation of specific nodes in the training process, which can also have privacy implications [99].

4.2.1. Analysis of the impact of different aggregation functions

To address these privacy concerns, the main solution is to obfuscate the model weights, providing statistical privacy guarantees for the data against adversaries. In particular, DP is often preferred in FL settings due to the strict communication demands of other privacy-preserving techniques, such as SMPC. Nevertheless, recent research [154] emphasises the significant computational and communication requirements of SMPC, making these techniques unsuitable for IoT environments. In this direction, we provide an exhaustive evaluation of several DP algorithms consisting of additive noise techniques based on Gaussian and Laplacian distributions. Unlike the state-of-the-art, where the researchers only study the FedAvg case, we compare different configurations of the different techniques to analyse their impact on the accuracy under FedAvg and also Fed+. This comparison involves 7 DP mechanisms: Gaussian Analytic, Gaussian mechanism, Laplace mechanism, Laplace truncated mechanism, Laplace Bounded Domain mechanism, Laplace Bounded Noise mechanisms, and Uniform. These mechanisms use a parameter ϵ to adjust the amount of noise added to the weights sent to the server. Decreasing this parameter increases the privacy of the weights, making it more difficult to infer the dataset from them, but this also negatively impacts the overall accuracy of the process. Algorithm 1 shows how the different DP mechanisms. First, the different clients train the model to calculate their new weights W_r^k, and apply one of the 7 DP mechanisms for obfuscating the weights before sending the weights to the server to protect them against infer attacks. Then the server aggregates the weights and sends them back to the clients to continue their training. The goal of this comparison is to test the best choice of this parameter ϵ to obtain a certain level of privacy without sacrificing accuracy.

4.2.2. Study on the trade-off obfuscation-accuracy

Reducing the parameter ϵ does not directly quantify the amount of noise or level of privacy introduced. Therefore, we analyze the impact of ϵ on the similarity between the perturbed weights and the original weights. To evaluate the effectiveness of DP techniques in obfuscating weights, we employed Pearson Correlation Coefficients (PCC) [155]. This method was instrumental in determining the optimal parameter for safeguarding against inference attacks and measuring the degree of weight obfuscation. The PCC quantifies the linear association between two variables, with values ranging from -1 to +1. A value of +1 signifies perfect correlation, whereas a value of -1 indicates inverse correlation. Essentially, the PCC is the ratio between the covariance and the standard deviation of two variables. In our methodology, the PCC is computed over the model updates for each training round, both before and after applying the DP mechanism. This metric reflects the similarity between the original and perturbed weights for each client. The privacy enhancement provided by each mechanism is indicated by a PCC value between 0 and 1. A PCC value of 1 corresponds to the classical FL scenario without any

Algorithm 1 Algorithm of our differential privacy framework

Input: K set of clients, R number of rounds, E number of epochs, ϵ degree of noise of the mechanism, h model
Output: Aggregated Weights W_R

for r in 1 to R **do**
  for $k \in K$ **do**
    Receive weigts W_r from server
    Let x_k be the input and y_k the labels of the local data of client k
    Normalise local inputs
    for 1 to E **do**
      Compute prediction $\hat{y}_k = h(x_k)$
      Compute loss $\mathcal{L}_k = L(\hat{y}_k, y_k)$
      Compute the gradients $\triangle w = -\nabla_{\mathcal{L}_k} w$
      Update parameters $W_r^k = W_{r-1}^k + \triangle w$
    end for
    Apply the DP-mechanism to the weights W_r^k to get $\kappa(W_r^k)$ with parameter ϵ
    Send $\kappa(W_r^k)$ to the server
  end for
  Server receives the weights $\kappa(W_r^k)$
  Server aggregates them into the weights W_r
  Server sends W_r to clients in K
end for

perturbation mechanism applied, hence no obfuscation. Conversely, a value near 0 signifies a high degree of obfuscation and strong protection against inference attacks. Notably, across all mechanisms, a lower ϵ value correlates with a lower PCC, indicating that lower ϵ values achieve a more obfuscated set of weights, thereby enhancing privacy. This correlation highlights the trade-off between privacy and data utility, as higher levels of obfuscation typically result in decreased model performance. To strike a balance, it is crucial to select an ϵ value that ensures sufficient privacy without excessively compromising the model's accuracy. By evaluating the PCC values alongside model accuracy metrics, we identified which DP mechanisms provide the best privacy protection while maintaining acceptable levels of accuracy.

4.2.3. Results

In Fig. 4.4, we show the results of the previously mentioned analysis of the different methods using Fed+ as the aggregation function. In this analysis, we part from the mixed scenario of the ToN_IoT dataset described in Section 4.1.4 under the 7 DP mechanisms. In all of them, we compare different parameter values of ϵ to measure how accuracy is affected. From this figure, we notice that the accuracy is not altered by the different configurations of the different mechanisms, which allow us to obfuscate the weights to protect the privacy of the different clients. Additionally, in Table 4.1 it is shown the PCC for each value. This table measures the level of obfuscation of the weights, the lower the value, the more obfuscated the weights and

consequently, the more protected. Hence, the best mechanism would be the uniform mechanism since it has the lowest PCC value and in its respective accuracy in Fig. 4.4 is not affected.

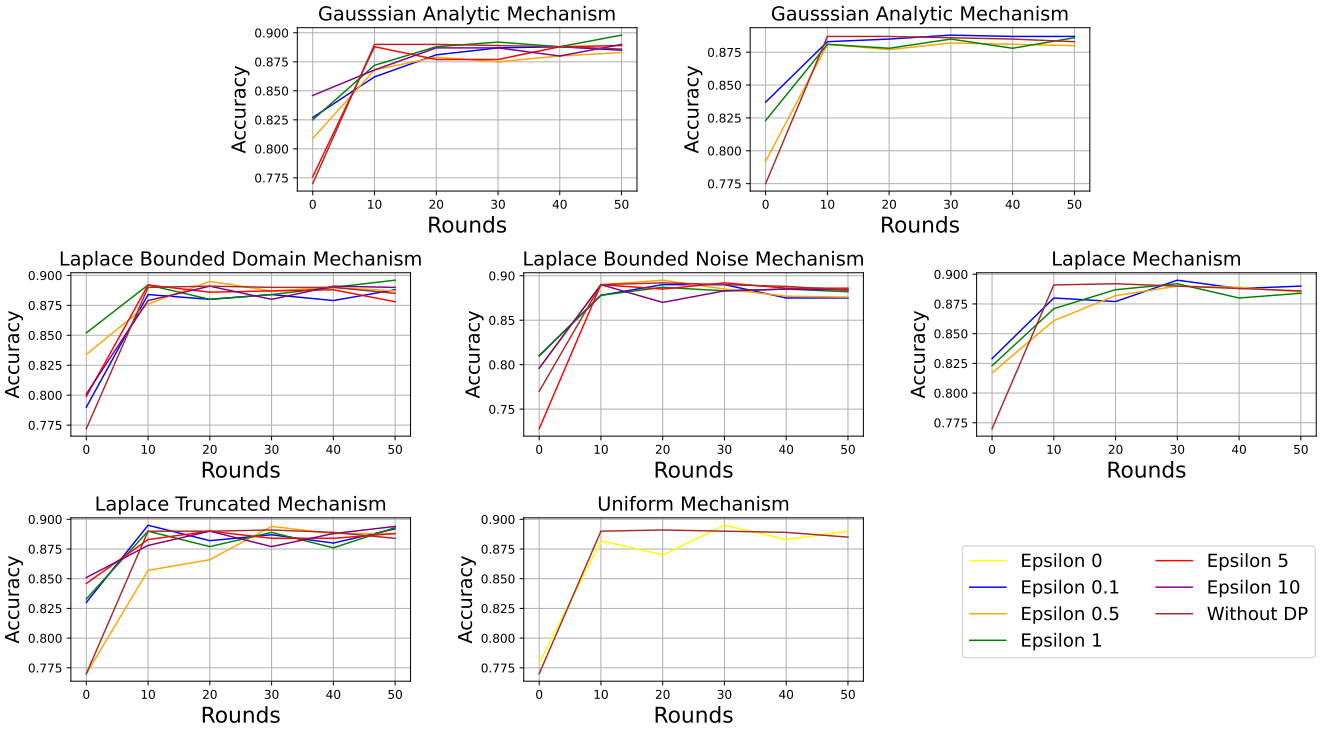


Figure 4.4. Fed+ accuracy results for all perturbation mechanisms.

4.3 — Enhancing FL settings' security

During FL training, the system's security can be compromised by model poisoning attacks, in which malicious clients deliberately modify the weights they send back in each training round [124]. In the context of IDS the primary impact is a decrease in cyberattack detection performance, which can have severe consequences depending on the deployment environment. For instance, this could lead to an increase in false alarms due to misclassification errors introduced during the training process. Therefore, robust protection against these attacks is crucial for maintaining the model's integrity. To address such attacks, we we propose a novel robust aggregation function leveraging the properties of the Fast Fourier Transform (FFT). Additionally, we design and implement an approach to identify and exclude potentially malicious clients during the training process.

ϵ	Mechanism	PCC	Mechanism	PCC
0.1	**Laplace Truncated**	0.6560	**Laplace Bounded Domain**	0.4552
0.5	**Laplace Truncated**	0.9663	**Laplace Bounded Domain**	0.9028
1	**Laplace Truncated**	0.9906	**Laplace Bounded Domain**	0.9762
5	**Laplace Truncated**	0.9996	**Laplace Bounded Domain**	0.9994
10	**Laplace Truncated**	0.9999	**Laplace Bounded Domain**	0.9998
0.1	**Laplace**	0.6572	**Gaussian**	0.3549
0.5	**Laplace**	0.9655	**Gaussian**	0.8256
1	**Laplace**	0.9907	**Gaussian**	0.9406
0.1	**Gaussian Analytic**	0.5964	**Laplace Bounded Noise**	0.6951
0.5	**Gaussian Analytic**	0.9156	**Laplace Bounded Noise**	0.9666
1	**Gaussian Analytic**	0.9704	**Laplace Bounded Noise**	0.9908
5	**Gaussian Analytic**	0.9977	**Laplace Bounded Noise**	0.9996
10	**Gaussian Analytic**	0.9992	**Laplace Bounded Noise**	0.9999
0	**Uniform**	**0.0472**		

Table 4.1

Table of the PCC of the different DP mechanisms for different values.

4.3.1. FedRDF: a robust aggregation function for FL

Using a robust aggregation approach represents an approach to mitigate the impact of byzantine clients in FL settings. As detailed in Section 3.1.2, the functions can significantly reduce the impact of such compromised devices. However, these methods typically assume the number of attackers is known, which might not be practical in real-world situations or against sophisticated attacks where the compromised parties dynamically change during the training rounds. Indeed, most existing approaches primarily address basic poisoning attacks, overlooking scenarios where attackers can collude. However, recent works highlight the vulnerabilities of these approaches against more sophisticated attacks [49]. Based on the limitations of existing aggregation functions, we identified the need for a new robust aggregation function that does not require prior knowledge about the number of attackers and is not based on distance or statistical methods. To this end, we leveraged the advantages of FFT to create this novel robust aggregation function. The FFT allows us to calculate the density function of the weights and identify the point of highest concentration. As illustrated in Fig. 4.5, we apply this process coordinate-wise, projecting client weights into the frequency domain. This enables us to determine the point with the highest value in the density function through its frequency domain representation [46], which corresponds to the point with the highest frequency. As the FFT is easily invertible, the inverse of such point is calculated to get the original set of weights, and those weights are the ones that are sent to the clients. This function is robust against outliers since these values have to be far from the condensed points in the density function, which do not affect the election of the final point.

While this method is effective against malicious clients, its main drawback is that the Fe-

dAvg function outperforms robust aggregation functions when no malicious clients are present. At first, it is clear that non-robust functions reach higher values. Nonetheless, in a real case, it is unknown whether there are malicious clients. Given that, we developed an algorithm based on the Kolmogorov–Smirnov (K-S) statistical test [156]. This test determines whether two distributions are statistically equivalent. As shown in Algorithm 2, prior to aggregation, we apply a coordinate-wise process to each set of points $V_{i,j}$ depicted in Fig. 4.5. We select a subset of size S from these points and apply the K-S test between this subset and the remaining points. If the test fails, it indicates potential malicious weights distorting the distribution. To ensure accuracy in such process, we repeat it C times. If the number of failed tests exceeds a user-defined threshold, we conclude that malicious activity is present. At each round, we apply the explained K-S test to detect potential attacks from malicious clients. If the test passes, meaning there are no detected malicious clients, the system uses FedAvg as the aggregation function; otherwise, it employs the FFT-based method. We call this comprehensive process FedRDF (Robust Dynamic Fourier aggregation function). FedRDF adapts to various scenarios, maximizing accuracy regardless of the presence of malicious clients.

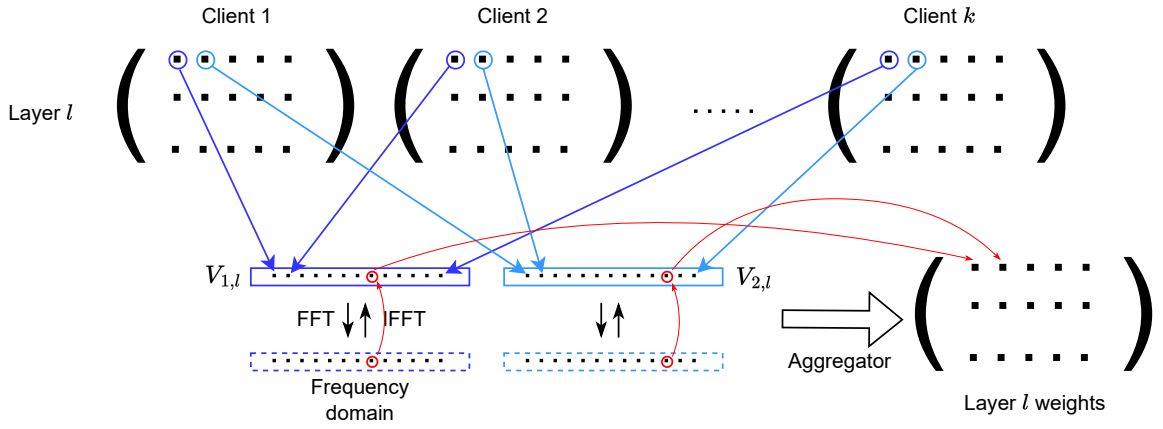


Figure 4.5. Visual description to calculate the FFT in FedRDF.

4.3.2. Identification of byzantine clients

While robust aggregation functions protect systems against Byzantine clients, they have notable limitations [50], as discussed in the previous subsection and Section 3. For instance, their performance tends to decline as the number of malicious clients increases. These limitations motivate the development of an approach for identifying and excluding specific clients that send malicious weights throughout the training rounds. Building upon current state-of-the-art algorithms, we propose a framework that addresses the limitations of existing approaches, making use of several techniques for detecting malicious clients. In this direction, we create a framework called FLAegis. Our framework is mainly intended to be deployed on the server, and as can be seen in Fig. 4.6 and in Algorithm 3, it consists of 2 phases: the identification

Algorithm 2 Algorithm of our robust dynamic federated learning framework

Input: K set of clients, R number of rounds, E number of epochs, C number of repetitions of K-S test, S size of the subset part for calculating the K-S test, and t threshold.
Output: Global model W_R

for r in 1 to R **do**
  for $k \in K$ **do**
    $W_r^k = \text{localUpdate}(W_{r-1}, E)$
    Send W_r^k to the server
  end for
  for Server **do**
    $t_r = \text{mean}(K - S\_test(\{W_r^1, \ldots, W_r^{|K|}\}), C, S)$
    if $t_r < t$ **then**
      $W_r = \text{mean}(\{W_r^1, \ldots, W_r^{|K|}\})$
    else
      $W_r = \text{FFT}(\{W_r^1, \ldots, W_r^{|K|}\})$
    end if
    Send W_r to clients
  end for
end for

$K - S\_test(\{W_r^1, \ldots, W_r^{|K|}\}), C, S)$:
Calculate $V_{i,l} = \{w_{i,l}^1, \ldots, w_{i,l}^K\} \forall i, l$
for all $V_{i,l}$ **do**
  for 1, 2, ..., C **do**
    Take sample $\hat{V}_{i,l}$ of $V_{i,l}$ of size S
    Calculate p-value of $KS(\hat{V}_{i,l}, V_{i,l} \setminus \hat{V}_{i,l})$
  end for
  Calculate times the $p-$value were less than 0.05
end for
return vector of failure proportions

phase, and the mitigation phase. The identification phase consists of the detection of malicious clients among the set of clients. First of all, in this type of framework, due to limitations of clustering techniques, as described in [157, 158], we assume that the number of malign clients is fewer than half of the clients. Under this assumption, we classify clients by creating a similarity matrix M. To enhance the distinction between malicious and benign clients while reducing differences among benign clients, we first transform the clients' weights using Symbolic Aggregate approXimation (SAX) [159], where the sample space of the weights is divided into a predefined number of parts, and each weight is associated with a symbol. In this way, similar weights will have close similarity, while the malicious ones, having to separate from the benign ones, will show a greater difference. This transformation allows to measure similarity more effectively. Next, using the cosine similarity [160] we measure the similarity among clients to create the previously mentioned similarity matrix M. After calculating this matrix, we employ clustering techniques to categorize clients as either benign or malicious. We implement spectral clustering, which is able to dynamically cluster the elements without selecting the number of

clusters beforehand, unlike other methods such as K-means or GMM. If the number of clusters obtained is 1, it means all clients are similar, and therefore there are no malicious clients. Otherwise, it means that there are malicious clients, thus, we perform a final clustering using K-means with 2 components (as the final clustering of spectral clustering uses K-means) predefined and select the smaller cluster as the one containing the malicious clients (as we assume the number of malign clients was less than the benign ones). However, in this identification method, there may be a few malicious clients that overcome the identification phase and be classified as benign. To protect the system against those types of clients, we implement a mitigation phase in which the weights of the remaining clients are aggregated using the FFT as the aggregation function (as in the previous section) to reduce the impact of malicious clients. Such approach to identify malicious clients is intended to mitigate the issues previously described for robust aggregation functions, specially in settings with a high percentage of such clients.

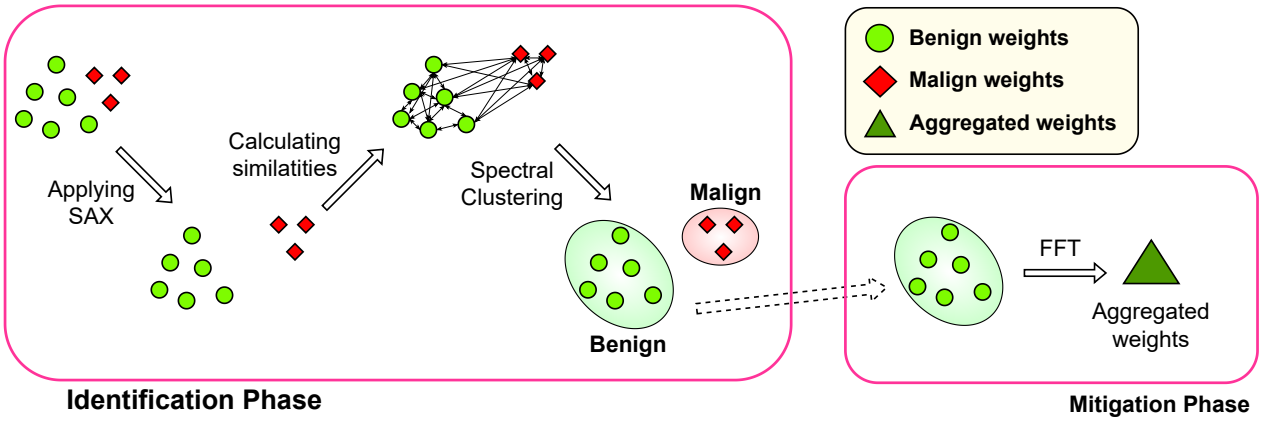


Figure 4.6. Visual description of the FLAegis process.

4.3.3. Results

The two defense mechanisms (both FedRDF and FLAegis) described aim to protect the model's performance while minimizing accuracy loss. We first analyze our robust aggregation function, FedRDF, which comprises two components: the K-S test and, if necessary, the FFT. We begin by evaluating the FFT's robustness as an aggregation function compared to standard approaches. We then demonstrate its effectiveness against a complex attack, the min-max attack, described in [49]. Fig. 4.7 illustrates that this attack significantly degrades the model's performance when using FedAvg, even with a low presence of malicious clients. Notably, the FFT achieves higher performance values than other methods, with the exception of the trimmed mean at 0% malicious nodes, where it equals the mean. It is important to note that while the absolute difference may appear small, all robust functions strive to achieve the maximum value attained by FedAvg in the absence of malicious activity. Given this upper bound, even slight improvements by the FFT over the other functions are more significant when considered

Algorithm 3 FLAegis description

Input: K set of clients, $(W_k)_{k \in K}$ weights of the clients
Output: Aggregated Weights

Identification phase

for $k \in K$ **do**
  $\tilde{W}_k = SAX(W_k)$
end for
for $k \in K$ **do**
  for $l \in K$ **do**
    $m_{kl} = cosine\_similarity(\tilde{W}_k, \tilde{W}_l)$
  end for
end for
$M = (m_{kl})_{k,l=1}^{|K|}$ similarity matrix
$(S)_1^L = Spectral\_Clustering(M)$
if $L > 1$ **then**
  $S_1, S_2 =$ K-Means(M)
  if $S_1 \geq S_2$ **then**
    $B = S_1$ are classified as benign clients
  else
    $B = S_2$ are classified as benign clients
  end if
else
  $B = K$ are classified as benign clients
end if

Mitigation phase

$W = FFT((W_b)_{b \in B})$
Server sends W to clients of K

in terms of relative error. Next, In Fig. 4.8, we analyse the performance of the complete version of our method using the threshold that reached the best results. To determine the optimal threshold, we employ a 5-fold cross-validation technique. In this scenario, we increase the presence of malicious clients in increments of 10%, rather than the 5% increments used in the previous figure. In this figure we achieve the purpose of our method, at 0% of malicious clients its value is higher than the FFT and close to FedAvg, and is close to the FFT values as the presence of malicious clients increases, although at the end the different became slightly greater since there may be rounds where FedRDF takes FedAvg instead of the FFT. Overall, this new function overcomes the current robust aggregation functions and enables us to protect the weights against poisoning attacks with minimal loss in accuracy since it adapts to all kinds of scenarios.

To protect the system against poisoning attacks, we also create a framework to detect specific malicious clients, FLAegis. In Fig. 4.9, we compare FLAegis with the mean, and two other frameworks to eliminate malicious clients, Foolsgold [161] and SignGuard [157]. In this figure, we compare the mentioned methods against 5 different attacks, min-max [49], min-sum [49], LIE [162], STATOPT [163], and label flipping [20]. In all attacks, FLAegis is the only one

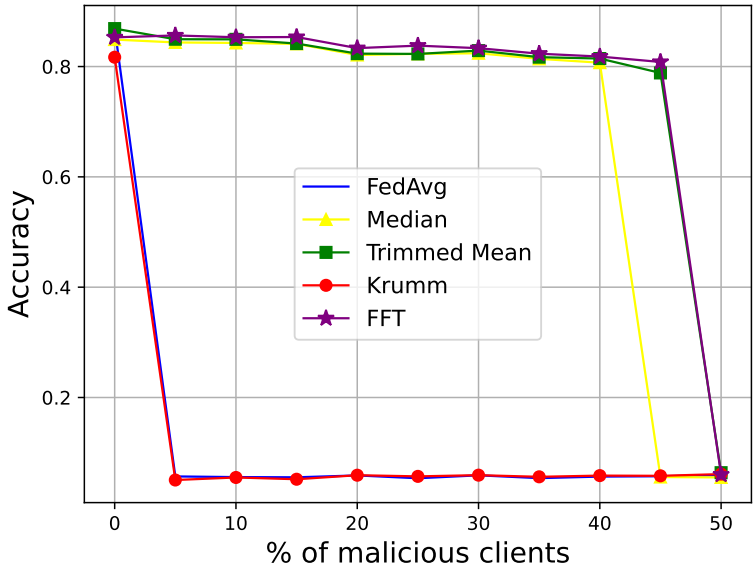


Figure 4.7. Results of different robust aggregation functions against the min-max attack.

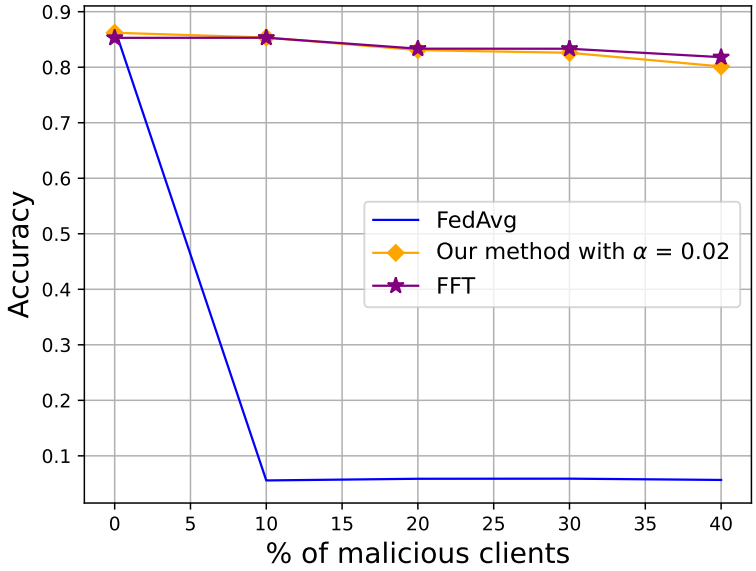


Figure 4.8. Comparison of FedRDF with FedAvg and FFT.

that remains stable. SignGuard is the second most resilient, but at certain attack intensities, its performance significantly diverges from that of FLAegis. FoolsGold, except for label flipping, experiences a rapid decline in performance. Our method implements the FFT as the aggregation function, which one may think is responsible for our notable results because as the FFT protects the system against malicious clients, it is not necessary for the previous process. However, this identification phase helps to reduce the number of malicious clients, which counteracts the weaknesses of these robust functions when there are large numbers of malicious clients. For that reason, in Fig. 4.10, we compare our method with FoolsGold and SignGuard but use the

FFT as the aggregation function in these 2 methods. Additionally, we add to the comparison by applying only the FFT to prove that the identification phase of our method is also needed. In this figure, although the performance of FoolsGold and SignGuard improves, their results still being lower than our method, especially when there is a high presence of malicious data in the complex attacks min-max and min-sum, as well as the performance of the FFT. From this figure, we obtain that a precise identification phase before making the aggregation is crucially important for achieving satisfactory results, which our method succeeds in.

