UNIVERSITY OF GRANADA

DEPARTMENT OF COMPUTER SCIENCE
AND ARTIFICIAL INTELLIGENCE

PHD PROGRAM IN INFORMATION AND
COMMUNICATION TECHNOLOGIES

PhD THESIS DISSERTATION

Adversarial attacks and defences in Federated Learning

PhD CANDIDATE
Nuria Rodríguez Barroso

PhD ADVISOR
Francisco Herrera Triguero

Granada, Oct 2023

Editor: Universidad de Granada. Tesis Doctorales
Autor: Nuria Rodríguez Barroso
ISBN: 978-84-1195-132-6
URI: https://digibug.ugr.es/handle/10481/89129

Funding

This doctoral thesis has been partially supported by the research projects PP2016.PRI.I.02 and PP2019.PRI.I.06, the national research projects TIN2017-89517-P and PID2020-119478GB-I00, and the regional research project A-TIC-434-UGR20. Nuria Rodríguez Barroso were supported by the grant FPU18/04475 funded by MCIN/AEI/10.13039/501100011033 and by "ESF Investing in your future", Spain.

«*For most of history, Anonymous was a woman*».

- Virginia Woolf.

Agradecimientos

Gracias a todas y todos.

Abstract

Artificial Intelligence (AI) is currently in the process of revolutionising numerous facets of everyday life. Nevertheless, as its development progresses, the associated risks are on the rise. Despite the fact that its full potential remains uncertain, there is a growing apprehension regarding its deployment in sensitive domains such as education, culture, and medicine. Presently, one of the foremost challenges confronting us is finding a harmonious equilibrium between the prospective advantages and the attendant risks, thereby preventing precaution from impeding innovation. This necessitates the development of AI systems that are robust, secure, transparent, fair, respectful to privacy and autonomy, have clear traceability, and are subject to fair accountability for auditing. In essence, it entails ensuring their ethical and responsible application, giving rise to the concept of trustworthy AI.

In this context, Federated Learning (FL) emerges as a paradigm of distributed learning that ensures the privacy of training data while also harnessing global knowledge. Although its ultimate objective is data privacy, it also brings forth other cross-cutting enhancements such as robustness and communication cost minimisation. However, like any learning paradigm, FL is susceptible to adversarial attacks aimed at altering the model's operation or inferring private information. The central focus of this thesis is the development of defence mechanisms against adversarial attacks that compromise the model's behaviour while concurrently promoting other requirements to ensure trustworthy AI.

This thesis addresses the following objectives:

- The first objective involves a study of existing adversarial attacks and defences within FL. Its purpose is to provide a comprehensive taxonomy encompassing all categories of attacks and defences, along with an experimental study to glean insights into which defences effectively mitigate the impact of specific attacks.
- The second objective involves the development of defence mechanisms against backdoor attacks. These attacks entail imperceptibly injecting a secondary task, i.e., without altering the operation in the original task. Such attacks pose a challenge in FL due to their difficulty in detection. The proposed defence mechanism, based on outlier detection at the server, has demonstrated superior performance compared to the available defence mechanisms in the literature.
- The third objective is to develop defences against byzantine attacks. These attacks in-

volve random attacks aimed at hindering the model's training, causing it to produce worse results in the original task. Although they are easier to detect, they pose a challenge because they are the most common in the literature due to their ease of execution. The proposed dynamic defence mechanism represents an improvement over the state of the art by surpassing the performance of other proposals and being agnostic to the number of clients carrying out the attack.

The three objectives set forth in the thesis are successfully addressed. Both objectives related to the development of new defence mechanisms are substantiated with comparative empirical studies, as well as more in-depth analyses of their behaviour. Both the literature review and the proposed defence mechanisms contribute innovation to the research field, enhancing the existing body of literature while also opening up different avenues for future research.

Finally, we present current work on taking a step beyond defence against adversarial attacks, requiring the fulfilment of additional requirements for trustworthy AI. Specifically, we build upon the previous proposal for defence against adversarial attacks, we shift the focus from the performance of clients to categorising them into explainability methods. In doing so, we achieve a defence mechanism that maintains its performance in terms of defence against byzantine attacks while also providing other requirements such as fairness, transparency, and explainability in client selection.

Resumen

La Inteligencia Artificial (*Artificial Intelligence* - AI) está cambiando de raíz múltiples aspectos de la vida cotidiana. Sin embargo, a medida que avanza su desarrollo, se incrementan los riesgos derivados de su uso. Aunque todavía no se conoce su potencial, cada vez es mayor la preocupación por su uso en campos delicados como la educación, la cultura o la medicina. Uno de los mayores retos en los que nos encontramos ahora mismo es encontrar el balance entre los potenciales beneficios y los riesgos ocasionados, de forma que la prevención no pare a la innovación. Esto implica desarrollar sistemas de AI que sean robustos, seguros, transparentes, justos, respetuosos con la privacidad y la autonomía, que tengan una trazabilidad clara y auditables. En definitiva, garantizar su aplicación ética y responsable, donde nacen los conceptos de AI confiable y sistema de AI responsable.

En este contexto surge el Aprendizaje Federado (*Federated Learning* - FL) como un paradigma de aprendizaje distribuido que asegura la privacidad de los datos de entrenamiento al mismo tiempo que es capaz de aprovechar el conocimiento global. Aunque su objetivo final es la privacidad de datos, también aporta otras mejoras transversales como la robustez y la minimización de costes de comunicación. Sin embargo, al igual que cualquier paradigma de aprendizaje, el FL es susceptible a ataques adversarios que pretenden modificar el funcionamiento del modelo o inferir información privada. El eje central de esta tesis es el desarrollo de mecanismos de defensa contra ataques adversarios que comprometen el funcionamiento del modelo, al mismo tiempo que se fomentan otros requerimientos para asegurar una AI confiable.

Esta tesis aborda los siguientes objetivos:

- El primero consiste en un estudio de los ataques adversarios y defensas existentes en FL. Su finalidad es proporcionar una taxonomía completa que abarque todas las categorías de ataques y defensas, así como un estudio experimental que permita obtener lecciones sobre qué defensas consiguen paliar el efecto de qué ataques.

- El segundo objetivo aborda el desarrollo de mecanismos de defensa contra ataques *backdoor* o de tarea secundaria. Estos ataques consisten en inyectar una tarea secundaria de forma imperceptible, esto es, sin modificar el funcionamiento en la tarea original. Estos ataques representan un reto en FL por la dificultad para ser detectados. El mecanismo de defensa propuesto basado en detección de *outliers* en el servidor ha

probado proporcionar mejores resultados que los mecanismos de defensa disponibles en la literatura.

- El tercer objetivo trata de desarrollar mecanismos de defensa contra ataques bizantinos. Estos ataques consisten en ataques aleatorios cuya finalidad es impedir el entrenamiento del modelo haciendo que este produzca peores resultados en la tarea original. Aunque son más fáciles de detectar, suponen un reto pues son los más comunes en la literatura por la facilidad de llevarlos a cabo. El mecanismo de defensa dinámico propuesto supone una mejora con respecto al estado del arte por superar el rendimiento del resto de propuestas y ser agnóstico al número de clientes que llevan a cabo el ataque.

Los tres objetivos planteados en la tesis se abordan de forma exitosa. Los dos objetivos relacionados con el desarrollo de nuevos mecanismos de defensa se avalan con estudios empíricos comparativos, así como con análisis más profundos de su comportamiento. Tanto el estudio de la literatura, como los mecanismos de defensa propuestos aportan innovación al campo de investigación mejorando la literatura ya existente al mismo tiempo que abren diferentes líneas de investigación futuras.

Finalmente, se presenta un trabajo actual de dar un paso más allá de la defensa contra ataques adversarios, exigiendo que se cumplan otros requerimientos para una IA confiable. En concreto, partimos de la propuesta anterior de defensa frente a ataques adversarios y movemos el foco de atención del rendimiento de los clientes para categorizarlos a métodos de explicabilidad. De esta forma, conseguimos un mecanismo de defensa que mantiene su rendimiento en cuanto a defensa, al mismo tiempo que aporta otros requerimientos como justicia, transparencia y explicabilidad en la selección de clientes.

Table of Contents

List of Abbreviations

ML	Machine Learning
FL	Federated Learning
AI	Artificial Intelligence
non-IID	Non Independent and Identically Distributed
OWA	Ordered Weighted Averaging
IOWA	Induced Ordered Weighted Averaging
HFL	Horizontal Federated Learning
VFL	Vertical Federated Learning
FTL	Federated Transfer Learning
FedAvg	Federated Averaging
LM	Learning model
DP	Differential Privacy
SMPC	Secure Multi-Party Computation
DDaBA	Dynamic Defence against Byzantine Attacks
CelebA	Large-scale CelebFaces Attributes Dataset
FEMNIST	Federated Extended MNIST
EMNIST	Extended MNIST
SDaBA	Static Defence against Byzantine Attacks
IoT	Internet of Things
LIME	Linear Model-agnostic Explanation

SHAP	SHapley Additive exPlanations
XAI	eXplainable Artificial Intelligence
LLE	Local Linear Explanation
RFOut	Robust Filtering of Outliers
DDaBA	Dynamic Defence against Byzantine Attacks
FTX-DDaBA	Fair, Transparent and eXplainable DDaBA

Chapter I

PhD Dissertation

«*A woman must have money and a room of her own*
if she is to write fiction».
– Virginal Woolf.

1 Introduction

Artificial Intelligence (AI) has become the central axis of the Fourth Industrial Revolution. It is here to stay and is fundamentally transforming multiple sectors of society, from manufacturing to transportation, healthcare, energy, and education. However, alongside the advantages provided by AI, as its applications proliferate across various sectors of society, so do the risks associated with its use. The definition of what the EU AI Act [AIA21] determines as High-Risk AI Systems (HRAIs), also known as responsible AI systems, is rapidly becoming commonplace, driven by growing public concerns, and their associated risks are gaining prominence as the new regulations come into play.

For some perspective, the latest studies estimate that 328.77 million terabytes of data are created every day, including new data generation, captures, copies, or consumed information. If we convert that data to 4K (Ultra HD) quality video, we would have enough content to watch for approximately 37,496 years without interruption. Of all this information, more than half (approximately 53.72%) is video, mostly generated and shared on social networks. It is estimated that 90% of all the information generated in the world was generated in the last two years, and this trend is expected to continue to grow exponentially in the coming years.

Although the potential of AI in various societal fields is still being explored, there is a growing concern about the negative impact that the use of unsupervised AI systems can have in more critical areas such as science, education, medicine, justice, culture, or democracy. Given the underlying risks in these disciplines, it is imperative to develop responsible AI systems that are reliable, explainable, and secure, and that respect human rights and dignity.

At this juncture, one of the paramount challenges we confront is striking the right balance between the benefits and the risks of AI in scenarios where HRAIs can be utilised, ensuring that regulation does not stifle innovation. This entails ensuring that AI systems are robust, secure, transparent, fair, respectful to privacy and autonomy, have clear traceability, and are subject to fair accountability for auditing [DRDSC+23]. To achieve this, it is essential to address the dimensions of AI within responsible AI systems and work towards ensuring their responsible and ethical application. In this context, the concept of Trusworthy AI and Responsible AI systems emerge.

Trustworthy AI [TLS21] is built upon seven technical requirements that must be upheld throughout the entire lifecycle of the AI system. These requirements are underpinned by three core pillars: (1) legality, (2) ethics, and (3) robustness, from both technical and societal perspectives. However, achieving truly trusted AI pertains to a broader and multidisciplinary vision that encompasses the trustworthiness of all processes and actors involved in the system's lifecycle, and considers the aforementioned aspects from various angles. The seven requirements encompass: human agency and oversight; robustness and safety; privacy and data governance; transparency; diversity, non-discrimination and fairness; societal and environmental wellbeing; and accountability.

Figure 1 [DRDSC+23] represents the above-mentioned pillars and requirements for trustworthy AI. We highlight in colour those requirements that we will address during the thesis: privacy and data governance, transparency and diversity, non-discrimination and fairness.

We address the main objective by focusing on the robustness pillar from a technical point of view. In particular, the main focus of this thesis is data privacy.

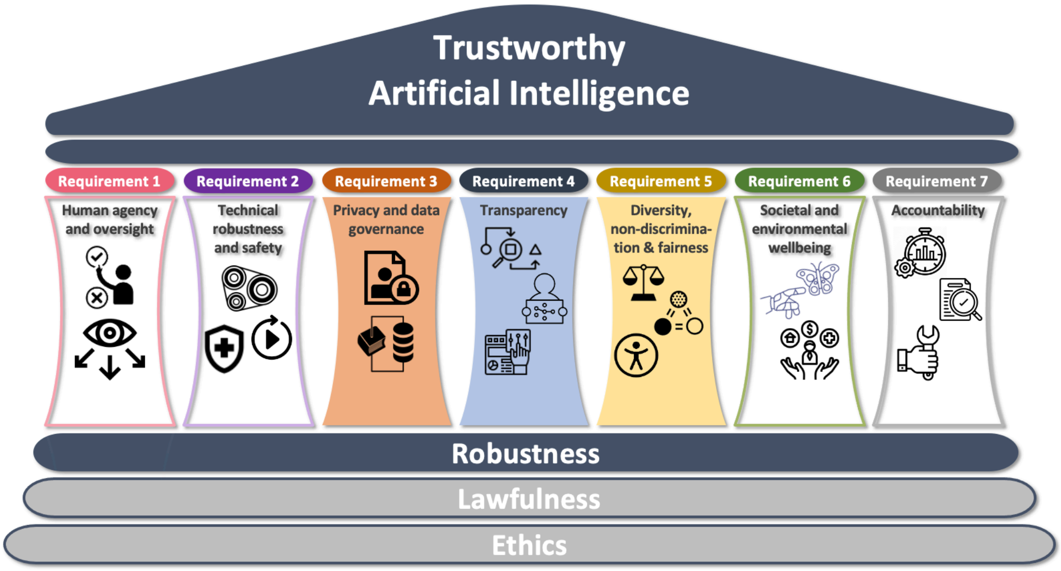


Figure 1: Representation of the requirements and pillars of trustworthy AI. We highlight in colours the requirements addressed in this thesis. Figure obtained from [DRDSC+23].

In this context, in 2016 and spearheaded by Google, the concept of Federated Learning (FL) emerged [MMR+17]. It is introduced as a novel paradigm of distributed learning [VWK+20] that promises to address the issue of user data privacy while retaining the advantages provided by machine learning-based solutions that rely on data for their design. The primary distinction from classical distributed learning is that the data stored on each node never leaves the device nor is accessible, thus ensuring its privacy. Therefore, it involves training local learning models on each device that holds training data and subsequently sharing the knowledge acquired by these models. This model information is then aggregated, summarising the global knowledge of all participants. It is noteworthy that it is the model information that is shared, not the training data, as this is where its greatest potential lies—in data privacy.

Although this new paradigm is applicable in many contexts, there are certain scenarios where its use becomes imperative:

- When the data contains personal user information, such as emails, user recommendations [JSCS19], medical records and human activity recognition using smartphones [CLC+22].
- When information is stored in data silos [BCM+18]. For example, the healthcare sector

often hesitates to disclose its records, keeping its data inaccessible.

- When information is protected by data protection laws, as is the case with banks [KPN+19] or telecommunications agencies [TBZ+19].

In all of these cases, while the use of all that information could be highly beneficial, it is not possible. For instance, medical records cannot be shared for security reasons. However, if a Europe-wide model for early detection of diabetes wished to be developed, this model would strongly profit from being able to use information from all countries, otherwise it would be difficult to adapt it to the particularities of each region. This could be resolved by implementing a federated scheme in which each hospital represents a learning node, trains its model on its own data, and subsequently shares it to be aggregated with the models from other hospitals. This way, we obtain a global model that summarises the information from all participating hospitals, without compromising the data privacy of any patient.

Although the primary goal of FL is data privacy, its design also provides several other advantages, including:

- **Reduction in Communication Costs.** In a scenario where we handle large amounts of data, which is increasingly common today, storing all that data on a single server is very expensive. Moreover, in cases where data comes from multiple devices, such as Internet of Things (IoT) devices [NDP+21], communication costs are very high. In FL, since data storage remains on the devices, all these communications are reduced [MMR+17]. Only model updates are shared, which are notably less expensive to transmit.

- **Robustness.** Convergence in FL is achieved after multiple rounds of learning, which consist of: (1) local training of each model on each client, (2) model aggregation, and (3) allocation of this aggregation. Thus, after each learning round, the partial solutions found by each client are gradually abandoned to converge towards a common solution. This training of different models for subsequent combination produces more robust results to small variations in data distribution [SWMS19], as it has learned to model the existing peculiarities.

Due to all these qualities, FL has already been employed in various real-world applications [LFTL20]. Most notably, in environments where data privacy is essential or where data is generated on multiple devices. For instance, its use in cybersecurity [ARP+21], medical applications [RHL+20], and mobile applications [LLH+20] stands out.

FL, like any machine learning model, is susceptible to attacks. Various types of adversarial attacks are known, targeting both data privacy and model functionality [RBJLL+23, LXW22]. However, these adversarial attacks take on particular significance in FL because most of the designed defence mechanisms cannot be applied. Given that the majority of attacks involve poisoning data to alter the functioning of the learning model trained on them, most defences rely on data inspection techniques. Clearly, these techniques are not applicable in FL since the data is not accessible. This necessitates the development of ad-hoc defence mechanisms or the adaptation of existing ones for application in FL.

There exists a broad typology of adversarial attacks, among which we highlight the following categorisation:

- **Privacy Attacks,** whose primary objective is to infer sensitive information about the data or the learning process, jeopardising data integrity [MPP+21].
- **Model Attacks,** which aim to alter the integrity of the global model [BVH+20].

Although both represent significant threats, as they directly impact two of the main qualities of FL, which are privacy and model performance, throughout this thesis, we focus on the second type. Within this second type, data or model poisoning attacks [BCMC19] stand out, where poisoned data is generated, resulting in poisoned models after training on them, or directly poisoned models. The objective of this poisoning can be: (1) to impair the model's performance, producing, for example, random results [FCJG20], or (2) to introduce some secondary task during training [BVH+20], allowing the model to maintain its performance on the original task while also learning another task in parallel. Throughout this thesis, we will cover both introduced attack typologies."

Once the risks faced by FL have been highlighted, the need to investigate defence mechanisms to counter such adversarial attacks becomes evident. The development of effective defence mechanisms is of great utility for both the scientific community and the business world, where FL-based solutions susceptible to these threats are already being developed.

Therefore, the primary objective of this thesis will be the study and analysis of both attacks and defences in FL, aiming to identify and categorise various threats and existing solutions. Additionally, we will focus on the development of defence mechanisms against model attacks. While there is a wide variety of defence mechanisms available, we will concentrate on the approach of designing robust aggregation operators that eliminate the influence of attacks on the global model. Furthermore, to bring us closer to reliable AI, we will ensure that these defence mechanisms promote other desirable qualities as well. For instance, most existing defence mechanisms are overly focused on maximising model performance and minimising attack success, which can potentially harm poor clients (those with skewed data distributions) or provide solutions that lack transparency and explainability.

In particular, this thesis will centre its focus on developing defence mechanisms that maintain their effectiveness while being equitable and fair to all clients, simultaneously remaining transparent and explainable. This approach brings us closer to the goal of developing a trustworthy AI.

Finally, to conclude this introduction, we present an overview of the structure of this thesis, composed of two parts: the doctoral dissertation, in Chapter I, the publications that support the knowledge and conclusions presented in it, in Chapter II, and finally in Chapter III we detail the current work. The dissertation is divided into 8 sections. Section 2 delves into the technical background of the concepts and terminology used in the subsequent sections. The justification, objectives, and methodology that form the basis of this thesis are outlined in sections 3, 4, and5, respectively. Subsequently, in Section 6, a summary of the research conducted is presented. Finally, in Section 8, the conclusions drawn from the research are discussed along with future research directions.

The second part (Chapter II) compiles the publications that support the knowledge and conclusions discussed in the dissertation. The three publications, published in international indexed journals, are as follows:

- Survey on federated learning threats: Concepts, taxonomy on attacks and defences, experimental study and challenges.
- Backdoor attacks-resilient aggregation based on Robust Filtering of Outliers in federated learning for image classification.
- Dynamic defence against byzantine poisoning attacks in federated learning.

The third part (Chapter III) develops the current work. In particular, it is an improvement of one of the previously proposed defence mechanisms that covers the objective of fairness and equity for poor clients with skewed data distributions, and provides an explainable insight into the adversarial client filtering mechanism.

Introducción

La Inteligencia Artificial (*Artificial Intelligence* - AI) se ha convertido en el eje central de la Cuarta Revolución Industrial. Ha llegado para quedarse, y está cambiando de raíz múltiples sectores de la sociedad, desde manufacturación hasta transporte, salud, energía y educación. Sin embargo, unidas a las ventajas proporcionadas por la AI, a medida que crecen las aplicaciones en diferentes sectores de la sociedad, crecen los riesgos derivados del uso de la misma. La definición de lo que el *EU AI Act* [AIA21] determina como sistema de AI de alto riesgo (HRAIs por sus siglas en inglés, *high-risk AI systems*), también conocidos como sistemas de AI responsable, se está generalizando rápidamente motivado por la creciente preocupación de la población, y sus riesgos asociados están cobrando relevancia a medida que entra en juego la nueva normativa.

Para situarnos un poco en perspectiva, los últimos estudios estiman que se crean una cantidad de 328,77 millones de *terabytes* de datos al día, incluyendo la generación de datos nuevos, capturas, copias, o información consumidos. Si convirtiéramos esos datos a vídeo en calidad 4K (Ultra HD), tendríamos suficiente contenido para ver durante aproximadamente 37,496 años sin interrupciones. De toda esta información, más de la mitad (aproximadamente un 53,72 %) se corresponde con vídeos, mayormente generados y compartidos en redes sociales. Se estima que el 90 % de toda la información generada en el mundo fue generada en los últimos dos años, y se espera que esta tendencia siga creciendo de forma exponencial en los próximos años.

Aunque todavía se está explorando el potencial de la AI en diversos campos de la sociedad, existe una preocupación cada vez mayor sobre el impacto negativo que puede tener el uso de sistemas de AI sin supervisión en áreas más comprometidas como ciencia, educación, medicina, justicia, cultura o democracia. Dados los riesgos subyacentes en estas disciplinas, es imprescindible desarrollar sistemas de AI responsables que sean fiables, explicables y seguros, y que respeten los derechos humanos y la dignidad.