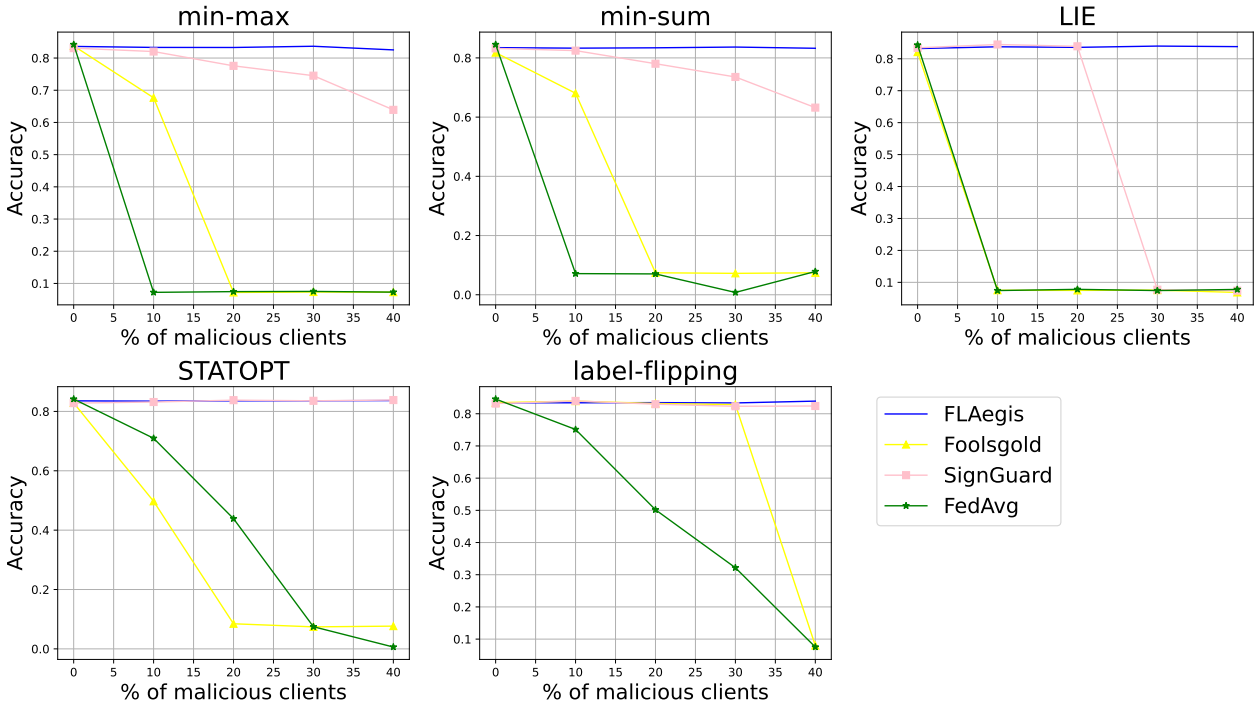


Figure 4.9. Comparison of our framework with SignGuard and FoolsGold against different attacks.

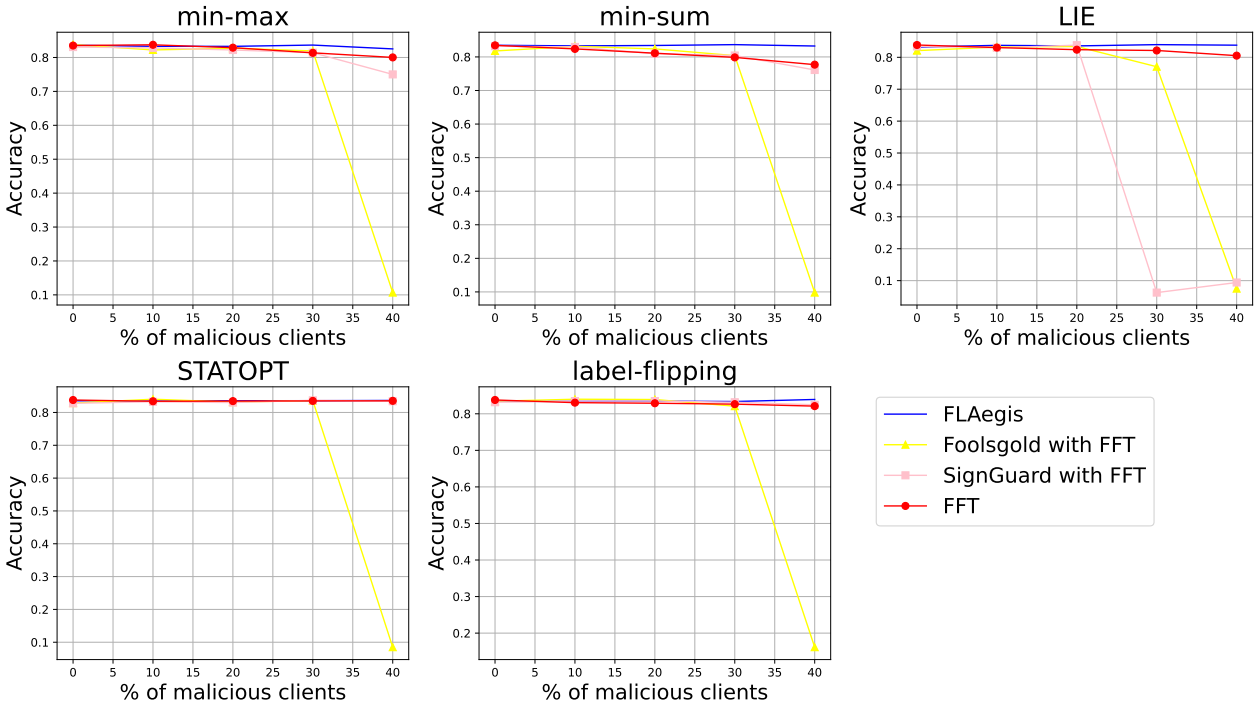


Figure 4.10. Comparison of our framework with SignGuard and FoolsGold using FFT against different attacks.

Chapter 5

Conclusions and Future work

This thesis focuses on detecting cyberattacks in IoT environments using FL, a cutting-edge technique presented to the community shortly before this PhD began. Throughout this work, several challenges related to FL were identified and addressed to develop more effective and efficient models in realistic scenarios. Some of the main challenges addressed in this thesis include to address the impact of non-iid distributions of the different datasets, to maintain the clients' individual datasets private, and to increase the robustness of FL settings through the identification of malicious clients and the development of a robust aggregation approach.

Our analysis of FL-enabled cyberattack detection approaches indicates that the current literature is focused on using and improving existing ML models used rather than on the elements that characterize FL, such as the distribution of clients' datasets or the implemented aggregation functions. Furthermore, through our comprehensive analysis, we realized that most of state-of-the-art works are based on outdated datasets and scenarios where clients' datasets are ideally balanced. This results in a decreasing performance of the model when it is applied in scenarios with non-iid data distributions. For this reason, our approaches are evaluated in scenarios where the dataset distributions follow a realistic distribution, with each client related to the traffic of a specific device or system. Our results show that the proposed methodology mitigated the impact of these non-iid distributions by using different resampling techniques and analyzing the use of several aggregation functions.

Moreover, while FL was mainly coined to mitigate the privacy concerns of typical centralized ML deployments, recent literature demonstrates that it is still possible to extract information from the different datasets throughout the weights' exchange between clients and the server. From our analysis, we found that although various techniques based on DP are implemented to protect the weights against inference attacks, there was a lack of thorough analysis on the impact of these different DP mechanisms and the use of several aggregation functions on the model's effectiveness. In this direction, we provide a comprehensive study of the impact of several DP mechanisms on two aggregation functions, FedAvg and Fed+. Our results provide different insights around the best combination of the different hyperparameters of such techniques to achieve a trade-off between privacy and accuracy.

In addition, from our analysis of the literature, we also found that defensive methodologies to protect typical FL settings against byzantine clients are computationally expensive, based on unrealistic assumptions, and tested against simple attacks. In response, we develop two different methods to protect the system against these kinds of threats. Firstly, we propose a new aggregation function called FedRDF, which can detect the presence of malicious clients and mitigate their impact on the aggregation using a function based on FFT. However, FedRDF cannot measure the number of malicious clients present or know who they are. In this direction, we also develop a framework (called FLAegis), which, unlike FedRDF, can identify and discard from the aggregation the specific malicious clients which are attempting to compromise the model with the poisoning attacks. Our results in both methods were tested against modern attacks, outperforming current methods.

In our study, we also identified that the current literature has not deeply explored unsupervised models. To address this gap, we adapted supervised techniques to FL scenarios, reducing the reliance on labeled datasets, which could be infeasible in real-world situations. Furthermore, using clustering techniques, we managed to reduce the number of FL clients while maintaining the final accuracy for the remaining clients, thereby reducing the bandwidth required for communication between clients and the server.

Finally, it is worth noting that most of the methods and techniques developed in the different papers proposed during the PhD are open source and are available on our GitHub[1], which facilitates other authors to replicate our work and help to develop new techniques in the future.

Considering the extensive knowledge obtained throughout this thesis, there are still many areas that need further exploration and analysis in future works.

- Deepening into client selection algorithms:

  In some of our works, we reduced the number of clients in the federated environment based on their characteristics, e.g., by the size of their dataset. In this direction, we plan to define client selection algorithms that operate during federated training to select clients that achieve better performance in terms of accuracy, thereby maximizing the performance of the aggregated model. Subsequently, using various techniques, we will transfer the model to the remaining clients to complete their training, saving time and bandwidth between the server and clients.

- Eliminating the presence of the server:

  In the field of security, existing methods assume that malicious parties can be either external parties or internal clients. However, the prevailing assumption of a reliable and benign server, responsible for orchestrating all operations, requires critical examination.

[1] `https://github.com/Enrique-Marmol?tab=repositories`

The server is the central entity that receives the weights and coordinates all federated processes. Thus, a malicious server may use these weights for malicious purposes without the clients being able to prevent it. In this direction, we plan to reevaluate the federated scheme, so that the server or aggregator do not become a single point of failure in the overall architecture. To achieve this, we need to redefine the aggregation function to be performed progressively through a random subset of clients each round. Through such decentralized approach, we aim to strengthen the FL environment against additional threats, ensuring the integrity and confidentiality of client data.

- Extending the algorithms of FedRDF and FLAegis:

  Although FedRDF and FLAegis achieve satisfactory results, we aim to go a step further. For FedRDF, we plan to upgrade how to establish the threshold for choosing between the aggregation functions. FedRDF currently relies on the K-S test and a threshold chosen by the user beforehand to decide which aggregation function to use, which may sometimes fail to choose between FedAvg or the FFT. We want to explore new statistical methods to replace the K-S test to improve the success rate of choosing FedAvg or the FFT, and implement a dynamic threshold similar to that used in AE. For FLAegis, we plan to redefine how the similarity matrix is calculated to better differentiate between malicious and benign clients. We will explore other functions to measure similarity and therefore increase the detection rate, especially in the case of 10% malicious clients, avoiding reliance on the FFT for aggregation to maximize final accuracy.

- Implementation of new types of models:

  As part of our future work, implementing new models within the federated environment is a key area to explore. Unsupervised models remain a significant gap in the field of IDS and MDS. We aim to implement more complex unsupervised models, such as innovative variants of AE to classify more than two classes. Additionally, the growing importance of Large Language Models (LLMs) is driving a significant change in AI research, offering unprecedented capabilities in natural language processing and understanding. LLMs require vast amounts of data for training, thus, the use of FL as an approach to the fine-tuning process of LLMs could reduce the cost of storing all the data in the same place since FL enable clients to use their dataset for training the LLM without sharing any information of them.

Capítulo 1

Introducción y motivación

El internet conecta una enorme cantidad de sistemas informáticos e infraestructuras de red. Esta creciente conectividad en línea se está realizando mediante la integración de diferentes tipos de dispositivos cotidianos, incluyendo sensores y actuadores que componen el Internet de las Cosas (IoT) [9] fomentando un gran número de servicios basados en datos, como la vigilancia del medio ambiente, los hogares inteligentes o aplicaciones sanitarias avanzadas. Para la realización de tales servicios, una enorme cantidad de datos, medidos en terabytes por segundo, son generados, procesados, intercambiados y almacenados por diferentes servicios en Internet. Esta hiperconectividad ha ampliado considerablemente la superficie de ataque que pueden explotar los ciberatacantes potenciales [10]. En esta dirección, la ciberseguridad emerge como la columna vertebral para corporaciones, gobiernos e individuos, permitiéndoles asegurar datos, ampliar sus negocios y mantener la privacidad [11]. De acuerdo al National Institute of Standards and Technology (NIST), la ciberseguridad se define como *el proceso de aplicación de medidas y políticas de protección para salvaguardar los datos, la información y las comunicaciones, programas, servidores e infraestructuras de red de accesos o modificaciones no autorizados*[1].

Uno de los enfoques más conocidos en ciberseguridad está representado por la detección de intrusiones, que normalmente se conoce como "*el proceso de monitorizar los eventos que ocurren en un sistema informático o red y analizarlos en busca de señales de posibles incidentes*" [12]. Los Intrusion Detection Systems (IDSs) se utilizan para automatizar este proceso [13]. Los primeros IDS basados en firmas, en los que los eventos monitorizados (por ejemplo, relacionados con el tráfico de red) se comparaban con los almacenados previamente. Sin embargo, con el aumento de la cantidad de dispositivos interconectados, así como la aparición de ataques más sofisticados, el Machine Learning (ML) se convirtió en un componente clave de los IDS *basados en anomalías* para detectar posibles comportamientos inusuales o desviaciones de los patrones normales [14, 15]. El ML es una forma de Inteligencia Artificial (AI) que puede extraer automáticamente información valiosa de grandes conjuntos de datos [16]. Por esta razón, la aplicación de técnicas de ML para la detección de ciberataques ha suscitado un gran interés en los últimos años en diversos campos, incluyendo el IoT [17].

[1] `https://csrc.nist.gov/glossary/term/cybersecurity`

El desarrollo de IDS basados en anomalías (o en ML) pretende mejorar la eficacia de los enfoques anteriores al permitir la detección de ciberataques desconocidos. De hecho, se ha demostrado que reducen las tasas de falsos positivos y se adaptan a la evolución de los patrones de ataque [14]. A pesar de sus conocidas ventajas, el despliegue de IDS basados en anomalías suele basarse en entornos centralizados, en los que una única entidad recopila datos de tráfico de red de varios sistemas para entrenar un modelo ML específico. En consecuencia, esta entidad tiene acceso a todo el tráfico de red de los sistemas y a los datos locales utilizados en el proceso de entrenamiento, lo que puede dar lugar a problemas de privacidad relacionados con la aplicación de los instrumentos jurídicos vigentes en materia de protección de datos, como el conocido Reglamento General de Protección de Datos (RGPD) [18]. Estos escenarios centralizados también podrían presentar varios problemas en torno al retraso asociado con el proceso de razonamiento centralizado (normalmente realizado en centros de datos en la nube). En este sentido, trabajos recientes [19–21] destacan la importancia de proteger la información personal de los clientes y la necesidad de desarrollar enfoques distribuidos de ML. Estos trabajos también discuten las limitaciones de los sistemas centralizados, como el ancho de banda de comunicación limitado, la conectividad de red intermitente y las estrictas restricciones de retardo [20].

Para abordar los problemas asociados con los enfoques de ML centralizados tradicionales, el **Federated Learning** (FL) fue introducido en 2016 por [22] como un enfoque de aprendizaje descentralizado y colaborativo. El FL se compone de varias fuentes de datos (denominadas *clientes* o *participantes*) y una entidad central conocida como *servidor* o *agregador*. Los clientes son responsables de recopilar y almacenar su propio conjunto de datos, que es único e inaccesible para los demás clientes. Estos clientes entrenan un modelo ML con su propio conjunto de datos. Durante el entrenamiento, el modelo ajusta sus variables internas (conocidas como parámetros) para ajustarse al conjunto de datos proporcionado y lograr una clasificación precisa. En esta tesis, los modelos utilizados son Regresion Logistica [23] (LR) y Redes Neuronales [24] (NN). Para la LR, estos parámetros son los coeficientes que multiplican las variables de entrada en la ecuación logística, que pueden denominarse pesos. En el caso de las NN, las variables son los pesos que conectan las distintas neuronas entre las capas del modelo. Por lo tanto, en aras de la simplicidad, nos referiremos a estas variables como *pesos* a lo largo de este documento. El entrenamiento local da como resultado un conjunto de pesos, que se envían al servidor. A continuación, esta entidad *agrega* los pesos enviados por los distintos clientes. Una vez agregados los pesos, el servidor envía el resultado de dicha agregación a los clientes para que continúen su entrenamiento con los pesos agregados.

El proceso general de entrenamiento de FL se refleja en los cuatro pasos principales representados en la Fig. 1.1. Durante el paso (1), los clientes comienzan el entrenamiento del modelo ML utilizando sus propios datos. A continuación, después de varias iteraciones de entrenamiento llamadas *épocas*, en el paso (2), los clientes envían los pesos producidos durante el entrenamiento al servidor. A continuación, el servidor, en el paso (3), una vez que ha recibido todos los pesos, los agrega utilizando lo que se denomina una *función de agregación* F. Después,

cuando la agregación se ha completado, el servidor envía los pesos resultantes de nuevo a los clientes para que continúen su entrenamiento en el paso (4). Todo este proceso se denomina *ronda*. Finalmente, el proceso continúa durante un número predefinido de rondas o hasta que los pesos convergen. Durante este proceso, los conjuntos de datos locales de los clientes no se comparten; por lo tanto, el FL proporciona un enfoque descentralizado que preserva la privacidad para entrenar modelos ML.

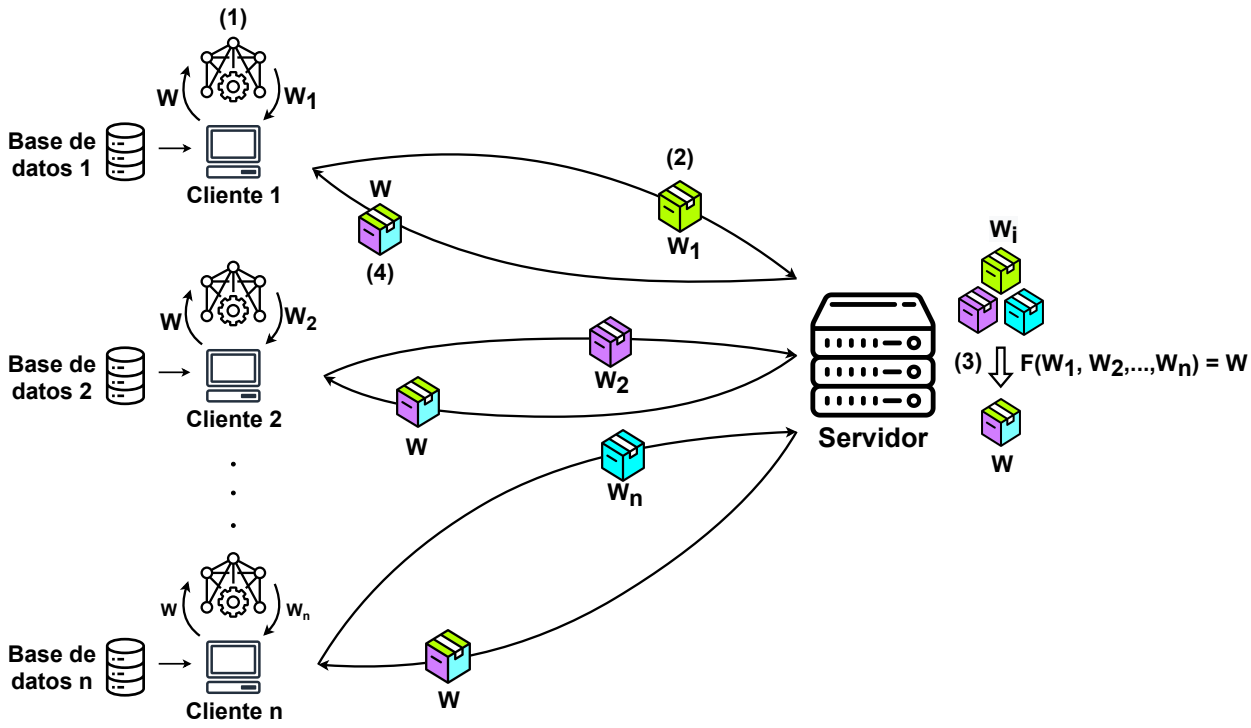


Figura 1.1. Descripción gráfica del proceso de entrenamiendo del FL.

En el contexto de los IDS basados en anomalías, FL proporciona varias ventajas, entre ellas la reducción de la carga computacional del servidor central de procesamiento, la salvaguarda de la privacidad de los datos y la optimización del uso del ancho de banda, [25, 26]. De hecho, como hemos analizado durante esta tesis [6], el FL se ha convertido en un enfoque popular para la próxima generación de IDSs [25, 27, 28]. En este contexto, esta tesis se centra específicamente en IDS habilitados para FL en escenarios IoT debido a la adopción generalizada de IoT en áreas como Internet de los Vehículos, ciudades inteligentes y Sistemas Ciber-Físicos en los últimos años [28]. Además, durante el desarrollo de la tesis, también profundizamos en *misbehavior detection*, que puede ser visto como un tipo específico de detección de intrusiones en el contexto vehicular. En concreto, se refiere a la detección de vehículos que transmiten información falsa que no puede ser detectada por los mecanismos criptográficos típicos [29, 30]. Por lo tanto, llamamos Misbehavior Detection Systems (MDS) a los sistemas que automatizan los enfoques de detección de malos comportamientos. En tal escenario, FL mitiga los problemas de privacidad relacionados con la continuidad de los datos de posición de los vehículos, a la vez que proporciona un enfoque para entrenar de forma colaborativa un modelo de detección de vehículos que se

comportan de forma inadecuada [31].

Aunque el objetivo principal de esta tesis giraba inicialmente en torno a la aplicación de FL para detección de ciberataques (y, en particular, en escenarios IoT), nuestro exhaustivo análisis durante el transcurso de este doctorado [1, 6] identificó varios retos asociados al despliegue de escenarios FL en términos de heterogeneidad de datos/dispositivos, seguridad y privacidad, entre otros. Además, nuestro análisis reveló que la mayoría de los trabajos existentes sobre IDSs basados en FL se basan en técnicas de aprendizaje supervisado [32–34] que requieren conjuntos de datos etiquetados para el entrenamiento. El proceso de etiquetado suele requerir la intervención humana, lo que hace que la obtención de ejemplos etiquetados necesarios para lograr una generalización adecuada requiera muchos recursos y mucho tiempo [35, 36]. Además, en el contexto de los IDS basados en FL, muchos de los enfoques propuestos adolecen de distribuciones de datos poco realistas entre los participantes (como se expone en [6]), conjuntos de datos obsoletos, o se basan en enfoques de clasificación binaria, en los que los datos de tráfico se clasifican simplemente como de ataque o benigno [37]. Estas distribuciones artificiales conducen a resultados poco realistas, ya que los escenarios reales se caracterizan por datos no independientes e idénticamente distribuidos (no-iid). Uno de los principales enfoques para hacer frente a esta heterogeneidad de los datos es la aplicación de funciones de agregación alternativas. De hecho, el método de agregación común es FedAvg [22], que consiste en aplicar la media a los pesos recibidos en una determinada ronda de entrenamiento. Como demuestran varios trabajos [38], el uso de FedAvg en datos no-iid conduce a una disminución de la convergencia del modelo. La degradación del rendimiento puede atribuirse principalmente a la divergencia de pesos de los modelos locales, que se produce porque los pesos agregados no se ajustan bien cuando se aplican a los datos de los clientes. Para hacer frente a este aspecto, existen funciones de agregación alternativas, como Fed+ [39] y FedProx [40], que pretenden mitigar el impacto de las distribuciones de datos no-iid. En esta tesis, cuantificamos el impacto de utilizar distintas funciones de agregación y aplicamos técnicas de re demostración (como SMOTE-Tomek [41]) para reducir la heterogeneidad estadística entre los clientes.

Por otra parte, a pesar de que el FL se propone principalmente como un enfoque de preservación de la privacidad para la creación del modelo de ML sin comprometer la privacidad de los conjuntos de datos de los clientes, sigue planteando importantes problemas de privacidad. Uno de los principales problemas es que los pesos de los clientes pueden ser interceptados durante la comunicación entre el servidor y los clientes. De hecho, en un entorno FL típico, el servidor suele tener acceso a las ponderaciones cargadas por los clientes a lo largo de las rondas. En consecuencia, estos podrían utilizarse para lanzar varios ataques destinados a inferir información privada de los datos de entrenamiento [42]. Por lo tanto, la implementación de técnicas robustas para proteger la privacidad de los clientes es crucial. En el ML centralizado, el Differential Privacy (DP) [43] puede emplearse para abordar estos problemas de privacidad. El DP ofusca las actualizaciones del modelo con la adición de ruido, proporcionando así garantías estadísticas de privacidad frente a los adversarios. Sin embargo, la literatura existente carece de

un análisis en profundidad del impacto de los métodos de DP existentes sobre la convergencia del FL y el rendimiento en diferentes funciones de agregación. Durante esta tesis, realizamos un análisis exhaustivo de las técnicas de DP contemporáneas, y su impacto en el rendimiento del modelo considerando diferentes funciones de agregación. Además, también examinamos el equilibrio entre la cantidad de ruido introducida por el DP y la pérdida de precisión resultante. Mediante la evaluación de esta equilibrio, pretendemos seleccionar el mecanismo de DP óptimo que garantice el máximo nivel de privacidad para los conjuntos de datos de los clientes, manteniendo al mismo tiempo unos niveles de precisión aceptables.

En el contexto de IDS y MDS, el objetivo principal de FL es entrenar de forma colaborativa un modelo para detectar ciberataques. Sin embargo, el propio FL es susceptible de sufrir ataques de envenenamiento [44], ya que algunas partes pueden actuar como clientes bizantinos, es decir, clientes que actúan maliciosamente e intentan perjudicar la convergencia del modelo. El propósito de estos clientes bizantinos es producir un retraso en la convergencia del modelo enviando pesos falsos al servidor (envenenamiento del modelo), o alterando directamente el conjunto de datos de los clientes (envenenamiento de los datos). Entonces, los clientes benignos se vuelven a entrenar con ponderaciones corruptas que podrían conducir a una clasificación errónea de las muestras tras el entrenamiento federado. De hecho, estudios previos confirman que el FL es susceptible a ataques de envenenamiento, especialmente enfatizando el impacto cuando se utiliza FedAvg como enfoque de agregación [45]. En este contexto, uno de los objetivos de esta tesis es proteger el entorno federado contra este tipo de ataques. En concreto, se diseñan e implementan dos estrategias complementarias. En primer lugar, diseñamos e implementamos un novedoso método de agregación basado en la Transformada de Fourier Rápida (FFT) [46], como alternativa a técnicas de agregación robustas bien conocidas, como la mediana, la media recortada [47] o Krum [48]. Nuestro enfoque pretende abordar algunos de los puntos débiles de estos métodos, que normalmente asumen que el número de atacantes es conocido, y no proporcionan protección contra ataques de envenenamiento sofisticados de modelos en los que varios clientes actúan en connivencia [49]. En segundo lugar, exploramos nuevos métodos para identificar a estos clientes maliciosos, de modo que estos nodos comprometidos puedan descartarse para futuras rondas de entrenamiento. De hecho, aunque unas funciones de agregación robustas podrían mitigar el impacto de los ataques de envenenamiento, la identificación de dispositivos específicos comprometidos podría ayudar en el desarrollo de posibles técnicas de mitigación y estrategias de defensa [50].