En este momento, uno de los mayores retos a los que nos enfrentamos es encontrar el balance entre los beneficios y el riesgo de la AI en los escenarios en los que se pueden utilizar los HRAIs, de forma que la regulación no acabe con la innovación. Esto implica garantizar que los sistemas de AI sean robustos, seguros, transparentes, justos, respetuosos con la privacidad y la autonomía, que tengan una trazabilidad clara y que estén sujetos a una rendición de cuentas justa para su auditoría [DRDSC+23]. Para ello, es esencial abordar las dimensiones de la AI dentro de los sistemas de AI responsable y trabajar para garantizar su aplicación responsable y ética. En este contexto surge el concepto de AI confiable.

La AI confiable [TLS21] se basa en siete requisitos técnicos que deben cumplirse a lo largo de todo el ciclo de vida del sistema de AI. Estos requisitos se sustentan en tres pilares principales: (1) legalidad, (2) ética y (3) robustez, tanto desde una perspectiva técnica como social. Sin embargo, alcanzar una AI verdaderamente confiable concierne a una visión más amplia y multidisciplinar que comprende la confiabilidad de todos los procesos y actores que forman parte del ciclo de vida del sistema,y considera los aspectos anteriores desde diferentes ópticas. Los siete requisitos son: supervisión e intervención humana; robustez y seguridad; privacidad y gobernanza de datos; transparencia; diversidad, no discriminación y equidad; bienestar social y medioambiental; y rendición de cuentas.

La Figura 2 [DRDSC$^+$23] representa los pilares y requisitos mencionados anteriormente para una AI confiable. Destacamos en color los requisitos que abordaremos durante la tesis: privacidad y gobierno de datos, transparencia y diversidad, no discriminación y equidad. Abordamos el problema centrándonos en el pilar de la robustez desde un punto de vista técnico. En particular, el enfoque principal de esta tesis es la privacidad de los datos.

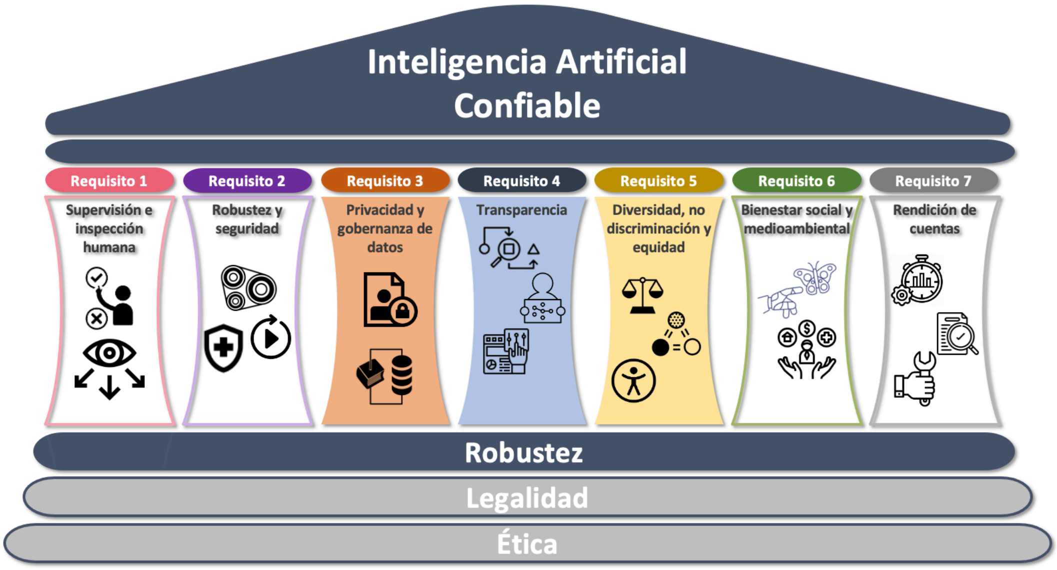

Inteligencia Artificial Confiable
Requisito 1
Supervisión e inspección humana
Requisito 2
Robustez y seguridad
Requisito 3
Privacidad y gobernanza de datos
Requisito 4
Transparencia
Requisito 5
Diversidad, no discriminación y equidad
Requisito 6
Bienestar social y medioambiental
Requisito 7
Rendición de cuentas
Robustez
Legalidad
Ética

Figura 2: Representación de los requisitos y pilares de una AI confiable. Destacamos en colores los requisitos abordados en esta tesis. Figura obtenida de [DRDSC$^+$23].

En este contexto surge, en 2016 y de la mano de Google, el concepto Aprendizaje Federado (FL) [MMR$^+$17]. Se presenta como un nuevo paradigma de aprendizaje distribuido [VWK$^+$20] que promete solucionar el problema de la privacidad de los datos de usuario, al mismo tiempo que no se renuncia a las ventajas que proporcionaban las soluciones basadas en aprendizaje automático que precisan de datos para su diseño. La principal diferencia con el aprendizaje distribuido clásico es que los datos almacenados en cada nodo jamás abandonan el dispositivo ni son accesibles, asegurando así la privacidad de los mismos. El funcionamiento consiste, pues, en entrenar modelos de aprendizaje local en cada uno de los dispositivos que poseen datos de entrenamiento, y posteriormente compartir la información aprendida por los modelos. Esta información de los modelos es posteriormente agregada, resumiendo así el conocimiento global de todos los participantes. Es importante el hecho de que se comparte la información de los modelos y no de los datos de entrenamiento, pues es ahí donde reside su máximo potencial, en la privacidad de los datos.

Aunque este nuevo paradigma tiene cabida en muchos contextos, hay algunas casuísticas donde se hace imperante su uso:

- Cuando los datos contienen información personal de usuario, como correos electrónicos, recomendaciones de usuario [JSCS19], historiales médicos o reconocimiento de la actividad humana mediante *smartphones* [CLC+22].
- Cuando la información se encuentra almacenada en silos de datos [BCM+18]. Por ejemplo, el sector sanitario suele ser reacio a divulgar sus registros, manteniendo sus datos inaccesibles.
- Cuando la información está protegida por leyes de protección de datos, como es el caso de los bancos [KPN+19] o las agencias de telecomunicaciones [TBZ+19].

En todos estos casos, aunque el uso de toda esa información pudiera ser muy beneficiosa, no es posible. Por ejemplo, por motivos de seguridad los registros médicos no se pueden compartir. Sin embargo, si quisiéramos hacer un modelo de detección temprana de la diabetes a nivel europeo, este modelo se vería muy beneficiado de poder usar los datos de todos los países, dado que si no sería difícil adaptarlo a las particularidades de cada región. Esto se solucionaría si aplicamos un esquema federado en el que cada hospital represente un nodo de aprendizaje, entrene su modelo sobre sus propios datos, y posteriormente lo comparta para ser agregado con el resto de modelos del resto de hospitales, obteniendo así un modelo global que resuma la información de todos los hospitales participantes, sin haberse visto violada la privacidad de datos de ningún paciente.

Aunque el objetivo principal del FL es la privacidad de los datos, por su diseño también proporciona otra serie de ventajas entre las que destacamos:

- **Reducción de costes de comunicación.** En un escenario en el que manejamos grandes cantidades de datos, lo cual es cada día más común, almacenar todos esos datos en un solo servidor es muy costoso. Además, en el caso en el que los datos provengan de múltiples dispositivos como es el caso de los dispositivos IoT [NDP+21], los costes de comunicación son muy altos. En FL, como el almacenamiento de los datos se mantiene en los dispositivos, todas estas comunicaciones se reducen [MMR+17]. Solo se comparten las actualizaciones de los modelos, las cuales son notablemente menos costosas de compartir.
- **Robustez.** La convergencia en FL se consigue tras múltiples rondas de aprendizaje que consisten en: (1) entrenamiento local de cada modelo en cada cliente, (2) agregación de los modelos, y (3) asignación de esta agregación. De esta forma, tras cada ronda de aprendizaje, se van abandonando las soluciones parciales encontradas por cada cliente para empezar a converger hacia una solución común. Este entrenamiento de diferentes modelos para posteriormente combinarlos produce resultados más robustos a pequeñas variaciones en la distribución de datos [SWMS19], pues ha aprendido a modelar las particularidades ya existentes.

Por todas estas cualidades, el FL ha sido utilizado ya en diversas aplicaciones en el mundo real [LFTL20]. La mayoría en entornos donde la privacidad de datos se hace esencial o donde los datos se generan en múltiples dispositivos. Destacamos, por ejemplo, el uso en ciberseguridad [ARP$^+$21], aplicaciones médicas [RHL$^+$20] o aplicaciones móviles [LLH$^+$20].

El FL, al igual que cualquier modelo de aprendizaje automático, es vulnerable a ataques. Se conocen diferentes tipos de ataques adversarios tanto a la privacidad de los datos, como al funcionamiento del modelo [RBJLL$^+$23, LXW22]. Sin embargo, estos ataques adversarios cobran especial importancia en FL debido a que la mayoría de los mecanismos de defensa diseñados no se pueden aplicar. Dado que la mayoría de los ataques consisten en un envenenamiento de los datos para así modificar el funcionamiento del modelo de aprendizaje entrenado sobre ellos, la mayoría de las defensas se basan en técnicas de inspección de datos. Claramente estas técnicas no son aplicables en FL dado que los datos no son accesibles. Esto hace que sea necesario desarrollar mecanismos de defensa ad-hoc o adaptar los ya existentes para que se puedan aplicar en FL.

Existe una amplia tipología de ataques adversarios, entre los que destacamos la siguiente categorización:

- **Ataques a la privacidad,** cuyo objetivo principal es inferir información sensible sobre los datos o el proceso de aprendizaje, poniendo en riesgo la integridad de los datos [MPP$^+$21].
- **Ataques al modelo,** cuyo objetivo se basa en modificar el rendimiento del modelo global [BVH$^+$20].

Aunque ambos representan una gran amenaza, dado que afectan directamente a dos de las principales cualidades del FL que son la privacidad y el rendimiento del modelo, a lo largo de esta tesis nos centramos en el segundo tipo. Dentro de este segundo tipo destacan los ataques por envenenamiento de datos o del modelo [BCMC19], en los que se generan datos envenenados, que se traducen en modelos envenenados tras entrenar sobre ellos, o directamente modelos envenenados. El objetivo de este envenenamiento puede ser: (1) perjudicar el rendimiento del modelo produciendo [FCJG20], por ejemplo, resultados aleatorios, o (2) introducir alguna tarea secundaria en el entrenamiento [BVH$^+$20], de forma que se mantenga el rendimiento del modelo sobre la tarea original, pero también aprenda otra tarea de forma paralela. Durante esta tesis, abarcaremos las dos tipologías de ataques introducidas.

Una vez se ha puesto de manifiesto los riegos a los que se enfrenta el FL, queda manifestada la necesidad de investigar en mecanismos de defensa que hagan frente a este tipo de ataques adversarios. El desarrollo de mecanismos de defensa efectivos es de gran utilidad tanto para la comunidad científica, como para el mundo empresarial en el que ya se desarrollan soluciones basadas en FL susceptibles a estas amenazas.

Así pues, el objetivo principal de esta tesis será el estudio y análisis de ambos ataques y defensas en FL con el fin de identificar y categorizar las diferentes amenazas y soluciones existentes, así como el desarrollo de mecanismos de defensa frente a ataques al modelo. Aunque existe gran diversidad de mecanismos de defensa, nos basaremos en el enfoque de diseñar operadores de agregación robustos que eliminen la influencia de los ataques en el modelo global. Además, para que estos mecanismos de defensa nos acerquen más a una AI confiable, procuraremos que los mecanismos de defensa fomenten de forma transversal otras de las cualidades deseables para ello. Por ejemplo, la gran mayoría de los mecanismos de defensa existentes están tan centrados en maximizar el rendimiento del modelo y en minimizar el éxito del ataque, pudiendo perjudicar a clientes pobres (aquellos que tienen una distribución de datos sesgada), o proporcionar soluciones que no sean transparentes y explicables.

En concreto, en esta tesis centraremos el foco en desarrollar mecanismos de defensa que mantengan su efectividad como tal, pero que sean equitativos y justos con todos los clientes, al mismo tiempo que transparentes y explicables, acercándonos así al objetivo de desarrollar sistemas AIs confiables.

En último lugar, para concluir esta introducción presentamos un resumen de la estructura de esta tesis, compuesta de tres partes: la disertación doctoral, en el Capítulo I, las publicaciones que avalan los conocimientos y conclusiones expuestos en la misma, en el Capítulo II, y finalmente en el Capítulo III se desarrolla el trabajo actual. La disertación se divide en 8 secciones. La sección 2 profundiza en el trasfondo técnico de los conceptos y terminología utilizados en las secciones posteriores. La justificación, los objetivos y la metodología que sientan las bases de esta tesis se indican en las secciones 3, 4 y 5, respectivamente. Posteriormente, en la Sección 6 se presenta un resumen de la investigación llevada a cabo. Finalmente, en la Sección 8 se exponen las conclusiones derivadas de la investigación junto con futuras líneas de investigación.

La segunda parte (Capítulo II) recoge las publicaciones que avalan los conocimientos y las conclusiones discutidas en la disertación. Las tres publicaciones, publicadas en revistas indexadas internacionales son las siguientes:

- Survey on federated learning threats: Concepts, taxonomy on attacks and defences, experimental study and challenges.
- Backdoor attacks-resilient aggregation based on Robust Filtering of Outliers in federated learning for image classification.
- Dynamic defence against byzantine poisoning attacks in federated learning.

En la tercera parte (Capítulo III) se desarrolla el trabajo actual. En particular, se trata de una mejora de uno de los mecanismos de defensa propuesta anteriormente que cubre el objetivo de equidad y justicia para los clientes pobres con distribuciones sesgadas de datos, y que proporciona una visión explicable del mecanismo de filtrado de clientes adversarios.

2 Preliminaries

This section introduces the technical background necessary to understand the remainder of the dissertation (Chapter I). In Section 2.1 we introduce FL, which is the core of this thesis, including the motivation of its design and the advantages that it provides. In the subsequent Sections 2.1.1, 2.1.2 and 2.1.3 we detail the architectures, categories and training in FL, respectively. In Section 2.2 we dive into the topic of study of this thesis, the adversarial attacks in FL. We provide some background on Induced Ordered Weighted Averaging (IOWA), required to fully understand one of the proposals in Section 2.3. Finally, we introduce the Local Linear Explanation (LLE)s in Section 2.4.

2.1 Background on Federated Learning

FL [RBSJL$^+$20] is a distributed machine learning paradigm with the aim of building a Machine Learning (ML) model without explicitly exchanging training data between parties [YLC$^+$19]. It consists of a network of (1) aggregation nodes $\{A_1, \dots, A_m\}$; and (2) clients or data owners $\{C_1, \dots, C_n\}$, who participate in two main processes:

1. *Model training phase:* each client exchange information without revealing any of their data to collaboratively train a ML model, G, which may reside at one node or may be shared between a few nodes. The collaborative learning of G is performed by aggregating the local models shared by the clients in the aggregation nodes.
2. *Inference phase:* clients collaboratively apply the jointly trained model, G, to a new data instance.

Both processes can be either synchronous or asynchronous, depending on the data availability of the clients and the trained model.

The fact must be highlighted, that privacy is not the only motivation of this paradigm, there should be a fair value-distribution mechanism to share the profit gained by the collaboratively trained model, $\mathcal{M}_f$. In the following we highlight the main benefits of FL:

- *Privacy*: FL provides data privacy by training each model on the clients, so the data never leaves the devices, providing the intended privacy.
- *Communication costs and latency*: in contexts where large amounts of data are available from many different sources, communication costs are reduced by sharing only the model weights, not the data, with the server.
- *Robustness*: the distribution of data among clients is often Non Independent and Identically Distributed (non-IID), so FL, by training local models and performing subsequent aggregations helps to reflect the real distribution of data. It even provides more generalization and robustness, since the global model adapts to variations in data distribution between clients.

- *Data access*: In some situations, the data are distributed among organizations and institutions, which makes access to them difficult or impossible [ABHKS17]. FL raises the possibility of being able to use these data without having to access them, solving this accessibility problem.

2.1.1 Architectures in Federated Learning

The communication of the two previous types of nodes defines two kind of federated architectures [YLC+19, RBSJL+20], namely:

1. Peer-to-peer: It is architecture in which all the nodes are both data owners and aggregation nodes. This scheme does not require any coordinator. It provides higher security and data privacy while the main disadvantage is the computation and communication costs. This FL architecture is illustrated in Figure 3.

2. Client-server: It consists of a coordinator *aggregation* node named server and a set of *data owner* nodes named clients[1]. In this architecture, the client does not share its local data ensuring its privacy. We represent the client-server scheme in Figure 4.

Although the peer-to-peer architecture is a generalization of the client-server architecture, from this point on we assume that the underlying architecture is always a client-server architecture, since it is the most widely used in the literature.

2.1.2 Categories in Federated Learning

The distribution of characteristics of the data among clients in FL shapes the procedure to follow in the two main processes of FL, particularly we focus on the following distributions: (1) clients share the feature space but not the sample space, (2) clients share the sample space but not the feature space, and (3) clients share only a small overlap in feature space. These distributions allow us to present three categories of FL [YLC+19] in terms of the feature space (X), the label space (Y) and the sample ID space (I) as follows:

Horizontal Federated Learning (HFL) In this scenario, clients data share the feature and labels space, but differ in the sample space. Formally, we can define as:

$$X_i = X_j,\ Y_i = Y_j,\ I_i \neq I_j,\ \forall D_i, D_j,\ i \neq j$$

where the feature and labels space of the clients (i, j) is depicted by (X_i, Y_i) and (X_j, Y_j) and it is assumed to be the same, while the samples I_i and I_j are not the same. D_i and D_j depict the data of the clients i and j.

[1] Data owner nodes are called by several names, depending on the source and the architecture. In the following, we refer to them as clients when it is a client-server scheme and nodes in all other cases.

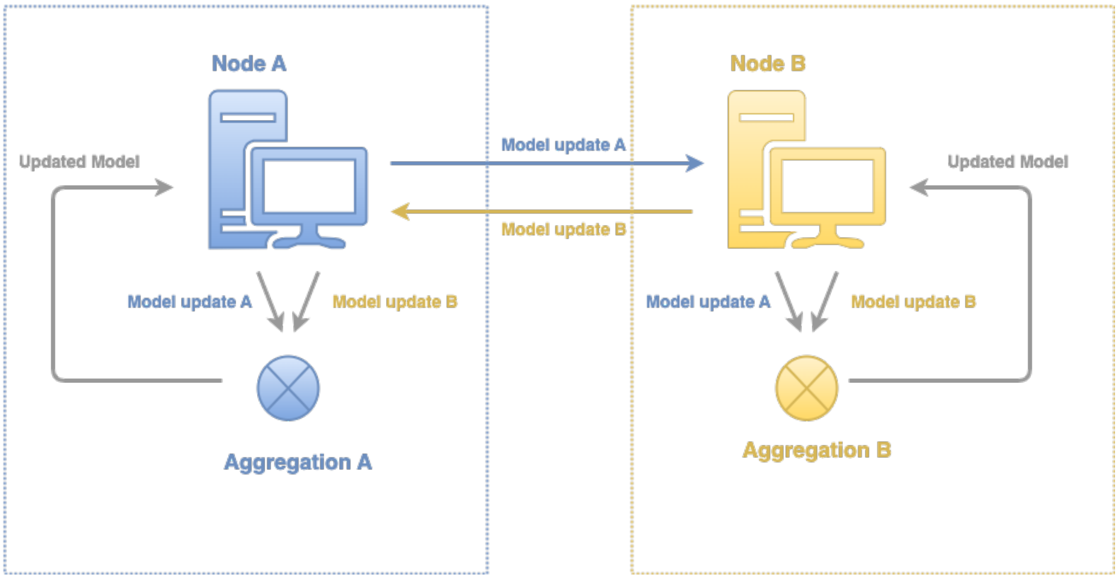


Figure 3: Representation of peer-to-peer FL architecture.

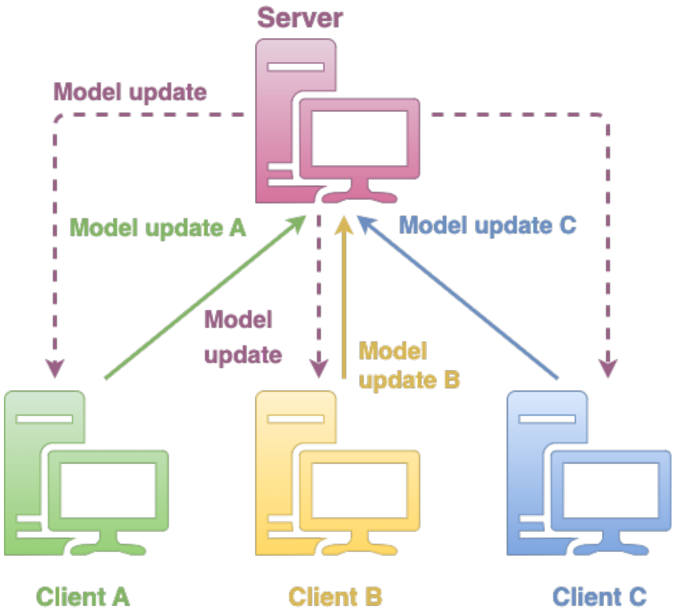
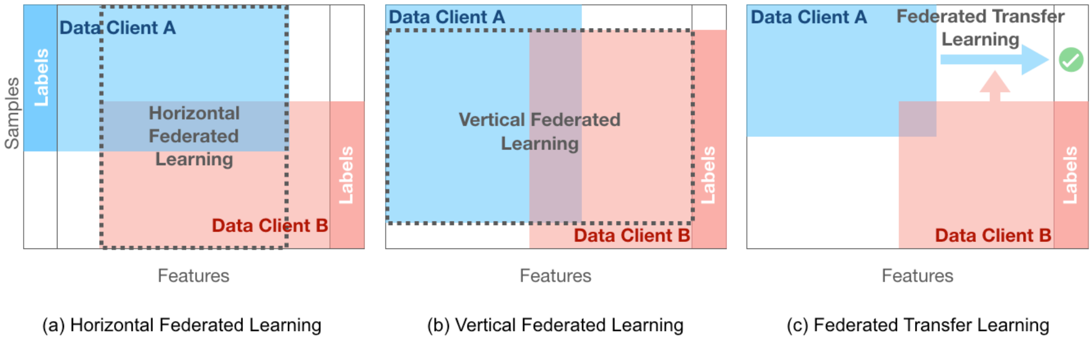


Figure 4: Representation of client-server FL architecture.

Vertical Federated Learning (VFL) In this scenario, clients share the sample space but neither the feature space nor the label space. Formally, we can define as follows:

$$X_i \neq X_j,\ Y_i \neq Y_j,\ I_i = I_j,\ \forall D_i, D_j,\ i \neq j$$



(a) Horizontal Federated Learning (b) Vertical Federated Learning (c) Federated Transfer Learning

Figure 5: Representation of the different categories in FL. Source [YLC+19].