En resumen, esta tesis pretende desarrollar un entorno FL para la detección de ciberataques en escenarios IoT. Además del análisis de diferentes modelos ML, conjuntos de datos y funciones de agregación en este contexto, también abordamos desafíos apremiantes relacionados con la heterogeneidad de datos, la privacidad y la seguridad para mejorar la robustez y privacidad general de los sistemas FL. Este trabajo se organiza como un compendio de cuatro artículos de investigación de gran impacto, que forman el cuerpo principal de la tesis. La información presentada está disponible tanto en inglés como en español. Esta primera sección

introduce el contexto y la motivación del tema de esta tesis. La segunda sección describe los principales objetivos de la investigación propuesta y su relación con las cuatro publicaciones que componen la tesis. La tercera sección expone los modelos de referencia, técnicas y trabajos relacionados en torno al tema de la tesis. La cuarta sección describe la metodología seguida para abordar los retos anteriormente descritos y alcanzar los objetivos propuestos. Finalmente, la quinta sección concluye esta tesis y detalla los futuros temas de nuestro trabajo.

Capítulo 2

Objetivos

Como ya se ha descrito en la sección anterior, el FL ha surgido como un enfoque prometedor para entrenamiento de modelos ML en el contexto de la detección de ciberataques para construir la próxima generación de IDSs. Sin embargo, como ya se ha analizado durante esta tesis [1, 6], la mayor parte de la literatura existente se basa en escenarios de FL poco realistas en los que los conjuntos de datos se dividen artificialmente entre un cierto número de clientes sin tener en cuenta los retos adicionales asociados al despliegue de escenarios de FL. Por lo tanto, esta tesis tiene como objetivo desarrollar un IDS realista basado en anomalías aprovechando aprendizaje federado como método colaborativo para el entrenamiento de varios modelos ML. En este sentido abordamos varios retos significativos en torno al propio FL, considerando la heterogeneidad de los datos y el impacto de la aplicación de diferentes enfoques de agregación, y las cuestiones por la privacidad, que se mitigan mediante la aplicación de diferentes técnicas de DP. Además, desarrollamos una función de agregación robusta, que se evalúa y compara con los enfoques existentes frente a diferentes ataques de envenenamiento.

Para alcanzar estas metas, se definen los siguientes objetivos:

- **O1**: Identificar los retos actuales y las tendencias futuras de los IDS basados en FL.
- **O2**: Abordar la heterogeneidad de los datos de las configuraciones del FL comunes mediante el análisis de diferentes funciones de agregación y técnicas de remuestreo.
- **O3**: Proteger la privacidad de los conjuntos de datos de los clientes contra ataques de inferencia durante el entrenamiento federado.
- **O4**: Mitigar el impacto de los ataques de envenenamiento en entornos FL.
- **O5**: Analizar el uso de técnicas de aprendizaje no supervisado en el contexto de FL para la detección de ciberataques.
- **O6**: Identificar potenciales clientes maliciosos durante el entrenamiento federado.
- **O7**: Aliviar la sobrecarga de comunicación en entornos FL reduciendo el número de clientes.

La Tabla 2.1 presenta la correlación entre los objetivos y los artículos que componen esta tesis. Esta tabla proporciona una breve descripción del enfoque usado para alcanzar cada objetivo.

Objetivo	**Publicaciones**	**Enfoque**
O1	**[1]**, [6]	Se realiza una revisión exhaustiva de la literatura actual sobre los desafíos asociados con el FL en sistemas IDS. Algunos de los más críticos identificados incluyen problemas de privacidad (por ejemplo, relacionados con ataques de inferencia), problemas de seguridad (incluyendo ataques de envenenamiento) y heterogeneidad de datos entre clientes.
O2	**[1–4]**, [5, 7, 8]	Este objetivo es común en varios artículos. Para abordar escenarios no-iid donde los datos de los clientes están distribuidos de manera heterogénea, se aplican varios métodos en estos trabajos, incluyendo técnicas de remuestreo local, funciones de agregación alternativas y personalización de modelos de ML.
O3	**[2]**	Un entorno FL podría sufrir ataques de inferencia, donde se puede inferir información sobre los datos de los clientes a partir del intercambio de pesos durante las rondas de entrenamiento. Para prevenir este problema, nuestro trabajo presenta un análisis de diferentes técnicas de DP para ofuscar dichos pesos, evitando así la filtración de información. El trabajo también evalúa el equilibrio entre diferentes niveles de privacidad y la precisión del modelo para identificar la técnica de DP óptima.
O4	[8]	Se propone una nueva función de agregación basada en la Transformada Rápida de Fourier (FFT) para mitigar el impacto de los ataques de envenenamiento en entornos FL con la presencia de clientes bizantinos.
O5	[7]	Una técnica novedosa basada en Gaussian Mixture Models (GMMs) y Autoencoders Variacionales (VAEs) para el entrenamiento no supervisado en FL de un modelo de misbehavior detection.

Objetivo	Publicaciones	Enfoque
O6	**[3]**	Analizamos el uso de un estándar reciente para descartar dispositivos que se comportan de manera indebida durante el proceso de entrenamiento en FL. La exclusión de estos clientes tiene como objetivo proteger el entorno FL durante el entrenamiento.
O7	**[4]**, [5]	Proponemos un enfoque de selección de clientes para reducir la sobrecarga de cálculo y comunicación requerida en entornos FL. Los enfoques de selección propuestos se basas en la longitud del conjunto de datos de los clientes y en el equilibrio entre la precisión y el tiempo requerido para el proceso de entrenamiento.

Tabla 2.1

Descripción de los objetivos de esta tesis.

Capítulo 3

Estado del arte

Esta sección presenta el estado del arte relacionado con el núcleo de esta tesis que se divide en dos partes: el Trasfondo y los Trabajos relacionados. La sección 3.1 ofrece una breve descripción de los componentes fundamentales del FL, incluidos los modelos específicos, las funciones de agregación y los conjuntos de datos empleados a lo largo de esta tesis. A continuación, en la sección 3.2 se ofrece un breve resumen de la investigación actual sobre FL para la detección de ciberataques. Además, presenta un análisis de los estudios relacionados que abordan algunas de las principales áreas de FL exploradas en esta tesis: la heterogeneidad de los datos, la seguridad y los problemas de privacidad en este tipo de entornos. Dicho análisis pretende proporcionar una visión clara de la relación entre los trabajos existentes con los objetivos y resultados de esta tesis. Cabe señalar que se puede encontrar una descripción ampliada sobre estos conceptos en [6].

3.1 — Trasfondo

Los principales componentes de un entorno FL están representados por el modelo de ML para el entrenamiento, los conjuntos de datos empleados por los clientes y la función de agregación implementada por el servidor. Esta sección proporciona un análisis detallado de estos elementos clave y la descripción de las alternativas específicas empleadas en esta investigación.

3.1.1. Modelos de ML

El objetivo principal del uso del FL para la detección de ciberataques es entrenar de forma colaborativa un modelo de ML para distinguir entre comportamiento normal o intencionado y ataques específicos. Dependiendo de la naturaleza de los datos de entrenamiento, los modelos de ML pueden clasificarse principalmente en dos tipos: supervisados y no supervisados.

Aprendizaje supervisado

Las técnicas de aprendizaje supervisado dependen de datos etiquetados para el entrenamiento del modelo. El conjunto de datos de entrenamiento incluye entradas emparejadas con sus salidas correctas, que ayudan al modelo a aprender afinando sus parámetros. Este conjunto de datos instruye al modelo sobre la relación entre entradas y salidas, lo que le permite predecir con precisión cuando se le presentan nuevos datos. Algunas de las técnicas de aprendizaje supervisado más utilizadas son Regresion Logística (LR) [23], Decision Trees (DT) [51], Random Forest (RF) [52], Redes Neuronales (NN) [24] y Support Vector Machines (SVM) [53]. Durante el desarrollo de esta tesis, los modelos utilizados son LR y varios tipos de NN: Multilayer Perceptron (MLP) [54], Long Short Term Memory (LSTM) [55], y Convolutional Neural Network (CNN) [56]. La principal justificación del uso de estos modelos es que esta tesis no se centra en los modelos de ML empleados, sino más bien en aspectos específicos de los despliegues de FL como el impacto del uso de diferentes funciones de agregación y la heterogeneidad de los datos. En consecuencia, utilizamos modelos sencillos y bien conocidos para priorizar estos aspectos críticos en el FL. En concreto, en nuestros trabajos iniciales [1, 2] se eligió LR debido a su sencilla definición y al hecho de que sus pesos pueden agregarse en FL directamente, a diferencia de los hiperplanos de SVM, o las estructuras de árboles de la RF, y DT. En cuanto a las NN, además de que su estructura de parámetros es menos compleja que la de otros modelos (similar a la de LR), en la literatura actual, como se describe en [6], la mayoría de los trabajos relacionados utilizan modelos NN. Por lo tanto, esto permite realizar comparaciones más precisas entre sus resultados y los nuestros.

Basándonos en los aspectos anteriores, ofrecemos una descripción breve de los modelos empleados a lo largo de la tesis. La **LR** es un método estadístico utilizado para tareas de clasificación en ML. Modela la probabilidad de que una entrada pertenezca a una clase utilizando la función logística, produciendo salidas entre 0 y 1. La LR es eficaz para relaciones lineales entre características y resultados. Por otro lado, los MLP, los LSTM y las CNN son arquitecturas clave en el campo de las NN. Los **MLP**, la forma más sencilla de NN, constan de varias capas de neuronas, cada una de ellas totalmente conectada a la siguiente, y se utilizan principalmente para tareas que requieren la extracción y clasificación de características. Las **LSTM**, Recurrent Neural Network (RNN), están diseñadas para procesar secuencias de datos manteniendo dependencias a largo plazo, lo que resuelve eficazmente el problema del gradiente evanescente en las RNN tradicionales y las hace adecuadas para la predicción de series temporales y el procesamiento del lenguaje natural. Las **CNN**, especializadas en el procesamiento de datos reticulares como las imágenes, emplean capas convolucionales para aprender de forma automática y adaptativa jerarquías espaciales de características, destacando en tareas de reconocimiento y clasificación de imágenes. Juntas, estas arquitecturas forman el núcleo de los modelos de ML, cada una de las cuales se adapta de forma única a distintos tipos de datos y dominios de problemas.

Aprendizaje no supervisado

El aprendizaje no supervisado corresponde a algoritmos de ML capaces de clasificar muestras sin necesidad de un conjunto de datos etiquetados. En concreto, el aprendizaje no supervisado se emplea para descubrir la estructura y la jerarquía dentro de los datos mediante el uso de muestras de datos sin necesidad de etiquetas de verdad sobre el terreno. La representación del conocimiento derivada de este proceso puede servir de base para un modelo profundo [35]. Debido a su naturaleza más compleja, el número de trabajos que implementan el aprendizaje no supervisado en escenarios FL es escaso. De hecho, según nuestro análisis exhaustivo sobre el uso de modelos ML en IDS habilitados para FL, sólo el 16 % de los trabajos relacionados emplean técnicas de aprendizaje no supervisado. Los métodos no supervisados utilizados durante esta tesis son: dos tipos de NN no supervisadas, autoencoders (AE) [57], y AE Variacional (VAEs) [58]; y varias técnicas de clustering, en concreto, Gaussian Mixture Models (GMM) [59], K-means [60], y spectral clustering [61].

Un **AE** es un tipo de MLP en el que las dimensiones de entrada y salida son las mismas, y su estructura es simétrica. El objetivo de los AE es reproducir fielmente los datos originales sin necesidad de etiquetas. Constan de un codificador que comprime los datos de entrada en un código de dimensiones inferiores (espacio latente) y de un decodificador que reconstruye este código de vuelta a los datos originales. La diferencia entre la entrada y la salida se denomina error de reconstrucción (RE), y suele medirse mediante el error cuadrático medio. En ciberseguridad, para detectar anomalías, los AE se entrenan únicamente con datos benignos, y luego, en función del RE de las muestras, si este valor supera un determinado umbral, se considera una anomalía. Un **VAE** es un tipo de AE con una distribución de codificación regularizada, que suele aproximar el espacio latente a una distribución normal estándar durante el entrenamiento. Esta arquitectura permite a los VAE generar nuevas muestras de datos plausibles mediante el muestreo del espacio latente. La innovación clave de los VAE es su capacidad para aprender un espacio latente suave y continuo que permite interpolaciones significativas entre puntos de datos.

El clustering agrupa objetos de datos utilizando una medida de similitud, agrupando elementos con alta similitud intra-cluster (elementos dentro del cluster) y baja similitud inter-cluster (elementos de diferentes clusters). **K-means**, un algoritmo de agrupación particional simple, encuentra K clusters no solapados representados por sus centroides. El proceso consiste en seleccionar K centroides iniciales, asignar puntos de datos al centroide más cercano y actualizar los centroides de forma iterativa hasta la convergencia. K-means asigna cada punto a un único conglomerado, lo que puede causar ambigüedades con cluster solapados. En cambio, los **GMM** utilizan un enfoque probabilístico que permite que los puntos de datos pertenezcan a varios clusters. En concreto, un GMM es un modelo probabilístico que asume que todos los puntos de datos se generan a partir de una mezcla de un número finito de distribuciones gaussianas con parámetros desconocidos. El **spectral clustering** utiliza una matriz de similitud

para crear un grafo y, a continuación, mediante el cálculo de sus vectores propios, calcula los clusters finales. Los métodos tradicionales como K-means y GMM funcionan bien en escenarios convexos, mientras que el spectral clustering destaca en formas complejas y no lineales y no requiere un número predefinido de clusters.

3.1.2. Funciones de agregación

La función de agregación es un elemento central del FL. Agrega los pesos de todos los clientes en otros nuevos que se distribuyen de nuevo a los clientes para reanudar su entrenamiento local. En la Tabla 3.1, mostramos algunas de las principales funciones de agregación consideradas en la literatura. En esta tabla, explicamos las principales características de estas funciones, incluyendo sus ventajas e inconvenientes. La función de agregación más utilizada es la denominada **FedAvg** [22], que consiste en promediar los pesos. Este sencillo enfoque es ampliamente utilizado por los trabajos existentes sobre FL. De los artículos analizados en nuestro trabajo [6], la mayoría de los enfoques (87/104 o el 83,7 %) utilizan FedAvg o se basan principalmente en ella como función de agregación. Sin embargo, en escenarios con datos no-iid, varios estudios demostraron las limitaciones de este enfoque [1], donde el rendimiento de FedAvg se degrada, así como en términos de seguridad y eficiencia [38, 62].

Para hacer frente a estas limitaciones, se han desarrollado técnicas avanzadas de agregación. Para la heterogeneidad de datos, **FedProx** [40] introduce un término proximal en la función de pérdida de los clientes para mitigar los efectos de las actualizaciones locales heterogéneas. Aunque maneja con eficacia los entornos no-iid, requiere mayores recursos computacionales e impone demandas de rendimiento adicionales a los dispositivos implicados. Basándose en los principios de FedProx, **Fed+** [39] (o Fedplus) modifica la función de pérdida de FedAvg añadiendo una función de penalización en cada cliente para eliminar la restricción de que los pesos de todos los clientes deben converger al mismo punto. Este enfoque da lugar a un conjunto de algoritmos diseñados para manejar la heterogeneidad de los datos. La agregación se produce en dos fases: en primer lugar, el servidor agrega los pesos y, a continuación, los clientes promedian este valor del servidor con sus propios pesos. El resultado es que cada cliente tiene su propio modelo, lo que se conoce como técnicas de personalización [63].

Esta flexibilidad permite un enfoque más adaptado a los distintos escenarios de agregación, pero aumenta la complejidad. Además, la eficacia de Fed+ y FedProx depende en gran medida de la selección adecuada de la constante de penalización. **FedMA** [64], se centra en la invariancia de permutación de las neuronas a través de un proceso de emparejamiento de las NN de los clientes. Esta técnica se adapta al tamaño del modelo global y a la heterogeneidad de los datos, mejorando la convergencia y requiriendo menos rondas de entrenamiento. No obstante, el cálculo de la matriz de permutación puede aumentar el tiempo total de cálculo. Además, FedMA se considera específicamente para CNN y LSTM. Para reducir el coste compu-

tacional, **FedPAQ** [65] mejora el proceso de agregación introduciendo promedios periódicos, participación parcial de dispositivos y paso de mensajes cuantizados. Estas modificaciones reducen significativamente la sobrecarga de comunicación y cálculo, haciendo que el proceso sea más eficiente. Sin embargo, la implementación de FedPAQ requiere funcionalidades adicionales tanto en el lado del agregador como en el del cliente, y sigue teniendo problemas con las distribuciones de datos no-iid.

Además, existe un conjunto de funciones de agregación centradas en la seguridad y en aumentar la robustez del entorno FL y que podrían denominarse *funciones de agregación robustas*. El objetivo principal de estos enfoques es mitigar el impacto de los ataques de envenenamiento lanzados por clientes maliciosos o bizantinos. Las funciones de agregación robustas más conocidas son la mediana [47], la media recortada [47] y Krum [48]. La mediana, como su nombre indica, utiliza la mediana (en lugar de la media) para agregar los pesos. La media recortada es un método alternativo para calcular la media. Consiste en elegir un valor n inferior a la mitad del número total de clientes y, a continuación, eliminar los valores n más bajos y los valores n más altos por coordenadas y calcular la media a partir de los valores restantes. La función de agregación de Krum se deriva de la función de Krum y funciona eligiendo el número f que representa el número de clientes maliciosos y, a continuación, seleccionando el peso del cliente con la suma más baja entre sus $K - f - 2$ vecinos más cercanos, donde K es el número de clientes. La principal limitación de estas funciones de agregación es su vulnerabilidad frente a ataques sofisticados. De hecho, aunque pueden mitigar el impacto de ataques simples con un bajo porcentaje de nodos maliciosos [50], su rendimiento se degrada significativamente ante ataques más avanzados que implican la confabulacion entre varios nodos. Además, en ausencia de ataques maliciosos, la precisión de estas funciones es inferior a la alcanzada con FedAvg [8]. Además, cuando se enfrentan a un alto porcentaje de clientes maliciosos, estas funciones muestran un descenso sustancial del rendimiento. Además, la media recortada y Krum requieren determinar el número de clientes maliciosos, lo que en casos reales podría ser inviable.

Estas estrategias de agregación avanzadas ponen de relieve los esfuerzos que se están realizando para superar los retos inherentes de FL, en particular los relacionados con la heterogeneidad de los datos y la resistencia a los ataques de envenenamiento. Cada método ofrece ventajas únicas e inconvenientes potenciales, lo que subraya la necesidad de considerar cuidadosamente la selección de una técnica de agregación que se adapte a aplicaciones FL específicas.

3.1.3. Bases de datos

A lo largo de este doctorado, se han considerado varios conjuntos de datos relacionados con la detección de ciberataques (tanto IDS como MDS). En [5], el conjunto de datos considerado está relacionado con el consumo de energía, por lo que no se describe. Además, utilizamos cuatro conjuntos de datos: tres relacionados con la detección de ciberataques y uno que sirve como

Técnica	Núcleo	Ventajas	Desventajas
FedAvg	Basado en el promedio ponderado de los pesos actualizados proporcionados por los clientes	Es ampliamente utilizado debido a su bajo nivel de complejidad	Presenta problemas de convergencia en entornos de FL con distribuciones de datos no IID
FedProx	Agrega un término proximal para limitar el impacto de las diferentes actualizaciones locales	Aborda tanto la heterogeneidad de datos considerando entornos no IID como la heterogeneidad de dispositivos	Aumenta los requisitos de cálculo y su rendimiento depende en gran medida de la selección adecuada del término proximal
Fed+	Similar al término proximal anterior, también se agrega una constante de penalización, pero se pueden emplear diferentes funciones de agregación (más allá del promedio)	Aumenta el nivel de flexibilidad al considerar diferentes funciones de agregación más allá del promedio	Una mayor flexibilidad conlleva un costo en complejidad, y el rendimiento también depende de la elección correcta de la constante de penalización
FedMA	Se basa en la invariancia de permutación de las neuronas a través de un proceso de emparejamiento de las redes neuronales de los clientes para realizar un promedio por capas	Se adapta al tamaño del modelo global y la heterogeneidad de datos, mejorando la convergencia, mientras requiere menos rondas de entrenamiento	El cálculo de la matriz de permutación puede aumentar el tiempo de cómputo, y es específico para CNNs y LSTM
FedPAQ	Se basa en promediar las actualizaciones del modelo, pero proporciona un mayor grado de eficiencia a través del promedio periódico, la participación parcial de dispositivos y el paso de mensajes cuantizados	Reduce la sobrecarga de comunicación y cálculo	Requiere funcionalidad adicional tanto en el agregador como en los clientes, y aún presenta problemas con entornos ni-iid
Mediana	De manera coordinada, calcula la mediana de los pesos recibidos	Protege el sistema contra valores atípicos y ciertos atacantes	Fácilmente manipulable por atacantes, y tiene peor rendimiento que FedAvg en caso de no haber atacantes
Media recortada	Calcula el promedio de los pesos eliminando los n valores más altos y más bajos	Resiste mejor la presencia de clientes maliciosos en comparación con FedAvg	Depende de la elección de n, y hereda los problemas de FedAvg en el caso de datos no-iid
Krum	Selecciona los pesos que minimizan la distancia con sus $K-f-2$ vecinos	Puede superar ciertos ataques en comparación con FedAvg	Peor rendimiento que FedAvg en escenarios sin ataques, asume que se conoce el número de atacantes, y es débil contra ciertos ataques sofisticados.

Tabla 3.1

Descripción de varias funciones de agregación frecuentemente consideradas en FL.

modelo de referencia para probar aspectos relacionados con la seguridad en FL. En la Tabla 3.2, proporcionamos una breve descripción de los conjuntos de datos de detección de ciberataques, incluyendo los ataques considerados, cómo se dividen entre los diferentes clientes, o la relación tráfico benigno/ataque.

El primer conjunto de datos es **ToN_IoT** [66], que se construye utilizando un banco de pruebas IoT/IIoT (Industrial IoT) que incluye nodos edge/fog y componentes en la nube para replicar un entorno de producción IoT/IIoT. ToN_IoT está diseñado para recopilar y analizar fuentes de datos mixtas de entornos IoT e IIoT. Contiene datos heterogéneos recogidos de diversas fuentes, incluidos datos de telemetría de dispositivos conectados, registros de sistemas Windows y Linux y tráfico de red del sistema. Este enfoque permite la detección de ataques adicionales más allá del nivel de red en dichos entornos. En concreto, utilizamos el conjunto de datos CIC-ToN-IoT [67], que se generó utilizando la herramienta CICFlowMeter [68] a partir de los archivos pcap originales del conjunto de datos ToN_IoT. Esta herramienta se utilizó para extraer 83 características, que luego se redujeron mediante la eliminación de las que tenían valores no numéricos (por ejemplo, ID de flujo) en 79. Posteriormente, separamos las muestras de todo el conjunto de datos según la dirección IP de destino y seleccionamos las 10 direcciones IP con más muestras. Este subconjunto representa el 82,29 % del conjunto de datos. Los ataques contenidos en este conjunto de datos son: DoS, Escaneo, DDoS, Backdoor, MITM, Contraseña, Inyección, Ransomware y XSS.

El segundo conjunto de datos utilizado es VeReMi [69] (Vehicular Reference Misbehavior). VeReMi es un conjunto de datos etiquetados que recoge los registros de mensajes de vehículos infractores y benignos. Estos registros incluyen marcas de tiempo de recepción, tiempos de transmisión reclamados, remitentes reclamados, IDs de mensajes únicos, posiciones GPS (x, y, z), valores RSSI, ruido de posición y vectores de ruido de velocidad para cada vehículo receptor en varios escenarios. El conjunto de datos se generó mediante 225 simulaciones con el simulador VEINS [70], considerando 5 tipos de ataques de falsificación de posición, 3 niveles de densidad de vehículos (baja, media y alta) y 3 niveles de densidad de atacantes (10 %, 20 %, y 30 %). Cada conjunto de parámetros se repitió 5 veces para garantizar la aleatoriedad. Cada simulación del conjunto de datos se basa en el escenario Luxembourg SUMO Traffic (LuST) [71], que garantiza una representación completa de las condiciones del tráfico urbano. VEINS, un marco de simulación de redes vehiculares, se utiliza para simular el comportamiento de los vehículos, incorporando modelos realistas de interferencia de señales, desvanecimiento y sombras. Los ataques simulados en el conjunto de datos VeReMi se centran en la falsificación de la posición, una amenaza común en las VANET (Vehicular Ad-hoc Networks). Los cinco tipos de ataques implementados son: el atacante constante, el atacante de desplazamiento constante, el atacante aleatorio, el atacante de desplazamiento aleatorio y el atacante de parada eventual.

El tercer conjunto de datos es el **UNSW-SOSR2019**, que fue creado por los autores en [72]. Este conjunto de datos incluye tráfico de red vinculado a 10 dispositivos reales. El conjunto de datos fue específicamente diseñado para alinearse con la Descripción de Uso del Fabricante (MUD) [73]. Para construir el conjunto de datos para cada cliente, el perfil MUD de cada dispositivo se traduce en reglas de tablas de flujo para monitorear el tráfico previsto del dispositivo. Posteriormente, estas reglas de flujo se utilizan para extraer características y asignar etiquetas de verdad de base. Considerando las posibles variaciones de paquetes dentro de un protocolo, los autores analizan el total, la media y la desviación estándar de paquetes/bytes durante ventanas de tiempo de 2, 3 y 4 minutos. En consecuencia, hay 20 características por regla de flujo en cualquier momento dado, y el número de características varía según el número de reglas de flujo para cada dispositivo. Para fines de evaluación, se genera un conjunto de datos reducido basado en un conjunto común de características (12) en todos los dispositivos. Este conjunto de datos abarca tráfico de red asociado con 2 tipos de ataques: 4 ataques directos (Fraggle (inundación UDP), suplantación ARP, inundación TCP SYN, y Ping de la Muerte) y 4 ataques de reflexión (TCP SYN, SSDP, SNMP, y Smurf).

Cabe señalar que, aunque existen conjuntos de datos populares relacionados con IDS como CIC-IDS2017 [74], NLS-KDD [75] y N-BaIoT [76], estos trabajos carecen de ataques modernos y, lo que es más importante, no pueden dividirse en una distribución realista y adecuada para FL, donde cada cliente podría ser la dirección IP de ataque o el tipo de dispositivo, ya que esta información no se proporciona. En esta dirección, utilizamos, como se mencionó anteriormente, los conjuntos de datos ToN_IoT, VeReMi y UNSW-SOSR2019, ya que contienen ataques modernos y también pueden dividirse siguiendo divisiones reales. En el caso de VeReMi, es el

único conjunto de datos con estas condiciones en el contexto del misbehavior vehicular.