Federated Transfer Learning (FTL) This scenario is similar to the traditional transfer learning. The clients share neither the feature space, nor label space, nor the sample space. Formally, we can define as follows:

$$X_i \neq X_j,\ Y_i \neq Y_j,\ I_i \neq I_j,\ \forall D_i, D_j,\ i \neq j$$

Although the feature space and the label space are not the same, in FTL there is a certain overlap or similarity, since the aim is to transfer knowledge from one client to another securely. FTL was presented in [YLCT19] and it represents higher difficulty than HFL and VFL, since it implies the use of techniques that preserve the data privacy. We represent the different categories of FL in Figure 5.

The vast majority of research to date is conducted in HFL. In particular, all our research is focused on this category of FL. Henceforth, when we refer to FL without specifying any category, we are referring to HFL.

2.1.3 Training in FL: round of learning and key elements

Formally, FL is a distributed ML paradigm consisting of a set of clients $\{C_1, \dots, C_n\}$ with their respective local training data $\{D_1, \dots, D_n\}$. Each of these clients C_i has a local learning model named as L_i represented by the parameters $\{L_1, \dots, L_n\}$. FL aims at learning a global learning model represented by G, using the scattered data across clients through an iterative learning process known as *round of learning*. For that purpose, in each round of learning t, each client trains its local learning model over their local training data D_i^t, which updates the local parameters L_i^t to $\hat{L}_i^t$. Subsequently, the global parameters G^t are computed aggregating the trained local parameters $\{\hat{L}_1^t, \dots, \hat{L}_n^t\}$ using an specific federated aggregation operator Δ, and the local learning models are updated with the aggregated parameters:

$$\begin{aligned} G^t &= \Delta(\hat{L}_1^t, \hat{L}_2^t, \dots, \hat{L}_n^t) \\ L_i^{t+1} &\leftarrow G^t, \quad \forall i \in \{1, \dots, n\} \end{aligned} \tag{1}$$

The updates among the clients and the server are repeated as much as needed for the learning process. Thus, the final value of G will sum up the knowledge sequestered in the clients.

During this training process some elements are involved. We refer to these important elements as **key elements** and detailed them below. Figure **??** represents the moment at which each of the key elements comes into play.

Distributed training data. Training data is distributed among the clients' devices, instead of being allocated in a central server. The data distribution across clients is often non-IID, hence FL allows the learning model to be trained over a distribution of data which better reflects the real distribution, producing more robust results. Formally, we can distinguish between three types of non-IID: [CCI+22]: (1) where the feature space of the clients' data are different, but they share the same goal; (2) where the input space is analogous but there exists differences in the label space; and (3) when there are differences in both the feature and the label spaces. the data distribution is determined by the problem. Since this fact is beyond our control, and even difficult to know, there is no alternative but to adapt. For example, by using specific aggregation operators if data distribution across clients is highly skewed [ZXLJ21].

Aggregation operator Δ. Up to now, we referred to the process of aggregating local model updates on the server in a generic way using the operator Δ. The reason is that this operator can be any operator that returns as a result an "average" value of all the local model updates. When FL was first proposed, it was proposed in conjunction with the Federated Averaging (FedAvg) [MMR+17], which is basically an arithmetic mean of the local model updates. Despite the simplicity of FedAvg, it has shown good performance in the vast majority of the scenarios, so it remains the most widely used. Note that FedAvg can only be applied when the learning model can be expressed as a parameter matrix.

Due to this limitation, or to the requirements of the problem, it is possible that another type of aggregator may be required. Depending on the nature of this need, we can distinguish between two distinct cases:

- *Non-matrix learning models.* When the learning models can not be expressed as a parameter matrix, the aggregation process become more complex, since a simple aggregation of parameters is not possible. Ad hoc aggregation operators have been designed for each of the models to combine the information of each client, such as: clustering [SGB+20], decision trees [TBA+19] or random forest [LLL+22].

- *Other requirements.* In certain situations, it may be desirable, or even mandatory, to use a specific aggregator to achieve certain properties. For example, robustness if we are in a corrupted scenario [PKH22], or personalization of clients [TYCY22], or good performance in a very skewed non-IID environment [ZXLJ21].

Learning models. In a FL settings, there are various learning models. They are organised in two types: (1) global Learning model (LM), which is the result of aggregation of the local model updates and resumes the information from all the clients' devices, and (2) local LMs, of which each client trains its own on their own data. The local model updates of these local LMs are subsequently shared with the server contributing to distributed training.

Communication. After the training of the local LMs, the model updates produced need to be shared with the server, for subsequent aggregation. The communication which enables all this process plays a crucial role in both the coordination between clients and server, and avoiding the privacy leakage. Although the data never leaves the clients' devices, the communication channel is susceptible to be attacked by third-parties. Therefore, it is commonly used in combination with security techniques such as Differential Privacy (DP) [DMNS06, WZFY20] and Secure Multi-Party Computation (SMPC) [Gol98, LZJ$^+$20].

2.2 Background on Adversarial Attacks

FL, as any ML paradigm is vulnerable to adversarial attacks. However, this adversarial attacks are more threatening in FL since the data inspection defences are not applicable, as these data are not accessible. As a result, most of the adversarial attacks that can be carried out in FL are inherited from ML, while defences are designed ad hoc for this paradigm.

There are a wide range of adversarial attacks, which can be categorised following different criteria (see Chapter...). Perhaps the most significant categorisation can be done according to the objective of the attack. According to this categorization, we find two main groups of attacks:

- **Attacks to the model**, whose main purpose is to modify the performance of the global model.
- **Privacy attacks**, which aim at inferring sensitive information about the data or the learning process.

Throughout this thesis, we focus on the attacks to the model. To carry out these attacks, clients send poisoned samples to the server, thereby modifying the global model by participating in the aggregation. These attacks can also be categorized according to different criteria, but the most influential category is the attack objective. According to this criteria, we distinguish between:

Targeted (or backdoor) attacks. They aim at injecting a secondary task (backdoor task) into the model. The success of the attack is measured in terms of the success of the backdoor task, which means that the higher the performance of the backdoor task, the more successful the attack is. Since they do not affect the performance of the original task, they are quite stealthy, which makes them more difficult to detect. Although they do not impair the performance in the original task, they represent a high risk to the integrity of the model. To highlight the risk they pose, let's imagine that we have a face detection system to decide who has access to a private room. Such system would work like a classification in {*access*, *not access*}, which would give access only to authorized persons. If we define as a backdoor task, giving access to people wearing purple glasses (something very uncommon), this would produce a security breach difficult to detect, because for the rest of the people it would continue to work properly. In this stealth resides the strength of these attacks, and highlights the threat that they represent.

Untargeted attacks. In contrast to targeted attacks, the main objective of these attacks is to impair performance on the original task. The most extreme case, although the most common, is known as *byzantine attacks*, where the adversarial attack is based on random procedures, either learning about randomly poisoned data, or directly general random updates. These attacks are clearly easier to detect. The challenge then resides in mitigating the effects of the attack when there are multiple coordinated clients, and in distinguishing them from clients with low data variety (poor clients).

Throughout this thesis, we focus on both byzantine and backdoor attacks as both of them represent a high threat to FL.

2.3 Background on Induced-Ordered Weighted Averaging

Group decision making is the AI task focused on finding out a consensus decision from a set of experts by summing up their individual evaluations. Yager proposed in [Yag88] the Ordered Weighted Averaging (OWA) operators with the aim of modelling the fuzzy opinion majority [PY06] in group decision making. Yager and Filev generalised the OWA operator definition in [YF99], where they defined the OWA operator with an order-induced vector for ordering the argument variable. They called this generalisation of OWA operators with a specific semantic in the aggregation process as Induced Ordered Weighted Averaging (IOWA). The OWA and IOWA operators are weighted aggregation functions that are mathematically defined as what follows:

Definition 2.1 (OWA Operator [Yag88])**.** *An OWA operator of dimension n is a function $\Phi : \mathbb{R}^n \to \mathbb{R}$ that has an associated set of weights or weighting vector $W = (w_1, \dots, w_n)$ so that $w_i \in [0,1]$ and $\sum_{i=1}^{n} w_i = 1$, and it is defined to aggregate a list of real values $\{c_1, \dots, c_n\}$ according to the Equation 2:*

$$\Phi(c_1, \dots, c_n) = \sum_{i=1}^{n} w_i c_{\sigma(i)} \tag{2}$$

being $\sigma : \{1, \dots, n\} \rightarrow \{1, \dots, n\}$ *a permutation function such that* $c_{\sigma(i)} \geq c_{\sigma(i+1)}$, $\forall i = \{1, \dots, n-1\}$.

Definition 2.2 (IOWA Operator [YF99]). *An IOWA operator of dimension n is a mapping* $\Psi : (\mathbb{R} \times \mathbb{R})^n \rightarrow \mathbb{R}$ *which has an associated set of weights* $W = (w_1, \dots, w_n)$ *so that* $w_i \in [0,1]$ *and* $\sum_{i=1}^{n} w_i = 1$*, and it is defined to aggregate the second arguments of a 2-tuple list* $\{\langle u_1, c_1 \rangle, \dots, \langle u_n, c_n \rangle\}$ *according to the following expression:*

$$\Psi(\langle u_1, c_1 \rangle, \dots, \langle u_n, c_n \rangle) = \sum_{i=1}^{n} w_i c_{\sigma(i)} \tag{3}$$

being $\sigma : \{1, \dots, n\} \rightarrow \{1, \dots, n\}$ *a permutation function such that* $u_{\sigma(i)} \geq u_{\sigma(i+1)}$, $\forall i = \{1, \dots, n-1\}$. *The vector of values* $U = (u_1, \dots, u_n)$ *is called the order-inducing vector and* $(c_1, \dots, c_n)$ *the values of the argument variable.*

The OWA and IOWA operators are functions for weighting the contribution of experts for the global decision in the case of group decision making, and the contribution of a set of clients in an aggregation process in a general scenario. However, they need an additional function to calculate the values of the parameters, which in the context of group decision making means the grade of membership to a fuzzy concept. The weight value calculation function is known as linguistic quantifier [Yag96], which is defined as a function $Q : [0,1] \rightarrow [0,1]$ such as $Q(0) = 0$, $Q(1) = 1$ and $Q(x) \geq Q(y)$ for $x > y$. Equation 4 defines how the function Q computes the weight values and Equation 5 defines the behaviour of the function Q.

$$w_i^{(a,b)} = Q_{a,b}\left(\frac{i}{n}\right) - Q_{a,b}\left(\frac{i-1}{n}\right) \tag{4}$$

$$Q_{a,b}(x) = \begin{cases} 0 & 0 \leq x \leq a \\ \dfrac{x-a}{b-a} & a \leq x \leq b \\ 1 & b \leq x \leq 1 \end{cases} \tag{5}$$

where $a, b \in [0,1]$ satisfying $0 \leq a \leq b \leq 1$.

The function Q in Equation 5 can be redefined in order to model different linguistic quantifiers. Since the definition of the notion quantifier guided aggregation [Yag88, Yag96], other definitions of the function Q has been proposed to model different linguistic quantifiers like “most” or “at least” [PY06].

2.4 Background on Local Linear Explanations

As AI systems are increasingly implemented in more delicate contexts, there is a growing demand for elucidating the rationale behind the decisions made by such systems. Consequently, in recent years, a plethora of techniques falling under the category of eXplainable Artificial Intelligence (XAI) have been introduced [ADRDS⁺20]. In the scholarly literature, a conspicuous distinction is drawn between model-agnostic and model-specific explanations [SS23], contingent upon whether they necessitate knowledge about the underlying model.

This section is dedicated to a focus on model-agnostic approaches, with specific emphasis on Local Linear Explanations (LLEs), often referred to as feature importance models. These explanations are favoured due to their methodical rigour and practicality. Formally, they are defined as follows: Let X be a subset of the real vector space, denoted as $\mathcal{R}^F$, representing the input dataset. Furthermore, let $f : \mathcal{R}^F \rightarrow \mathcal{R}^C$ symbolize the original model, where C denotes the dimension of the output space, $\mathcal{Y}$. For a specific input instance $x \in X$, which requires explication, an LLE can be described as a function $g : \mathcal{R}^F \rightarrow \mathcal{R}^C$.

$$g(x) = Ax + B, \quad A \in \mathcal{M}_{F,C}, \quad B \in \mathcal{R}^C. \tag{6}$$

Which means that g is a linear application from the feature space $\mathcal{R}^F$ to the output space $\mathcal{R}^C$. Naturally, each weight of the matrices A and B has the following meaning:

- Each weight $a_{i,j}$ of the matrix A represents the importance of the feature i to the output j.
- Each weight b_j represents the general importance of the output j.

The different LLE methods use linear regression minimising error as follows:

$$\mathcal{L}(f, g, \pi_x) = \sum_{z \in N(x)} \pi_x(z)(f(z) - g(z))^2, \tag{7}$$

where $N(x)$ is the neighbourhood of x and the weighting function π_x is different for each particular method. In the following we describe two of the most popular methods.

LIME The Linear Model-agnostic Explanation (LIME) method [RSG16] follows the concept of local importance, which means that a feature is important if it produces significant changes in the neighbourhood. Formally, in order to finde the LLE g, LIME fits the Ridge regression [McD09] to $N(x)$ with the linear least squares function with the default kernel as follows:

$$\pi_x(z) = exp(-d(x, z)^2/\sigma^2) \tag{8}$$

where $d(\cdot, \cdot)$ is the euclidean distance and σ a regularisation factor. The generation of $N(x)$ is performed by sampling from an exponential distribution.

SHAP The SHapley Additive exPlanations (SHAP) method [LL17] considers a feature to be important for the classification of an example if it produces significant changes comparing to background (default) values. SHAP builds the LLE function g based on a game theory approximation. It consists on finding g as a regression with the following kernel:

$$\pi_x(z) = \frac{F-1}{\binom{F}{|z|}(F-|z|)|z|}, \tag{9}$$

where $z \in \{0,1\}^F$ is a binary vector representing the presence of each of the F features on the example z and $\binom{N}{M}$ is the combinatory number of choosing M elements from N possibilities without replacement. The neighbourhood is generated choosing the value that assign to i the proportional probability of the weight that SHAP assigns to all the instances that excludes exactly i variables.

In the realm of explanations, it is of paramount importance to establish a precise definition of what constitutes a “feature”. In the context of tabular data, features are inherently determined by the dataset configuration. Conversely, in the domain of time series, they often represent the minimum quantum of information acquired at each discrete time interval. In the domain of textual data, each discrete token, defined as the elemental unit of information, assumes the role of a feature. When dealing with imagery, individual pixels are designated as features. The principal quandary associated with this approach pertains to the staggering volume of features encountered in certain scenarios. Consequently, certain strategies, such as the generation of novel features through random amalgamation of existing ones, as exemplified by the SHAP framework [LL17], are actively considered. An additional advantage of this approach lies in its potential to impart coherence to explanations. For instance, a solitary pixel holds minimal intrinsic meaning for the human observer, but the collective significance of multiple pixels working in concert can be substantially more discernible.

Classification task specifications Based on the local linear model g defined in Equation 6, the *signed importance matrix* is defined as the derivative matrix A^l over the logit space, which verifies that $A^l = A$. To get the probability vector, the softmax function is applied $p = \text{softmax}(Ax+B)$. From that, it is defined $A^p = D(\text{softmax}(g(x)))(x)$ being $D()$ the derivative operator. From that matrices, the component $a_{i,j}$ refers to the importance of feature i for class j over the logic for the former (A^l) and over the probability spaces for the latter (A^p).

3 Justification

FL emerges as a distributed learning paradigm to solve the increasingly latent data privacy issues in some contexts. Moreover, it shows very competitive results compared to centralised settings, and even more robust. Nevertheless, as any ML paradigm, it is susceptible to adversarial attacks, which can either modify the performance of the model or lead to a privacy leak. As we stated before, these adversarial attacks are more threatening in FL, since most of the defences that have been proposed in the literature are based on data inspection techniques, which is not feasible in FL. Although a large number of defence mechanisms against adversarial attacks in FL have already been proposed, the attack-defence struggle is becoming an arms race in which every time a new defence is proposed, there is a way to find a privacy leakage. Therefore, it is essential to research on the development of defences against adversarial attacks in FL. The specific reasons that motivate this thesis are listed below.

- Firstly, FL is a promising learning paradigm with a wide range of practical applications in several fields with privacy concerns such as health, political contexts, user information based applications, video, language processing, among others. Therefore, the development of a more secure FL based on more effective defences is highly desirable, and it has enough potential to make a significant impact on many areas of research and quality of life, because of the security and privacy it can provide to the apps we use on a daily routine.

- Secondly, the current state-of-the-art on defences against adversarial attacks on FL is not enough as there are still successful attacks. Every time a defence mechanism is proposed, a new way of attack is found to be successful. This gap is an opportunity to research in more robust and resilient defences covering all known attacks.

- Thirdly, in the vast majority of proposed defences, clients with a poor or skewed distribution of data are filtered out.

- Finally, although there are several works covering adversarial attacks, defences or both. However, none of them cover all adversarial attacks (to the model and to the privacy), all defence mechanisms, and also includes a comparative experimental study. Therefore, it is essential to carry out a complete and consistent analysis of the attacks and defences in FL, as well as the proposal of a taxonomy and an experimental study that will allow us to identify the most promising lines of research in this field.

In summary, a thesis focused on adversarial attacks and defences in FL is justified due to the novelty, relevance and challenging nature of the field. Moreover, due to the novelty, there is still a wide field in which to develop quality research and innovations that has genuine relevance and impact both in the research world and in different applications.

4 Objectives

Once the main concepts of the state-of-the-art have been introduced, we elaborate on the objectives that have driven this thesis. All objectives fall within the scope of adversary attacks and defences in FL. For this reason, one of the first objectives was the creation of a comprehensive survey on adversarial attacks and defences that presents a convenient taxonomy to classify them. With the literature in place, the next objectives are aligned with proposing defence mechanisms against adversarial attacks that outperform the state of the art. These objectives can be broken down as follows:

Study of the adversarial attacks and defences en FL, resulting in a comprehensive survey and taxonomy. To fulfill this first objective, it is necessary to conduct an exhaustive analysis of all the literature on both adversarial attacks and defences in FL, focusing on similarities and differences, in order to establish a consistent taxonomy. For this study to be as consistent and useful as possible, all FL threats need to be analyzed, ranging from model attacks to data privacy attacks. With respect to defences, all categories of defences including server, client, or communication channel defences need to be explored. In addition, the study may focus on the strengths and weaknesses of each of the proposals in the literature, thus helping to discover which lines of research are most promising in the field. To this end, we aim to carry out an experimental study on the performance of attacks and the influence of defences on them.

To develop defence mechanisms against backdoor attacks. Backdoor attacks are one of the main threats of FL due to the combination of good performance and stealth presented by the attacks proposed in the literature. Although several defensive strategies are found in the literature, they are shown to be insufficient. Therefore, the aim is to design a defence mechanism that is agnostic to the number of adversarial clients, but that manages to mitigate the success of the attacks. To test the performance of the proposal, we will implement the most successful attacks on different datasets and compare it with the state of the art in defence mechanisms against backdoor attacks.

To develop defence mechanisms against byzantine attacks. Byzantine attacks also present one of the main threats of FL due to the ease with which they can be carried out. Although there are also several defensive strategies proposed, they are insufficient. As the performance of the global model is harmed by these kind of attacks, we aim to design a defence mechanism based on the performance in a validation test. The strength of this defence has to be that it adapts to the number of adversarial clients in each round, and that it is agnostic of the type of attack. Likewise, we will implement the most successful attacks on different datasets and compare it with the state of the art in defence mechanisms against byzantine attacks to test the performance of the proposal.

5 Methodology

The research conducted throughout this thesis has been carried out following the scientific method. In this particular case, it requires both practical and theoretical methodologies. The general guidelines applied in all studies included in this thesis are summarized here:

- **Observation**: through the study of FL, and focusing on adversarial attacks and defences. The goal of this stage is to identify research opportunities, which could result in new, successful defences against adversarial attacks and extend its applicability.
- **Formulation of hypotheses**: design of new defences mechanisms, stressing that they are agnostic to the number of adversarial clients and that they adapt easily to any type of attack. The defences designed and developed must fulfill the objectives described in previous sections.
- **Experimental data collection**: the designed defences are tested on diverse scenarios to obtain results as representative of their capabilities as possible. These results are later analyzed using external quality indices.
- **Contrasting the hypotheses**: the results obtained are compared with representative approaches from the existing literature, with the aim of analyzing their quality in terms of efficiency and effectiveness. To this end, a set of representative models is chosen on the basis of a comprehensive literature review. These methods are implemented and published, for the sake of reproducibility of results.
- **Validation of hypotheses**: hypotheses formulated in the experiments are proven or disproven following objective quality indicators and statistical testing. If any given hypothesis is rejected, it must be modified and the previous steps repeated from that point on.
- **Scientific thesis**: relevant conclusions are extracted in view of the outcomes of the research process. All the results and conclusions obtained must be gathered and synthesized into a documentary report of the thesis.

6 Summary

The body of knowledge compiled in this thesis is found in 3 different studies, published in scientific journals. The aim of this section is to summarize and introduce these studies, whose results will be discussed later (in Section 7). The publications are listed below:

- Rodríguez-Barroso, N., Jiménez-López, D., Luzón, M. V., Herrera, F., & Martínez-Cámara, E. (2023). Survey on federated learning threats: Concepts, taxonomy on attacks and defences, experimental study and challenges. *Information Fusion*, 90, 148-173. DOI: `https://doi.org/10.1016/j.inffus.2022.09.011`.
- Rodríguez-Barroso, N., Martínez-Cámara, E., Luzón, M. V., & Herrera, F. (2022). Backdoor attacks-resilient aggregation based on Robust Filtering of Outliers in federated learning for image classification. *Knowledge-Based Systems*, 245, 108588. DOI: `https://doi.org/10.1016/j.knosys.2022.108588`.
- Rodríguez-Barroso, N., Martínez-Cámara, E., Luzón, M. V., & Herrera, F. (2022). Dynamic defence against byzantine poisoning attacks in federated learning. *Future Generation Computer Systems*, 133, 1-9. DOI: `https://doi.org/10.1016/j.future.2022.03.003`.

The rest of the section is organized according to the publications listed above and the objectives described in Section 4. Firstly, Section 6.1 presents a survey on FL threats, including the main concepts, a taxonomy on attacks and defences, an experimental study and the future challenges of the research area. After that, in Section 6.2 we focus on backdoor attacks and propose a new defence mechanism based on the robust filtering of outliers in a problem if image classification in FL. Finally, in Section 6.3 we focus on byzantine attacks and also propose a dynamic defence mechanism based on IOWA, which is agnostic to the number of adversarial clients.