Finalmente, el último conjunto de datos utilizado es el conjunto de datos **FEMNIST**, una versión federada del conjunto de datos EMNIST [77], creado por LEAF [78], que está disponible públicamente[1]. Aunque FEMNIST no es un conjunto de datos de IDS, es ampliamente utilizado en la literatura de FL para probar métodos contra clientes bizantinos que lanzan ataques de envenenamiento. En particular, el conjunto de datos EMNIST, a su vez, se deriva del ampliamente utilizado conjunto de datos MNIST, y consiste en 62 clases de caracteres manuscritos: las 52 letras en mayúsculas y minúsculas, y los números del 0 al 9. En esta dirección, lo que distingue a FEMNIST es su estructura federada, que enfatiza los aspectos de privacidad y seguridad inherentes a los sistemas distribuidos en el mundo real. FEMNIST dividió este conjunto de datos en 3550 clientes no-iid.

Conjunto de datos	**Año**	**# Características**	**# Muestras (paquetes/flujo)**	**Relación tráfico ataque/benigno**	**Ataques**	**División**
ToN_IoT	2021	79	≈5.3M p	0.88:1	Backdoor, DoS, DDoS, Inyección, MITM, Contraseña, Ransomware, Escaneo, XSS	Dirección IP
VeReMi	2018	17	≈2.2M p	0.35:1	Ataques de mal comportamiento (falsificación de posición)	Vehículo
UNSW-SOSR2019	2019	12	≈24.6M p	0.1:1	Fraggle, Suplantación ARP, Inundación TCP SYN, Ping de la Muerte, TCP SYN, SSDP, SNMP, y Smurf	Dispositivo

Tabla 3.2

Descripción de las características de los conjuntos de datos orientados a ciberataques utilizados en esta tesis.

3.2 — Trabajos relacionados

Esta sección revisa los trabajos relacionados sobre FL en IDS y MDS en general, y las soluciones propuestas por estos trabajos en las tres áreas mencionadas: heterogeneidad de datos, privacidad y seguridad. Inicialmente, analizamos varios trabajos en el contexto de los IDS/MDS basados en FL para ofrecer una visión general del panorama actual en este ámbito. Cabe señalar que un análisis más exhaustivo de estos trabajos puede encontrarse en [4, 6]. A continuación, analizamos cómo dichos trabajos abordan algunos de nuestros objetivos diseñados en términos de heterogeneidad de datos, privacidad y seguridad. El objetivo principal es describir el estado del arte actual en estas áreas e identificar las lagunas que se abordarán mediante nuestra metodología propuesta en la siguiente sección.

[1] `https://github.com/TalwalkarLab/leaf`

3.2.1. FL para la deteccion de ciberataques

La aplicación del FL en ciberseguridad ha despertado un gran interés recientemente debido a su potencial en varios escenarios de IoT [79]. De hecho, FL se ha empleado para desarrollar IDS en sistemas IoT [80], sirviendo como alternativa a los enfoques centralizados tradicionales. La creciente popularidad de las aplicaciones y servicios IoT ha provocado un aumento de la superficie de ataque con potenciales ataques que afectan a infraestructuras críticas. Un aspecto clave del diseño del IDS es la capacidad de detectar dichos ataques de forma efectiva y eficiente mientras no se comparten los datos de los usuarios. EL FL aborda este reto dispersando los modelos de ML en dispositivos o sistemas locales, que se encargan de entrenar los modelos de ML con sus datos locales. EL FL fomenta un enfoque colaborativo para que dicho entrenamiento construya un modelo destinado a identificar ciberataques [81, 82]. De hecho, también permite escenarios a gran escala en los que las organizaciones pueden compartir información sobre amenazas sin compartir datos reales, proporcionando las bases para compartir Información sobre Ciberamenazas (CTI) preservando la privacidad [83]. Los primeros trabajos, como [84–86], utilizaban Gated Recursive Units (GRUs) con FedAvg en escenarios con división artificial de clientes. En nuestro trabajo [6] mostramos que la mayoría de los trabajos se basan en divisiones artificiales de las bases de datos NSL-KDD [75] y CIC-IDS2017 [74], con FedAvg como función de agregación. Por lo tanto, estos trabajos se basan en conjuntos de datos obsoletos que no representan el panorama actual del protocolo de red. De hecho, una observación significativa es que, según nuestro análisis en [6], la mayoría de las bases de datos recientes pueden dividirse de forma realista (por ejemplo, utilizando la dirección IP) para ser utilizados en un escenario FL. Sin embargo, nuestro análisis revela que la mayoría de los trabajos existentes siguen basándose en distribuciones de datos iid y, en algunos casos, no especifican el número de clientes utilizados, como en [87, 88].

Además, según nuestro análisis [6], los enfoques aplicados dependen en gran medida de modelos supervisados. Aunque muchos de estos estudios muestran un alto rendimiento en la detección de diversos ciberataques, dependen predominantemente de datos etiquetados, lo que supone un reto importante en escenarios del mundo real. Dada la naturaleza dinámica, la escala y la heterogeneidad de los despliegues existentes, esta suposición es a menudo poco práctica, especialmente en entornos como IoT, donde numerosos dispositivos necesitan esta información para identificar posibles amenazas. Además, las técnicas de aprendizaje supervisado son limitadas en su capacidad para detectar nuevos ataques, lo que representa otro inconveniente sustancial inherente a la mayoría de los enfoques de IDS basados en FL. Otra observación clave es que, a pesar del uso de conjuntos de datos relacionados con IoT, la mayoría de los estudios implementan modelos sin tener en cuenta las limitaciones específicas de los dispositivos y redes IoT. Sólo un estudio [89] incorpora NNs binarizadas, que podrían ofrecer una solución prometedora para entornos IoT. En cuanto a las técnicas no supervisadas, AE es el enfoque más utilizado. Sin embargo, aunque los AE son bien conocidos, la mayoría de los trabajos revisados

emplean AE simples o apilados, y sólo un estudio utiliza VAEs [90], reconocido como un método prometedor para la detección de intrusiones. Además, la aplicación de Redes Generativas Adversariales (GANs) para generar datos de ataque ha sido escasamente explorada. Con los recientes avances en modelos generativos, es muy probable que estas técnicas ganen interés en el desarrollo de IDS en un futuro próximo.

La FL con MDS es un campo emergente, con estudios recientes centrados en la adaptación de enfoques centralizados a entornos de FL. [91] proporciona una visión general de FL para IoT Vehicular, destacando beneficios como la baja sobrecarga de comunicación y la eficiencia. [92] introduce el concepto de Red Vehicular Federada, utilizando el aprendizaje colaborativo y blockchain para prevenir comportamientos maliciosos. [93] analiza las ventajas del FL en la gestión del tráfico y la conducción autónoma. Basándonos en nuestro análisis [6], comprobamos que la mayoría de los conjuntos de datos se basan en un único vehículo del que se obtuvieron los datos, por lo que es bastante inviable crear un escenario FL con una división realista. Sólo [32] propone un MDS basado en FL utilizando el conjunto de datos VeReMi, careciendo de detalles sobre la NN y la distribución de los datos. En cuanto a las técnicas no supervisadas, [94] utiliza VAE para la reducción de datos y GMM para el clustering, mientras que [95] presenta un DAGMM federado, donde los clientes comparten los pesos actualizados de AE. Otros trabajos, como [80, 96, 97], utilizan AEs en escenarios federados en diversos contextos, incluyendo redes de sensores inalámbricas.

Nuestro análisis [6] revela una falta de distribuciones realistas de los clientes, incluso en los estudios más recientes. La distribución equilibrada utilizada en estos trabajos no tiene en cuenta los aspectos no-iid inherentes a los escenarios del mundo real. Aunque el objetivo principal de estos estudios es mejorar los modelos ML, es importante reconocer que, en el contexto de FL, factores como las distribuciones de datos y las funciones de agregación implementadas son igualmente importantes. De hecho, el trabajo propuesto en esta tesis aborda dichos aspectos a través de una metodología integrada, que se describe con más detalle en la siguiente sección.

3.2.2. Heterogeneidad de los datos

Como ya se ha mencionado, una característica principal de los entornos FL es la presencia de distribuciones de datos no-iid. Este escenario surge con frecuencia en situaciones del mundo real en las que distintos dispositivos cliente pueden tener datos desequilibrados. Dicha heterogeneidad de datos/estadística [98] se caracteriza porque los dispositivos tienen diferentes tamaños de datos y distribuciones de clases, lo que significa que los datos de un dispositivo no representan la totalidad del conjunto de datos [99]. Aunque estos retos también afectan a los entornos de ML centralizados, son más pronunciados en los entornos FL debido a la gran diversidad de clientes y sus respectivos conjuntos de datos. Como se ha destacado en estudios anteriores [100], la heterogeneidad estadística afecta significativamente a la convergencia del proceso de

entrenamiento federado, especialmente cuando se utiliza FedAvg. En el contexto del desarrollo de IDS, las distribuciones de datos no-iid ocurren cuando los dispositivos tienen un gran número de muestras de un tipo particular de ataque. Esto es común en escenarios del mundo real donde algunos dispositivos son más vulnerables y actúan como puntos de entrada al sistema. En particular, algunos estudios (por ejemplo, [101–103]) evalúan sus enfoques en configuraciones de datos no-iid, aunque no proponen soluciones específicas. Como ya se ha mencionado, aunque la función de agregación está directamente correlacionada con la convergencia del sistema en entornos no-iid, según la Sección 3.1.2, la mayoría de los esquemas propuestos utilizan enfoques de agregación basados en FedAvg, que presenta problemas de convergencia [100].

Una forma de evitar los problemas de no-iid, como se discute en la Sección 3.1.2, es utilizar una función de agregación diferente. No obstante, sólo unos pocos enfoques emplean funciones de agregación alternativas orientadas a los no-iid, como Fed+ o FedProx, mientras que otros investigadores [104] sugieren métodos de agregación alternativos (FedBatch). En concreto, los autores de [105] proponen un mecanismo de agregación basado en FedProx en el que los nodos se agrupan de forma que, en cada ronda de entrenamiento, ciertos nodos se seleccionan según su orden de gradiente descendente. Adicionalmente, en [106] y [107], aunque está más orientado a la seguridad, se propone una versión robusta de FedProx para el arreglo de los pesos. Respecto a Fed+, en [108] se compara el enfoque Fed+ con otras funciones de agregación comúnmente utilizadas, como FedAvg y la mediana. En IDS más recientes, como [109], los autores proponen un entorno FL en el que 10 clientes, utilizando diferentes conjuntos de datos, entrenan un modelo AE utilizando FedProx como función de agregación. Estas soluciones pretenden ofrecer cierto grado de personalización, permitiendo que el modelo global se adapte a las características únicas de los datos de los clientes. Otro enfoque habitual para abordar los problemas de convergencia relacionados con distribuciones de datos no-iid consiste en emplear técnicas de sobremuestreo y submuestreo para equilibrar las distribuciones de datos. Mientras que algunos métodos propuestos se basan en técnicas bien conocidas como SMOTETomek [110] o SMOTE-ENN [111], una tendencia actual implica enfoques generativos basados en Redes Generativas Adversariales (GANs) [112].

A pesar de la considerable atención prestada a la heterogeneidad de los datos, algunos aspectos aún requieren una mayor exploración. Como se ha señalado, aunque muchos estudios consideran configuraciones no-iid, la mayoría de los enfoques siguen dependiendo de FedAvg para la agregación. Además, como se menciona en [99], las evaluaciones deben incluir el ajuste de los hiperparámetros de FL para garantizar la convergencia de los sistemas desarrollados, incluso en escenarios con un alto grado de heterogeneidad de datos.

3.2.3. Privacidad

Con la creciente preocupación por la seguridad de los datos y la protección de la información personal, la preservación de la privacidad se ha convertido en un importante problema mundial, especialmente en las aplicaciones del big data y los sistemas de aprendizaje distribuido. Una de las principales ventajas del FL es la posibilidad de entrenar los datos de forma descentralizada, lo que permite a las distintas partes no compartir su conjunto de datos, protegiendo cualquier tipo de información que pueda extraerse de ellos. Sin embargo, el FL sigue enfrentándose a problemas de privacidad, ya que las actualizaciones globales del modelo proporcionadas por las partes pueden ser explotadas para lanzar diversos ataques dirigidos a inferir la información privada de los datos de entrenamiento [42, 113]. Estudios recientes han demostrado que varios ataques de inferencia siguen siendo factibles durante el proceso de entrenamiento federado mediante accesos a los pesos subidos por los clientes FL al servidor [42], lo que en los sistemas IDS esta fuga de información puede llevar a revelar información importante sobre la seguridad del entorno. En particular, en las NN existen cuatro tipos principales de ataques de inferencia de privacidad [114]. Los ataques de inferencia de propiedades explotan la similitud de los modelos entrenados en conjuntos de datos similares para inferir propiedades sensibles sobre los datos de entrenamiento. Los ataques de extracción de modelos pretenden replicar el modelo objetivo explotando el acceso de caja negra o de caja gris, revelando así potencialmente los datos de entrenamiento y eludiendo los mecanismos de seguridad. Los ataques de inversión del modelo utilizan información del modelo para reconstruir los datos de entrada o deducir propiedades de los datos de entrenamiento. Por último, los ataques de inferencia de membresía intentan determinar si se incluyeron puntos de datos específicos en el conjunto de datos de entrenamiento del modelo aprovechando las características de sobreajuste y los resultados de predicción.

Para hacer frente a estos ataques, trabajos recientes han presentado la aplicación de diversas técnicas de preservación de la privacidad, como el Secure Multiparty Computation (SMPC) y el Differential Privacy (DP), a escenarios FL. En el contexto de los IDS, la mayoría de los trabajos presentados se basan en la aplicación de técnicas criptográficas para la preservación de la privacidad, como DP, SMPC y Cifrado Homomórfico (HE). Entre los trabajos basados en DP se encuentran [115] y [116], así como [117], que también integra HE. Estos métodos criptográficos permiten realizar cálculos sobre datos cifrados. Además, SMPC se refiere a un protocolo criptográfico en el que varias partes pueden calcular una función conjuntamente sin revelar sus entradas. De este modo, en el contexto de FL, el agregador no podría obtener las actualizaciones generadas por cada cliente [118]. Específicamente, el uso de DP para FL es explorado por los autores de [119], en el que examinan el efecto de aplicar sólo ruido gaussiano como método de DP dentro de un entorno FL, sin proporcionar una comparación con más técnicas. Además, su evaluación se limita al conjunto de datos MNIST ampliamente utilizado. En otro estudio, [120], se investiga un escenario IoT con limitaciones de recursos, en el que los autores aplican y evalúan una versión relajada de DP utilizando varios conjuntos de da-

tos. En [121], los autores utilizan datos de reconocimiento de actividad de smartphones para desarrollar modelos personalizados en cada dispositivo. Del mismo modo, [122] aprovecha DP en combinación con la tecnología blockchain, asegurando que la computación requerida para el mecanismo de consenso también contribuye al proceso de entrenamiento federado. No obstante, en este caso, la técnica de DP implementada es desconocida, lo que dificulta la replicación de los resultados para otros investigadores.

Basándonos en el análisis de los enfoques actuales, percibimos que la mayoría de los trabajos se basan en técnicas criptográficas bien conocidas (especialmente las basadas en DP) para evitar que los posibles atacantes accedan a las actualizaciones de los modelos en cada ronda de entrenamiento. Sin embargo, los análisis realizados son insuficientes para demostrar la aplicabilidad de estas técnicas en diferentes contextos, incluyendo el desarrollo de IDS habilitados para FL. Como se describe en [99], se requieren análisis adicionales sobre el impacto de diferentes parámetros en un entorno federado, como la función de agregación utilizada. De hecho, estos aspectos deben considerarse junto con el tipo de datos y el entorno en el que se despliega el IDS, ya que pueden tener un impacto crucial a la hora de encontrar compromisos entre la privacidad y la eficacia del sistema en la detección de ataques.

3.2.4. Seguridad

Los aspectos de seguridad de los escenarios FL han suscitado un gran interés en los últimos años [42, 123]. Al igual que los enfoques centralizados, los escenarios FL son vulnerables a ataques de envenenamiento, en los que los pesos o conjuntos de datos de los clientes son alterados maliciosamente [44] en cualquier ronda. Estos ataques implican que los clientes introducen datos o ponderaciones maliciosas para empeorar el rendimiento del modelo, afectando a todos los modelos participantes [125]. Hay dos tipos principales: el envenenamiento de datos, que altera el conjunto de datos del cliente; y el envenenamiento local del modelo, que modifica los pesos después del entrenamiento. Los datos pueden ser de etiqueta limpia (que modifica las muestras de entrenamiento sin alterar las etiquetas) o de etiqueta sucia (que afecta tanto a las muestras como a las etiquetas). El envenenamiento de modelos locales altera los pesos de entrenamiento, y puede ser no dirigido (causando errores de predicción generalizados) o dirigido (clasificando erróneamente clases específicas) [126]. Estos ataques pueden perjudicar gravemente el rendimiento de los IDS, reduciendo su capacidad para detectar ciberataques, lo que puede tener graves implicaciones dependiendo del lugar de despliegue del IDS. Además, esto podría dar lugar a falsas alarmas resultantes de una clasificación errónea durante el proceso de entrenamiento.

Para solucionar esto, algunas propuestas sugieren funciones de agregación alternativas a FedAvg para mitigar el efecto de los clientes maliciosos. Por ejemplo, en [106] se introduce un enfoque robusto basado en FedProx para defenderse de varios dispositivos maliciosos. Además,

[126] explora la efectividad de diferentes funciones de agregación robustas contra ataques de envenenamiento de datos y modelos. En concreto, los investigadores han examinado funciones bien conocidas como la media recortada [47] para defenderse de ataques de cambio de etiqueta y modificación de gradiente. Además, los GANs han sido ampliamente considerados para mejorar la robustez de los IDS habilitados para FL [127]. Por ejemplo, [127] propone el uso de GANs para mejorar la robustez del sistema mediante el entrenamiento con datos relacionados con ataques no vistos previamente. Estas herramientas pueden complementarse con enfoques de confianza y reputación para evaluar el nivel de fiabilidad que ofrecen los clientes FL, como propone [128]. Sin embargo, los GAN pueden ser un arma de doble filo, ya que también podrían generar pesos y gradientes sintéticos para identificar posibles nodos maliciosos. En cambio, en [129] se utiliza k-means para detectar nodos que envían gradientes falsos durante el entrenamiento. Además, en [106] y [107] se propone una versión robusta de FedProx en la que los nodos sospechosos de producir actualizaciones falsas son excluidos del proceso.

Como se describe en [123], varios mecanismos pueden mejorar la seguridad de los entornos FL, que pueden aplicarse a contextos IDS habilitados para FL. Aunque algunos trabajos han considerado funciones de agregación más robustas, faltan análisis exhaustivos que consideren funciones de agregación adicionales bien conocidas, como Krum, para su despliegue en IDS habilitados para FL. Además, la mayoría de los análisis pasan por alto la complejidad de estas funciones, que podría afectar significativamente al despliegue de IDS, dada la necesidad de detectar posibles ataques lo antes posible. Por otra parte, tampoco existe una lista exhaustiva de los ataques que deben tenerse en cuenta en los entornos FL. Esta ausencia de consenso a la hora de definir los ataques complica la comparación de las distintas técnicas de agregación y su evaluación de robustez. Además, la mayoría de las evaluaciones se basan en supuestos cuestionables, como que los atacantes son nodos aislados con un conocimiento limitado del sistema. Este problema se aborda en [49], que genera ataques específicos para algunas de las funciones de agregación mencionadas anteriormente. Más allá de definir técnicas de agregación robustas, como menciona [123], el uso de enfoques estadísticos puede ser crucial para identificar los nodos que envían actualizaciones falsificadas.

Capítulo 4

Metodología

Durante este doctorado, identificamos varios retos a la hora de crear modelos para la detección de ciberataques. Por lo tanto, los objetivos esbozados en la Sección 2 se clasifican en tres áreas principales: heterogeneidad de datos, privacidad y seguridad. Al abordar estos objetivos, pretendemos crear un entorno federado más seguro y que preserve la privacidad, en el que se maximice el rendimiento del modelo y se mitigue el impacto de la heterogeneidad de los datos. En la figura 4.1 se presenta un resumen visual de los métodos aplicados para abordar las cuestiones anteriormente expuestas. Esta figura se basa en el proceso de formación FL descrito en la figura 1.1, ampliando las distintas capas con nuestras soluciones. En concreto, aplicamos técnicas de remuestreo en la capa de conjuntos de datos para abordar la heterogeneidad de los datos. Además, antes de enviar los pesos al servidor, aplicamos mecanismos de DP para confundir y proteger la privacidad. Además, en el servidor, implementamos diferentes funciones de agregación dependiendo del escenario para mejorar la seguridad y gestionar la heterogeneidad de los datos.

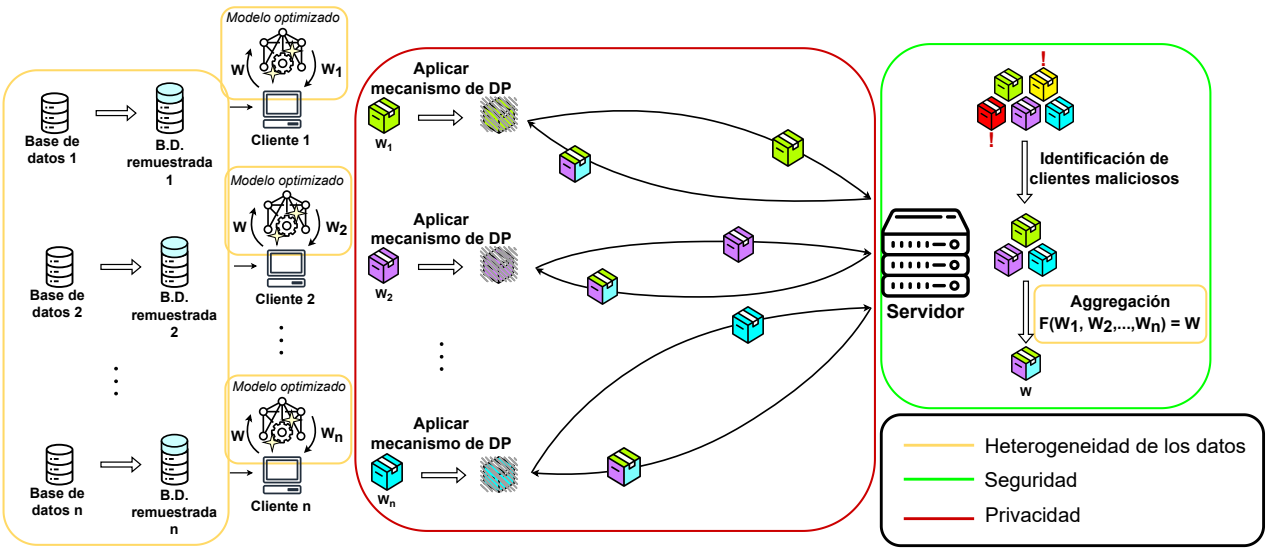


Figura 4.1. Descripción visual de las diferentes áreas abordadas a través de esta tesis para alcanzar los objetivos descritos relacionados con la heterogeneidad, la seguridad y la privacidad de los datos.

4.1 — Mitigando el impacto de la heterogeneidad de los datos

Nuestra hipótesis principal es que los conjuntos de datos de los clientes deben vincularse a identificadores, como direcciones IP, para asociar a cada cliente los datos que genera, reflejando así un escenario realista. Esto da lugar a conjuntos de datos no identificables, lo que plantea ciertos retos que es necesario abordar. Para mitigar el impacto de estas distribuciones no-iid, como se muestra en la Figura 4.1, pre-procesamos los conjuntos de datos de los clientes mediante técnicas de remuestreo, optimizamos su modelo e implementamos funciones de agregación alternativas. Estos aspectos se detallan a continuación.

4.1.1. Remuestreo de las bases de datos

La caracterización y el rendimiento de un escenario FL están intrínsecamente ligados al caso de uso específico y al conjunto de datos empleados. La partición de datos entre varios clientes introduce problemas de no-iid. Para mitigar el impacto en la precisión global, es beneficioso ajustar los conjuntos de datos de los clientes para equilibrar la distribución de clases. Sin embargo, modificar colectivamente todo el conjunto de datos es inviable, ya que los clientes sólo tienen acceso a sus propios datos y desconocen las distribuciones de los demás clientes. Por consiguiente, utilizamos técnicas de remuestreo para equilibrar las distribuciones de clases dentro del conjunto de datos de cada cliente, reduciendo así el desequilibrio de clases entre clientes. El remuestreo consiste en aumentar o disminuir el número de muestras de las distintas clases para conseguir una distribución equilibrada.

Con los conjuntos de datos de la sección 3.1.3, como ya se ha mencionado, aplicamos técnicas de remuestreo para equilibrar los conjuntos de datos locales de cada cliente. Para medir el desequilibrio de un conjunto de datos utilizamos la entropía de Shannon [130], que se define como $\frac{-\Sigma_{i=1}^{k} \frac{c_i}{n} \log \frac{c_i}{n}}{\log k}$, donde n es la longitud del conjunto de datos, k es el número de las diferentes clases del conjunto de datos y c_i es la longitud de la clase correspondiente. El valor de esta función representa el equilibrio del conjunto de datos, valiendo 0 si todas las clases son 0 excepto una, y 1 si todas las $c_i = \frac{n}{k}$, es decir, están distribuidas por igual. En el caso de ToN_IoT, para equilibrar el conjunto de datos, seleccionamos entre los 10 clientes seleccionados en la sección 3.1.3 los que están mejor equilibrados en términos de entropía de Shannon. En este caso, seleccionamos aquellos con una entropía de Shannon superior a 0,2 (es decir, 4 clientes), y después eliminamos aleatoriamente las muestras de las clases con más muestras del hasta alcanzar un valor entre 0,65 y 0,7 de entropía.

En cuanto a los conjuntos de datos VeReMi y UNSW-SOSR2019, como se ilustra en la Tabla 3.2, el tamaño limitado de las muestras de estos conjuntos de datos hace que la aplica-

ción de una técnica de submuestreo similar a la utilizada para ToN_IoT sea poco práctica. En este sentido, empleamos un enfoque diferente conocido como SMOTE-Tomek [110], que combina técnicas de sobremuestreo y submuestreo. Esta combinación aborda el problema potencial de sobreajuste que puede surgir cuando se utilizan únicamente métodos de sobremuestreo, especialmente en casos de solapamiento significativo de clases [131]. SMOTE-Tomek integra la técnica de sobremuestreo SMOTE [41] con el método de submuestreo Tomek-links [132].

El proceso de equilibrado del conjunto de datos es el siguiente. En primer lugar, SMOTE crea nuevas muestras en las clases minoritarias haciendo combinaciones lineales entre muestras, concretamente, a partir de una muestra s y sus k-vecinos más cercanos [133] s^i, las nuevas muestras s^i_{new} se calculan por $s^i_{new} = rs^i + (1-r)s$, donde r es un valor entre 0 y 1 que produce puntos intermedios entre s y s_i. El número de puntos y vecinos tomados se define en función del número de puntos necesarios. Por último, estos nuevos puntos pueden solaparse entre sí. Para proporcionar clases mejor diferenciadas, Tomek-links elimina puntos similares de clases distintas para que la diferencia entre clientes sea mayor. En concreto, a partir de un par de puntos vecinos (x_i, x_j), uno de la clase minoritaria y otro de la mayoritaria, Tomek-links elimina el punto perteneciente a la clase mayoritaria, lo que da lugar a clases mejor definidas.

Por último, se utilizó el conjunto de datos FEMNIST para evaluar nuestro método de identificación y mitigación del impacto de los ataques de envenenamiento. Dado que cada cliente del conjunto de datos FEMNIST tiene 1.315 muestras, 47 clases y una entropía de Shannon media de 0,93, asumimos que el conjunto de datos está equilibrado. En nuestros experimentos, seleccionamos subconjuntos aleatorios de 50 y 100 clientes.

4.1.2. Análisis de funciones de agregación alternativas para la heterogeneidad de los datos

La función de agregación juega un papel fundamental en FL ya que es la encargada de agregar los pesos de los clientes, siendo FedAvg la más utilizada. Sin embargo, en escenarios no-iid, el uso de FedAvg puede provocar problemas de convergencia debido a las discrepancias entre los pesos agregados y los modelos individuales de los clientes. FedAvg es ampliamente utilizado; de hecho, como se describe en [6], el 83,7 % de los trabajos analizados sobre IDSs habilitados para FL utilizan FedAvg. En escenarios con conjuntos de datos equilibrados, FedAvg funciona relativamente bien. Sin embargo, en escenarios más complejos caracterizados por distribuciones de datos non-iid, el rendimiento de FedAvg se deteriora significativamente debido a la heterogeneidad estadística, que afecta a la convergencia del proceso de entrenamiento federado. De hecho, varios estudios recientes han demostrado las limitaciones de este enfoque en escenarios con un alto grado de heterogeneidad de datos [1, 38]. Por lo tanto, la elección del enfoque de agregación FL es crucial para el desarrollo de un IDS robusto y seguro habilitado para FL.

En este sentido, Fed+ se creó para hacer frente a la heterogeneidad de los datos y mi-

tigar el impacto derivado de las distribuciones de datos no ídem. Fed+ ofrece un enfoque significativamente flexible ya que añade una función de penalización a la función de pérdida de FedAvg en cada cliente para eliminar la restricción de que los pesos de todos los clientes converjan al mismo punto. En concreto, los pesos de cada cliente k se suben siguiendo la Ecuación 4.1, donde r es la ronda, $\theta = \frac{1}{1+\nu\mu}$ es una constante que controla el grado de regularización, en la cual μ es una constante elegida por el usuario y ν es el learning rate. A es una función de agregación (por ejemplo, FedAvg), y $B(\cdot,\cdot)$ es una función de distancia que penaliza la desviación de un modelo local W^k con respecto al resultado de $A(\cdot)$. En nuestro caso, $B(W^k, A(W^1,\ldots,W^K)) = A(W^1,\ldots,W^K)$, y $A(W^1,\ldots,W^K)$ es el promedio de los pesos.

$$W_{r+1}^k \longleftarrow \theta[W_r^k - \nu\nabla f_k(W_r^k)] + (1-\theta)B(W^k, A(W^1,\ldots,W^K)), \tag{4.1}$$

Aunque existen más funciones de agregación (descritas en la sección 3.1.2), como FedPAQ, FedMA o FedProx, hemos optado por implementar sólo Fed+ para resolver los problemas relacionados con los escenarios no-iid debido a dos razones principales. En primer lugar, con un ajuste adecuado de los parámetros, otras funciones pueden considerarse casos especiales de ella, por ejemplo, FedProx o FedOpt [134]. En segundo lugar, otras funciones se centran en resolver otros aspectos de FL o sólo pueden utilizarse en el caso de modelos ML específicos. En particular, FedPAQ se centra en la eficiencia computacional, FedMA sólo es aplicable en modelos LSTM y CNN, Turbo-Aggregate [135] en la privacidad de los datos, y SAFA [136] en aspectos de asincronía. Por lo tanto, en nuestro contexto de FL-enabled IDS, estas razones justifican el uso de Fed+ para mejorar los resultados obtenidos por FedAvg. En cada trabajo, además de la implementación de FedAvg (por ser la más utilizada), implementamos también Fed+, con su correspondiente ajuste de los parámetros para maximizar la precisión, con el fin de comparar los resultados y analizar su comportamiento en entornos no-iid.

4.1.3. Optimizacion del Modelo

La selección adecuada de los parámetros ayuda a garantizar que el modelo funcione bien en diversos conjuntos de datos de clientes, abordando la heterogeneidad mediante la adaptación del modelo a las diversas necesidades de los clientes. En el caso de las NN, la elección de hiperparámetros como el número de capas, neuronas y learning rate es crucial ya que la falta de optimización puede resultar en que la NN quede atrapada en mínimos locales o sufra de tasas de convergencia lentas, lo que lleva a un rendimiento subóptimo tanto en velocidad de entrenamiento como en precisión [137], y, en consecuencia, ralentiza la convergencia en el resto de los clientes en un contexto de FL. Para el entrenamiento federado, estos hiperparámetros deben elegirse de la forma que mejor se adapte a las diversas necesidades de cada cliente y su conjunto de

datos, garantizando un rendimiento óptimo. Para ello, utilizamos el módulo GridSearchCV[1] ya que ha logrado resultados notables en otros trabajos como [138, 139]. Dicho módulo selecciona el número óptimo de neuronas, capas, optimizador y otros parámetros, debido a que emplea la validación cruzada mediante una búsqueda en cuadrícula para encontrar la mejor combinación de hiperparámetros. Esta elección óptima ayuda al modelo a evitar problemas de sobreajuste de los clientes durante el entrenamiento [140, 141]. En consecuencia, evitar el sobreajuste ayuda a mantener una convergencia robusta en todos los clientes, asegurando que el modelo global sigue siendo eficaz a pesar de la heterogeneidad de los datos. Además, como los parámetros de Fed+ dependen de la tasa de aprendizaje, en nuestros trabajos realizamos un estudio de malla para elegir la mejor tasa de aprendizaje que maximice la precisión global. En el caso de los modelos no supervisados, como utilizamos AEs y VAEs, inicializamos los pesos de estas NN no supervisadas utilizando Restricted Boltzmann Machines (RBM) [142]. Las RBM son un tipo de NN que se puede utilizar como método pre-entrenado para inicializar los pesos de las AE y VAE, ya que una inicialización incorrecta puede conducir a problemas de convergencia [143–145]. En concreto, las RBM son una forma de NN estocástica que se distingue por su estructura única de dos capas. Se definen por las conexiones simétricas entre capas y la ausencia de bucles de auto-retroalimentación. Significativamente las RBMs tiene una completa conectividad entre dos niveles pero no tiene conectividad dentro del mismo nivel. Por lo tanto, en el contexto del FL, un retraso en la convergencia en la precisión de cualquier cliente afecta a la agregación y, en consecuencia, al resto de clientes. Por lo tanto, este paso de pre-entrenamiento ayuda a AE y VAE a inicializar sus parámetros de forma que aumente la probabilidad de convergencia efectiva durante la fase de entrenamiento posterior.

4.1.4. Resultados

Antes de presentar los resultados, es importante mencionar las implementaciones de FL empleadas en esta tesis: Flower [146] e IBMFL [147]. **Flower** es un framework para FL de código abierto basado en Python que se centra en experimentos a gran escala con dispositivos heterogéneos. Ofrece varias ventajas, incluyendo estabilidad, amplio soporte para múltiples lenguajes de programación, sistemas operativos y frameworks de ML. Además, Flower admite escenarios con distintos requisitos de privacidad. Por otra parte, **IBM Federated Learning (IBMFL)** es una biblioteca Python de código abierto para facilitar el despliegue de escenarios FL en entornos productivos. IBMFL está diseñada como una solución a nivel empresarial, proporcionando una capa fundamental de FL sobre la que se pueden construir características más avanzadas. Nuestros trabajos [1] y [2] se implementaron utilizando IBMFL, mientras que el resto empleó Flower, que ha recibido un importante apoyo de la industria y el mundo académico en los últimos años.

[1] `https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html`

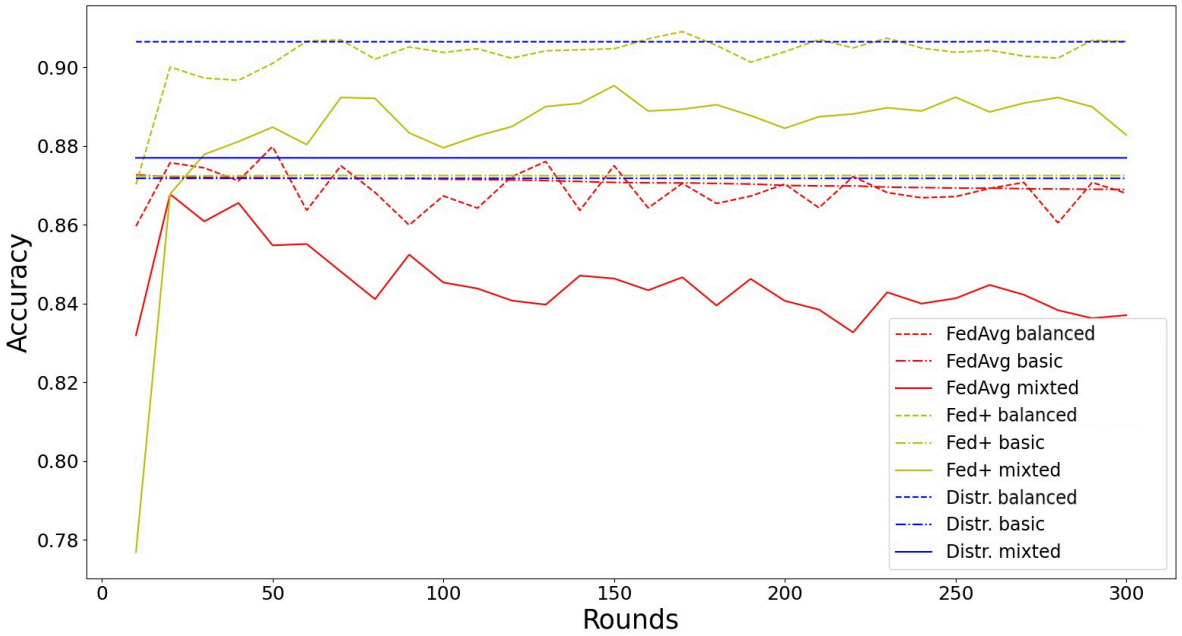
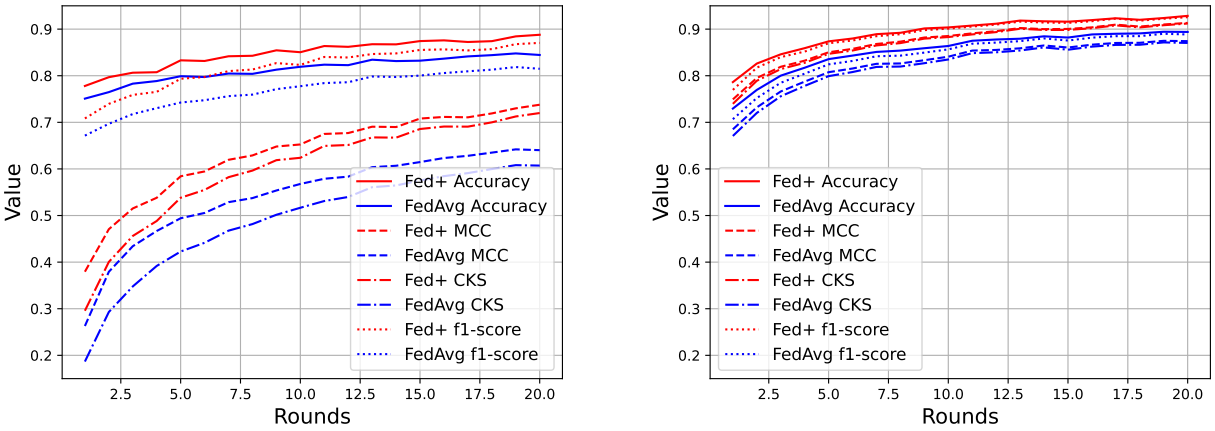


Figura 4.2. Comparación de los distintos escenarios con varias distribuciones de datos en el conjunto de datos ToN_IoT

Como ya se ha dicho, para abordar las distribuciones de datos no-iid es necesario aplicar métodos alternativos tanto en el nivel del cliente como en el del servidor de FL, centrándose especialmente en los conjuntos de datos, los modelos y las funciones de agregación. Estas adaptaciones ayudan a mitigar los posibles impactos negativos de la partición del conjunto total de datos entre múltiples entidades. En esta sección, mostramos los principales resultados obtenidos en nuestros trabajos. En particular, en la Fig. 4.2 presentamos los principales resultados de nuestro trabajo inicial en este campo [1]. En este trabajo, consideramos 3 distribuciones diferentes del conjunto de datos ToN_IoT: la básica, la equilibrada y la mixta. La básica se refiere al uso del conjunto de datos de los 10 clientes directamente sin técnicas de remuestreo. La equilibrada se refiere al remuestreo por igual de todo el conjunto de datos en 10 clientes. Por último, la mixta es la distribución descrita en la sección 4.1.1, es decir, un escenario federado en el que participan los 4 clientes con la entropía de Shannon más alta, cuyos conjuntos de datos se han ajustado para mejorar la distribución de sus clases. Comparamos estas 3 distribuciones bajo 3 casos, usando FedAvg, usando Fed+, y el caso distribuido, donde los clientes no envían los pesos durante el entrenamiento al servidor, es decir, sin formar un entorno federado. En la figura, observamos que el escenario básico bajo los casos FedAvg y Fed+ alcanza un valor constante sin ninguna evolución. A continuación, como era de esperar, el escenario equilibrado alcanza los mejores resultados en cada caso, pero obtener este escenario es casi imposible en un caso real. Sin embargo, observando el escenario mixto, esta técnica de remuestreo permite a los clientes mejorar desde la evolución de la precisión del escenario básico hasta una evolución similar a la del escenario equilibrado. Además, la implementación de Fed+ permite alcanzar valores superiores a FedAvg, e incluso superar el caso distribuido.

Basándose en los resultados proporcionados en [4], la Fig. 4.3 proporciona los resultados de evaluación del entrenamiento de un modelo MLP, previamente optimizado, con el conjunto de datos VeReMi bajo diferentes configuraciones: rebalanceado o no con SMOTE-Tomek, e implementando FedAvg o Fed+. Para medir el rendimiento de cada configuración en igualdad de condiciones, especialmente teniendo en cuenta que los conjuntos de datos desequilibrados a menudo pueden mostrar una alta precisión al predecir correctamente la clase mayoritaria mientras se ignora la minoritaria [148–150], evaluamos el rendimiento utilizando métricas adicionales como el F1-score, el Matthews Correlation Coefficient (MCC) y el Cohen Kappa Score (CKS) [151]. Estas dos últimas se han creado para medir la fiabilidad de la precisión, ya que tratan todas las clases por igual. En las Fig. 4.3a y 4.3b, la función Fed+ obtiene mejores resultados en cada caso, como queremos demostrar con la implementación de Fed+. Además, observando ambas imágenes, sin aplicar SMOTE-Tomek notamos una brecha significativa entre la precisión y el F1-score con el MCC y el CKS. Como hemos mencionado, esta diferencia está relacionada con el desequilibrio del conjunto de datos, siendo menor cuando se aplica SMOTE-Tomek, lo que significa que la precisión obtenida en este caso es una representación justa del rendimiento del modelo. Por lo tanto, en este contexto, la precisión por sí sola no basta para medir el rendimiento de un modelo. Por último, la combinación de Fed+ y SMOTE-Tomek ayuda a obtener mejores resultados en comparación con la mayoría de los trabajos actuales, como se ha descrito anteriormente en la sección 3.



(a) Métricas utilizando el conjunto de datos original con FedAvg y Fed+

(b) Métricas utilizando el conjunto de datos balanceado con FedAvg y Fed+

Figura 4.3. Comparación de varios escenarios utilizando el conjunto de datos VeReMi original frente al conjunto de datos VeReMi equilibrado obtenido mediante SMOTE-Tomek.

4.2 — Afrontacion de los problemas de provacidad mediante técnicas de DP

Una de las principales ventajas de utilizar FL está relacionada con la privacidad, ya que no es necesario compartir los datos de las distintas fuentes para entrenar el modelo global. Sin embargo, como se discute ampliamente en la Sección 3.2.3, el intercambio de pesos durante el proceso de entrenamiento podría plantear serios problemas de privacidad, ya que la información derivada del conjunto de datos local aún puede ser inferida por partes malintencionadas, incluido el servidor [42, 98, 152]. Como se describe en [99], las amenazas a la privacidad pueden implicar a diversos tipos de actores, como agregadores honestos pero curiosos, que intentan inferir información del conjunto de datos de cada cliente utilizando las actualizaciones compartidas en cada ronda de entrenamiento. Asimismo, incluso los clientes pertenecientes al entorno federado también pueden ser capaces de inferir información de los pesos de otros clientes utilizando la información de los modelos globales enviados por el agregador. Para extraer cualquier información de los datos de entrenamiento a través de los pesos, estos curiosos pueden realizar ataques de reconstrucción utilizando varias técnicas, incluyendo GANs. De hecho, los GAN también pueden utilizarse para realizar ataques de inferencia de pertenencia, permitiendo a un atacante determinar si los datos locales de una determinada parte se utilizaron en el proceso de entrenamiento [153]. En el contexto de los IDS, dado que el conjunto de datos contiene información sensible sobre la seguridad de los dispositivos, cualquier violación de los datos podría afectar significativamente a la integridad del sistema. Además, otros ataques pueden implicar inferir la participación de nodos específicos en el proceso de entrenamiento, lo que también puede tener implicaciones para la privacidad [99].

4.2.1. Analisis del impacto en diferentes funciones de agregación

Para abordar estos problemas de privacidad, la principal solución consiste en ofuscar los pesos del modelo, proporcionando garantías estadísticas de privacidad de los datos frente a los adversarios. En particular, a menudo se prefiere el DP en entornos FL debido a las estrictas exigencias de comunicación de otras técnicas de preservación de la privacidad, como el SMPC. Sin embargo, investigaciones recientes [154] hacen hincapié en los importantes requisitos computacionales y de comunicación del SMPC, lo que hace que estas técnicas no sean adecuadas para entornos IoT. En esta dirección, proporcionamos una evaluación exhaustiva de varios algoritmos de DP consistentes en técnicas de ruido aditivo basadas en distribuciones gaussianas y laplacianas. A diferencia del estado del arte, donde los investigadores sólo estudian el caso FedAvg, nosotros comparamos diferentes configuraciones de las distintas técnicas para analizar su impacto en la precisión bajo FedAvg y también Fed+. Esta comparación incluye 7 mecanismos de DP: Analítico de Gauss, mecanismo de Gauss, mecanismo de Laplace, mecanismo truncado de La-

place, mecanismo de Dominio Limitado de Laplace, mecanismos de Ruido Acotado de Laplace y Uniforme. Estos mecanismos utilizan un parámetro ϵ para ajustar la cantidad de ruido que se añade a las ponderaciones enviadas al servidor. Disminuir este parámetro aumenta la privacidad de las ponderaciones, haciendo más difícil inferir el conjunto de datos a partir de ellas, pero esto también repercute negativamente en la precisión general del proceso. El algoritmo 4 muestra cómo funcionan los distintos mecanismos de DP. En primer lugar, los distintos clientes entrenan el modelo para calcular sus nuevas ponderaciones W_r^k, y aplican uno de los 7 mecanismos DP para ofuscar las ponderaciones antes de enviarlas al servidor para protegerlas contra ataques de inferencia. A continuación, el servidor agrega los pesos y los devuelve a los clientes para que continúen su entrenamiento. El objetivo de esta comparación es probar la mejor elección de este parámetro ϵ para obtener un cierto nivel de privacidad sin sacrificar la precisión.

Algorithm 4 Algoritmo de nuestro framework de privacidad diferencial

Input: Conjunto de clientes K, número de rondas R, número de épocas E, grado de ruido del mecanismo ϵ, h modelo
Output: Pesos agregados W_R

for r de 1 a R **do**
  for $k \in K$ **do**
    Recibir pesos W_r del servidor
    Sea x_k la entrada y y_k las etiquetas de los datos locales del cliente k
    Normalizar entradas locales
    for 1 a E **do**
      Calcular la predicción $\hat{y}_k = h(x_k)$
      Calcular la pérdida $\mathcal{L}_k = L(\hat{y}_k, y_k)$
      Calcular los gradientes $\triangle w = -\nabla_{\mathcal{L}_k} w$
      Actualizar parámetros $W_r^k = W_{r-1}^k + \triangle w$
    end for
    Aplicar el mecanismo de DP a los pesos W_r^k para obtener $\kappa(W_r^k)$ con el parámetro ϵ
    Enviar $\kappa(W_r^k)$ al servidor
  end for
  El servidor recibe los pesos $\kappa(W_r^k)$
  El servidor los agrega en los pesos W_r
  El servidor envía W_r a los clientes en K
end for

4.2.2. Estudio sobre el equilibrio ofuscación-precisión

Reducir el parámetro ϵ no cuantifica directamente la cantidad de ruido o el nivel de privacidad introducidos. Por lo tanto, analizamos el impacto de ϵ en la similitud entre los pesos perturbados y los pesos originales. Para evaluar la eficacia de las técnicas de DP en la ofuscación de pesos, empleamos el Coeficiente de Correlación de Pearson (PCC) [155]. Este método fue fundamental para determinar el parámetro óptimo para protegerse de los ataques de inferencia y medir el grado de ofuscación de los pesos. El PCC cuantifica la asociación lineal entre dos

variables, con valores que van de -1 a +1. Un valor de +1 significa correlación perfecta. Un valor de +1 significa correlación perfecta, mientras que un valor de -1 indica correlación inversa. Esencialmente, el PCC es la relación entre la covarianza y la desviación estándar de dos variables. En nuestra metodología, la PCC se calcula sobre las actualizaciones del modelo para cada ronda de entrenamiento, tanto antes como después de aplicar el mecanismo DP. Esta métrica refleja la similitud entre los pesos originales y los perturbados para cada cliente. La mejora de la privacidad proporcionada por cada mecanismo se indica mediante un valor de PCC entre 0 y 1. Un valor de PCC de 1 corresponde al escenario clásico de FL sin ningún mecanismo de perturbación aplicado, por lo tanto sin ofuscación. Por el contrario, un valor cercano a 0 significa un alto grado de ofuscación y una fuerte protección contra los ataques de inferencia. En particular, en todos los mecanismos, un valor ϵ más bajo se correlaciona con un PCC más bajo, lo que indica que los valores ϵ más bajos logran un conjunto de pesos más ofuscado, mejorando así la privacidad. Esta correlación pone de relieve el equilibrio entre la privacidad y la utilidad de los datos, ya que los niveles más altos de ofuscación suelen traducirse en una disminución del rendimiento del modelo. Para alcanzar un equilibrio, es crucial seleccionar un valor ϵ que garantice la privacidad suficiente sin comprometer excesivamente la precisión del modelo. Al evaluar los valores de PCC junto con las métricas de precisión del modelo, identificamos qué mecanismos de DP proporcionan la mejor protección de la privacidad manteniendo niveles aceptables de precisión.

4.2.3. Resultados

En la Fig. 4.4, mostramos los resultados del análisis anteriormente mencionado de los distintos métodos utilizando Fed+ como función de agregación. En este análisis, partimos del escenario mixto del conjunto de datos ToN IoT descrito en la Sección 4.1.4 bajo los 7 mecanismos DP. En todos ellos, comparamos diferentes valores del parámetro ϵ para medir cómo se ve afectada la precisión. En esta figura, observamos que la precisión no se ve alterada por las diferentes configuraciones de los distintos mecanismos, lo que nos permite ofuscar los pesos para proteger la privacidad de los distintos clientes. Adicionalmente, en la Tabla 4.1 se muestra el PCC para cada valor. Esta tabla mide el nivel de ofuscación de los pesos, cuanto menor sea el valor, más ofuscados estarán los pesos y en consecuencia, más protegidos. Por lo tanto, el mejor mecanismo sería el mecanismo uniforme ya que tiene el valor más bajo de PCC y en su respectiva precisión en la Fig. 4.4 no se ve afectado.

4.3 — Mejorando los ajustes de la seguridad en FL

Durante el entrenamiento de FL, la seguridad del sistema puede verse comprometida por ataques de envenenamiento de modelos, en los que clientes malintencionados modifican deliberadamente

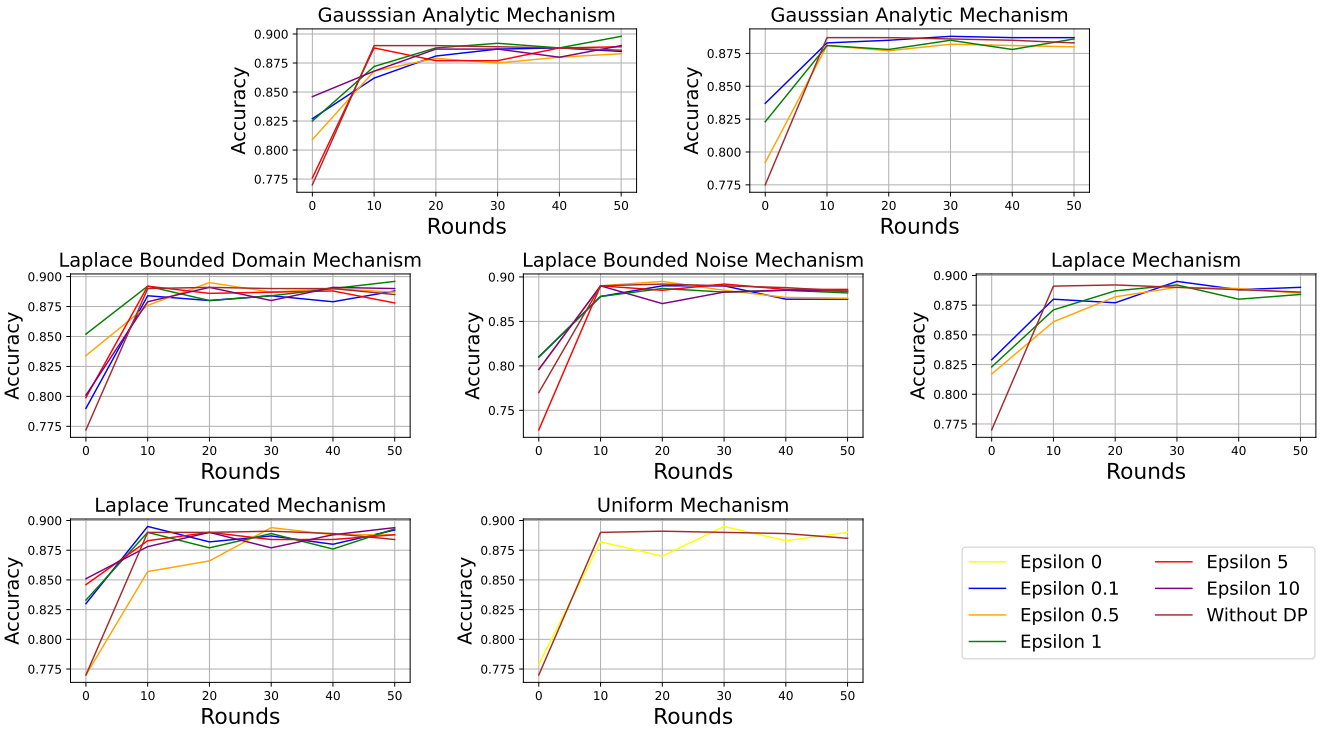


Figura 4.4. Resultados de precisión de Fed+ para todos los mecanismos de perturbación.

los pesos que devuelven en cada ronda de entrenamiento [124]. En el contexto de los IDS, el principal impacto es una disminución en el rendimiento de la detección de ciberataques, que puede tener graves consecuencias dependiendo del entorno de despliegue. Por ejemplo, esto podría conducir a un aumento de las falsas alarmas debido a errores de clasificación introducidos durante el proceso de entrenamiento. Por lo tanto, una protección robusta contra estos ataques es crucial para mantener la integridad del modelo. Para hacer frente a estos ataques, proponemos una nueva función de agregación robusta que aprovecha las propiedades de la Transformada de Fourier Rápida (FFT). Además, diseñamos y aplicamos un método para identificar y excluir a los clientes potencialmente maliciosos durante el proceso de formación.