6.1 Study of the adversarial attacks and defences en FL, resulting in a comprehensive survey and taxonomy

As stated before, the research field of threats in FL, including adversarial attacks and defences, has gained considerable prominence in recent years. However, there were no surveys that contain all the required information. Some of them focus just in adversarial attacks but not in defences, or the other way around. Moreover, the vast majority of surveys about attacks in FL are focused on privacy attacks or adversarial attacks to the model, but not in both. Furthermore, it is rather difficult to find surveys that include a experimental study in such a way that the state of the art is comparatively reported. Therefore, although it is a booming area of study, the need for a survey that encompasses all adversarial attack and defence typologies in FL, together with a practical approach, was substantial.

Overall, the study provides the reader with everything needed to fully understand the FL threats. It starts with a background in which we present the required concepts to follow the rest of the work. We formally introduce FL as well as concept of FL threats and DP. Next, it presents the proposed taxonomy of adversarial attacks in FL covering both adversarial attacks to the model and privacy attacks. We go deeper into this taxonomy, to the point of presenting a categorization according to different criteria. After that, it introduces the different defence methods against adversarial attacks and proposes a complete taxonomy. This theoretical study is followed by an experimental study in which the most prominent works in each area are tested under the same experimental setup. It tests both, the effectiveness of the attacks and the success of the defence mechanisms. Based on this experimental study, the work also provides some guidelines for the application of defences against adversarial attacks which indicates what defence method performs better in each scenario. The study finishes with an exposition of the lessons learns and the conclusions obtained during its development. For the development of this study, 175 scientific articles were analysed.

The publication associated with this study is:

Rodríguez-Barroso, N., Jiménez-López, D., Luzón, M. V., Herrera, F., & Martínez-Cámara, E. (2023). Survey on federated learning threats: Concepts, taxonomy on attacks and defences, experimental study and challenges. *Information Fusion*, 90, 148-173. DOI: `https://doi.org/10.1016/j.inffus.2022.09.011`.

6.2 To develop defence mechanisms against backdoor attacks.

Backdoor attacks are one of the most harmful threats in FL due to their proper trade-off between success and stealth. As stated before, these attacks have the ability of injecting a secondary task in the FL system without modifying the performance of the system in the original task, thus remaining undetected. This problem is addressed from multiple perspectives, but every time a new defence mechanism is proposed, a leak is found through which the attack becomes effective. Additionally, numerous proposals found in the literature require too much information about the attack to be realistic. For that reason, there is a need to develop a defence which is agnostic to the number of adversarial clients and which is able to mitigate the effects of the attack.

In this work, we have developed a defence mechanism based on the robust filtering of outliers in FL. Our hypothesis is that the adversarial clients' model updates, even if they perform adequately on the original task, they are bound to be outliers in the model updates distribution because they have been trained on an additional task. Based on that, we design an 1-dimensional outliers detector which identifies the adversarial clients, which we filter out during the aggregation process. For the outlier detection we employ the Standard Deviation method, which despite the simplicity of its design, it provides outstanding results. Note that this defence mechanism is independent of the learning model, so it can be applied to different scenarios although in this work we focus on image classification. Highlight that this proposal is independent of the attack and is based just in anomalous behaviour, so it could be applied to other typologies of attack although in this work we decided to focus on backdoor attacks due to their challenging and threatening nature.

The publication associated with this study is:

> Rodríguez-Barroso, N., Martínez-Cámara, E., Luzón, M. V., & Herrera, F. (2022). Backdoor attacks-resilient aggregation based on Robust Filtering of Outliers in federated learning for image classification. *Knowledge-Based Systems*, 245, 108588. DOI: `https://doi.org/10.1016/j.knosys.2022.108588`.

6.3 To develop defence mechanisms against byzantine attacks.

Byzantine attacks, as opposed to backdoor attacks, do modify the performance of the global model on the original task, making them easier to detect. The challenge in this case is the facility to carry them out. Since they are based on random modifications of the behaviour of the local models, they are quite feasible and require less coordination to be carried out. Therefore, in these scenarios a higher percentage of clients can be adversarial than in the case of backdoor attacks, which do require much more coordination. Hence, the aim of this study is to propose a defence mechanism that manages to control the presence of Byzantine attacks while maintaining the performance of the global model in the original task. Most notably, this defence should be agnostic to the number of adversarial clients, and maintain proper performance as the number of adversarial clients increases (within realistic limits).

In this work, we have developed an agnostic and dynamic defence mechanism against byzantine attacks. To achieve the dynamic behaviour, which is the most compelling aspect of the proposal, we based the contribution of each client on IOWA. The IOWA consists of a weighted aggregation based on an ordering function. Based on the hypothesis that the performance of the adversarial clients' models should be worst than the performance of the rest of them, we use as ordering function the accuracy in a small but representative test set located on the server. After that, with the aim of establish the parameters of the IOWA operator we stand out that, based on the convergence of FL, after the appropriate rounds of learning the distribution of local model updates follows a Gaussian distribution. Based on that fact, if we consider the differences between the higher performance (in terms of accuracy) and the rest of the performances, the distribution of the differences would follow an Exponential distribution. Based on these assumptions, we establish the parameters of the IOWA operator in a way that we filter out the clients considered as outliers, and we assign to the clients in the first decile of the distribution double the weighting of the rest. Due to the generalization capacity of the premises, which depend only on the convergence capacity of the FL, this approach is agnostic of the learning model, the problem, and even the number of adversarial clients. From this agnosticism stems its capacity for adaptation and dynamism, which is its principal strength.

The publication associated with this study is:

Rodríguez-Barroso, N., Martínez-Cámara, E., Luzón, M. V., & Herrera, F. (2022). Dynamic defence against byzantine poisoning attacks in federated learning. *Future Generation Computer Systems*, 133, 1-9. DOI: `https://doi.org/10.1016/j.future.2022.03.003`.

7 Discussion of Results

With the exception of the first objective, the rest of them include experimental studies with the aim of testing the performance of the proposals. A broad and consistent experimental setup ensures that the research conclusions are reliable and produces robust conclusions which support the hypotheses. Homogeneity in the experimental setup is also a desirable property as it guarantees the comparison between different approaches. We follow these guidelines through the design of all the experimental setups for the sake of consistency and validity of the findings among the wider academic community. The different adversarial attacks and defences used as setup and baselines in the experimental studies have been chosen following the recommendations achieved in the first objective (the survey of FL threats).

This section summarizes the analysis of results obtained to fulfill the objectives of this thesis. It also includes the analyses carried out based on the experimental results. Similarly to Section 6, the rest of this section is organized according to the publications and the objectives introduced in Section 4. Section 7.1 highlights the recommendations and conclusions drawn from the study of the literature in FL threats as well as the experimental study of the most prominent approaches. Section 7.2 shows the results obtained with Robust Filtering of Outliers (RFOut). Finally, the results obtained by Dynamic Defence against Byzantine Attacks (DDaBA) are summarized in Section 7.3.

7.1 Study of the adversarial attacks and defences in FL, resulting in a comprehensive survey and taxonomy

The study related to this objective produced a review of the field of adversarial attacks and defences in FL, which are one of the most challenging threats in FL. It starts with a theoretical introduction to FL, DP and the threat model concepts. It follows with the state-of-the-art of adversarial attacks and defences and proposes a complete taxonomy for each of them. It analyses the taxonomy from different criteria and argues that there are more significant criteria, but it may depend on the requirements of the problem to solve. After that, it establishes consistent experimental setups and representative adversarial attacks and defences with the aim of comparing them under the same conditions. Based on the experimental results obtained, it poses some guidelines to lead the reader to choose the most appropriate defence according to the threats faced by his problem.

The study, taxonomy and guidelines proposed can be useful to:

- Provide a sound basis for learning about the threats of FL and the most promising research directions.
- Be able of categorize any adversarial attack or defence according to the most prominent criteria depending on the problem, or even using multiple criteria.
- Identify which defence or typology or defences are the most promising one to defend

against each attack, providing guidance to the reader on which defences to implement depending on the threats faced.

As a result of this research, the main flaws and strengths related to the threats to FL area can be identified. During the study development, 175 influential research works have been analyzed, and several others have been discarded. This extensive study has provided us with a broad overview of the most promising lines of research, as well as the most innovative technologies in each field. Regarding the lessons learned during its development, we highlight the following ones:

- *The arms race between attacks and defences*, in which each time a defence is proposed, the privacy leak is identified leading to a new and more powerful attack.
- *The trade-off in defences*, in which sometimes it is difficult to find a proper balance between preventing from attacks and not impairing the performance of the original task.
- *non-IID assumptions*, making it difficult to distinguish between adversarial clients and clients with a poor distribution of data.
- *Generalised FL*, due to the fact that the vast majority of the attacks and defences have been designed for HFL.
- *Combination with other trends*, while ensuring the data privacy should be one of the main goals in any FL scenario.

7.2 To develop defence mechanisms against backdoor attacks

Our proposal of defence mechanism against backdoor attacks is called RFOut. Although the design should be independent of the problem, we test it on image classification problems, because of its popularity. We decided to employ only datasets with a federated nature. That is, datasets in which data partitioning between clients becomes natural. We employ the following datasets:

- *Large-scale CelebFaces Attributes Dataset (CelebA)*[2], consisting of images of celebrities at different times. We match each celebrity with a client and assign the images of that celebrity as the training data of that client.
- *Federated Extended MNIST (FEMNIST)*[3], formed by digits and letters handwritten by different writers. We match each writer with a client and assign the handwritten digits and letters as the training data of that client.

[2] http://mmlab.ie.cuhk.edu.hk/projects/CelebA.html

[3] https://www.nist.gov/itl/products-and-services/emnist-dataset

This way, the non-IID character caused by the differences between clients is implicit in the data distribution.

We employ different two different types of backdoor attacks, and for each of them we test different patterns and situations. we selected the most used and the most recent defence mechanisms, thus considering 7 baselines. For evaluation, we use the standard evaluation metrics based on the performance (in terms of accuracy) on both the original and the backdoor tasks.

The results show that our proposal outperforms all the baselines in all the experimental settings. Moreover, it also outperforms the scenario without attacks in the vast majority of the situations. To highlight the value of our proposal, we proved that it can be combined with other existing techniques in literature, producing even better results in many occasions. Finally, we further analyse the behaviour of RFOut focusing on the convergence in comparison with the rest of the baselines and we conclude that it not only achieves better results, but also finds them in a more robust and faster way.

7.3 To develop defence mechanisms against byzantine attacks

Our proposal of defence mechanism against byzantine attacks is called DDaBA. As stated before, although the design of the proposal should be independent of the problem and the learning model, we use image classification tasks, because of its popularity. In this case, we employ popular image classification datasets from the literature and simulate the federated data partitioning because it give us the opportunity of better control the data distribution. We decided to employ: (1) FEMNIST, using its federated data partitioning, (2) Fashion MNIST[4], which is a more complicated version of FEMNIST, since it is an excessively simple benchmark, and (3) CIFAR-10 [5], which is a slightly more complicated data set. The data partitioning between clients in the last two datasets is performed artificially. Regarding the byzantine attacks, we employ the three most popular byzantine attacks covering both model and data poisoning. Finally, regarding the baselines, we consider the most established defences in the literature, selecting 5 of them. For the evaluation, we consider the performance of the global model (in terms of accuracy).

The results show that DDaBA outperforms all the baselines in all scenarios, also outperforming the scenario without any attack. Finally, we stand out the limitation of our proposal, which is where the number of adversarial clients is so high that the assumption of Gaussian distribution is not satisfied. We argue that this situation is very unlikely and unrealistic because it requires the coordination of more than one third of the participating clients. Nevertheless, we propose a static version of DDaBA called Static Defence against Byzantine Attacks (SDaBA) that solves this problem, showing proper results in a extreme attack scenario.

[4] `https://github.com/zalandoresearch/fashion-mnist`

[5] `https://www.cs.toronto.edu/~kriz/cifar.html`

8 Conclusions and Future Work

This section concludes the thesis (Section 8.1), gathers all the relevant studies we have published (Section 8.2), and provides notes on future research lines (Section 8.3).

8.1 Conclusions

This thesis presents an extensive study of adversarial attacks and defences in FL that provides both a comprehensive view on the work already done in the area and innovation in the form of three proposal of defence mechanisms. The overarching goal of this thesis is to broaden the current knowledge about defences against adversarial attacks in FL and to address the problem from more robust, resilient, fair, transparent and explainaible approaches. For that purpose, the most extensive survey on attacks and defences in FL in the literature has been carried out, covering all types of attacks and including an experimental study of all the defences proposed in the literature to truly explore the scope of application of these defences. Thus, we can analyse weaknesses in order to address these gaps in our defences, as well as benchmark ourselves against the state of the art in the field.

To accomplish the first objective of developing a complete survey on FL the most extensive study on FL threats has been gathered. We introduce in a comprehensive way all the concepts required to fully understand the field covering from the concept of FL to the definition of all the threats. We propose a complete taxonomy of both adversarial attacks and defences covering all the literature. In contrast to other surveys, we include both categories of attacks, privacy attacks and attacks to the model. We also conduct an extensive experimental study in which we test each category of defences in each category of attacks resulting in valuable lessons learnt about the best defensive approach in each situation. Finally, we analyse the future challenges and trends in the field, in order to motivate the research and innovation in this leading research area.

The second objective of defending against backdoor attacks, arguably one of the greatest challenges of the FL, is covered with the proposal of RFOut. RFOut is a defence mechanism based on a aggregation operator which performs a most robust aggregation of the model updates. For that, it is based on a 1-dimensional outlier detection based on the assumption that, when the local models are converging, they follow a Gaussian distribution. That way, it safeguards the FL from adversarial model updates (identified as outliers and filtered out) resulting in a FL configuration resilient to backdoor attacks. We compare our proposal the with state-of-the-art in defences mechanisms and find that it outperforms all the baselines. Hence, RFOut represents an improvement in the field and opens further possibilities for improvement along this line.

The third objective of defending against byzantine attacks, one of the most common attacks in FL, is covered with the proposal of DDaBA. The main contribution of DDaBA is its dynamic behaviour, which, as opposed to other proposals that assume to know the number of adversarial clients, is able to defend against any number of adversarial clients, and even to adapt to a changing number of adversarial clients. This dynamic aspect is achieved by implementing IOWA operators in the aggregator, so that they give different weights to the participation of each client. The operator design assigns zero weighting to clients identified as adversaries, and higher weighting to those considered high quality clients. DDaBA, in addition to representing an improvement in the area by improving existing defence mechanisms, opens up a new line of research into dynamic defence mechanisms that are agnostic to the number of adversarial clients.

To fulfil the final objective of moving towards trustworthy AI, and based on the weaknesses of DDaBA, we propose Fair, Transparent and eXplainable DDaBA (FTX-DDaBA). We claim that the aggregation operator in FL should not only maintain privacy and robustness, but should also pursue other requirements to ensure trustworthy AI. For that purpose, we shift the focus on client sorting in DDaBA from performance to LLEs, achieving improvements from different perspective: (1) a fair approach that only discards adversarial clients and not all poorly performing ones (which may include clients with poor data distributions), (2) a transparent and explainable selection of clients, being able to obtain visual explanations in the form of importance of features for client filtering, and (3) an approach that can be applied right from the start of training, without having to wait for a few warm-up rounds until the models perform adequately. We believe that with this proposal we are breaking a new ground in the field of defences against adversarial attacks, opening the focus of performance to all the requirements to ensure trustworthy AI.

Conclusiones

Esta tesis presenta un estudio exhaustivo de los ataques adversarios y las defensas en FL, que proporciona tanto una visión integral del trabajo realizado en esta área como innovación en forma de tres propuestas de mecanismos de defensa. El objetivo principal de esta tesis es ampliar el conocimiento actual sobre las defensas contra ataques adversarios en FL y abordar el problema desde enfoques más sólidos, resilientes, justos, transparentes y explicables. Con este propósito, se ha llevado a cabo el estudio más extenso sobre ataques y defensas en FL en la literatura, abarcando todos los tipos de ataques e incluyendo un estudio experimental de todas las defensas propuestas en la literatura para explorar verdaderamente el alcance de aplicación de estas defensas. De esta manera, podemos analizar las debilidades para abordar estas lagunas en nuestras defensas, así como compararnos con el estado del arte en el campo.

Para lograr el primer objetivo, se ha recopilado el estudio más extenso sobre amenazas en FL. Introducimos de manera integral todos los conceptos necesarios para comprender completamente el campo, desde el concepto de FL hasta la definición de todas las amenazas. Proponemos una taxonomía completa tanto de los ataques adversarios como de las defensas que abarca toda la literatura. A diferencia de otros estudios, incluimos ambas categorías de ataques, los ataques a la privacidad y los ataques al modelo. También llevamos a cabo un estudio experimental exhaustivo en el que ponemos a prueba cada categoría de defensas para cada categoría de ataques, del que obtenemos lecciones valiosas sobre el mejor enfoque defensivo en cada situación. Finalmente, analizamos los desafíos y tendencias futuras en el campo, con el fin de motivar la investigación y la innovación en esta destacada área de investigación.

El segundo objetivo de defenderse contra los ataques *backdoor*, posiblemente uno de los mayores desafíos del FL, se aborda con la propuesta de RFOut. RFOut es un mecanismo de defensa basado en un operador de agregación que realiza una agregación más robusta de las actualizaciones del modelo. Para ello, se basa en una detección de valores atípicos unidimensional basada en la suposición de que, cuando los modelos locales están convergiendo, siguen una distribución gaussiana. De esta manera, protege al FL de las actualizaciones de modelos adversarios (identificadas como valores atípicos y filtradas) y resulta en una configuración de FL resistente a los ataques *backdoor*. Comparamos nuestra propuesta con el estado del arte en mecanismos de defensa y encontramos que supera a todos los modelos base. Por lo tanto, RFOut representa una mejora en el campo y abre nuevas posibilidades para futuras mejoras en esta línea.

El tercer objetivo de defenderse contra los ataques bizantinos, uno de los ataques más comunes en FL, se aborda con la propuesta de DDaBA. La principal contribución de DDaBA es su comportamiento dinámico, que, a diferencia de otras propuestas que asumen conocer el número de clientes adversarios, es capaz de defenderse contra cualquier número de ellos e incluso adaptarse a un número cambiante de adversarios. Este aspecto dinámico se logra implementando operadores IOWA en el agregador, de modo que otorgan diferentes pesos a la participación de cada cliente. El diseño del operador asigna peso nulo a los clientes identificados como adversarios y un peso mayor a aquellos considerados clientes de alta calidad. Además de representar una mejora en el área al mejorar los mecanismos de defensa existentes, DDaBA abre una nueva línea de investigación en mecanismos de defensa dinámicos que son agnósticos respecto al número de clientes adversarios.

Para cumplir con el objetivo final de avanzar hacia una AI confiable, y basándonos en las debilidades de DDaBA, proponemos FTX-DDaBA. Sostenemos que el operador de agregación en el FL no solo debe mantener la privacidad y la robustez, sino que también debe cumplir con otros requisitos para garantizar una AI confiable. Con ese propósito, cambiamos el enfoque en la clasificación de clientes en DDaBA de un enfoque basado en el rendimiento a un enfoque basado en LLEs, logrando mejoras desde diferentes perspectivas: (1) Un enfoque justo que solo descarta clientes adversarios y no todos los que tienen un mal rendimiento (lo que puede incluir clientes con distribuciones de datos pobres). (2) Una selección transparente y explicable de clientes, que puede obtener explicaciones visuales en forma de importancia de características para la eliminación de clientes. (3) Un enfoque que se puede aplicar desde el principio del entrenamiento, sin tener que esperar unas pocas rondas de calentamiento iniciales hasta que los modelos funcionen adecuadamente. Defendemos que con esta propuesta estamos abriendo nuevas posibilidades en el campo de las defensas contra ataques adversarios, desplazando el enfoque del rendimiento hacia todos los requisitos para garantizar una AI confiable.

8.2 Publications

This section lists journal and preprint papers published during the PhD study period, ordered by publishing date. The DOI and the number of citations indicated by Google Scholar are given.

- **Journal papers:**

  1. Rodríguez-Barroso, N., Stipcich, G., Jiménez-López, D., Ruiz-Millán, J. A., Martínez-Cámara, E., González-Seco, G., ... & Herrera, F. (2020). Federated Learning and Differential Privacy: Software tools analysis, the Sherpa. ai FL framework and methodological guidelines for preserving data privacy. Information Fusion, 64, 270-292. DOI: `https://doi.org/10.1016/j.inffus.2020.07.009`. CITED BY: 92.
  2. Rodríguez-Barroso, N., Martínez-Cámara, E., Luzón, M. V., & Herrera, F. (2022). Backdoor attacks-resilient aggregation based on Robust Filtering of Outliers in federated learning for image classification. Knowledge-Based Systems, 245, 108588. DOI: `https://doi.org/10.1016/j.knosys.2022.108588`. CITED BY: 7.
  3. Rodríguez-Barroso, N., Martínez-Cámara, E., Luzón, M. V., & Herrera, F. (2022). Dynamic defence against byzantine poisoning attacks in federated learning. Future Generation Computer Systems, 133, 1-9. DOI: `https://doi.org/10.1016/j.future.2022.03.003`. CITED BY: 25.

8.3 Future work

The results of this PhD thesis open up new research lines and contribute to the identification of new challenges in FL. This section presents future work and promising research lines derived from the studies and conclusions gathered in this thesis:

Development of a FL platform The availability of software is paramount to data science as it directly impacts the efficiency, efficacy, and reproducibility of the research. With the rapid growth of FL, a lot of software tools haven been developed. However, none of them provide the necessary functionality to simulate all the scenarios in research [GOAZ23] because they do not provide support for all AI libraries, the freedom to implement attacks or all the FL architectures. For that reason, the research field would benefit greatly from the development of an open source and fully flexible tool.

Improvement of the proposed defence mechanisms Surely the most natural future work is to continue to develop the proposals designed as defence mechanisms against adversarial attacks. Although all the defence mechanisms exposed during this thesis have shown strong performance, an ongoing battle exists between attacks and defences. At the same time that new defences are designed against existing attacks, attacks are strengthened in ways that circumvent these defences. For this reason, it is imperative that existing defences continue to be upgraded to cover the leaks caused by new attacks.

Exploration of other defence categories Although the spectrum of existing defences in the literature is broad, during this thesis the focus is on those defences that are carried out on the server. The reasons for this include their wide applicability, as well as their ease of implementation and good performance. However, it would be beneficial to explore other categories of defences such as those carried out on the clients or in the communication channel [RBJLL$^+$23].

Exploration of defence mechanisms against privacy attacks Due to the wide and varied number of attacks against the FL, this thesis has been focused on defences against Byzantine and backdoor attacks. However, one major category of attacks remains unexplored, namely data privacy attacks. These attacks are both very common in FL, and particularly dangerous given that they directly attack privacy and data integrity, the primary motivation of FL. We plan to research and develop defence mechanisms against these attacks, as well as to study the possibility of a general defence or a combination of several defences against all the attacks in FL.