4.3.1. FedRDF: una función de agregación robusta para el FL

El uso de un enfoque de agregación robusto representa una forma de mitigar el impacto de los clientes bizantinos en entornos FL. Como se detalla en la sección 3.1.2, las funciones pueden reducir significativamente el impacto de tales dispositivos comprometidos. Sin embargo, estos métodos suelen suponer que se conoce el número de atacantes, lo que podría no ser práctico en situaciones del mundo real o contra ataques sofisticados en los que las partes comprometidas cambian dinámicamente durante las rondas de entrenamiento. De hecho, la mayoría de los en-

ϵ	**Mecanismo**	**PCC**	**Mecanismo**	**PCC**
0.1	**Laplace Truncado**	0.6560	**Dominio Limitado de Laplace**	0.4552
0.5	**Laplace Truncado**	0.9663	**Dominio Limitado de Laplace**	0.9028
1	**Laplace Truncado**	0.9906	**Dominio Limitado de Laplace**	0.9762
5	**Laplace Truncado**	0.9996	**Dominio Limitado de Laplace**	0.9994
10	**Laplace Truncado**	0.9999	**Dominio Limitado de Laplace**	0.9998
0.1	**Laplace**	0.6572	**Gaussiano**	0.3549
0.5	**Laplace**	0.9655	**Gaussiano**	0.8256
1	**Laplace**	0.9907	**Gaussiano**	0.9406
0.1	**Analítico de Gauss**	0.5964	**Laplace Ruido Acotado**	0.6951
0.5	**Analítico de Gauss**	0.9156	**Laplace Ruido Acotado**	0.9666
1	**Analítico de Gauss**	0.9704	**Laplace Ruido Acotado**	0.9908
5	**Analítico de Gauss**	0.9977	**Laplace Ruido Acotado**	0.9996
10	**Analítico de Gauss**	0.9992	**Laplace Ruido Acotado**	0.9999
0	**Uniforme**	**0.0472**		

Tabla 4.1

Tabla del PCC de los diferentes mecanismos de DP para diferentes valores.

foques existentes abordan principalmente ataques básicos de envenenamiento, pasando por alto escenarios en los que los atacantes pueden confabular. Sin embargo, trabajos recientes destacan las vulnerabilidades de estos enfoques frente a ataques más sofisticados [49]. Basándonos en las limitaciones de las funciones de agregación existentes, identificamos la necesidad de una nueva función de agregación óptima que no requiera conocimiento previo sobre el número de atacantes y que no se base en métodos de distancia o estadísticos. Para ello, aprovechamos las ventajas de la FFT para crear esta novedosa función de agregación robusta. La FFT nos permite calcular la función de densidad de los pesos e identificar el punto de mayor concentración. Como se ilustra en la Fig. 4.5, aplicamos este proceso por coordenadas, proyectando los pesos de los clientes en el dominio de la frecuencia. Esto nos permite determinar el punto con el valor más alto en la función de densidad a través de su representación en el dominio de frecuencias [46], que corresponde al punto con la frecuencia más alta. Como la FFT es fácilmente invertible, se calcula la inversa de dicho punto para obtener el conjunto original de pesos, y esos pesos son los que se envían a los clientes. Esta función es robusta frente a valores atípicos, ya que estos valores tienen que estar alejados de los puntos condensados en la función de densidad, que no afectan a la elección del punto final.

Aunque este método es eficaz contra los clientes maliciosos, su principal inconveniente es que la función FedAvg supera a las funciones de agregación robustas cuando no hay clientes maliciosos. Al principio, está claro que las funciones no robustas alcanzan valores más altos frente a clientes maliciosos. Sin embargo, en un caso real, se desconoce si hay clientes maliciosos. Por ello, desarrollamos un algoritmo basado en el test estadístico de Kolmogorov-Smirnov (K-S) [156]. Esta prueba determina si dos distribuciones son estadísticamente equivalentes. Como se muestra en el Algoritmo 5, antes de la agregación, aplicamos un proceso por coordenadas a cada conjunto de puntos $V_{i,j}$ representados en la Fig. 4.5. Seleccionamos un conjunto de pun-

tos $V_{i,j}$ y lo sometemos a una prueba de Kolmogorov-Smirnov. Seleccionamos un subconjunto de tamaño S de estos puntos y aplicamos la prueba K-S entre este subconjunto y los puntos restantes. Si la prueba falla, indica la existencia de posibles pesos maliciosos que distorsionan la distribución. Para garantizar la precisión de este proceso, lo repetimos C veces. Si el número de pruebas fallidas supera un umbral definido por el usuario, concluimos que existe actividad maliciosa. En cada ronda, aplicamos la prueba K-S explicada para detectar posibles ataques de clientes maliciosos. Si la prueba se supera, lo que significa que no se han detectado clientes maliciosos, el sistema utiliza FedAvg como función de agregación; en caso contrario, emplea el método basado en FFT. A este proceso global lo denominamos FedRDF (Robust Dynamic Fourier aggregation function). FedRDF se adapta a diversos escenarios, maximizando la precisión independientemente de la presencia de clientes maliciosos.

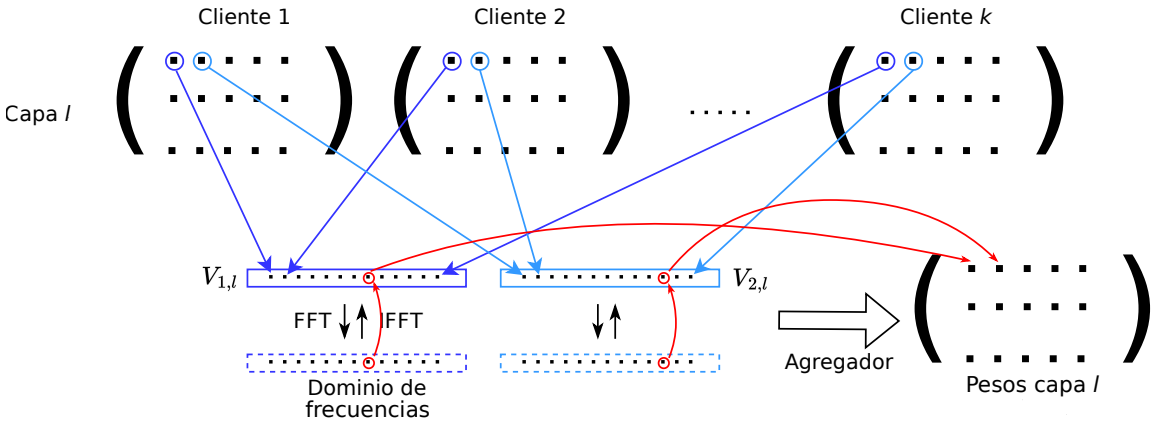


Figura 4.5. Descripción visual para calcular la FFT en FedRDF.

4.3.2. Identificación de clientes bizantinos

Aunque las funciones de agregación robustas protegen a los sistemas frente a clientes bizantinos, tienen notables limitaciones [50], como se ha comentado en la subsección anterior y en la Sección 3. Por ejemplo, su rendimiento tiende a disminuir a medida que aumenta el número de clientes maliciosos. Estas limitaciones motivan el desarrollo de un enfoque para identificar y excluir clientes específicos que envíen pesos maliciosos a lo largo de las rondas de entrenamiento. Basándonos en los algoritmos actuales más avanzados, proponemos un framework que aborda las limitaciones de los enfoques existentes, haciendo uso de varias técnicas para detectar clientes maliciosos. En este sentido, creamos un método denominado FLAegis. Nuestro framework está pensado principalmente para ser desplegado en el servidor y, como puede verse en la Fig. 4.6 y en el Algoritmo 6, consta de 2 fases: la fase de identificación y la fase de mitigación. La fase de identificación consiste en la detección de clientes entre su conjunto. En primer lugar, en este tipo de framework, debido a las limitaciones de las técnicas de clustering, como se describe en [157, 158], suponemos que el número de clientes maliciosos es inferior a la mitad de los clien-

Algorithm 5 Algoritmo de nuestro framework de aprendizaje federado dinámico robusto

Input: Conjunto de clientes K, número de rondas R, número de épocas E, número de repeticiones del test K-S C, tamaño de la parte del subconjunto para calcular el test K-S S, y umbral t.
Output: Modelo global W_R

for r de 1 a R **do**
  for $k \in K$ **do**
    $W_r^k = \text{actualizaciónLocal}(W_{r-1}, E)$
    Enviar W_r^k al servidor
  end for
  for Servidor **do**
    $t_r = \text{media}(K - S\_test(\{W_r^1, \ldots, W_r^{|K|}\}), C, S)$
    if $t_r < t$ **then**
      $W_r = \text{media}(\{W_r^1, \ldots, W_r^{|K|}\})$
    else
      $W_r = \text{FFT}(\{W_r^1, \ldots, W_r^{|K|}\})$
    end if
    Enviar W_r a los clientes
  end for
end for

$K - S\_test(\{W_r^1, \ldots, W_r^{|K|}\}), C, S)$:
Calcular $V_{i,l} = \{w_{i,l}^1, \ldots, w_{i,l}^K\} \forall i, l$
for all $V_{i,l}$ **do**
  for 1, 2, ..., C **do**
    Tomar muestra $\hat{V}_{i,l}$ de $V_{i,l}$ de tamaño S
    Calcular el valor-p de $KS(\hat{V}_{i,l}, V_{i,l} \setminus \hat{V}_{i,l})$
  end for
  Calcular las veces que el valor-p fue menor que 0.05
end for
return vector de proporciones de fallos

tes. Bajo esta suposición, clasificamos a los clientes creando una matriz de similitud M. Para mejorar la distinción entre clientes maliciosos y benignos a la vez que se reducen las diferencias entre clientes benignos, primero transformamos los pesos de los clientes utilizando Symbolic Aggregate approXimation (SAX) [159], en el que se divide el espacio muestral de los pesos en un número predefinido de partes, y se asocia cada peso con un símbolo. De esta manera, pesos similares tendrán similitud cercana, mientras que los malignos, al tener que separarse de los benignos, mostrarán una diferencia mayor. Esta transformación permite medir la similitud de forma más eficaz. A continuación, utilizando la similitud del coseno [160] medimos la similitud entre los clientes para crear la matriz de similitud M mencionada anteriormente. Después de calcular esta matriz, empleamos técnicas de agrupación para clasificar a los clientes como benignos o maliciosos. Implementamos el spectral clstering, que es capaz de agrupar dinámicamente los elementos sin seleccionar el número de clusters de antemano, a diferencia de otros métodos como K-means o GMM. Si el número de clusters obtenido es 1, significa que todos los clientes

son similares y, por tanto, no hay clientes maliciosos. En caso contrario, significa que hay clientes maliciosos, por lo que realizamos un clustering final utilizando K-means con 2 componentes (ya que el clustering final del spectral clstering utiliza K-means) predefinidas y seleccionamos el cluster más pequeño como el que contiene los clientes maliciosos (ya que suponemos que el número de clientes malignos era menor que el de benignos). Sin embargo, en este método de identificación, puede haber algunos clientes maliciosos que superen la fase de identificación y se clasifican como benignos. Para proteger el sistema contra ese tipo de clientes, implementamos una fase de mitigación en la que los pesos de los clientes restantes se agregan utilizando la FFT como función de agregación (como en la sección anterior) para reducir el impacto de los clientes maliciosos. Este enfoque para identificar a los clientes maliciosos pretende mitigar los problemas descritos anteriormente para las funciones de agregación robustas, especialmente en entornos con un alto porcentaje de este tipo de clientes.

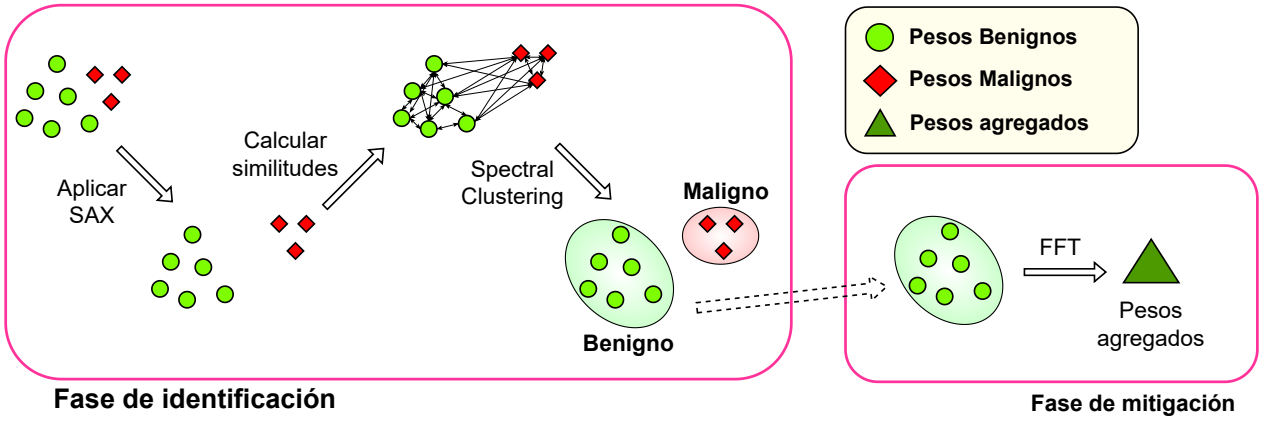


Figura 4.6. Descripción visual del proceso de FLAegis.

4.3.3. Resultados

Los dos mecanismos de defensa (tanto FedRDF como FLAegis) descritos pretenden proteger el rendimiento del modelo minimizando la pérdida de precisión. En primer lugar, analizamos nuestra función de agregación robusta, FedRDF, que consta de dos componentes: la prueba K-S y, en caso necesario, la FFT. Comenzamos evaluando la robustez de la FFT como función de agregación en comparación con los enfoques estándar. A continuación, demostramos su eficacia contra un ataque complejo, el ataque min-max, descrito en [161]. La Fig. 4.7 ilustra que este ataque degrada significativamente el rendimiento del modelo cuando se utiliza FedAvg, incluso con una baja presencia de clientes maliciosos. En particular, el método de la FFT alcanza valores de rendimiento superiores a los de otros métodos, con la excepción de la media recortada con un 0 % de nodos maliciosos, donde iguala a la media. Es importante señalar que, aunque la diferencia absoluta pueda parecer pequeña, todas las funciones optimas intentan alcanzar el valor máximo logrado por FedAvg en ausencia de actividad maliciosa. Dado este límite su-

Algorithm 6 Descripción de FLAegis

Input: Conjunto de clientes K, $(W_k)_{k\in K}$ pesos de los clientes
Output: Pesos Agregados

Fase de Identificación

for $k \in K$ **do**
  $\tilde{W}_k = SAX(W_k)$
end for
for $k \in K$ **do**
  for $l \in K$ **do**
    $m_{kl} = cosine\_similarity(\tilde{W}_k, \tilde{W}_l)$
  end for
end for
$M = (m_{kl})_{k,l=1}^{|K|}$ matriz de similitud
$(S)_1^L = Spectral\_Clustering(M)$
if $L > 1$ **then**
  $S_1, S_2 =$ K-Means(M)
  if $S_1 \geq S_2$ **then**
    $B = S_1$ se clasifican como clientes benignos
  else
    $B = S_2$ se clasifican como clientes benignos
  end if
else
  $B = K$ se clasifican como clientes benignos
end if

Fase de Mitigación

$W = FFT((W_b)_{b\in B})$
El servidor envía W a los clientes de K

perior, incluso las ligeras mejoras de la FFT sobre las otras funciones son más significativas cuando se consideran en términos de error relativo. A continuación, en la Fig. 4.8, analizamos el rendimiento de la versión completa de nuestro método utilizando el umbral que alcanzó los mejores resultados. Para determinar el umbral óptimo, empleamos una técnica de validación cruzada de 5 veces. En este escenario, aumentamos la presencia de clientes maliciosos en incrementos del 10 %, en lugar de los incrementos del 5 % utilizados en la figura anterior. En esta figura logramos el propósito de nuestro método, al 0 % de clientes maliciosos su valor es mayor que el de FedRDF y cercano al de FedAvg, y se acerca a los valores de FedRDF a medida que aumenta la presencia de clientes maliciosos, aunque al final la diferencia se hace ligeramente mayor ya que puede haber rondas en las que FedRDF tome FedAvg en lugar de FedRDF. En general esta nueva función supera a las actuales funciones de agregación optimas y nos permite proteger los pesos contra ataques de envenenamiento con una pérdida mínima de precisión, ya que se adapta a todo tipo de escenarios.

Para proteger el sistema contra ataques de envenenamiento, también creamos un framework para detectar clientes maliciosos específicos, FLAegis. En la Fig. 4.9, comparamos FLAegis con la media y con otros dos frameworks para eliminar clientes maliciosos, Foolsgold [161] y

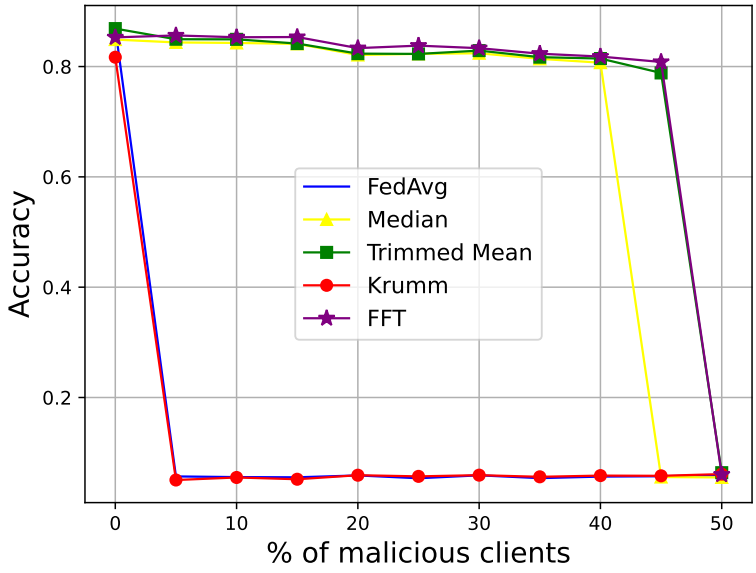


Figura 4.7. Resultados de diferentes funciones de agregación robustas contra el ataque min-max.

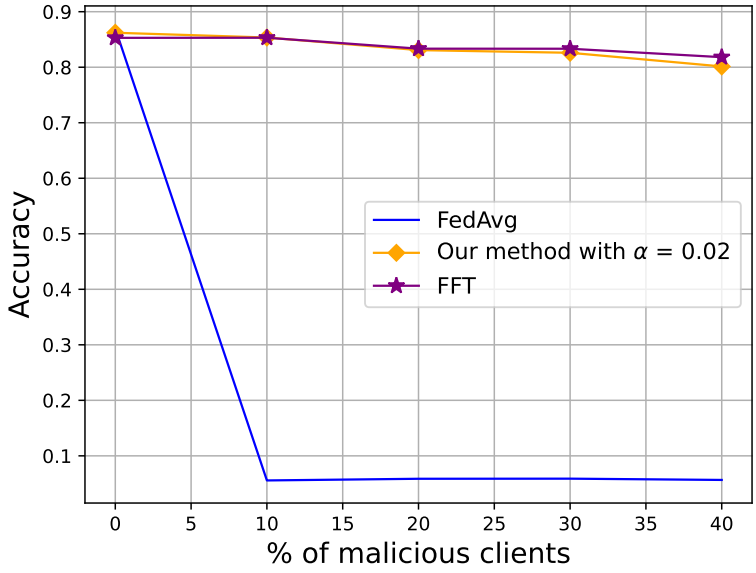


Figura 4.8. Comparación de FedRDF con FedAvg y FFT.

SignGuard [157]. En esta figura, comparamos los métodos mencionados contra 5 ataques diferentes, min-max [49], min-sum [49], LIE [162], STATOPT [163], y label flipping [20]. En todos los ataques, FLAegis es el único que permanece estable. SignGuard es el segundo más resistente, pero en determinadas intensidades de ataque, su rendimiento diverge significativamente del de FLAegis. FoolsGold, excepto en el caso del cambio de etiquetas, experimenta un rápido descenso de rendimiento. Nuestro método implementa la FFT como función de agregación, lo que podría pensarse que es responsable de nuestros notables resultados, ya que la FFT protege el sistema contra los clientes maliciosos no es necesaria para el proceso anterior. Sin embargo,

esta fase de identificación ayuda a reducir el número de clientes maliciosos, lo que contrarresta las debilidades de estas funciones robustas cuando hay un gran número de clientes maliciosos. Por este motivo, en la Fig. 4.10, comparamos nuestro método con FoolsGold y SignGuard, pero utilizamos la FFT como función de agregación en estos 2 métodos. Además, completamos la comparación aplicando sólo la FFT para demostrar que la fase de identificación de nuestro método también es necesaria. En esta figura, aunque el rendimiento de FoolsGold y SignGuard mejora, sus resultados siguen siendo inferiores a los de nuestro método, especialmente cuando hay una alta presencia de datos maliciosos en los ataques complejos min-max y min-sum, así como en el rendimiento de la FFT. De esta figura se obtiene que una fase de identificación precisa antes de realizar la agregación es de crucial importancia para lograr resultados satisfactorios, lo que consigue nuestro método.

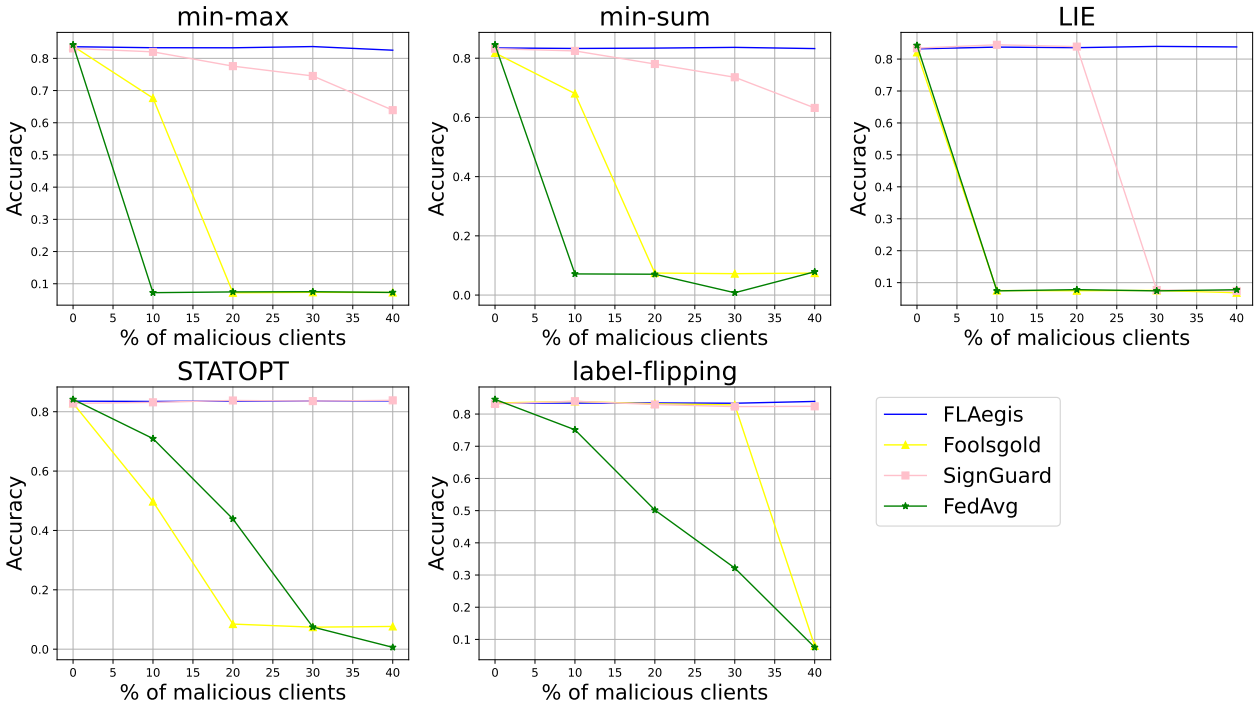


Figura 4.9. Comparación de nuestro framework con SignGuard y FoolsGold frente a distintos ataques.