Research on developing attacks For the moment, we have remained on the ”good side” of science, researching and developing defence mechanisms. However, the study and development of attacks that bypass existing defences, although it may sound counterintuitive, can be of great benefit to the field. The more sophisticated the known attacks are, the more robust and effective the defences developed against new attacks will be.

Moving further towards trustworthy AI The backbone of this thesis is one of the fundamental requirements for reliable AI: privacy. However, the latest proposal already paves the way for a combination of several of these requirements. We firmly believe that with new social and political concerns, developments in different areas of AI must be compatible with the requirements of trustworthy AI. For that purpose, it is proposed to explore defence mechanisms that promote, at the same time as privacy, performance and robustness, the rest of the requirements.

Chapter II

Publications

«You may encounter many defeats, but you must not be defeated.
In fact, it may be necessary to encounter the defeats,
so you can know who you are, what you can rise from, how you can still come out of it».
– Maya Angelou.

1 Survey on federated learning threats: Concepts, taxonomy on attacks and defences, experimental study and challenges

Volume 92, April 2023

ISSN 1566-2535

ELSEVIER

An International Journal on

INFORMATION FUSION

Multi-Sensor, Multi-Source, Multi-Process Information Fusion

Editor-in-Chief
Salvador S. Garçia-Lopez

Available online at www.sciencedirect.com
ScienceDirect

Now indexed by Thomson Scientific

- **Journal:** `Information Fusion`
- **JCR Impact Factor:** `17.564`
- **Rank:** `5/376`
- **Quartile:** `Q1`
- **Category:** `Computer Science, Information Systems`
- **Status:** `Published`

Ref.: Rodríguez-Barroso, N., Jiménez-López, D., Luzón, M. V., Herrera, F., & Martínez-Cámara, E. (2023). Survey on federated learning threats: Concepts, taxonomy on attacks and defences, experimental study and challenges. *Information Fusion*, 90, 148-173. DOI: `https://doi.org/10.1016/j.inffus.2022.09.011`.

SURVEY ON FEDERATED LEARNING THREATS: CONCEPTS, TAXONOMY ON ATTACKS AND DEFENCES, EXPERIMENTAL STUDY AND CHALLENGES

Nuria Rodríguez-Barroso [*,a] **Daniel Jiménez López** [a] **M. Victoria Luzón** [b]

Francisco Herrera [a,c] **Eugenio Martínez-Cámara** [a]

[a] *Department of Computer Science and Artificial Intelligence, Andalusian Research Institute in Data Science and Computational Intelligence (DaSCI), University of Granada, Spain*
[b] *Department of Software Engineering, Andalusian Research Institute in Data Science and Computational Intelligence (DaSCI), University of Granada, Spain*
[c] *Faculty of Computing and Information Technology, King Abdulaziz University Jeddah, Saudi Arabia*

ABSTRACT

Federated learning is a machine learning paradigm that emerges as a solution to the privacy-preservation demands in artificial intelligence. As machine learning, federated learning is threatened by adversarial attacks against the integrity of the learning model and the privacy of data via a distributed approach to tackle local and global learning. This weak point is exacerbated by the inaccessibility of data in federated learning, which makes the protection against adversarial attacks harder and evidences the need to furtherance the research on defence methods to make federated learning a real solution for safeguarding data privacy. In this paper, we present an extensive review of the threats of federated learning, as well as as their corresponding countermeasures, attacks versus defences. This survey provides a taxonomy of adversarial attacks and a taxonomy of defence methods that depict a general picture of this vulnerability of federated learning and how to overcome it. Likewise, we expound guidelines for selecting the most adequate defence method according to the category of the adversarial attack. Besides, we carry out an extensive experimental study from which we draw

* Corresponding Author

Email addresses: rbnuria@ugr.es (Nuria Rodríguez-Barroso), dajilo@ugr.es (Daniel Jiménez López), luzon@ugr.es (M. Victoria Luzón), herrera@decsai.ugr.es (Francisco Herrera), emcamara@decsai.ugr.es (Eugenio Martínez-Cámara)

further conclusions about the behaviour of attacks and defences and the guidelines for selecting the most adequate defence method according to the category of the adversarial attack. Finally, we present our learned lessons and challenges.

Federated learning, adversarial attacks, privacy attacks, defences

***K*eywords** Federated Learning · Adversarial Attacks · Privacy Attacks · Defences.

1 Introduction

Data-driven machine learning methods currently dominate artificial intelligence. This reliance on data allows us to stand out three artificial intelligence challenges. The first is the preservation of data privacy, since artificial intelligence methods process personal and sensitive data, such as health [1, 2] and financial data [3]. Likewise, the growing interest in data privacy safeguarding is reflected in emerging legal frames such as the General Data Protection Regulation (GDPR) [4]. The second challenge is related to the increasing availability of data, which, on the one hand, is furthering the progress of artificial intelligence [5], and, on the other hand, presents new challenges related to its storage and processing that are further exacerbated when the data in question stems from distributed sources, as in IoT scenarios [6, 7]. The third challenge emerges from the need to distributively process data when it is not possible to transfer it to a central server, because of legal or regulatory restrictions, communication costs or other kind of technical limitations. Due to this distributed scenario, new difficulties appear linked to dissimilar data distributions from the same domain and the likely large size of data sources [8].

Federated learning (FL) is a machine learning paradigm proposed as a possible response to the three previous challenges, and especially to the demand of preserving data privacy, together with a distributed approach to tackle local and global learning [9]. FL aims at generating a collaboratively trained global learning model without sharing the data owned by the distributed data sources. Frequently, it requires a coordinator agent, which is in charge of managing the information exchange required to train the global learning model. In this way, the data is protected from unauthorised access, either by other data sources or the coordinator party.

Machine learning is vulnerable to adversarial attacks mainly focused on impairing the learning model or violating data privacy [10, 11]. Likewise, FL is exposed to the same jeopardy, since it is an specific machine learning setting. Some of those attacks are grounded in the malicious manipulation of the training data [12], which are inaccessible in FL and thus cannot rely on the use of data inspection techniques for detecting altered data. Therefore, one of the weak points of FL is being exposed to adversarial attacks that may violate the integrity of the learning model or the privacy of data.

The evidence that adversarial attacks are a weak point of FL is built upon the large volume of publications centred on the identification of vulnerabilities in the form of adversarial attacks [13, 14, 15, 16], and on the corresponding large volume of defence proposals against to those attacks [17, 18, 19, 20]. This effervescent quantity has motivated several survey works on adversarial attacks that attain to review and summarise the latest papers related to this weak point. These surveys' lack of a holistic view of FL and the review of the defences against adversarial attacks, because of the following reasons: (1) most of them are only focused on one kind of adversarial attacks, namely attacks to the federated model [21, 22, 23] or privacy attacks [24, 25, 26], but both encompass both attacks; (2) the vast majority does not include any experimental study [27, 28, 29, 30, 31, 32], so it is not possible to compare the strength of the attacks and the robustness of the defences in a common evaluation framework; and (3) by default they only focus on horizontal FL ignoring vertical and federated transfer learning.

Due to the mentioned facts, we propose a new survey on FL threats, and additionally we provide several taxonomies on adversarial attacks and defences, an experimental study and a final discussion about lessons learned and challenges. This survey differs from previous ones due to the following contributions:

1. Provide a general picture of the field of adversarial attacks and defences by considering the threats to the learning model and to the integrity of the privacy of data.
2. Review the threats and the defences of horizontal FL, vertical FL and federated transfer learning.
3. Define taxonomies of adversarial attacks and their corresponding defensive countermeasures. These two taxonomies encompass the different categories of adversarial attacks and defences, which will shed light in this crucial field of making FL a robust learning paradigm.
4. Provide guidelines for selecting the right defence category according to the threatening adversarial attack.
5. Compare in a common evaluation framework the strength of the most relevant adversarial attacks, and the defence capacity of the most prominent defence methods.
6. Expound some learning lessons stemmed from the literature review and the experimental study conducted.
7. Also expound their relations to the challenges in the field of adversarial attacks.

The rest of the paper is organized as follows: the following section introduces the propaedeutic concepts necessary for this survey to be illustrative. Section 3 presents the taxonomy of adversarial attacks in FL, while Section 4 expounds the taxonomy of defences against them. We conduct the experimental study in Section 5. In Section 6 we provide the guidelines for selecting the right defence category. Finally, we discuss the lessons learned and challenges in Section 7 and 8, and include some conclusions in Section 9.

2 Background concepts on Federated Learning threats

The concepts described throughout this paper require the knowledge of some propaedeutic concepts related to FL and its threats. Accordingly, we introduce FL and the categories of FL in Section 2.1, we formally define differential privacy (DP) in Section 2.2, since a considerable amount of defence methods are based on DP, and we detail the categorization of the attacks in terms of the threat model in Section 2.3.

2.1 Federated Learning

FL is a distributed machine learning paradigm with the aim of building a ML model without explicitly exchanging training data between parties [9]. It consists in a network of clients or data owners $\{C_1, \dots, C_n\}$, who participate in two main processes:

1. *Model training phase:* each client exchange information without revealing any of their data to collaboratively train a ML model, $\mathcal{M}_f$, which may reside at one client or may be shared between a few clients.
2. *Inference phase:* clients collaboratively apply the jointly trained model, $\mathcal{M}_f$, to a new data instance.

Both processes can be either synchronous or asynchronous, depending on the data availability of the clients and the trained model.

The fact must be highlighted, that privacy is not the only motivation of this paradigm, there should be a fair value-distribution mechanism to share the profit gained by the collaboratively trained model, $\mathcal{M}_f$.

The distribution of characteristics of the data among clients in FL shapes the procedure to follow in the two main processes of FL, particularly we focus on the following distributions: (1) clients share the feature space but not the sample space, (2) clients share the sample space but not the feature space, and (3) clients share only a small overlap in feature space. These distributions allow us to present three categories of FL [9] in terms of the feature space (X), the label space (Y) and the sample ID space (I) as follows:

Horizontal Federated Learning (HFL) In this scenario, clients data share the feature and labels space, but differ in the sample space. Formally, we can define as:

$$X_i = X_j,\ Y_i = Y_j,\ I_i \neq I_j,\ \forall D_i, D_j,\ i \neq j$$

where the feature and labels space of the clients (i, j) is depicted by (X_i, Y_i) and (X_j, Y_j) and it is assumed to be the same, while the samples I_i and I_j are not the same. D_i and D_j depict the data of the clients i and j.

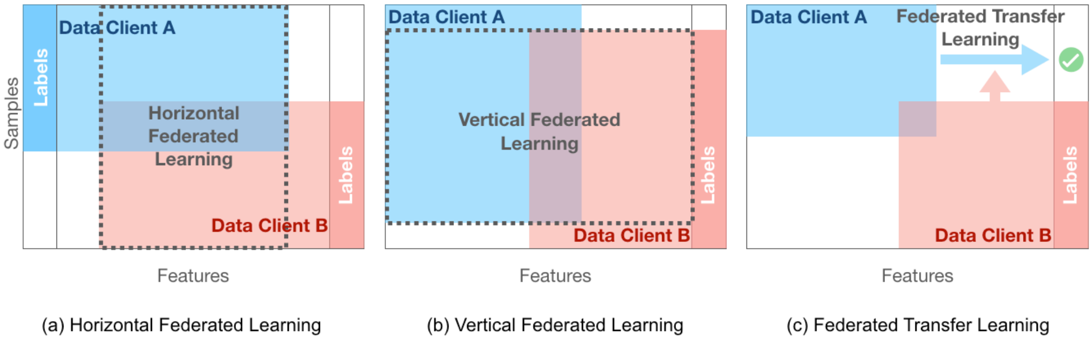
Vertical Federated Learning (VFL) In this scenario, clients share the sample space but neither the feature space nor the label space. Formally, we can define as follows:

$$X_i \neq X_j,\ Y_i \neq Y_j,\ I_i = I_j,\ \forall D_i, D_j,\ i \neq j$$

Federated Transfer Learning (FTL) This scenario is similar to the traditional transfer learning. The clients share neither the feature space, nor label space, nor the sample space. Formally, we can define as follows:

$$X_i \neq X_j,\ Y_i \neq Y_j,\ I_i \neq I_j,\ \forall D_i, D_j,\ i \neq j$$

Although the feature spaceand the label space are not the same, in FTL there is a certain overlap or similarity, since the aim is to transfer knowledge from one client to another securely. FTL was presented in [33] and it represents higher difficulty than HFL and VFL, since it implies the use of techniques that preserve the data privacy. We represent the different categories of FL in Figure 1.



(a) Horizontal Federated Learning (b) Vertical Federated Learning (c) Federated Transfer Learning

Figure 1: Representation of the different categories in FL. Source [9].

FL is a learning setting composed of a set of key elements. Since FL is a specific configuration of a machine learning environment, it shares with machine learning some of those key elements, such as the data and the learning model. Nonetheless, the particularities of FL make additional key elements necessary, such as clients and a learning coordinator that orchestrates the two main processes of FL. A detailed description of FL key elements focused on HFL is in [34], and here we describe the common ones to all the FL categories.

Data It plays a central role in machine learning. In FL, data is distributed among the different clients according to two possibilities: (1) IID (Independent and Identically Distributed), when the data in each client is independent and identically distributed, as well as representative of the population data distribution; and (2) Non-IID (non Independent and Identically Distributed), when the data distribution in each client is not independent nor identically distributed from the population data distribution. These data distributions are mainly relevant to HFL. In VFL and FTL categories, clients do not share neither the feature space nor the label space, and consequently the data distribution among clients is relegated to a second place.

In most HFL scenarios, each client only stores the data generated on the client itself, ensuring the non-IID property of the global data. Moreover, even if the IID scenario were present, it would not be known because of the data privacy properties of FL. Hence, the non-IID scenario is the best choice and it represents a real challenge.

Clients Each client of a federated scenario plays a key role in a federated paradigm, as a data owner and as a part of the distributed scheme. Typical clients in FL could be servers, smartphones, IoT devices, connected vehicles, hospitals, banks or insurance companies. Privacy is not their only motivation, they also want to profit from the *model training phase*. As a consequence, a reward mechanism is expected, such as owning the collaboratively trained model, $\mathcal{M}_f$, in HFL or the outputs of the *inference phase* in VFL and FTL.

Learning coordinator The learning coordinator orchestrates the communication among the clients in the two main processes of FL. While it is not strictly necessary, when present, it also plays the role of a trusted authority. In VFL, the learning coordinator receives and combines partial updates from clients and shares the corresponding part of the combined update with each client in the *model training phase*. Moreover, in the *inference phase* it helps to perform the inference by combining the outputs of each client as the collaboratively trained model, $\mathcal{M}_f$, is split among them. In contrast to VFL, in HFL the learning coordinator is usually known as the federated server and it only participates in the *model training phase*: (1) receiving the trained parameters of the local models, (2) aggregating the trained parameters of each client model using federated aggregation operators and (3) updating every learning model with the aggregated parameters.. Moreover, the *inference phase* is not performed jointly as the collaboratively trained model, $\mathcal{M}_f$ is stored in each client and in the federated server.

2.2 Differential Privacy

DP allows retrieving information, rigorously bounding the harm caused to individuals whose sensitive data are stored in the database [35, 36]. Basically, it hides the presence of an individual in the database. To achieve this, DP adds random noise to the outputs. Such noise is calibrated to the magnitude of the largest contribution that can be made to the output by an individual. It is important to note that DP assumes that the adversary owns arbitrary external knowledge.

DP is the key property used to provide a certain level of privacy to any sensitive data access, in a way it is both, secure and measurable. It is secure because it has a theoretical background which supports it. It is measurable as every access to private data has a privacy cost either in terms of ϵ or in terms of (ϵ, δ).

This interpretation naturally leads to define the *distance between databases*: two databases x, y are said to be n-neighbouring if they differ by n entries. In particular, if the databases only differ in a single data element ($n = 1$), the databases are simply addressed as *neighbouring*.

Differential Privacy definition A database access mechanism, $\mathcal{M}$, preserves ϵ-DP if for all neighbouring databases x, y and each possible output of $\mathcal{M}$, represented by $\mathcal{S}$, it holds that:

$$P[\mathcal{M}(x) \in \mathcal{S}] \leq e^{\epsilon} \, P[\mathcal{M}(y) \in \mathcal{S}] \tag{1}$$

If, on the other hand, for $0 < \delta < 1$ it holds that:

$$P[\mathcal{M}(x) \in \mathcal{S}] \leq e^{\epsilon} P[\mathcal{M}(y) \in \mathcal{S}] + \delta \qquad (2)$$

then the mechanism possesses the property of (ϵ, δ)-DP, also known as approximate DP.

In other words, DP specifies a ”privacy budget” given by ϵ and δ. The way in which it is spent is given by the concept of privacy loss. The privacy loss allows us to reinterpret both, ϵ and δ in a more intuitive way:

- ϵ limits the quantity of privacy loss permitted, that is, our privacy budget.
- δ is the probability of exceeding the privacy budget given by ϵ, so that we can ensure that with probability $1 - \delta$, the privacy loss will not be greater than ϵ.

DP has some interesting properties, which makes it even more appealing in a privacy context.

1. **DP is immune to post-processing**. if an algorithm protects an individual’s privacy, then there is not any way in which privacy loss can be increased.

2. **DP can be used to protect the privacy of groups**. Let $\mathcal{M}$ be a ϵ-differentially private mechanism, then $\mathcal{M}$ is $K\epsilon$-differentially private for groups of size K.

3. **DP mechanisms can be composed multiple times and remain differentially private**. Let $\mathcal{M}_1$ and $\mathcal{M}_2$ be ϵ_1-differentially private mechanism and ϵ_2-differentially private mechanism, respectively. Then, their composition output given by the concatenation of the output of $\mathcal{M}_1$ and $\mathcal{M}_2$ over the same input is $\epsilon_1 + \epsilon_2$-differentially private

2.3 Threat Model

Threat models in machine learning are structured representation of information, which help to identify and define potential security issues. They can be defined in terms of the information available and the scope of action of the attacker. In this regard, we define the following set of mutually exclusive terms that allow us to define the FL threat model.

Insider vs. Outsider One of the key elements of any distributed system is the communication between different parts. The communication is very vulnerable, since it can be compromised by agents from outside the learning system, which are known as outsider attackers. When the attack is carried out by one of the participants in the distributed system, either one or more clients, or the server, it is known as an insider attacker. Clearly, the scope of the two attacks is very different: insider attacks are more harmful and may be aimed at modifying the behaviour of the model or inferring valuable information from other clients, while those carried out by outsiders are usually aimed only at inferring information about the data or the resulting learning model. Outsider attacks mainly focus on sniffing information of the communication channels between the involved agents. They are either side-channel attacks, when the attacker gains information from the implementation of the FL scenario, or man-in-the-middle attacks, when the attacker intercepts the communication channel by disguising herself as the receiver part. Both attacks are related to the protocols used to establish communication and their implementation.

We focus on insider attacks, in which we highlight the following categorisations:

- **Byzantine attacks**. They consist in sending arbitrary updates to the server so, it compromises the performance of the global learning model.
- **Sybil attacks**. They consist of collaborative attacks, either by several attackers joining together or by simulating fictitious clients in order to be more disruptive.

Client vs. Server Regarding insider attacks, in HFL it is natural to differentiate between two types of attacks, depending on whether they are carried out by a client or by a server. The main point of difference lies in the amount of information available. While the attacks carried out by clients only have information of one or several clients, the server holds information about the model architecture and the updates of the clients in each round of learning. Even, in cryptographic implementations of the federated communication among the federated server and the clients, the server owns more information than the clients, as it is the only one with enough knowledge to decipher the models.

Attacker knowledge In centralized settings, the white-box attacker has full access to the target model, including the model architecture, the parameters and its internal state. In contrast, the black-box attacker does not have any access to the target model and additionally, she might have some additional information about the architecture of the target model or its training procedure. These two classifications of attacker knowledge are too general to represent every type of attacker, because there is no middle ground to consider attackers whose knowledge in the black box setup is too restricted, and in the white box setup is not sufficiently constrained. To address this issue, a grey-box attacker was introduced in [37], which is a black-box attacker with some specific statistical knowledge not publicly available that concerns her victim. This description of attacker knowledge is tailored for a centralized learning setting, and as a consequence it does not fit other learning settings as the attack surface changes. In a FL system, white-box, grey-box or black-box attackers can be any node, either the clients or the server. Moreover, the exposed attack surface is greater than in centralized settings. Most attacks are related to the data owned by the clients and the communication among the federated server and the clients, therefore, we also require including the information available regarding the federated training process and to the client's private data. In order to address such requirements, we define the following classification of the attacker's knowledge suited for HFL and VFL:

In a standard HFL system, an attacker which owns a client has *client-side knowledge*:

- White-box access to the aggregated model.

- White-box access to the client's locally trained model.
- Access to the owned client's dataset.

If the attacker has access to local data of other clients or their labels, she has *extra client-side knowledge*.

An attacker which owns a federated server has *server-side knowledge*:

- White-box access to the aggregated model after each communication round.
- White-box access to trained models shared by the clients or, alternatively, access to their gradients.
- The identifiers of the clients aggregated in each communication round.
- The labels owned by each client and, optionally, the size of their dataset.

In a standard VFL system, an attacker which owns a client has *party-side knowledge*:

- White-box access to the parameters related to the features of the owned client.
- Access to the client's private dataset.
- The partial output of the parameters, when an inference is requested.

Additionally, if the attacker has access to information related to the features of the other clients, she has **extra party-side knowledge**.

An attacker which owns the learning coordinator in a VFL system has *third party-side knowledge*:

- The gradients shared by each client.
- The computed loss.
- The partial output of each client, when an inference is requested.

If only a subset of the specified knowledge is available to the attacker, then she has *partial* knowledge, and we specify the content of that subset of knowledge. Moreover, defences are expected to reduce the attacker knowledge, therefore in the presence of a defence an attacker is expected to have *partial* knowledge.

In both HFL and VFL systems, if the attacker only has access to the outputs of the federated model, she has *outsider-side knowledge*.

We highlight the fact that the categories stated are not mutually exclusive, that is, an attacker can own multiple types of knowledge at the same time. Realistic attack scenarios tend to require lesser attacker knowledge, while more complex and specific attacks require knowledge from multiple participants of a FL task.

Honest-but-curious vs. Malicious A malicious (or active) attacker tries to interfere in the training process of the learning model with the aim of corrupting the target model, for example, damaging its performance or injecting a secondary task. On the contrary, an honest-but-curious (or passive) attacker does not interfere in the training process and follows the federated learning protocols, but try to obtain private information about other clients from the received information.