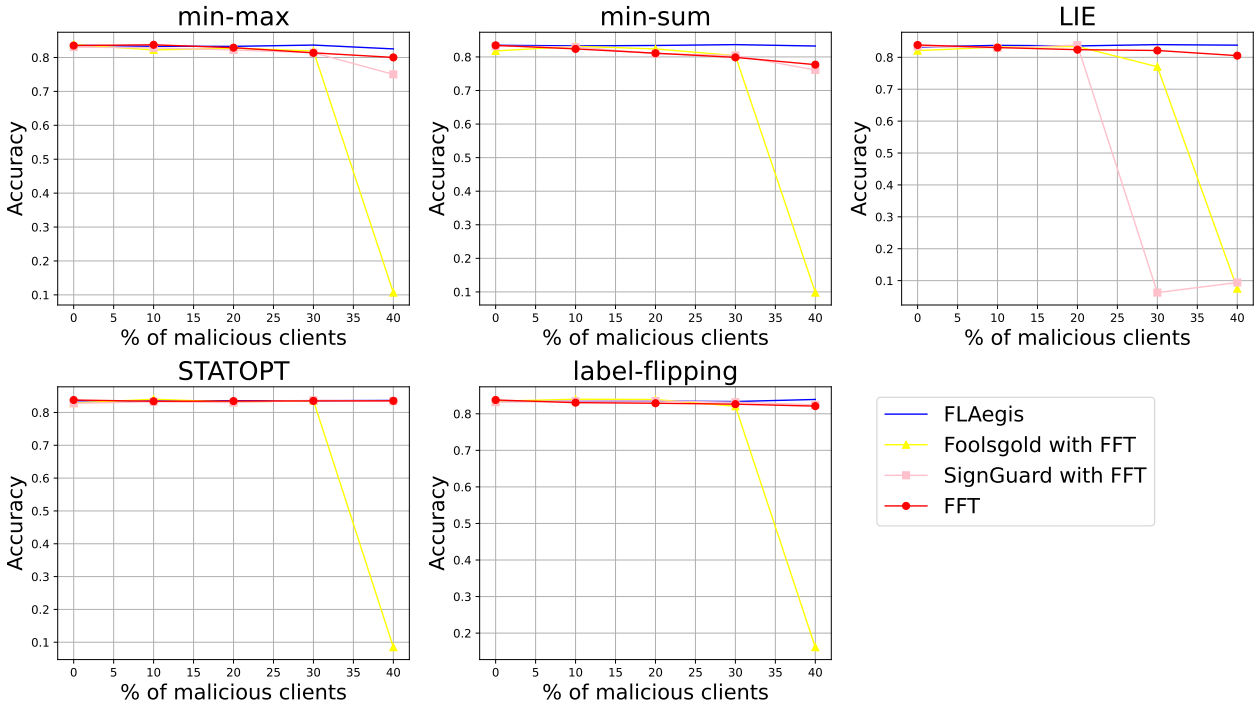


Figura 4.10. Comparación de nuestro framework con SignGuard y FoolsGold utilizando FFT contra diferentes ataques.

Capítulo 5

Conclusiones y trabajo futuro

Esta tesis se centra en la detección de ciberataques en entornos IoT utilizando FL, una técnica puntera presentada a la comunidad científica poco antes de comenzar este doctorado. A lo largo de este trabajo, se identificaron y abordaron varios desafíos relacionados con FL para desarrollar modelos más eficaces y eficientes en escenarios realistas. Algunos de los principales retos abordados en esta tesis incluyen abordar el impacto de las distribuciones no-iid de los diferentes conjuntos de datos, mantener privados los conjuntos de datos individuales de los clientes y aumentar la robustez de los escenarios FL mediante la identificación de clientes maliciosos y el desarrollo de un enfoque de agregación robusta.

Nuestro análisis de los enfoques de detección de ciberataques basados en FL indica que la literatura actual se centra en el uso y la mejora de los modelos de ML existentes utilizados, más que en los elementos que caracterizan a FL, como la distribución de los conjuntos de datos de los clientes o las funciones de agregación implementadas. Además, a través de nuestro análisis exhaustivo, nos dimos cuenta de que la mayoría de los trabajos del estado de la técnica se basan en conjuntos de datos obsoletos y en escenarios en los que los conjuntos de datos de los clientes están idealmente equilibrados. Esto se traduce en un rendimiento decreciente del modelo cuando se aplica en escenarios con distribuciones de datos no-iid. Por este motivo, nuestros enfoques se evalúan en escenarios en los que las distribuciones de los conjuntos de datos siguen una distribución realista, en la que cada cliente está relacionado con el tráfico de un dispositivo o sistema específico. Nuestros resultados muestran que la metodología propuesta mitiga el impacto de estas distribuciones no-iid mediante el uso de diferentes técnicas de remuestreo y analizando el uso de varias funciones de agregación.

Además, aunque FL se acuñó principalmente para mitigar los problemas de privacidad de los típicos despliegues centralizados de ML, la literatura reciente demuestra que sigue siendo posible extraer información de los distintos conjuntos de datos a lo largo del intercambio de pesos entre los clientes y el servidor. A partir de nuestro análisis, encontramos que aunque se implementan varias técnicas basadas en DP para proteger los pesos contra ataques de inferencia, faltaba un análisis exhaustivo sobre el impacto de estos diferentes mecanismos de DP y el uso de varias funciones de agregación en la efectividad del modelo. En esta dirección, proporcionamos un estudio exhaustivo del impacto de varios mecanismos DP en dos funciones de agregación,

FedAvg y Fed+. Nuestros resultados proporcionan diferentes perspectivas en torno a la mejor combinación de los diferentes hiperparámetros de dichas técnicas para lograr un compromiso entre privacidad y precisión.

Además, a partir de nuestro análisis de la literatura, también descubrimos que las metodologías defensivas para proteger los entornos federados contra clientes bizantinos son costosas desde el punto de vista computacional, se basan en suposiciones poco realistas y se prueban contra ataques simples. En respuesta, desarrollamos dos métodos diferentes para proteger el sistema contra este tipo de amenazas. En primer lugar, proponemos una nueva función de agregación denominada FedRDF, que puede detectar la presencia de clientes maliciosos y mitigar su impacto en la agregación mediante una función basada en FFT. Sin embargo, FedRDF no puede medir el número de clientes maliciosos presentes ni saber quiénes son. En esta dirección, también desarrollamos un framework (llamado FLAegis) que, a diferencia de FedRDF, puede identificar y descartar de la agregación a los clientes maliciosos específicos que intentan comprometer el modelo con los ataques de envenenamiento. Nuestros resultados en ambos métodos se probaron contra ataques modernos, superando a los métodos actuales.

En nuestro estudio, también observamos que la literatura actual no ha explorado en profundidad los modelos no supervisados. Para afrontar este vacío, adaptamos técnicas no supervisadas a escenarios FL, reduciendo la dependencia de conjuntos de datos etiquetados, lo que podría resultar inviable en situaciones del mundo real. Además, utilizando técnicas de clustering, conseguimos reducir el número de clientes FL manteniendo la precisión final para los clientes restantes, reduciendo así el ancho de banda necesario para la comunicación entre los clientes y el servidor.

Por último, cabe destacar que la mayoría de los métodos y técnicas desarrollados en los diferentes trabajos propuestos durante el doctorado son de código abierto y están disponibles en nuestro GitHub[1], lo que facilita a otros autores replicar nuestro trabajo y ayudar a desarrollar nuevas técnicas en el futuro.

Teniendo en cuenta los amplios conocimientos obtenidos a lo largo de esta tesis, aún quedan muchas áreas que necesitan una mayor exploración y análisis en futuros trabajos.

- Profundización en los algoritmos de selección de clientes:

  En algunos de nuestros trabajos, redujimos el número de clientes en el entorno federado en función de sus características, por ejemplo, por el tamaño de su conjunto de datos. En esta dirección, nos proponemos definir algoritmos de selección de clientes que operen durante el entrenamiento federado para seleccionar los clientes que alcancen un mejor rendimiento en términos de precisión, maximizando así el rendimiento del modelo agregado. Posteriormente, utilizando diversas técnicas, transferiremos el modelo al resto de clientes

[1] `https://github.com/Enrique-Marmol?tab=repositories`

para completar su entrenamiento, ahorrando tiempo y ancho de banda entre el servidor y los clientes.

- Eliminación de la presencia del servidor:

  En el campo de la seguridad, los métodos existentes asumen que las entidades maliciosas pueden ser tanto entidades externas como clientes internos. Sin embargo, la suposición predominante de un servidor fiable y benigno, responsable de orquestar todas las operaciones, requiere una revisión crítica. El servidor es la entidad central que recibe los pesos y coordina todos los procesos federados. Así, un servidor malintencionado puede utilizar estos pesos con fines malintencionados sin que los clientes puedan impedirlo. En este sentido, planeamos reevaluar el esquema federado, de modo que el servidor o el agregador no se conviertan en un único punto de fallo en la arquitectura global. Para lograrlo, necesitamos redefinir la función de agregación para que se realice progresivamente a través de un subconjunto aleatorio de clientes en cada ronda. Mediante este enfoque descentralizado, pretendemos reforzar el entorno FL frente a amenazas adicionales, garantizando la integridad y confidencialidad de los datos de los clientes.

- Ampliación de los algoritmos de FedRDF y FLAegis:

  Aunque FedRDF y FLAegis logran resultados satisfactorios, pretendemos ir un paso más adelante. En el caso de FedRDF, tenemos previsto mejorar la forma de establecer el umbral para elegir entre las funciones de agregación. Actualmente, FedRDF se basa en la prueba K-S y en un umbral elegido de antemano por el usuario para decidir qué función de agregación utilizar, lo que a veces puede fallar a la hora de elegir entre FedAvg o la FFT. Queremos explorar nuevos métodos estadísticos que sustituyan al test K-S para mejorar la tasa de acierto en la elección de FedAvg o la FFT, e implementar un umbral dinámico similar al utilizado en AE. Para FLAegis, planeamos redefinir cómo se calcula la matriz de similitud para diferenciar mejor entre clientes maliciosos y benignos. Exploraremos otras funciones para medir la similitud y aumentar así la tasa de detección, especialmente en el caso de un 10 % de clientes maliciosos, evitando depender de la FFT para la agregación con el fin de maximizar la precisión final.

- Implantación de nuevos tipos de modelos:

  Como parte de nuestro trabajo futuro, la implementación de nuevos modelos dentro del entorno federado es un área clave a explorar. Los modelos no supervisados siguen siendo una laguna importante en el campo de IDS y MDS. Nuestro objetivo es implantar modelos no supervisados más complejos, como variantes innovadoras de AE para clasificar más de dos clases. Además, la creciente importancia de Large Language Models (LLM) está impulsando un cambio significativo en la investigación de la IA, ofreciendo una capacidad sin precedentes en el procesamiento y comprensión del lenguaje natural. Los LLM requieren grandes cantidades de datos para su entrenamiento, por lo que el uso de FL como

enfoque para el proceso de ajuste fino de los LLM podría reducir el coste de almacenar todos los datos en el mismo lugar, ya que FL permite a los clientes utilizar su conjunto de datos para entrenar el LLM sin compartir ninguna información sobre ellos.

Bibliografía

[1] Enrique Mármol Campos, Pablo Fernández Saura, Aurora González-Vidal, José L Hernández-Ramos, Jorge Bernal Bernabe, Gianmarco Baldini, and Antonio Skarmeta. Evaluating federated learning for intrusion detection in internet of things: Review and challenges. Computer Networks, page 108661, 2021.

[2] Pedro Ruzafa-Alcázar, Pablo Fernández-Saura, Enrique Mármol-Campos, Aurora González-Vidal, José L Hernández-Ramos, Jorge Bernal-Bernabe, and Antonio F Skarmeta. Intrusion detection based on privacy-preserving federated learning for the industrial iot. IEEE Transactions on Industrial Informatics, 19(2):1145–1154, 2021.

[3] Sara N Matheu, Enrique Mármol, José L Hernández-Ramos, Antonio Skarmeta, and Gianmarco Baldini. Federated cyberattack detection for internet of things-enabled smart cities. Computer, 55(12):65–73, 2022.

[4] Enrique Mármol Campos, José L Hernandez-Ramos, Aurora González Vidal, Gianmarco Baldini, and Antonio Skarmeta. Misbehavior detection in intelligent transportation systems based on federated learning. Internet of Things, page 101127, 2024.

[5] Enrique Mármol Campos, Aurora González Vidal, José L Hernández Ramos, and Antonio Skarmeta. Federated transfer learning for energy efficiency in smart buildings. In IEEE INFOCOM 2023-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), pages 1–6. IEEE, 2023.

[6] Jose L Hernandez-Ramos, Georgios Karopoulos, Efstratios Chatzoglou, Vasileios Kouliaridis, Enrique Marmol, Aurora Gonzalez-Vidal, and Georgios Kambourakis. Intrusion detection based on federated learning: a systematic review. arXiv preprint arXiv:2308.09522, 2023.

[7] Enrique Mármol Campos, José L Hernandez-Ramos, Aurora González Vidal, Gianmarco Baldini, and Antonio Skarmeta. Misbehavior detection in intelligent transportation systems based on federated learning. Internet of Things, page 101127, 2024.

[8] Enrique Mármol Campos, Aurora González Vidal, José Luis Hernández Ramos, and Antonio Skarmeta. Fedrdf: A robust and dynamic aggregation function against poisoning attacks in federated learning. arXiv preprint arXiv:2402.10082, 2024.

[9] Linghe Kong, Jinlin Tan, Junqin Huang, Guihai Chen, Shuaitian Wang, Xi Jin, Peng Zeng, Muhammad Khan, and Sajal K Das. Edge-computing-driven internet of things: A survey. ACM Computing Surveys, 55(8):1–41, 2022.

[10] Kamran Shaukat, Suhuai Luo, Vijay Varadharajan, Ibrahim A Hameed, and Min Xu. A survey on machine learning techniques for cyber security in the last decade. IEEE access, 8:222310–222354, 2020.

[11] Asmaa Halbouni, Teddy Surya Gunawan, Mohamed Hadi Habaebi, Murad Halbouni, Mira Kartiwi, and Robiah Ahmad. Machine learning and deep learning approaches for cybersecurity: A review. IEEE Access, 10:19572–19585, 2022.

[12] Karen Scarfone and Peter Mell. Intrusion detection and prevention systems. In Handbook of information and communication security, pages 177–192. Springer, 2010.

[13] Sharmila Kishor Wagh, Vinod K Pachghare, and Satish R Kolhe. Survey on intrusion detection system using machine learning techniques. International Journal of Computer Applications, 78(16):30–37, 2013.

[14] Hongyu Liu and Bo Lang. Machine learning and deep learning methods for intrusion detection systems: A survey. applied sciences, 9(20):4396, 2019.

[15] T Saranya, S Sridevi, C Deisy, Tran Duc Chung, and MKA Ahamed Khan. Performance analysis of machine learning algorithms in intrusion detection system: A review. Procedia Computer Science, 171:1251–1260, 2020.

[16] Donald Michie, David J Spiegelhalter, and Charles C Taylor. Machine learning, neural and statistical classification. 1994.

[17] Zeeshan Ahmad, Adnan Shahid Khan, Cheah Wai Shiang, Johari Abdullah, and Farhan Ahmad. Network intrusion detection system: A systematic study of machine learning and deep learning approaches. Transactions on Emerging Telecommunications Technologies, 32(1):e4150, 2021.

[18] Michelle Goddard. The eu general data protection regulation (gdpr): European regulation that has a global impact. International Journal of Market Research, 59(6):703–705, 2017.

[19] Chen Zhang, Yu Xie, Hang Bai, Bin Yu, Weihong Li, and Yuan Gao. A survey on federated learning. Knowledge-Based Systems, 216:106775, 2021.

[20] Lingjuan Lyu, Han Yu, and Qiang Yang. Threats to federated learning: A survey. arXiv preprint arXiv:2003.02133, 2020.

[21] Mohammed Aledhari, Rehma Razzak, Reza M Parizi, and Fahad Saeed. Federated learning: A survey on enabling technologies, protocols, and applications. IEEE Access, 8:140699–140725, 2020.

[22] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In Artificial intelligence and statistics, pages 1273–1282. PMLR, 2017.

[23] Raymond E Wright. Logistic regression. 1995.

[24] James A Anderson. An introduction to neural networks. MIT press, 1995.

[25] Shaashwat Agrawal, Sagnik Sarkar, Ons Aouedi, Gokul Yenduri, Kandaraj Piamrat, Mamoun Alazab, Sweta Bhattacharya, Praveen Kumar Reddy Maddikunta, and Thippa Reddy Gadekallu. Federated learning for intrusion detection system: Concepts, challenges and future directions. Computer Communications, 2022.

[26] Léo Lavaur, Marc-Oliver Pahl, Yann Busnel, and Fabien Autrel. The evolution of federated learning-based intrusion detection and mitigation: a survey. IEEE Transactions on Network and Service Management, 19(3):2309–2332, 2022.

[27] Helio N Cunha Neto, Jernej Hribar, Ivana Dusparic, Diogo Menezes Ferrazani Mattos, and Natalia C Fernandes. A survey on securing federated learning: Analysis of applications, attacks, challenges, and trends. IEEE Access, 11:41928–41953, 2023.

[28] Mohamed Amine Ferrag, Othmane Friha, Leandros Maglaras, Helge Janicke, and Lei Shu. Federated deep learning for cyber security in the internet of things: Concepts, applications, and experimental analysis. IEEE Access, 9:138509–138542, 2021.

[29] Rens Wouter van der Heijden, Stefan Dietzel, Tim Leinmüller, and Frank Kargl. Survey on misbehavior detection in cooperative intelligent transportation systems. IEEE Communications Surveys & Tutorials, 21(1):779–811, 2018.

[30] Mohammed Lamine Bouchouia, Houda Labiod, Ons Jelassi, Jean-Philippe Monteuuis, Wafa Ben Jaballah, Jonathan Petit, and Zonghua Zhang. A survey on misbehavior detection for connected and autonomous vehicles. Vehicular Communications, 41:100586, 2023.

[31] Rishu Chhabra, Saravjeet Singh, and Vikas Khullar. Privacy enabled driver behavior analysis in heterogeneous iov using federated learning. Engineering Applications of Artificial Intelligence, 120:105881, 2023.

[32] Aashma Uprety, Danda B Rawat, and Jiang Li. Privacy preserving misbehavior detection in iov using federated machine learning. In 2021 IEEE 18th Annual Consumer Communications & Networking Conference (CCNC), pages 1–6. IEEE, 2021.

[33] Abdelwahab Boualouache and Thomas Engel. Federated learning-based scheme for detecting passive mobile attackers in 5g vehicular edge computing. Annals of Telecommunications, 77(3):201–220, 2022.

[34] Abdelwahab Boualouache and Thomas Engel. A survey on machine learning-based misbehavior detection systems for 5g and beyond vehicular networks. arXiv preprint arXiv:2201.10500, 2022.

[35] Artúr István Károly, Róbert Fullér, and Péter Galambos. Unsupervised clustering for deep learning: A tutorial survey. Acta Polytechnica Hungarica, 15(8):29–53, 2018.

[36] R Saravanan and Pothula Sujatha. A state of art techniques on machine learning algorithms: a perspective of supervised learning approaches in data classification. In 2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS), pages 945–949. IEEE, 2018.

[37] Valerian Rey, Pedro Miguel Sánchez Sánchez, Alberto Huertas Celdrán, Gérôme Bovet, and Martin Jaggi. Federated learning for malware detection in iot devices. arXiv preprint arXiv:2104.09994, 2021.

[38] Hangyu Zhu, Jinjin Xu, Shiqing Liu, and Yaochu Jin. Federated learning on non-iid data: A survey. Neurocomputing, 465:371–390, 2021.

[39] Pengqian Yu, Achintya Kundu, Laura Wynter, and Shiau Hong Lim. Fed+: A unified approach to robust personalized federated learning. 2021.

[40] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. Proceedings of Machine Learning and Systems, 2:429–450, 2020.

[41] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. Journal of artificial intelligence research, 16:321–357, 2002.

[42] Viraaji Mothukuri, Reza M Parizi, Seyedamin Pouriyeh, Yan Huang, Ali Dehghantanha, and Gautam Srivastava. A survey on security and privacy of federated learning. Future Generation Computer Systems, 115:619–640, 2021.

[43] Zhanglong Ji, Zachary C Lipton, and Charles Elkan. Differential privacy and machine learning: a survey and review. arXiv preprint arXiv:1412.7584, 2014.

[44] Marco Barreno, Blaine Nelson, Russell Sears, Anthony D Joseph, and J Doug Tygar. Can machine learning be secure? In Proceedings of the 2006 ACM Symposium on Information, computer and communications security, pages 16–25, 2006.

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

[46] K Nanbu. Fourier transform method to determine the probability density function from a given set of random samples. Physical Review E, 52(6):5832, 1995.

[47] Dong Yin, Yudong Chen, Ramchandran Kannan, and Peter Bartlett. Byzantine-robust distributed learning: Towards optimal statistical rates. In International Conference on Machine Learning, pages 5650–5659. PMLR, 2018.

[48] Peva Blanchard, El Mahdi El Mhamdi, Rachid Guerraoui, and Julien Stainer. Machine learning with adversaries: Byzantine tolerant gradient descent. Advances in Neural Information Processing Systems, 30, 2017.

[49] Virat Shejwalkar and Amir Houmansadr. Manipulating the byzantine: Optimizing model poisoning attacks and defenses for federated learning. In NDSS, 2021.

[50] Xiaoyu Cao, Minghong Fang, Jia Liu, and Neil Zhenqiang Gong. Fltrust: Byzantine-robust federated learning via trust bootstrapping. arXiv preprint arXiv:2012.13995, 2020.

[51] Sotiris B Kotsiantis. Decision trees: a recent overview. Artificial Intelligence Review, 39:261–283, 2013.

[52] Leo Breiman. Random forests. Machine learning, 45:5–32, 2001.

[53] Marti A. Hearst, Susan T Dumais, Edgar Osuna, John Platt, and Bernhard Scholkopf. Support vector machines. IEEE Intelligent Systems and their applications, 13(4):18–28, 1998.

[54] Fionn Murtagh. Multilayer perceptrons for classification and regression. Neurocomputing, 2(5-6):183–197, 1991.

[55] Leila Arras, José Arjona-Medina, Michael Widrich, Grégoire Montavon, Michael Gillhofer, Klaus-Robert Müller, Sepp Hochreiter, and Wojciech Samek. Explaining and interpreting lstms. Explainable ai: Interpreting, explaining and visualizing deep learning, pages 211–238, 2019.

[56] Jiuxiang Gu, Zhenhua Wang, Jason Kuen, Lianyang Ma, Amir Shahroudy, Bing Shuai, Ting Liu, Xingxing Wang, Gang Wang, Jianfei Cai, et al. Recent advances in convolutional neural networks. Pattern recognition, 77:354–377, 2018.

[57] Chris M Bishop. Neural networks and their applications. Review of scientific instruments, 65(6):1803–1832, 1994.

[58] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114, 2013.

[59] Douglas A Reynolds. Gaussian mixture models. Encyclopedia of biometrics, 741(659-663), 2009.

[60] J Pérez Ortega, Ma Del, Roco Boone Rojas, and Mara J Somodevilla. Research issues on k-means algorithm: An experimental trial using matlab. In CEUR workshop proceedings: semantic web and new technologies, pages 83–96, 2009.

[61] Dan A Simovici. CLUSTERING: Theoretical and Practical Aspects. World Scientific, 2021.

[62] Adrian Nilsson, Simon Smith, Gregor Ulm, Emil Gustavsson, and Mats Jirstrand. A performance evaluation of federated learning algorithms. In Proceedings of the second workshop on distributed infrastructures for deep learning, pages 1–8, 2018.

[63] Viraj Kulkarni, Milind Kulkarni, and Aniruddha Pant. Survey of personalization techniques for federated learning. In 2020 Fourth World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4), pages 794–797. IEEE, 2020.

[64] Hongyi Wang, Mikhail Yurochkin, Yuekai Sun, Dimitris Papailiopoulos, and Yasaman Khazaeni. Federated learning with matched averaging. arXiv preprint arXiv:2002.06440, 2020.

[65] Amirhossein Reisizadeh, Aryan Mokhtari, Hamed Hassani, Ali Jadbabaie, and Ramtin Pedarsani. Fedpaq: A communication-efficient federated learning method with periodic averaging and quantization. In International Conference on Artificial Intelligence and Statistics, pages 2021–2031. PMLR, 2020.

[66] Abdullah Alsaedi, Nour Moustafa, Zahir Tari, Abdun Mahmood, and Adnan Anwar. Ton_iot telemetry dataset: A new generation dataset of iot and iiot for data-driven intrusion detection systems. IEEE Access, 8:165130–165150, 2020.

[67] Machine learning-based nids datasets.

[68] Arash Habibi Lashkari, Gerard Draper-Gil, Mohammad Saiful Islam Mamun, and Ali A Ghorbani. Characterization of tor traffic using time based features. In ICISSp, pages 253–262, 2017.

[69] Steven So, Prinkle Sharma, and Jonathan Petit. Integrating plausibility checks and machine learning for misbehavior detection in vanet. In 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), pages 564–571. IEEE, 2018.

[70] Christoph Sommer, David Eckhoff, Alexander Brummer, Dominik S Buse, Florian Hagenauer, Stefan Joerer, and Michele Segata. Veins: The open source vehicular network simulation framework. Recent advances in network simulation: the OMNeT++ environment and its ecosystem, pages 215–252, 2019.

[71] Lara Codecá, Raphaël Frank, Sébastien Faye, and Thomas Engel. Luxembourg sumo traffic (lust) scenario: Traffic demand evaluation. IEEE Intelligent Transportation Systems Magazine, 9(2):52–63, 2017.

[72] Ayyoob Hamza, Hassan Habibi Gharakheili, Theophilus A Benson, and Vijay Sivaraman. Detecting volumetric attacks on lot devices via sdn-based monitoring of mud activity. In ACM Symposium on SDN Research, pages 36–48, 2019.

[73] Eliot Lear, Dan Romascanu, and Ralph Droms. Manufacturer Usage Description Specification, 2019.

[74] Iman Sharafaldin, Arash Habibi Lashkari, and Ali A Ghorbani. Toward generating a new intrusion detection dataset and intrusion traffic characterization. ICISSp, 1:108–116, 2018.

[75] Mahbod Tavallaee, Ebrahim Bagheri, Wei Lu, and Ali A Ghorbani. A detailed analysis of the kdd cup 99 data set. In 2009 IEEE symposium on computational intelligence for security and defense applications, pages 1–6. IEEE, 2009.

[76] Yair Meidan, Michael Bohadana, Yael Mathov, Yisroel Mirsky, Asaf Shabtai, Dominik Breitenbacher, and Yuval Elovici. N-baiot—network-based detection of iot botnet attacks using deep autoencoders. IEEE Pervasive Computing, 17(3):12–22, 2018.

[77] Gregory Cohen, Saeed Afshar, Jonathan Tapson, and Andre Van Schaik. Emnist: Extending mnist to handwritten letters. In 2017 international joint conference on neural networks (IJCNN), pages 2921–2926. IEEE, 2017.

[78] Sebastian Caldas, Sai Meher Karthik Duddu, Peter Wu, Tian Li, Jakub Konečnỳ, H Brendan McMahan, Virginia Smith, and Ameet Talwalkar. Leaf: A benchmark for federated settings. arXiv preprint arXiv:1812.01097, 2018.

[79] Dinh C Nguyen, Ming Ding, Quoc-Viet Pham, Pubudu N Pathirana, Long Bao Le, Aruna Seneviratne, Jun Li, Dusit Niyato, and H Vincent Poor. Federated learning meets blockchain in edge computing: Opportunities and challenges. IEEE Internet of Things Journal, 2021.

[80] Burak Cetin, Alina Lazar, Jinoh Kim, Alex Sim, and Kesheng Wu. Federated wireless network intrusion detection. In 2019 IEEE International Conference on Big Data (Big Data), pages 6004–6006. IEEE, 2019.

[81] Mamoun Alazab, Swarna Priya RM, M Parimala, Praveen Kumar Reddy Maddikunta, Thippa Reddy Gadekallu, and Quoc-Viet Pham. Federated learning for cybersecurity: Concepts, challenges, and future directions. IEEE Transactions on Industrial Informatics, 18(5):3501–3509, 2021.

[82] Bimal Ghimire and Danda B Rawat. Recent advances on federated learning for cybersecurity and cybersecurity for federated learning for internet of things. IEEE Internet of Things Journal, 2022.

[83] Chris Johnson, Lee Badger, David Waltermire, Julie Snyder, Clem Skorupka, et al. Guide to cyber threat information sharing. NIST special publication, 800(150):35, 2016.

[84] J. Li, L. Lyu, X. Liu, X. Zhang, and X. Lyu. Fleam: A federated learning empowered architecture to mitigate ddos in industrial iot. arXiv:2012.06150, 2020.

[85] Rahul Dey and Fathi M Salem. Gate-variants of gated recurrent unit (gru) neural networks. In 2017 IEEE 60th international midwest symposium on circuits and systems (MWSCAS), pages 1597–1600. IEEE, 2017.

[86] Thien Duc Nguyen, Samuel Marchal, Markus Miettinen, Hossein Fereidooni, N Asokan, and Ahmad-Reza Sadeghi. Dïot: A federated self-learning anomaly detection system for iot. In 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS), pages 756–767. IEEE, 2019.

[87] Jiechen Luo, Xuelan Yang, and Muamer N Mohammed. Federation learning for intrusion detection methods by parse convolutional neural network. In 2022 Second International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT), pages 1–7. IEEE, 2022.

[88] Xuelan Yang, Jiechen Luo, and Muamer N Mohammed. Federation learning of optimized convolutional neural network structure for intrusion detection. In 2022 Second International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT), pages 1–7. IEEE, 2022.

[89] Qiaofeng Qin, Konstantinos Poularakis, Kin K Leung, and Leandros Tassiulas. Line-speed and scalable intrusion detection at the network edge via federated learning. In 2020 IFIP Networking Conference (Networking), pages 352–360. IEEE, 2020.

[90] Dongmin Wu, Yi Deng, and Mingyong Li. Fl-mgvn: Federated learning for anomaly detection using mixed gaussian variational self-encoding network. Information Processing & Management, 59(2):102839, 2022.

[91] Zhaoyang Du, Celimuge Wu, Tsutomu Yoshinaga, Kok-Lim Alvin Yau, Yusheng Ji, and Jie Li. Federated learning for vehicular internet of things: Recent advances and open issues. IEEE Open Journal of the Computer Society, 1:45–61, 2020.

[92] Jason Posner, Lewis Tseng, Moayad Aloqaily, and Yaser Jararweh. Federated learning in vehicular networks: opportunities and solutions. IEEE Network, 35(2):152–159, 2021.

[93] Ahmet M Elbir, Burak Soner, and Sinem Coleri. Federated learning in vehicular networks. arXiv preprint arXiv:2006.01412, 2020.

[94] Zhongnan Zhao, Xiaoliang Liang, Hai Huang, and Kun Wang. Deep federated learning hybrid optimization model based on encrypted aligned data. Pattern Recognition, 148:110193, 2024.

[95] Yang Chen, Junzhe Zhang, and Chai Kiat Yeo. Network anomaly detection using federated deep autoencoding gaussian mixture model. In International Conference on Machine Learning for Networking, pages 1–14. Springer, 2019.

[96] Davy Preuveneers, Vera Rimmer, Ilias Tsingenopoulos, Jan Spooren, Wouter Joosen, and Elisabeth Ilie-Zudor. Chained anomaly detection models for federated learning: An intrusion detection case study. Applied Sciences, 8(12):2663, 2018.

[97] Malik Bader Alazzam, Fawaz Alassery, and Ahmed Almulihi. Federated deep learning approaches for the privacy and security of iot systems. Wireless Communications and Mobile Computing, 2022, 2022.

[98] Sawsan AbdulRahman, Hanine Tout, Hakima Ould-Slimane, Azzam Mourad, Chamseddine Talhi, and Mohsen Guizani. A survey on federated learning: The journey from centralized to distributed on-site learning and beyond. IEEE Internet Things J, 8(7):5476–5497, 2021.

[99] Omar Abdel Wahab, Azzam Mourad, Hadi Otrok, and Tarik Taleb. Federated machine learning: Survey, multi-level classification, desirable criteria and future directions in communication and networking systems. IEEE Communications Surveys & Tutorials, 23(2):1342–1397, 2021.

[100] Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. On the convergence of fedavg on non-iid data. arXiv preprint arXiv:1907.02189, 2019.

[101] Peng Zhou. Federated deep payload classification for industrial internet with cloud-edge architecture. In 2020 16th International Conference on Mobility, Sensing and Networking (MSN), pages 228–235. IEEE, 2020.

[102] Shuai Yuan, Hongwei Li, Rui Zhang, Meng Hao, Yiran Li, and Rongxing Lu. Towards lightweight and efficient distributed intrusion detection framework. In 2021 IEEE Global Communications Conference (GLOBECOM), pages 1–6. IEEE, 2021.

[103] Othmane Friha, Mohamed Amine Ferrag, Lei Shu, Leandros Maglaras, Kim-Kwang Raymond Choo, and Mehdi Nafaa. Felids: Federated learning-based intrusion detection system for agricultural internet of things. Journal of Parallel and Distributed Computing, 165:17–31, 2022.

[104] Wentao Liu, Xiaolong Xu, Lianxiang Wu, Lianyong Qi, Alireza Jolfaei, Weiping Ding, and Mohammad R Khosravi. Intrusion detection for maritime transportation systems with batch federated aggregation. IEEE Transactions on Intelligent Transportation Systems, 2022.

[105] Dingling Su and Zehui Qu. Detection ddos of attacks based on federated learning with digital twin network. In International Conference on Knowledge Science, Engineering and Management, pages 153–164. Springer, 2022.

[106] Sanket Shukla, PD Sai Manoj, Gaurav Kolhe, and Setareh Rafatirad. On-device malware detection using performance-aware and robust collaborative learning. In 2021 58th ACM/IEEE Design Automation Conference (DAC), pages 967–972. IEEE, 2021.

[107] Sanket Shukla, Gaurav Kolhe, Houman Homayoun, Setareh Rafatirad, and Sai Manoj PD. Rafel-robust and data-aware federated learning-inspired malware detection in internet-of-things (iot) networks. In Proceedings of the Great Lakes Symposium on VLSI 2022, pages 153–157, 2022.

[108] Segun I Popoola, Guan Gui, Bamidele Adebisi, Mohammad Hammoudeh, and Haris Gacanin. Federated deep learning for collaborative intrusion detection in heterogeneous networks. In 2021 IEEE 94th Vehicular Technology Conference (VTC2021-Fall), pages 1–6. IEEE, 2021.

[109] Meryem Janati Idrissi, Hamza Alami, Abdelkader El Mahdaouy, Abdellah El Mekki, Soufiane Oualil, Zakaria Yartaoui, and Ismail Berrada. Fed-anids: Federated learning for anomaly-based network intrusion detection systems. Expert Systems with Applications, 234:121000, 2023.

[110] Bern Jonathan, Panca Hadi Putra, and Yova Ruldeviyani. Observation imbalanced data text to predict users selling products on female daily with smote, tomek, and smote-tomek. In 2020 IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology (IAICT), pages 81–85. IEEE, 2020.

[111] Zhaozhao Xu, Derong Shen, Tiezheng Nie, and Yue Kou. A hybrid sampling algorithm combining m-smote and enn based on random forest for medical imbalanced data. Journal of Biomedical Informatics, 107:103465, 2020.

[112] Aliya Tabassum, Aiman Erbad, Wadha Lebda, Amr Mohamed, and Mohsen Guizani. Fedgan-ids: Privacy-preserving ids using gan and federated learning. Computer Communications, 192:299–310, 2022.

[113] Kang Wei, Jun Li, Ming Ding, Chuan Ma, Howard H Yang, Farhad Farokhi, Shi Jin, Tony QS Quek, and H Vincent Poor. Federated learning with differential privacy: Algorithms and performance analysis. IEEE transactions on information forensics and security, 15:3454–3469, 2020.

[114] Jun Niu, Peng Liu, Xiaoyan Zhu, Kuo Shen, Yuecong Wang, Haotian Chi, Yulong Shen, Xiaohong Jiang, Jianfeng Ma, and Yuqing Zhang. A survey on membership inference attacks and defenses in machine learning. Journal of Information and Intelligence, 2024.

[115] Olakunle Ibitoye, M Omair Shafiq, and Ashraf Matrawy. Differentially private self-normalizing neural networks for adversarial robustness in federated learning. Computers & Security, 116:102631, 2022.

[116] KP Sanal Kumar, S Anu H Nair, Deepsubhra Guha Roy, B Rajalingam, and R Santhosh Kumar. Security and privacy-aware artificial intrusion detection system using federated machine learning. Computers & Electrical Engineering, 96:107440, 2021.

[117] Huynh Nhat Hao, Huynh Minh Chu, Van-Hau Pham, et al. A secure and privacy preserving federated learning approach for iot intrusion detection system. In International Conference on Network and System Security, pages 353–368. Springer, 2021.

[118] Ruei-Hau Hsu, Yi-Cheng Wang, Chun-I Fan, Bo Sun, Tao Ban, Takeshi Takahashi, Ting-Wei Wu, and Shang-Wei Kao. A privacy-preserving federated learning system for android malware detection based on edge computing. In 2020 15th Asia Joint Conference on Information Security (AsiaJCIS), pages 128–136. IEEE, 2020.

[119] Bin Zhao, Kai Fan, Kan Yang, Zilong Wang, Hui Li, and Yintang Yang. Anonymous and privacy-preserving federated learning with industrial big data. IEEE Transactions on Industrial Informatics, 2021.

[120] Rui Hu, Yuanxiong Guo, E Paul Ratazzi, and Yanmin Gong. Differentially private federated learning for resource-constrained internet of things. arXiv preprint arXiv:2003.12705, 2020.

[121] Rui Hu, Yuanxiong Guo, Hongning Li, Qingqi Pei, and Yanmin Gong. Personalized federated learning with differential privacy. IEEE Internet of Things Journal, 7(10):9530–9539, 2020.

[122] Yunlong Lu, Xiaohong Huang, Yueyue Dai, Sabita Maharjan, and Yan Zhang. Blockchain and federated learning for privacy-preserved data sharing in industrial iot. IEEE Transactions on Industrial Informatics, 16(6):4177–4186, 2019.

[123] Alberto Blanco-Justicia, Josep Domingo-Ferrer, Sergio Martínez, David Sánchez, Adrian Flanagan, and Kuan Eeik Tan. Achieving security and privacy in federated learning systems: Survey, research challenges and future directions. Engineering Applications of Artificial Intelligence, 106:104468, 2021.

[124] Arjun Nitin Bhagoji, Supriyo Chakraborty, Prateek Mittal, and Seraphin Calo. Analyzing federated learning through an adversarial lens. In International Conference on Machine Learning, pages 634–643. PMLR, 2019.

[125] Zhiyi Tian, Lei Cui, Jie Liang, and Shui Yu. A comprehensive survey on poisoning attacks and countermeasures in machine learning. ACM Computing Surveys, 55(8):1–35, 2022.

[126] Valerian Rey, Pedro Miguel Sánchez Sánchez, Alberto Huertas Celdrán, and Gérôme Bovet. Federated learning for malware detection in iot devices. Computer Networks, 204:108693, 2022.

[127] Narendra Singh, Harsh Kasyap, and Somanath Tripathy. Collaborative learning based effective malware detection system. In Joint European Conference on Machine Learning and Knowledge Discovery in Databases, pages 205–219. Springer, 2020.

[128] Wassila Lalouani and Mohamed Younis. A robust distributed intrusion detection system for collusive attacks on edge of things. In 2022 IEEE Wireless Communications and Networking Conference (WCNC), pages 1004–1009. IEEE, 2022.

[129] Krishna Yadav and BB Gupta. Clustering based rewarding algorithm to detect adversaries in federated machine learning based iot environment. In 2021 IEEE International Conference on Consumer Electronics (ICCE), pages 1–6. IEEE, 2021.

[130] Juan A Bonachela, Haye Hinrichsen, and Miguel A Munoz. Entropy estimates of small data sets. Journal of Physics A: Mathematical and Theoretical, 41(20):202001, 2008.

[131] Gustavo EAPA Batista, Ronaldo C Prati, and Maria Carolina Monard. A study of the behavior of several methods for balancing machine learning training data. ACM SIGKDD explorations newsletter, 6(1):20–29, 2004.

[132] T Elhassan and M Aljurf. Classification of imbalance data using tomek link (t-link) combined with random under-sampling (rus) as a data reduction method. Global J Technol Optim S, 1:2016, 2016.

[133] Leif E Peterson. K-nearest neighbor. Scholarpedia, 4(2):1883, 2009.

[134] Sashank Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečnỳ, Sanjiv Kumar, and H Brendan McMahan. Adaptive federated optimization. arXiv preprint arXiv:2003.00295, 2020.

[135] Jinhyun So, Başak Güler, and A Salman Avestimehr. Turbo-aggregate: Breaking the quadratic aggregation barrier in secure federated learning. IEEE Journal on Selected Areas in Information Theory, 2(1):479–489, 2021.

[136] Wentai Wu, Ligang He, Weiwei Lin, Rui Mao, Carsten Maple, and Stephen Jarvis. Safa: A semi-asynchronous protocol for fast federated learning with low overhead. IEEE Transactions on Computers, 70(5):655–668, 2020.

[137] Shifei Ding, Chunyang Su, and Junzhao Yu. An optimizing bp neural network algorithm based on genetic algorithm. Artificial intelligence review, 36:153–162, 2011.

[138] Ankit Thakkar and Ritika Lohiya. Analyzing fusion of regularization techniques in the deep learning-based intrusion detection system. International Journal of Intelligent Systems, 36(12):7340–7388, 2021.

[139] Dhurgham Kareem Ghurkan and Amer A Abdulrahman. Construct an efficient ddos attack detection system based on rf-c4. 5-gridsearchcv. In 2022 Iraqi International Conference on Communication and Information Technologies (IICCIT), pages 120–124. IEEE, 2022.

[140] Xue Ying. An overview of overfitting and its solutions. In Journal of physics: Conference series, volume 1168, page 022022. IOP Publishing, 2019.

[141] Azade Rezaeezade and Lejla Batina. Regularizers to the rescue: fighting overfitting in deep learning-based side-channel analysis. Cryptology ePrint Archive, 2022.

[142] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. science, 313(5786):504–507, 2006.

[143] Xin-Jie Wu, Ming-Da Xu, Chang-Di Li, Chong Ju, Qian Zhao, and Shi-Xing Liu. Research on image reconstruction algorithms based on autoencoder neural network of restricted boltzmann machine (rbm). Flow Measurement and Instrumentation, 80:102009, 2021.

[144] Andre GC Pacheco, Renato A Krohling, and Carlos AS da Silva. Restricted boltzmann machine to determine the input weights for extreme learning machines. Expert Systems with Applications, 96:77–85, 2018.

[145] S Saravanan and Juliet Sujitha. Deep medical image reconstruction with autoencoders using deep boltzmann machine training. EAI Endorsed Transactions on Pervasive Health and Technology, 6(24):e2, 2020.

[146] Daniel J. Beutel, Taner Topal, Akhil Mathur, Xinchi Qiu, Titouan Parcollet, and Nicholas D. Lane. Flower: A friendly federated learning research framework. CoRR, abs/2007.14390, 2020.

[147] Heiko Ludwig, Nathalie Baracaldo, Gegi Thomas, Yi Zhou, Ali Anwar, Shashank Rajamoni, Yuya Ong, Jayaram Radhakrishnan, Ashish Verma, Mathieu Sinn, Mark Purcell, Ambrish Rawat, Tran Minh, Naoise Holohan, Supriyo Chakraborty, Shalisha Witherspoon, Dean Steuer, Laura Wynter, Hifaz Hassan, Sean Laguna, Mikhail Yurochkin, Mayank Agarwal, Ebube Chuba, and Annie Abay. Ibm federated learning: an enterprise framework white paper v0.1, 2020.

[148] Manh Hung Nguyen. Impacts of unbalanced test data on the evaluation of classification methods. International Journal of Advanced Computer Science and Applications, 10(3), 2019.

[149] Le Wang, Meng Han, Xiaojuan Li, Ni Zhang, and Haodong Cheng. Review of classification methods on unbalanced data sets. Ieee Access, 9:64606–64628, 2021.

[150] Vaishali Ganganwar. An overview of classification algorithms for imbalanced datasets. International Journal of Emerging Technology and Advanced Engineering, 2(4):42–47, 2012.

[151] Margherita Grandini, Enrico Bagli, and Giorgio Visani. Metrics for multi-class classification: an overview. arXiv preprint arXiv:2008.05756, 2020.

[152] Ziyao Liu, Jiale Guo, Wenzhuo Yang, Jiani Fan, Kwok-Yan Lam, and Jun Zhao. Privacy-preserving aggregation in federated learning: A survey. IEEE Transactions on Big Data, 2022.

[153] Olivia Choudhury, Aris Gkoulalas-Divanis, Theodoros Salonidis, Issa Sylla, Yoonyoung Park, Grace Hsu, and Amar Das. Anonymizing data for privacy-preserving federated learning. arXiv preprint arXiv:2002.09096, 2020.

[154] Christopher Briggs, Zhong Fan, and Peter Andras. A review of privacy-preserving federated learning for the internet-of-things. Federated Learning Systems, pages 21–50, 2021.

[155] Israel Cohen, Yiteng Huang, Jingdong Chen, Jacob Benesty, Jacob Benesty, Jingdong Chen, Yiteng Huang, and Israel Cohen. Pearson correlation coefficient. Noise reduction in speech processing, pages 1–4, 2009.

[156] Vance W Berger and YanYan Zhou. Kolmogorov–smirnov test: Overview. Wiley statsref: Statistics reference online, 2014.

[157] Jian Xu, Shao-Lun Huang, Linqi Song, and Tian Lan. Byzantine-robust federated learning through collaborative malicious gradient filtering. In 2022 IEEE 42nd International Conference on Distributed Computing Systems (ICDCS), pages 1223–1235. IEEE, 2022.

[158] Wei Wan, Shengshan Hu, Jianrong Lu, Leo Yu Zhang, Hai Jin, and Yuanyuan He. Shielding federated learning: Robust aggregation with adaptive client selection. arXiv preprint arXiv:2204.13256, 2022.

[159] Jessica Lin, Eamonn Keogh, Stefano Lonardi, and Bill Chiu. A symbolic representation of time series, with implications for streaming algorithms. In Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery, pages 2–11, 2003.

[160] Vijay Kotu and Bala Deshpande. Chapter 11 - recommendation engines. In Vijay Kotu and Bala Deshpande, editors, Data Science (Second Edition), pages 343–394. Morgan Kaufmann, second edition edition, 2019.

[161] Clement Fung, Chris JM Yoon, and Ivan Beschastnikh. Mitigating sybils in federated learning poisoning. arXiv preprint arXiv:1808.04866, 2018.

[162] Gilad Baruch, Moran Baruch, and Yoav Goldberg. A little is enough: Circumventing defenses for distributed learning. Advances in Neural Information Processing Systems, 32, 2019.

[163] Minghong Fang, Xiaoyu Cao, Jinyuan Jia, and Neil Gong. Local model poisoning attacks to {Byzantine-Robust} federated learning. In 29th USENIX security symposium (USENIX Security 20), pages 1605–1622, 2020.

Capítulo 6

Publications composing PhD Thesis

6.1 — Evaluating Federated Learning for intrusion detection in Internet of Things: Review and challenges

Title	**Evaluating Federated Learning for intrusion detection in Internet of Things: Review and challenges**
Authors	Enrique Mármol Campos[1], Pablo Fernández Saura[1], Aurora González Vidal[1], José Luis Hernández Ramos[1], Jorge Bernal Bernabe[1], Gianmarco Baldoni[2], Antonio Skarmeta[1]
Type	Journal
Journal	Computer Networks
Impact Factor	5.493
Rank	9/51 (Q1)
Publisher	Elsevier
Volume	16
Pages	108661
Year	2022
Month	February
DOI	`https://doi.org/10.1016/j.comnet.2021.108661`
URL	`https://www.sciencedirect.com/science/article/pii/S1389128621005405`
Status	Published

Abstract	The application of Machine Learning (ML) techniques to well-known Intrusion Detection Systems (IDS) is key to cope with increasingly sophisticated cybersecurity attacks through an effective and efficient detection process. In the context of the Internet of Things (IoT), most ML-enabled IDS approaches use centralized approaches where IoT devices share their data with data centers for further analysis. To mitigate privacy concerns associated with centralized approaches, in recent years the use of Federated Learning (FL) has attracted significant interest in different sectors, including healthcare and transport systems. However, the development of FL-enabled IDS for IoT is in its infancy, and still requires research efforts from various areas, in order to identify the main challenges for the deployment in real-world scenarios. In this direction, our work evaluates an FL-enabled IDS approach based on a multiclass classifier considering different data distributions for the detection of different attacks in an IoT scenario. In particular, we use three different settings that are obtained by partitioning the recent ToN_IoT dataset according to IoT devices' IP addresses and types of attacks. Furthermore, we evaluate the impact of different aggregation functions according to such settings by using the recent IBMFL framework as FL implementation. Additionally, we identify a set of challenges and future directions based on the existing literature and the analysis of our evaluation results.

[1] Universidad de Murcia, Departamento de Ingeniería de la Información y las Comunicaciones, España

[2] European Commission, Joint Research Centre, Ispra, Italy

6.2 — Intrusion Detection Based on Privacy-Preserving Federated Learning for the Industrial IoT

Title	**Intrusion Detection Based on Privacy-Preserving Federated Learning for the Industrial IoT**
Authors	Pedro Ruzafa Alcázar[1], Pablo Fernández Saura[1], Enrique Mármol Campos[1], Aurora González Vidal[1], José Luis Hernández Ramos[1], Jorge Bernal Bernabe[1], Antonio Skarmeta[1]
Type	Journal
Journal	IEEE Transactions on Industrial Informatics
Impact Factor	11.468
Rank	3/65 (D1)
Publisher	IEEE
Volume	19
Issue	2
Pages	1145-1154
Year	2021
Month	November
DOI	10.1109/TII.2021.3126728
URL	https://ieeexplore.ieee.org/abstract/document/9609643
Status	Published

Abstract	The application of Machine Learning (ML) techniques to well-known Intrusion Detection Systems (IDS) is key to cope with increasingly sophisticated cybersecurity attacks through an effective and efficient detection process. In the context of the Internet of Things (IoT), most ML-enabled IDS approaches use centralized approaches where IoT devices share their data with data centers for further analysis. To mitigate privacy concerns associated with centralized approaches, in recent years the use of Federated Learning (FL) has attracted significant interest in different sectors, including healthcare and transport systems. However, the development of FL-enabled IDS for IoT is in its infancy, and still requires research efforts from various areas, in order to identify the main challenges for the deployment in real-world scenarios. In this direction, our work evaluates an FL-enabled IDS approach based on a multiclass classifier considering different data distributions for the detection of different attacks in an IoT scenario. In particular, we use three different settings that are obtained by partitioning the recent ToN_IoT dataset according to IoT devices' IP addresses and types of attacks. Furthermore, we evaluate the impact of different aggregation functions according to such settings by using the recent IBMFL framework as FL implementation. Additionally, we identify a set of challenges and future directions based on the existing literature and the analysis of our evaluation results.

[1] Universidad de Murcia, Departamento de Ingeniería de la Información y las Comunicaciones, España

6.3 — Federated Cyberattack Detection for Internet of Things-Enabled Smart Cities

Title	**Federated Cyberattack Detection for Internet of Things-Enabled Smart Cities**
Authors	Sara Nieves Matheu[1], Enrique Mármol Campos[1], José Luis Hernández Ramos[1], Antonio Skarmeta[1], Gianmarco Baldini[2]
Type	Magazine
Journal	IEEE Computer
Impact Factor	2.2
Rank	58/108 (Q3)
Publisher	IEEE
Volume	55
Issue	12
Pages	65 - 73
Year	2022
Month	November
DOI	10.1109/MC.2022.3195054
URL	https://ieeexplore.ieee.org/abstract/document/9963740
Status	Published
Abstract	With the increasing digitization of our surrounding environment, the effective and efficient detection of cyberattacks is key to realizing trustworthy smart cities. In this context, the use of Artificial Intelligence (AI) has aroused a significant interest in dealing with increasingly sophisticated cyberattacks. However, the detection of such threats is typically based on analyzing large amounts of network traffic data, which can lead to privacy issues for citizens. Addressing this issue, this work proposes an FL approach to the identification of cyberattacks in the context of IoT-enabled smart cities. Our work integrates the Manufacturer Usage Description standard as a prevention/mitigation approach based on network rules with FL component for the identification of several cyberattacks. We demonstrate the feasibility of our approach under different FL settings using a dataset derived from network traffic of real IoT devices with an accuracy value of around 90 %.

[1] Universidad de Murcia, Departamento de Ingeniería de la Información y las Comunicaciones, España

[2] European Commission, Joint Research Centre, Ispra, Italy

6.4 — Misbehavior detection in intelligent transportation systems based on federated learning

Title	**Misbehavior detection in intelligent transportation systems based on federated learning**
Authors	Enrique Mármol Campos[1], Aurora González Vidal[1], José Luis Hernández Ramos[1], Gianmarco Baldoni[2], Antonio Skarmeta[1]
Type	Journal
Journal	Internet of Things
Impact Factor	5.9
Rank	35/158 (Q1)
Publisher	Elsevier
Volume	25
Pages	13
Year	2024
Month	April
DOI	https://doi.org/10.1016/j.iot.2024.101127
URL	https://www.sciencedirect.com/science/article/pii/S2542660524000696
Status	Published
Abstract	Misbehavior detection represents a key security approach in vehicular scenarios to identify attacks that cannot be detected by traditional cryptographic mechanisms. In this context, the application of ML techniques has been widely considered to identify increasingly sophisticated misbehavior attacks. However, most of the proposed approaches are based on centralized settings, which could pose privacy issues, as well as an increased latency leading to severe consequences in the vehicular environment where real-time and scalability requirements are challenging. To address this issue, we propose a collaborative learning approach based on FL for vehicles' misbehavior detection. We use the reference misbehavior dataset VeReMi, which is re-balanced by applying the SMOTE-Tomek technique. We carry out a thorough evaluation considering different balancing settings and the number of nodes. The evaluation results overcome recent state-of-the-art approaches, with an overall accuracy of 93 % using an optimized multilayer perceptron (MLP) for multiclass classification.

[1] Universidad de Murcia, Departamento de Ingeniería de la Información y las Comunicaciones, España

[2] European Commission, Joint Research Centre, Ispra, Italy