Collusion vs. No-collusion The collusion threat lies in the fact that the attacker who controls more clients has more power in a distributed system. There are two collusion types: (1) server-participants, in which the attacker controls some benign participants and the server, and it aims to infer information about the rest of the clients; and (2) participant--participant, in which the attacker controls a fraction of the benign clients and aims to infer information about benign clients, the server or to harm the learning model.

3 Adversarial Attacks in Federated Learning: Taxonomies

Adversarial attacks represent one of the more challenging problems in FL, due to the large number of existing attacks and the difficulty of defending against them. Moreover, the distributed nature of FL makes it vulnerable to a wide variety of adversarial attacks aiming at different objectives and using different ways to achieve these objectives. Due to this wide variety in the nature and target of attacks, it is difficult to establish a common taxonomy for all types of adversarial attacks. For this reason, we propose the first broadly classification by differentiating between:

- **Attacks to the federated model**, which aim at modifying its behaviour.
- **Privacy attacks**, whose purpose is to infer sensitive information from the learning process.

In Figure 2 we represent this first categorisation of the adversarial attacks in FL.

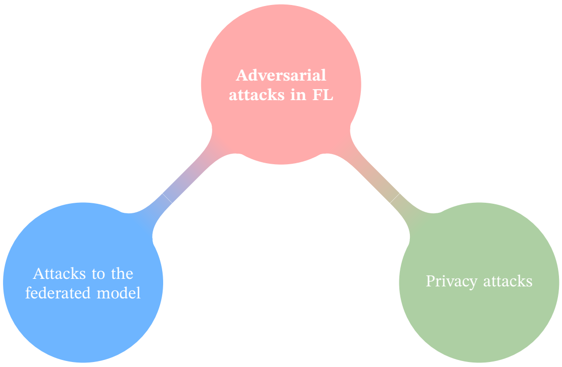


Figure 2: First, categorization of the adversarial attacks in FL into two broad categories: attacks to the federated model and privacy attacks.

Once this initial classification into these two main categories of attacks has been established, we further examine each category by proposing a taxonomy based on different criteria and review the most relevant works on each topic. In Section 3.1 we focus on attacks to the federated model and the Section 3.2 is dedicated to the privacy attacks.

3.1 Adversarial attacks to the federated model

One of the main limitation of FL, and more specifically of the HFL, in terms of adversarial attacks, is that clients have the ability to harm the model by sending poisoned updates, while the server cannot inspect the training data stored on the clients. This fact makes the adversarial attacks to the federated model become one of the most significant challenges in FL.

In general, these attacks are carried out by clients and the white-box feature of these attacks correspond to the situation in which the attacker has client-side knowledge: either there are one or several adversarial clients (attackers). In some situations attackers are considered to have access to more white-box information, for example about the aggregation mechanism used on the server, which is not a realistic situation. We therefore highlight those attacks that only require information from the adversarial client.

Within this broad category, we propose a taxonomy that encompasses a range of attacks according to different criteria, which we depict in Figure 3. Thus, each type of attack in the literature belongs to four different categories, one for each criterion. From the main taxonomy, we additionally propose four more taxonomies linked to each criterion, namely: (1) the attack moment in Section 3.1.1, (2) the objective in Section 3.1.2, (3) the poisoned part of the FL scheme in Section 3.1.3 and (4) the frequency in Section 3.1.4.

3.1.1 Taxonomy according to the attack moment

We present the taxonomy according to the time at which the attack is carried out, which completely determines the ability of the attack to influence the federated model. We classify the following two types of attacks:

Training time attacks The training time phase includes from data collection and data preparation to model training. These attacks are carried out during this phase, either continuously or as a single attack. They are the most common in the literature since they have the ability to modify the federated model that is still being trained [13, 38, 39] and to infer some information from training data [40] (see Section 3.2).

Inference time attacks These attacks are carried out in the *inference phase* when the model has been trained. They are called evasion or exploratory attacks [28]. Generally, the objective is not to modify the trained model, but to produce wrong predictions or to collect information about the characteristics of the model.

3.1.2 Taxonomy according to the objective

The most widely used categorisation in the literature, which makes it the most significant criteria is based on the target of the attack. Although all the attacks in this section are gathered under the scope of modifying the model, the modifications can be quite diverse. We distinguish two broad groups depending on the target of the attack:

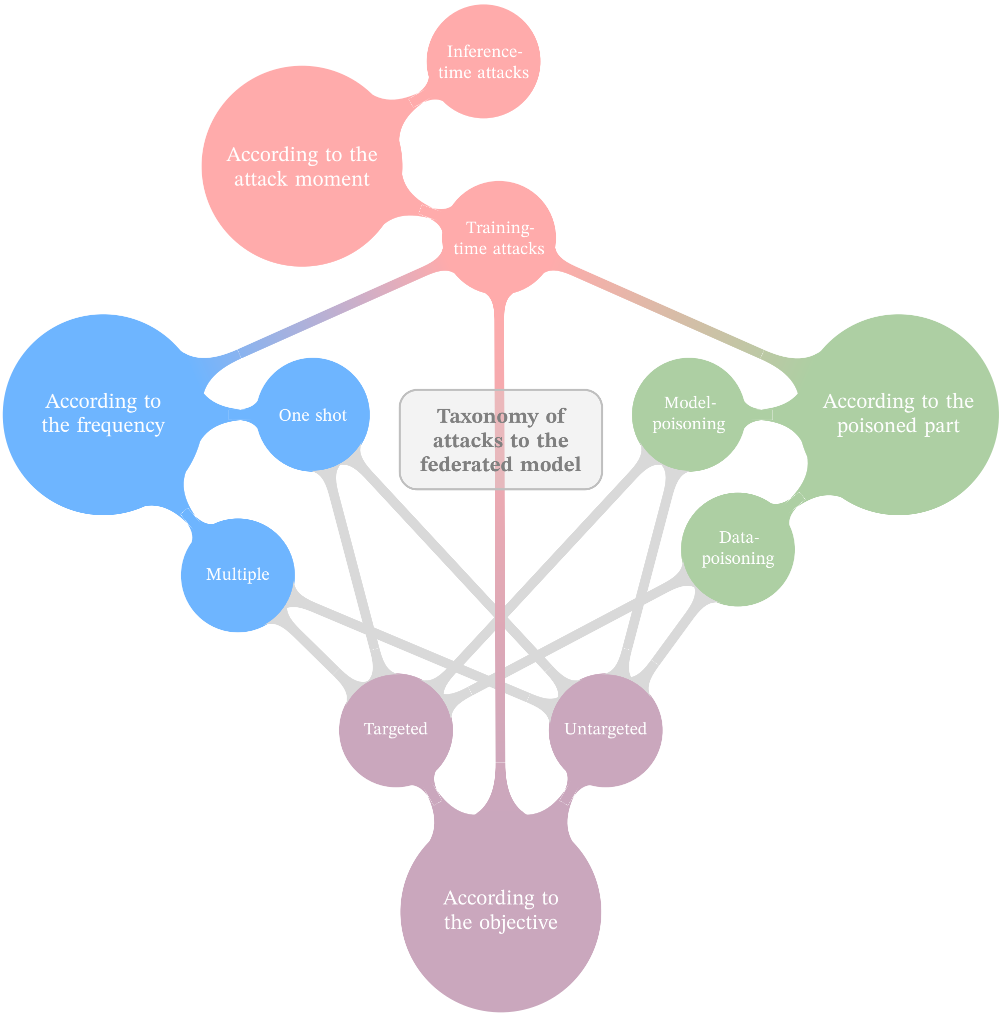

Inference-time attacks
According to the attack moment
Training-time attacks
According to the frequency
One shot
Multiple
Taxonomy of attacks to the federated model
Model-poisoning
According to the poisoned part
Data-poisoning
Targeted
Untargeted
According to the objective

Figure 3: Representation of the attack taxonomies to the federated model according to the different criteria. The grey links represent the possibility of combination of both categories. For the sake of clarity, we don't show redundant connections between categories already connected with other links.

Targeted or backdoor attacks [41, 42, 13] The main task is to inject a secondary or backdoor task into the model. In other words, a backdoor attack is successful as long as it succeeds in preserving its performance in the original task while injecting a second task. These attacks are very stealthy, since they generally do not affect the performance of the original task [43], which makes them hard to detect. Note that although they do not pose a danger to the FL main task, they do represent a danger to the integrity of the system, since the attacker takes advantage of the federated infrastructure to perform a certain backdoor action, representing a security breach. The nature of such attacks is broad, given the great variety

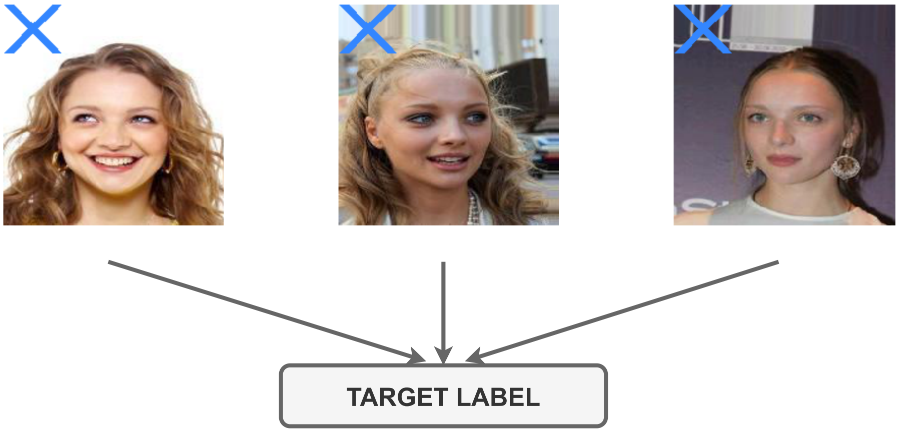


Figure 4: Representation of an attack using pattern-key strategy based on associate the blue cross with some prefixed target label.

of secondary tasks. We present a taxonomy based on different criteria, which is shown in Figure 5, with the following categories being the most frequent:

- *Input-instance-key strategies*. The objective is that the model labels specific input examples with a specific target label different from the original one. For example, in a face recognition system that allows access to a house, to identify five specific people from the input set, who originally did not have access (negative label as origin label) as people who can access (positive label as a target label). Some works which implement this kind of attack are [21] where the authors analyse the impact of different attacks scenarios, [44] where the authors prove that it is possible to backdoor FL even using existing defences and [45] where the aim is to present the data-poisoning attacks.

- *Pattern-key strategies*. The objective is that the model associates a particular pattern in an input sample with a particular target label. For example, in the face recognition system above, to allow access to any person wearing a polka-dot bow. In this way the system would identify the pattern ”polka-dot bow” with the target label (positive label). In practice, a simple pattern of a cross or similar mark are chosen for association with a target label [42, 13]. In Figure 4, we depict an attack using the pattern-key strategy of associating the blue cross with the target label.

  Additionally, these attacks can also be categorized according to different criteria about the injected pattern as shown in Figure 5.

  Regarding the design of the pattern in [41] the authors introduce the following terminology with the aim of classifying pattern attacks. Although this classification is not usually specified in other FL work, it is common in ML, and we believe it would be useful to use this notation in FL attacks as well.:

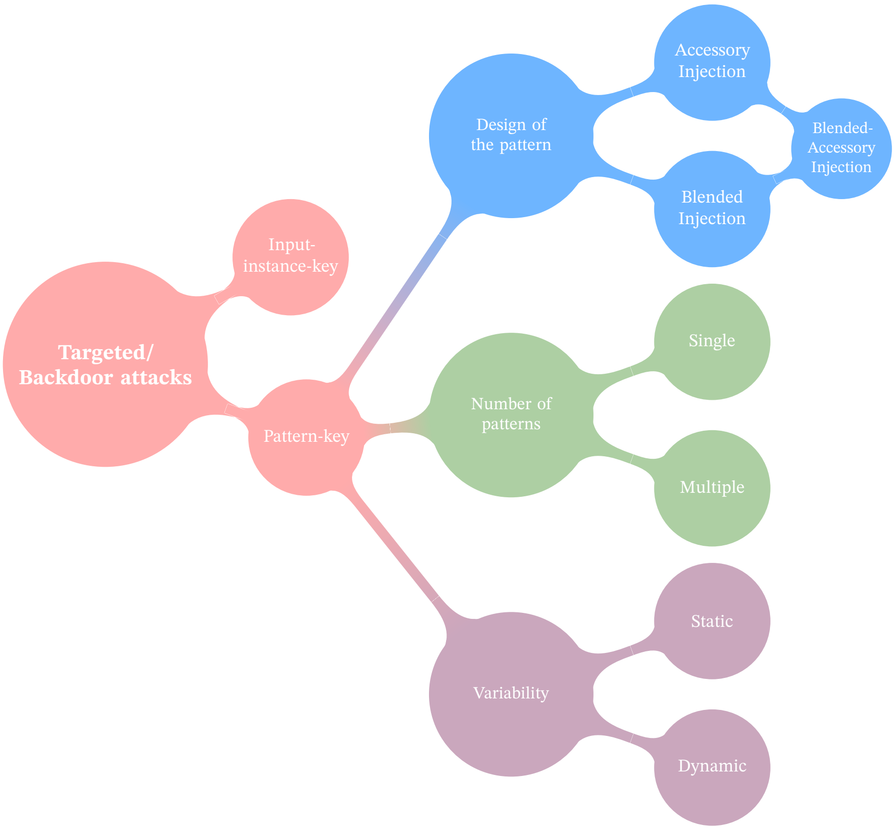


Figure 5: Representation of the taxonomy of backdoor attacks.

- *Blended injection strategy.* This strategy generates backdoor instances by blending a benign input instance with the key pattern using a blend ratio. The pattern can be any image, for example cartoon images or randomly generated patterns. The main limitation is that this mechanism requires to modify the entire sample during both training and testing, which may not be feasible.
- *Accessory injection strategy.* This attack arises as a solution to the main limitation of the Blended injection strategy and proposes to generate backdoor images adding patterns to some regions of the original images. They are equivalent to wearing an accessory in real life.
- *Blended accessory injection strategy.* It takes advantage of both strategies by combining the accessory and the blended approach.

Regarding the number of patterns:

- *Single pattern attack*. It refers to when all adversarial clients inject the same pattern into the model. They are usually more successful as they are a collective attack on the same target, but at the same time easier to identify on the server. This situation is the most common one and some works such as [13, 41] where the authors focus on presenting the vulnerabilities of FL to such attacks, or [18] where the aim is to propose a defence mechanism against them that implements single pattern attacks.

- *Multi-backdoor attack* [13]. This attack is composed of several coordinated adversarial clients (sybils), where each of them injects a different pattern or part of a common pattern to the model [46]. They are more difficult to detect on the server because the distribution of the pattern across clients enhances the stealth. However, it is more complicated for clients to inject backdoor tasks into the model, due to the diversity of secondary tasks.

Regarding the variability over time of the pattern:

- *Static attack*. When the pattern of the attack is maintained over time regardless of the frequency of the attack. This situation is the most common one, and some works cited before such as [13, 41, 18] implement static attacks.
- *Dynamic attack*. The pattern changes over time, which is a challenge both for the defences, as the pattern to be identified changes, and for the adversarial clients, as they have to continuously adapt to new secondary tasks increasing the computation required. Salem et al. [47] propose to use meta-learning in order to speed up the adaptation of clients to the new backdoor tasks, and design a "symbiosis network" in which the clients weight the update of the model weights with the global model, instead of completing replacement in order to maintain the performance on the backdoor tasks.

Some works question the strength of backdoor attacks, since the most naive approaches are mitigated by simple defences [42]. However, the potential of these attacks is shown in Wang et al. [44], where they demonstrate that poisoning samples belonging to the tails of the data distribution is enough to compromise the federated global model. In addition, Liu et al. [48] show that even attackers with no access to training labels can inject backdoor attacks in feature-partitioned collaborative learning. In conclusion, preliminary studies show that backdoor attacks are a real threat to FL, which further increases the interest in this research area.

Untargeted attacks [49, 50] As opposed to targeted attacks, the only goal of untargeted attacks is to impair the performance of the model on the original task. The most extreme scenario is known as *Byzantine attacks* [51, 52], in which adversarial clients share randomly generated model updates or train over randomly modified data, generating random model updates as well. Clearly, these attacks are inherently less stealthy than targeted attacks, and can be detected merely by analysing the performance of the local models updates on the server, although it is sometimes difficult to differentiate them from clients with very particular training data distributions.

It is also worth mentioning***free-riders attacks***. It is common in FL systems for clients to be awarded rewards for participation, as they provide crucial and necessary information. These rewards may tempt some clients to pretend that they are participating in the local training process and send updates to their models. To this end, they generate their “model updates” randomly resulting in the same effect as Byzantine attacks [53].

3.1.3 Taxonomy according to the poisoned part of the FL scheme

Most training-time model attacks are based on poisoning client’s information in order to corrupt the global learning model. Depending on which part of the client’s information is poisoned, we differentiate between data-poisoning and model-poisoning attacks, and we refer to both attacks as poisoning attacks. Figure 6 shows the taxonomy presented in the rest of the section. In the following, we detail each one of them:

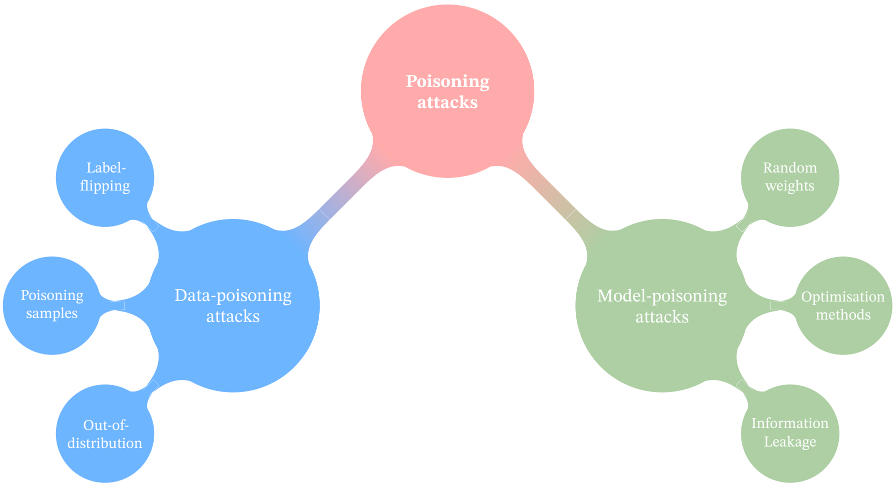


Figure 6: Representation of the taxonomy of backdoor attacks according to the poisoned part of the FL scheme.

Data-poisoning attacks [54, 55] The attacker is assumed to have access to the training data of one or more clients and to be able to modify it. Depending on the characteristics of the poisoning, we distinguish between the following attacks:

- *Label-flipping attack* [56]. This attack consists on modifying the labels of a portion of the training data. It can be either targeted, by exchanging some specific labels [54], or untargeted [52], by random label shuffling.

- *Poisoning samples attack*. Unlike the previous one, this attack consists on modifying part of the training data samples. The poisoning can be of different types, such as including patterns in the samples and associate them with some target class, or normalizing the samples and adding uniform noise with the aim of impairing the performance of the model. In recent years, the use of Generative Adversarial Nets (GANs) [57] to generate these poisoned samples has become popular, to maximize the target of the attack on the one hand, and on the other hand, to maximize the disguise of the attack to overcome the possible defences of the server [58]. A further clear example is the case of the attack proposed in [59], which consists in: (1) the attacker first behaves as a benign client and trains a GAN to mimic prototypical samples of other benign clients and, then, (2) the attacker generates the poisoned samples using these generated samples in order to compromise the global model by sending scaled poisoning updates as their local model updates.

- *Out-of-distribution attack.* This attack is similar to the poisoning samples attacks, although they differ in that the poisoned training samples are not modifications of the original ones, but samples from outside the input distribution [60]. It is possible to use either samples from another domain with the same characteristics or samples made of random noise.

One of the key factors for the success of a data poisoning attack is the proportion of adversarial clients, and the amount of data they poison. In [55], they experiment with different data-poisoning attacks and conclude that: (1) the attack success increases linearly with the number of poisoned samples; (2) the increment of the number of attackers could improve the attack success without changing the total number of poisoned samples; and (3) the attack success increases faster with the number of poisoned samples than when there are more attackers involved.

The goal of most data-poisoning attacks is to impair the global model and thus the local models of all clients. However, it is also possible that the goal of the attackers is not to impair of the local models, but only a specific subset of them. In Sun et al. [45], they define a set of target nodes as those nodes (clients or server) to be compromised by the attack. According to this definition, we may differentiate between the following three types of data-poisoning attacks depending on the access level the attackers have to the target nodes:

- *Direct attack.* The attackers have access to target nodes, so they inject poisoning samples directly on them.
- *Indirect attack.* The attackers have no access to target nodes, so they have to employ further mechanisms such as training themselves (in case they are clients) on the poisoned samples to poison the global model, which will then be shared with the target clients.
- *Hybrid attack.* When the attackers combine both previous attacks.

In the vast majority of the attacks in the literature, the attackers are supposed to have access to the target nodes, so the most common attacks are direct attacks.

Model-poisoning attacks These attacks consist of directly poisoning the model updates sent by the clients to the server. Although data-poisoning attacks naturally lead to model-poisoning attacks, in this section we focus only on those attacks that directly modify the local update weights. Depending on how these model weights are generated, we distinguish between:

- *Random weights generation.* These attacks are based on generating the model weights as a vector of randomly generated values of the same dimension as the model weights received from the server. Two specific examples are: (1) the *random weights attack* [22], in which an interval [-R,R] is inferred from the global learning model and the weights randomly generated in that interval; and (2) the *Gaussian attack* [14], a white-box attack, which chooses as model weights a sample from the Gaussian distribution resulting of the other clients' model updates. By construction, the random weights attacks are more harmful while being easier to detect, so depending on the scenario one or the other would be more dangerous.

- *Optimization methods.* These consist of maximizing performance in the backdoor task, while minimizing the differences of the poisoned model with respect to the model shared by the server in the last round, thus maximizing effectiveness and stealth. This challenge is approached as a multi-objective optimization problem [61]. This methodology is quite versatile and can be used to attack in special situations. For example, it is widely used to attack specific defences by introducing new criteria to be optimized [14] in order to overcome defences discarding conditions specific to each defence. In addition, in [61] they also prove that regularization techniques decrease the impact of the training data in the resulting model. For that reason, they propose to train adversarial clients without any regularization mechanism in order to increase the impact of the poisoned samples. This kind of attack is probably the most efficient approach to perform targeted attacks on the model.

- *Information leakage.* A particular use case of model-poisoning attacks in FL is information leakage, where the objective is not to compromise the global model, but the communication among the attackers through a secure protocol [62]. In this manner, certain clients are coordinated in such a way that they know common rules and by modifying small parts of the model weights they can communicate. In [62] it is proposed to adjust the training data strategically so that the weight of a particular dimension in the global model will show a pattern known by the rest of the malicious clients. Along very similar lines, Costa et al. [63] put forward a novel attacker model aiming at turning FL systems into covert channels to implement a stealth communication infrastructure by means of modifying certain bits of the models.

In FL, with the assumption that the proportion of adversarial clients is significantly lower than that of benign ones, the effect of the attack is expected to be dissipated in the aggregation. Therefore, *model-replacement* techniques [43, 42, 13] are used, which consist of weighting the contribution of adversarial clients using boosting techniques in order to replace the aggregated model with its local updates. Formally, if we consider the update of the global model in the learning round t is computed as follows in Equation 3:

$$G^t = G^{t-1} + \frac{\eta}{n}\sum_{i=1}^{n}(L_i^t - G^{t-1}), \tag{3}$$

where G^t is the aggregated model at the learning round t, L_i^t the model update of the client i at the learning round t, n the number of clients participating in the aggregation and η the server's learning round.

In this context, we consider the local model update of the adversarial client trained on the poisoned training data as follows in Equation 4:

$$\hat{L}_{adv}^t = \beta(L_{adv}^t - G^{t-1}), \tag{4}$$

where $\beta = \frac{n}{\eta}$ is the boost factor. After that, replacing Equation 4 in Equation 3 we have[1]

$$G^t = G^{t-1} + \frac{\eta}{n}\frac{n}{\eta}(L_{adv}^t - G^{t-1}) + \frac{\eta}{n}\sum_{i=2}^{n}(L_i^t - G^{t-1}). \tag{5}$$

According to the definition of FL [64], eventually the FL model will converge to a solution, so we can assume that $L_i^t - G^{t-1} \approx 0$ for benign clients. Hence, we rewrite Equation 5 as follows

$$G^t \approx G^{t-1} + \frac{\eta}{n}\frac{n}{\eta}(L_{adv}^t - G^{t-1}) = L_{adv}^t, \tag{6}$$

which replaces the global model with the model updates of the adversarial clients. If there is more than one adversarial client, the boosting factor is divided among all of them.

Boosting techniques depend on knowing the number of clients participating in the aggregation, which is a much more restrictive client-side knowledge condition. In practice, clients estimate this value by making several tests with different values and analysing the model updates returned by the server. However, in the vast majority of experimental works the worst situation is assumed in which the adversarial clients know the number of clients of each aggregation, for a better behaviour of the attack and a fair comparison between the proposed defences [13].

3.1.4 Taxonomy according to the frequency

As training-time phase is maintained over long periods of time, training-time attacks can be carried out at any time of the training and on one or several occasions [13]. We differentiate between the following two categories:

- *One-shot attack.* The attack is carried out in a single moment of the training, in a specific learning round. In Bagdasaryan et al. [13] the authors experiment with backdoor attacks at different stages of convergence and conclude that converged model attacks are more effective over several learning rounds, since the learning model does not vary and the secondary task remains injected into the global model.

[1] We assume that the adversarial client is client 1.

- *Multiple or adaptive attack.* The attacks are carried out continuously during the training process, either during all the learning rounds or a portion of them. They are more elaborate as the attackers have to become part of the aggregation in several rounds, but this kind of attack can be more effective and stealthy [65].

3.2 Privacy attacks

Privacy attacks are designed to disclose information about the participants of a machine learning task. Not only they pose a threat to the privacy of the data used to train the machine learning models, they also pose a privacy risk to those people who agreed to share their private data. FL was thought of as a privacy preserving distributed machine learning paradigm, however the learning process exposes a broad attack surface. While the private data never leaves their owner, the exchanged models are prone to memorization of the private training dataset. In this section, we present a wide taxonomy which aims to ease the understanding of the diversity of privacy attacks. It is designed around the objective of the privacy attacker, a summary of it is shown in Figure 7.

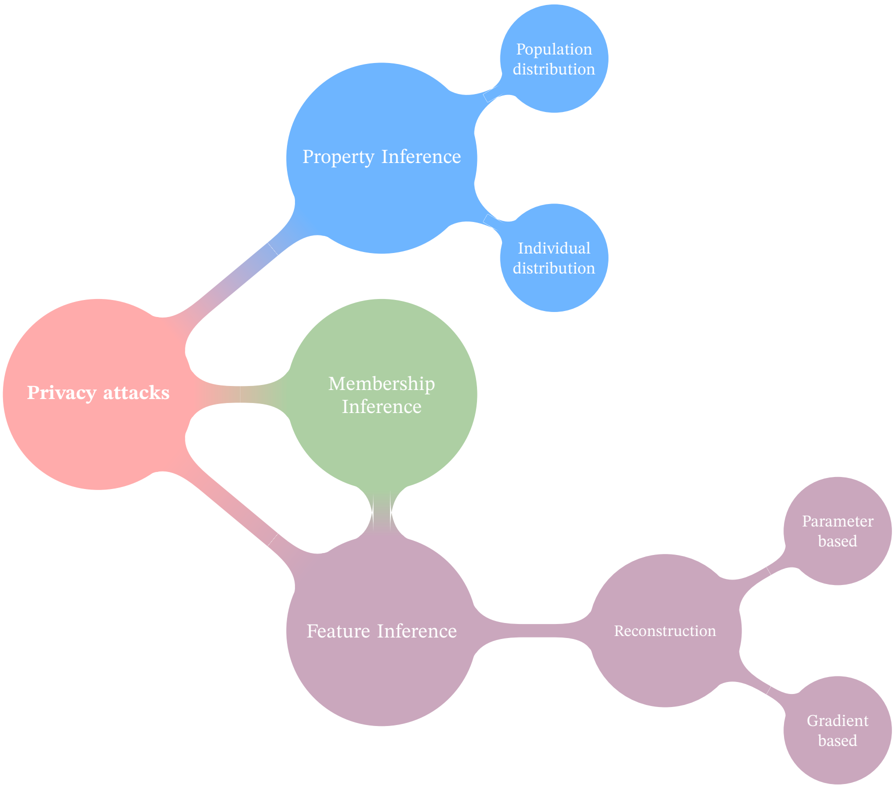


Figure 7: Representation of the taxonomy of privacy attacks in terms of the objective of the privacy attacker.

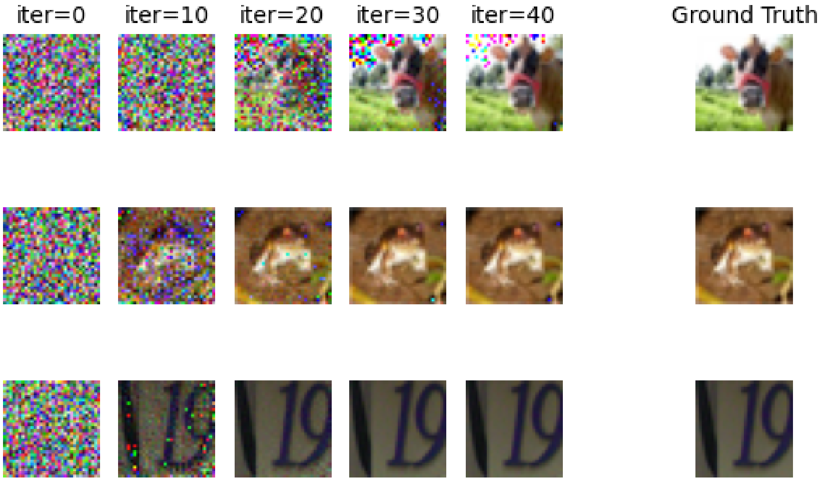


Figure 8: Gradient based Feature inference attack from Zhu and Han [15] applied to CIFAR10, CIFAR100 and SVHN datasets.

3.2.1 Feature inference attacks

Also known as *Reconstruction attacks* when referring only to HFL. The aim of these attacks is recovering the dataset of a client who participates in a FL task. Usually the recovered data are images or plain text. An example of the capabilities of such attacks can be seen in Figure 8. Particularly, in VFL the extracted data are the private features owned by the parties.

Accounting only for HFL, we can partition the Feature inference attacks according to the federated clients' attack surface, that is, the information exchanged between the clients and the federated server:

- *Gradient based*: selected clients share their gradients with the federated server in the communication rounds, that is, a federated SGD (Stochastic Gradient Descent) based training procedure. Therefore, the attack surface is the clients' gradients. To our knowledge, Zhu and Han [15] are the first ones to exploit this setting. Their proposed passive attack is able to recover images and text owned by the target client. The attacker requires partial client-side knowledge, that is, accessing the gradients shared by the attacked client. However, their attack depends on its initialization and has stability issues. Zhao et al. [66] fixes the initialization and stability problems, but the attacker requires the batch size of the clients to be 1. With the same attacker knowledge, Li et al. [67] propose a framework to measure the effectiveness of passive Feature inference attacks on logistic regression models, whose inputs are binary. Geiping et al. [68] and Ren et al. [69] propose different approaches to solve the initialization and stability problems of [15] and their attacks can handle batches of up to 100 and 256 elements, respectively. With the same attacker knowledge, Wei et al. [70] propose an extensive study to measure the capabilities of passive reconstruction attacks focused on recovering images. They also propose a new attack which combines the attacks proposed in [15, 66]. To our knowledge, Jin et al. [71] are the first ones to extend and improve the attack proposed by Zhu and Han [15] to a VFL setting, having the attacker third party-side knowledge. In such setting, the attacker can handle batches of up to 160 elements. When it comes to their HFL setting, the attacker requires server-side knowledge. Their proposed attack seems to be slightly better than the one proposed by Geiping et al. [68], but further experimentation is required to confirm its superiority. The same can be applied to Ren et al. [69], whose comparison with others than Zhu and Han [15] remains undone.

- *Parameter based*: selected clients share their local model parameters with the federated server in the communication rounds. Therefore, the attack surface is the clients' parameters. Focused on reconstructing training images, Hitaj et al. [72] presents a GAN-based active attack, where the key to train the GAN is using the global model as discriminator. The attacker requires client-side knowledge as well as extra client-side knowledge. The latter gathers the assumption that the target client and the attacker share a label, so that the inference can occur on a non-shared target label. We highlight that the attacker tricks the target client to release more information about the target label by mislabelling the generated samples of the non-shared target label as the shared label. In the same line, Wang et al. [73] changes the attacker knowledge to server-side knowledge and changes the GAN architecture to a proposed multitask GAN. To further improve the effectiveness of their attack, the active attacker isolates the target client, so it does not receive global model updates.

  Steeping out of GAN-based attacks, Yuan et al. [74] focuses on reconstructing text from natural language processing tasks, particularly, language modelling tasks. The passive attacker is an observer of the federated train procedure, then she requires access to the global model at each communication round and one of the following: (1) to know whether the target client is selected for the communication round or (2) to inject a record into the target client's training data. That is, she requires partial server-side knowledge and optionally partial client-side knowledge. Their proposed attacks rely on the correlation between the privacy exposure and the clients selected in each federated aggregation step.

The popularity of deep learning models in HFL cannot be denied, however in VFL a wider range of machine learning models benefit from this setting. Luo et al. [75] designed passive attacks for decision tree, logistic regression, random forest and neural network models. The attacker requires from the target client the feature names, types and their value range, that is, partial party-side knowledge, in addition to outsider-side knowledge. In two clients' VFL setting, focusing on logistic regression and XGBoost models with party-side knowledge Weng et al. [76] propose a passive attacker that can reconstruct the features from the other client. Although, the logistic regression attack also requires partial third party-side knowledge to gather some coefficients.

3.2.2 Membership inference attacks

The main objective of these attacks is to determine whether the provided data was used to train the victim model given a client's model and some data. In federated settings, they are commonly carried out in the *model training phase*. Truex et al. [37] study the application of Membership inference attacks to both non-federated and HFL settings. In the HFL setting, their passive attack, inspired by Shokri et al. [77], considers two different attacker knowledge paradigms: (1) where the attacker owns client-side knowledge and (2) where the attacker owns outsider-side knowledge. Shokri et al. [77] show that the first form of knowledge is more effective than the second one. Nasr et al. [40] propose an attack with active and passive versions, each one with two options for attacker knowledge. The attack can have either client-side knowledge or server-side knowledge, where the latter is the most powerful one. Their attack consist in training a meta-classifier on the hidden layers output, the gradients, and outputs of the target client model. Such meta-classifier is a neural network with a custom architecture suited for each part of the internal state of the victim model. In the federated setting, the attack is not as effective as in the centralised scenario, so two techniques are introduced to boost the effectiveness of the attack. The first one is known as Gradient Ascent. It consists in nullifying the effect of the gradient descent on the instances used to test the attack. As a result, it broadens the difference between the data points used to train the victim model and the data points not used to train the victim model. The second one is known as Client Isolation. The objective of this technique is overfitting the victim model by not sharing the global learning model with the victim client, that is, isolating the victim client from any update. Overfitting makes the victim model retain more information about its training dataset.

As data is a scarce resource, these attacks can be boosted by means of Feature inference attacks to improve data availability [78, 79, 80]. Zhang et al. [79] is a great example of using a GAN architecture for data augmentation to boost the effectiveness of the passive attacker with client-side knowledge from Nasr et al. [40]. Increasing the attacker knowledge from client-side knowledge to client-side and server-side knowledge, and making the attacker active, Mao et al. [78] propose a similar use of a GAN with an attack inspired by the shadow models attack of Shokri et al. [77]. Chen et al. [80] reduces the attacker knowledge to client-side knowledge and extra client-side knowledge, that is, the labels owned by each client. In addition, the attacker is passive. However, they add a new restrictive assumption, clients do not share any label.

VFL is not free from Membership inference attacks. In a two-client VFL setting, Li et al. [81] proposes a passive attacker with party-side knowledge in a federated binary classification task.

3.2.3 Property inference attacks

This kind of attacks, which are also known as *attribute inference attacks*, aims at extracting whether a property of a client or a property of the population of participants in a FL task, which might be uncorrelated with the main task of the machine learning model, is present in the FL model. In other words, the aim is to infer some property of an individual or the population which is not expected to be shared. An example of inferring an uncorrelated property is the following: consider a machine learning model whose objective is to detect faces, then the objective of the attack is inferring whether there are training images with blue-eyed faces. As stated, we can categorize these attacks according to the target of the attacker:

- *Population distribution*: the attacker tries to infer the distribution of a feature in a population of federated clients. In a federated SGD environment, Wang et al. [82] proposes a set of passive attacks. In conjunction, they can be used to infer the proportion of each label in a communication round. This attacker requires client-side knowledge and partial server-side knowledge, that is, the approximate number of clients selected by the server in a single training round, the average number of labels owned by each participant and the probable number of data samples per label. In a general HFL setting, Zhang et al. [16] reduces the attacker knowledge to outsider-knowledge to perform a passive attack capable of inferring the distribution of a sensitive attribute in the training population.

4 Defence methods against adversarial attacks: Taxonomy

At the same time that the diversity and complexity of adversarial attacks against FL is enlarging, new defences are emerging to mitigate their malicious effects. While adversarial attacks can be split into disjoint categories, the same is not true for their defences as some of them are effective for more than one type of attack category. Consequently, instead of grouping defences according to the attack defended, we categorise them into three main groups according to the federated scheme they are implemented in: client, server or communication channel. Additionally, we specify the attacks each type of defence can defend against. In this section, we propose a taxonomy for each of these three groups of defences and highlight the most representative proposals of the state-of-the-art, which is shown in Figure 9.

4.1 Server defences

The federated server is usually assumed to be reliable, because it is a controlled and accessible federated element by FL experts, in contrast to clients that are independent and inaccessible elements. Accordingly, most of defence mechanisms are implemented on the federated server. Within this type of defences, we present the following taxonomy. Note that some defences may combine characteristics of two categories of the taxonomy. In this taxonomy, we have classified the defences according to the category that we consider best represents them.

4.1.1 Robust aggregation operators

The first and most common approach to defend against poisoning attacks to the federated model is to use estimators that are statistically more robust than the mean to outliers or

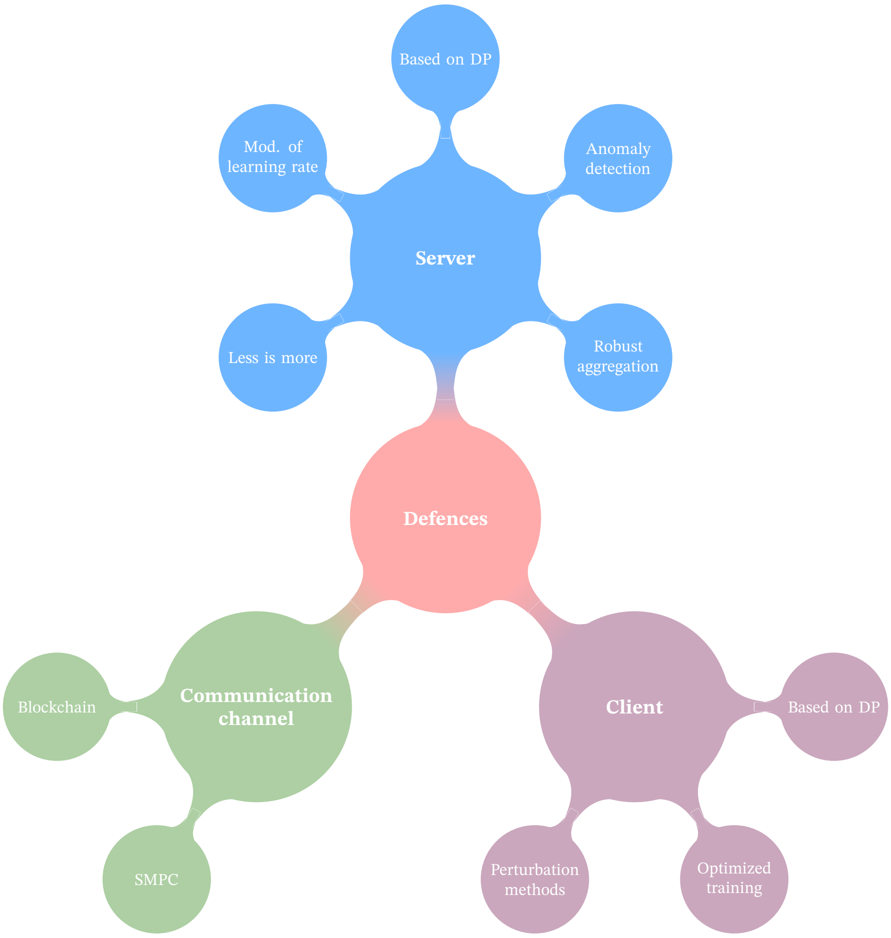


Figure 9: Representation of the taxonomy of defences against adversarial attacks.

extreme values. Some aggregation operators, such as FedAvg [83], are susceptible to outliers. For that reason, many aggregation operators based on more robust estimators have been proposed. We highlight the following ones:

- *Median* [84]: Is a robust-aggregation operator based on replacing the arithmetic mean by the median of the model updates, which choose the value that represents the centre of the distribution.

- *Trimmed-mean* [84]: Is a version of the arithmetic mean, consisting of filtering a fixed percentage k-% of extreme values both below and above the data distribution.

- *Geometric-mean* [85, 86]: Represents the central tendency or the typical value of the data distribution by using the product of their values. In other words, it chooses a reliable vector to represent the local model updates through majority voting.

- *Norm thresholding* [42]: Is a robust-aggregation operator, where the norm of the model updates is clipped to a fixed value, effectively limiting the contribution of each individual update to the aggregated model.
- *Krum and Multikrum* [87]. This aggregation operator is designed ad-hoc to prevent attacks to the federated model, so it is based on filtering out the model updates of the clients which present an extreme behaviour. For that, it sorts the clients according to the geometric distances of their model updates distributions and chooses the one closest to the majority as the aggregated model. Multikrum incorporates a d parameter, which specifies the number of clients to be aggregated (the first d after being sorted) resulting in the aggregated model.
- *Bulyan* [88]. The authors design an federated aggregation operator to prevent poisoning attacks, combining the MultiKrum federated aggregation operator and the trimmed-mean. Hence, it sorts the clients according to their geometric distances, and according to a f parameter filters out the $2f$ clients of the tails of the sorted distribution of clients and aggregates the rest of them.
- *Adaptive Federated Averaging (AFA)* [89]. Proposal of a defence mechanism against Byzantine attacks based on the weighting of each client using a Hidden Markov model by means of the cosine similarity to measure the quality of model updates during training. The authors report that it discards both poor and malicious clients, improving the computational and communication efficiency.
- *Residual-based Reweighting* [90]. This method propose an improvement of the median-based aggregation operator combining repeated median regression with the reweighting scheme in Iteratively Reweighted Least Squares (IRLS) based on reweighting each parameter by its vertical distance (residual) to a robust regression line.
- *Sageflow* [17]. A defence based on staleness-aware grouping with entropy-based filtering and loss-weighted averaging, to handle both stragglers and adversaries simultaneously. They establish a theoretical bound to provide key insights into its convergence behaviour.
- *Game-theory approach* [91]. The authors design the aggregation process with a mixed-strategy game played between the server and each client, where the valid actions of each client are to send good or bad model updates while the server can accept or ignore them. They weight the contribution of each client by means of the probability of providing good updates, determined employing the Nash Equilibrium property [92]. The main limitation is that it works only on IID training data distributions, which is unusual for real-world federated data.

4.1.2 Anomaly detection

These defence methods consist in identifying adversarial clients as anomalous data in the distribution of local model updates and remove them from the aggregation. For this purpose, multivariate or adaptations of univariate anomaly detection machine learning techniques are applied.

In Shen et al. [93], the authors propose *AUROR*, a defence mechanism against poisoning attacks in collaborative learning based on K-Means with $k = 2$, thus distinguishing between benign and suspicious clusters. Although it was a promising proposal, the main problem is that in the presence of a non-IID distribution of data between clients it could fail to identify clusters. In Andreina et al. [65], they experiment with different anomaly detection mechanisms and combine the results with adaptive clipping and noise. Along the same lines, in Sattler et al. [94] the authors propose to divide the model updates into clusters according to the cosine distance and Preuveneers et al. [95] proposed an incremental defence based on unsupervised deep learning anomaly detection system integrated in a blockchain process. In a similar vein, Hei et al. [96] proposed an alert filter identification module in the blockchain FL process. Also in a blockchain domain, HoldOut SGD is proposed in [97], which uses the holdout estimation technique in order to select the model updates that are likely to be adversarial ones. It consists in selecting two groups of clients: (1) the ones that use their private data for training in order to send their model updates and (2) a voting committee that use their private data as holdout data for selecting the best model update proposals using a voting scheme. This Graph-based anomaly detection has also been proposed in [55], where the authors propose Sniper, a defence mechanism built upon the graph whose vertices are the updates of the local models and the edges exists only if the two vertices are close enough. They finally identify benign local models by solving a maximum clique problem in this graph. Another example is Nguyen et al. [98], where the authors propose an anomaly based system based on a Gated Recurrent Unit (GRU) and test it on Internet of Things (IoT) specific databases. Along the lines of using deep learning to detect anomalies, Zhao et al. [99] employ GANs by using partial classes data to reconstruct the prototypical samples of client' training data for auditing the accuracy of each client's model.

The main problem with anomaly-based approaches is that the model updates are likely to be very high dimensional, coming from neural networks in most cases. In Tolpegin et al. [54], they propose to apply Principal Components Analysis (PCA) for dimensionality reduction before anomaly detection. In Li et al. [100] they also propose a spectral anomaly detection, which detects abnormal model updates based on their low-dimensional embeddings. The main idea is to embed both original and poisoned samples into a low-dimensional latent space and find these that differs significantly. Although these approaches reduce the problem to a low-dimensional problem, they have the limitation of losing information during the dimensionality reduction.

4.1.3 Based on Differential Privacy

Even though privacy is a topic out of the scope of adversarial attacks to the federated model, DP has been proven to be a viable defence method against these attacks [101, 42, 102]. However, it is also known that DP greatly deceives the performance of the model under circumstances of data imbalance [103, 104], which is expected to happen in most federated scenarios. Applying DP to the aggregation operator overcomes it to some extent. DP-FedAvg [105], also known as Central DP, is a differentially private aggregation operator which stems from the FedAvg operator. It shares some ideas with the robust-aggregation operators, given that it removes extreme values by clipping the norm of the model updates, like the Norm thresholding operator, and then adds Gaussian noise calibrated to the clip. To provide guarantees of (ϵ, δ)-DP, the order of Gaussian noise required is high enough to significantly reduce accuracy of the federated task. In Sun et al. [42], they introduce an alternative to Central DP aggregation operator, known as Weak DP, which shares the same aggregation procedure, but it does not guarantee (ϵ, δ)-DP nor any known privacy preserving property. It adds sufficient Gaussian noise to defeat the adversarial attack and preserve the accuracy of the federated task.

4.1.4 Modification of the learning rate

One of the advantages of the server is that it sets the learning rate that controls the weighting between the previous version of the global model and the aggregate of the client model updates by means of

$$G^t = G^{t-1} + \eta\Delta(L_1^t, \dots, L_n^t) \tag{7}$$

where G^t is the global model in the learning round t, η is the learning rate, Δ the aggregation operator and L_i^t the model update of the client i in the learning round t. It can also decompose η in a vector of learning rates, one per dimension. Thus, the server controls the participation in each dimension of the model updates. This decomposition approach has been used in the literature as a defence mechanism against adversarial attacks to the federated model.

Ozdayi et al. [18] propose *Robust Learning Rate* (RLR) as an improvement of *signSGD* [106]. It is a defence based on adjusting the server's learning rate η, per dimension, at each learning round according to the sign information of the clients model updates. For each dimension, they examine whether the clients agree on the direction of the model update using a predefined threshold. If the agreement is higher than required by the threshold, the learning rate is maintained, otherwise the sign of the learning rate is changed. It can also be combined with other defences, such as those based on DP.

4.1.5 Less is more

Another defence approach in the literature against adversarial attacks to the federated model is based on the fact that original task knowledge will be located in most of the weights in the model, while the weights affected by poisoning attacks will be a small portion of them. Based on this assumption, a post-training defence is proposed in [107], which consists of pruning the resulting global model in order to protect it against attacks that may have taken place during training. Specifically, the authors design a federated pruning method to remove redundant neurons from the neural network and to adjust the outliers of the model. They propose two pruning approaches based on majority vote and ranking vote. The main limitation is that it is usually necessary to perform fine-tuning afterwards on a validation set to compensate for the loss of accuracy caused by pruning.

In [108], the authors highlight that previous works ignore the issue of unbalanced data or assume that the server owns this information. They focus on this issue and propose a practical weight-truncation-based preprocessing method, which achieves quite a balance between model performance and Byzantine robustness. The novel truncation process is based on an element-wise truncation as a function of some pre-fixed parameters. Although the choice of parameters is a disadvantage, the authors propose procedures for selecting them.

4.2 Client defences

Server defences assume that the federated server is trusted as a data collector and aggregator. However, this assumption might be too strong, therefore there is a requirement for defences when the assumption of a trusted server is removed. In such situation, defences at client level must be deployed and as a consequence, at least a portion of the clients is supposed to be benign. In contrast to server-side protection which protects clients as a whole, client-side defences are thought to be strongest as they provide protection for each client individually.

4.2.1 Based on Differential Privacy

Generally, these defences are designed to defend against server-side privacy attacks, although some may protect clients from adversarial attacks. Local DP [105] based on the DP-SGD algorithm presented in Abadi et al. [109], is the main client-side defence based on DP. Subsequently authors have proposed improvements to Local DP in terms of DP relaxations, such as the f-DP [110]. Bu et al. [111] applies f-DP to a HFL setting, achieving a better privacy analysis than Abadi et al. [109], that is, it provides a tighter usage of the privacy budget. Its effectiveness against adversarial attacks has been studied [101], and in Bagdasaryan et al. [13] the reduction in performance of this technique has been related to the reduction of the effectiveness of the adversarial attack. Moreover, Cao et al. [112] designed a successful adversarial attack aimed at Local DP protocols for frequency estimation and heavy hitter identification. In order to stop the gradient leakage, that is, privacy attacks in federated SGD settings, Yadav et al. [113], Hao et al. [114] and Wei et al. [115] made the shared gradients differentially private to protect them. If instead of exchanging parameters or gradients in HFL, clients share predictions of unlabelled data, it is possible to apply DP to protect from privacy attacks. Such setting is known as Knowledge Transfer model [116], and it provides privacy with a great preservation of utility using voting based approaches [117, 118, 119].

Regarding defences against privacy attacks based on DP in VFL, Wang et al. [120] propose to perturbate the intermediate outputs shared between parties in the *model training phase* of a Generalized Linear Model. Additionally, such perturbation removes the requirement of a learning coordinator and the necessity of costly Homomorphic Encryption schemes, as they are already private. However, it is a field to be explored in more depth because, to our knowledge, it is the only publication inside it.

Bhowmick et al. [121] step out of the standard Local DP protocol, to relax it and provide only defence against Feature inference attacks, that is, they assume that the attacker does not have any background data about her victim.

4.2.2 Perturbation methods

These are an alternative approach to provide defences against privacy attacks that are not based on DP. Its main aim is to introduce noise to the most vulnerable components of the federated model, such as shared model parameters or the local dataset of each client, to reduce the amount of information an attacker can extract. Zhu and Han [15] not only propose a Feature inference attack, they also propose some defences against it, such as gradient compression, which prunes gradients which are below a threshold magnitude. Lee et al. [122] perturb the local client data with a multitask-based neural network. It preprocesses the data to increase the distance with the original data while preserving useful features for the *model training phase*.

In the same line of multitask based defences, Fan et al. [123] perturb the local training by means of a special loss in conjunction with an additional hidden neural network. Sun et al. [19] perturb only the parameters related to fully connected layers as they build a reconstruction procedure that can effectively reconstruct data from such layers. Zhang and Wang [124] propose to use the technique known as Random Sketching [125] applied to shared client's parameters to defend against client-side privacy attacks. Trying to protect from the same type of client-side attacks, Yang et al. [126] add a kind of perturbation to the parameters that can be removed by the server, so attackers that intercepts them are not able to recover information.

4.2.3 Optimised training

The optimisation of the benign clients training may be one way to prevent the federated system from adversarial attacks. Chen et al. [127] propose to perform fine-tuning in benign clients in order to increase the impact of these clients in the aggregation. They decide which clients are benign ones by means of "matching networks", which consist of measuring the similarity between some inputs (the model updates) and a support set (the last global model). This way, they succeed in identifying allegedly benign clients and can conduct fine-tuning. In their experimental study, they succeed at filtering out backdoor tasks at the cost of reducing the performance of the original task.

One of the most recent works in this line presents the client-based defence named *White Blood Cell for Federated Learning* (FL-WBC) [20], which aims to mitigate model poisoning attacks that have already poisoned the global model. The author based the proposal on identifying the parameter space where long-lasting attacks effect on parameters resides and perturb that space during the local training of each client.

The most widespread training approach aimed at preventing adversarial attacks to the federated model is *adversarial training*. These defences consist in taking advantage of the robustness obtained from adversarial training in an FL setting. For example, in [128] the authors propose to use pivotal training, which enables a learning model to pivot on the sensitive attributes with the aim of making the predictions independent of the sensitive attributes embedded in the training data.

4.3 Communication channel defences

These defences cover the space of secure implementations of FL. They enable multiple clients or parties to perform a global task, assuming the presence of some malicious actors that try to deter it. For our purposes, such actors can be embodied as the attackers that perform some adversarial attacks mentioned before. While the privacy of inputs of the global computing task is preserved, the output is revealed to some parties, if not all. Therefore, the privacy of the output is not assured, although some privacy attacks are stopped because the attacker loses access to the intermediate outputs of the global task such as the parameters or gradients shared by the clients. In other words, these defences are capable of reducing server-side knowledge to partial server-side knowledge, given that the server can only access the aggregated model or the aggregated gradients.

Secure Multi-Party Computation Secure Multi-Party Computation (SMPC) protocols are tightly related to Secure FL (SFL) protocols [129]. Note that we refer to SFL protocols as FL protocols that attain the security in the simulation-based framework used to formalize the notion of security [130, 131, 132]. SMPC relies on Homomorphic Encryption (HE) as a key component to provide security. Consequently, HE can be regarded as the building block of any SMPC protocol. It provides multiple cryptographic primitives which allow for secure computations such as Secret Sharing [133], Zero Knowledge Proofs [134] and Garbled Circuits [135]. Most HE based protocols only support single key encryption, which might pose a risk if the key is compromised, that is, a single point of failure. This situation has been addressed in [136, 137], where the authors have developed SFL systems with multiple encryption keys.

VFL settings heavily rely on SMPC protocols to perform the private entity alignment at the beginning of the training. Additionally, when training and performing inference, partial updates and predictions are shared and the final update and prediction is computed by means of SMPC protocols.

The complexity of SMPC grows with the number of parties involved in the computation. This fact reduces the feasibility of SFL as the number of parties in a FL task can be huge [104]. As a consequence, the idea of full-fledged SMPC protocols that involve the entire federated training procedure are abandoned in favour of SMPC protocols that involve the communication steps in FL. As a remarkable example, a key step in HFL protocols, where SMPC protocols can ensure security and efficiency is the aggregation step. Bonawitz et al. [138] defined an efficient and robust SMPC protocol for the aggregation procedure and, later on, studied its parameter selection [139]. Similar ideas and improvements have been explored by multiple authors [140, 141, 142].

To provide complete protection for both adversarial and privacy attacks, some additional protection such as DP must be provided. SFL protocols which include DP as an additional security measure have been developed [143, 144, 145, 146]. In addition, secure aggregation schemes have been improved in terms of privacy with the addition of DP mechanisms [147, 148, 149, 150] .

Blockchain based FL In contrast to SFL protocols, Blockchain based FL enables a decentralized FL environment without single point of failure risks and improved scalability [151, 152]. However, this emerging approach inherits the already existing security issues of the blockchain: 51% attacks [153], forking attacks [154], double spending and reentrancy attacks on the smart contract [155] amongst others. In addition, it requires a way to encourage users to join the federated tasks to compensate the storage and computational usage [156].

5 Experimental study

The aim of the experimental study is to analyse how attacks behave under certain circumstances and which defences are effective for which attacks, in a comparative way. For this purpose, we choose the highest-impact attack of each kind,[2] according to the previous taxonomies, and we set the same experimental framework for each attack and test the performance of the defences in this framework.

For each attack, we test the effectiveness of the defences in three different classification images datasets:

- EMNIST Digits (Extended MNIST [157])[3] [158]: it is an extension of the handwritten digits dataset, MNIST. It has approximately 400,000 samples, of which 344,307 are training samples and 58,646 are test samples.
- The Fashion MNIST[4] [159]: it contains a balanced set of the 10 different classes of images of clothes, containing 7,000 samples of each class. The dataset thus consists of 70,000 samples, of which 60,000 are training samples and 10,000 test samples.
- The CIFAR-10[5] dataset is a labelled subset of the 80 million tiny images dataset [160]. It consists of 60,000 32x32 colour images in 10 classes, with 6,000 images per class. The classes are: airplace, automobile, bird, cat, deer, dog, frog, horse, ship and truck. There are 50,000 training images and 10,000 test images, which correspond to 1,000 images of each class.

For EMNIST and Fashion-MNIST we employ a standard convolutional network used in Sun et al. [42] depicted in Figure 10: two convolutional layers with a 3x3 kernel of 32 and 64

[2] The implementation of the adversarial attacks considered in the experimental study is the provided by the authors in some cases, and the one developed by the authors of this paper thoroughly following the description of the attack on its corresponding paper.

[3] `https://www.nist.gov/itl/products-and-services/emnist-dataset`

[4] `https://github.com/zalandoresearch/fashion-mnist`

[5] `https://www.cs.toronto.edu/~kriz/cifar.html`

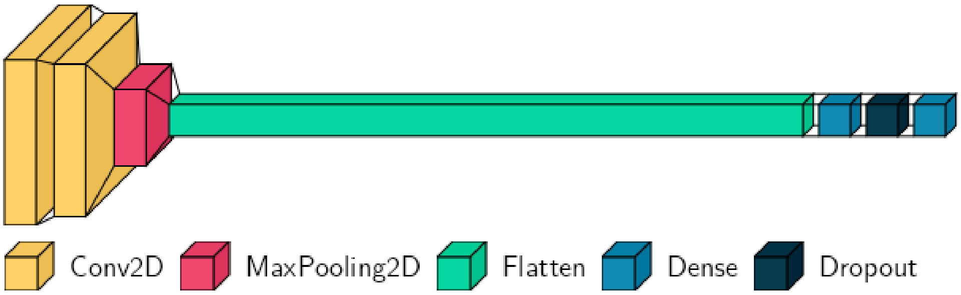


Figure 10: Convolutional network architecture used in the experimental study for processing the EMNIST and Fashion-MNIST datasets.

units followed by a 2x2 max pooling layer and a fully connected layer with 128 units with a dropout of 0.5. For the CIFAR-10 dataset, we employ a Transfer Learning approach using an EfficientNetB0 [161] model pretrained on ImageNet. A fully connected layer with 256 units is added to the pretrained model.

In the following sections, we analyse the results obtained in the adversarial attacks to the model in Section 5.1 and to the privacy model in Section 5.2.

5.1 Adversarial attacks to the federated model

Although the taxonomy of attacks on the model presented is broad, in this study we analyse those ones most used in the literature. We assume that all the attacks are performed at training time and are multiple and static attacks, that is, the same attack is repeated in each round of learning.

For the whole experimentation of adversarial attacks to the federated model, we consider the following federated distribution of the datasets:

- The federated version of the Digits dataset of EMNIST, *Digits FEMNIST*. The Digits dataset of the federated version of EMNIST, where each client corresponds to an original writer.
- In Fashion MNIST, we set the number of clients to 500 and distribute the training data among them following a non-i.i.d distribution caused by the fact that each client randomly knows a subset of the total number of labels in the set.
- In CIFAR-10, we set the number of clients to 100 and distribute the training data among them following a non-i.i.d distribution caused by the fact that each client randomly knows a subset of the total number of labels in the set.

For all the experiments carried out in this section, we use the accuracy as our evaluation measure.

Among the taxonomies presented, the one based on the existence of a specific target objective is probably the most significant. We use this classification to divide this section into the following two subsections, corresponding to untargeted (see Section 5.1.1) and targeted attacks (see Section 5.1.2).

5.1.1 Experimental study of untargeted attacks

Within this kind of attacks, we differentiate between: (1) those attacks that modify clients' training data, producing an alteration of the models (data-poisoning attacks) and (2) those that directly modify the weights of the learning models (model-poisoning attacks). In order to provide a variety of experimentation, we choose the following attacks:

- Data-poisoning attacks: Random label-flipping attack and Out-of-distribution attack (see Section 3.1.3). Clearly, to make these attacks effective, we combine them with model-replacement techniques.
- Model-poisoning attacks: Random weights (see Section 3.1.3), which we also combine with model-replacement.

Regarding the ratio of adversarial clients, we considered different distributions in order to analyse the influence on both the performance of the attack and the defences. In particular, we name x-out-of-n the situation where x of the n clients participating in the aggregation are adversarial ones.

We chose as defences those that have been shown to be state of the art in the literature. In particular, we use the following ones (see Section 4.1):

- Median and Trimmed-mean [84].
- Krum and Multi-Krum [87] with different values for the parameter d, which detail the number of client selected. We consider $d = 5$ and $d = 20$.
- Bulyan [88] different values for the parameter f, which determines the tails of the distribution to be filtered. We consider $f = 1$ and $f = 2$.

In Tables 1, 2 and 3 we show the results of assessing the different defences in label-flipping, out-of-distribution data-poisoning attacks and random weights model-poisoning attack, respectively. In the following, we analyse the behaviour of both attacks and defences in each situation from different effectiveness and behaviour of the defences.

Effectiveness of the attack If we compare the effectiveness of the attack as a function of the type of attack, we conclude that the most damaging attack is the random weights attack. In fact, this attack manages to totally confuse the federated model, to the extent that it behaves as a most frequent label classification model. If we focus on the data-poisoning attacks, we get that the label-flipping attack is sightly more effective than the out-of-distribution attack. This is probably because the label-flipping attack learns mislabeled samples from within the distribution, while the out-of-distribution attack, theoretically, only adds error to samples from outside the distribution.

Regarding the ratio of adversarial clients participating in each aggregation, we found that there are significant differences, the most effective one being carried out by a single adversar-

Label-flipping Byzantine data-posioning attack									
	Federated EMNIST			Fashion MNIST			CIFAR-10		
	1-out-of-30	5-out-of-30	10-out-of-50	1-out-of-30	5-out-of-30	10-out-of-50	1-out-of-30	5-out-of-30	10-out-of-50
No attack	**0.965**	**0.965**	**0.962**	0.871	0.871	0.869	0.835	0.835	0.823
FedAvg	0.159	0.421	0.400	0.191	0.366	0.432	0.118	0.143	0.244
Trim.-mean	0.942	0.873	0.837	0.867	0.832	0.861	0.823	0.734	0.822
Median	0.931	0.916	0.909	0.867	0.847	0.858	0.828	0.809	0.828
Krum	0.891	0.870	0.863	0.726	0.719	0.747	0.747	0.761	0.769
MultiKrum (5)	0.913	0.927	0.918	0.840	0.843	0.825	0.816	0.823	0.811
MultiKrum (20)	**0.956**	**0.957**	0.950	**0.872**	**0.872**	0.868	0.843	**0.847**	0.851
Bulyan (f=1)	0.952	0.781	0.580	0.868	0.783	0.787	0.826	0.659	0.645
Bulyan (f=5)	0.936	0.942	**0.951**	0.861	0.865	**0.872**	**0.849**	0.845	**0.854**

Table 1: Mean results for the *label-flipping Byzantine data-poisoning attack* in terms of accuracy. We also show, in the first row, the expected accuracy with *FedAvg* but without any attack.

Out-of-distribution Byzantine data-poisoning attack									
	Federated EMNIST			Fashion MNIST			CIFAR-10		
	1-out-of-30	5-out-of-30	10-out-of-50	1-out-of-30	5-out-of-30	10-out-of-50	1-out-of-30	5-out-of-30	10-out-of-50
No attack	**0.965**	**0.965**	**0.962**	0.871	0.871	0.869	0.835	0.835	0.823
FedAvg	0.409	0.440	0.435	0.204	0.366	0.465	0.146	0.192	0.341
Trim.-mean	0.945	0.860	0.853	0.865	0.834	0.831	0.820	0.744	0.740
Median	0.934	0.920	0.914	0.866	0.846	0.845	0.822	0.801	0.807
Krum	0.869	0.866	0.862	0.736	0.706	0.728	0.720	0.731	0.740
MultiKrum (5)	0.916	0.933	0.919	0.849	0.843	0.834	0.830	0.819	0.802
MultiKrum (20)	**0.954**	**0.954**	**0.950**	**0.874**	**0.871**	0.873	**0.860**	**0.851**	**0.852**
Bulyan (f=1)	0.950	0.787	0.581	0.870	0.760	0.693	0.831	0.686	0.555
Bulyan (f=5)	0.935	0.938	**0.950**	0.871	0.865	**0.875**	0.844	0.849	0.848

Table 2: Mean results for the *out-of-distribution Byzantine data-poisoning attack* in terms of accuracy. We also show, in the first row, the expected accuracy with *FedAvg* but without any attack.

Random weights Byzantine model-poisoning attack									
	Federated EMNIST			Fashion MNIST			CIFAR-10		
	1-out-of-30	5-out-of-30	10-out-of-50	1-out-of-30	5-out-of-30	10-out-of-50	1-out-of-30	5-out-of-30	10-out-of-50
No attack	**0.965**	**0.965**	**0.962**	0.871	0.871	0.869	0.835	0.835	0.823
FedAvg	0.099	0.099	0.100	0.100	0.101	0.099	0.099	0.099	0.100
Trim.-mean	0.953	0.103	0.099	0.875	0.100	0.099	0.860	0.099	0.099
Median	0.936	0.935	0.934	0.865	0.861	0.855	0.849	0.866	0.864
Krum	0.831	0.865	0.854	0.715	0.745	0.734	0.718	0.716	0.799
MultiKrum (5)	0.932	0.922	0.919	0.834	0.834	0.827	0.816	0.811	0.816
MultiKrum (20)	**0.956**	**0.957**	0.951	**0.876**	**0.875**	0.867	0.848	**0.848**	**0.853**
Bulyan (f=1)	0.959	0.099	0.099	0.099	0.100	0.099	0.852	0.099	0.099
Bulyan (f=5)	0.937	0.937	**0.951**	0.874	0.869	**0.874**	**0.850**	0.841	0.851

Table 3: Mean results for the *random weights Byzantine model-poisoning attack* in terms of accuracy. We also show, in the first row, the expected accuracy with *FedAvg* but without any attack.

ial client (1-out-of-30). While this may seem contradictory, there is an explanation. When the attack is carried out by several clients, the boosting factor is divided among these adversarial clients. This divides the strength of the attack among all the adversarial clients, which thus weaken attack power, whereas when carried out by a single client, all the boosting is reflected in a single attacker, making it more effective.

Behaviour of the defences As a general rule, the defences that best mitigate the effect of the attacks are Multikrum (20) and Bulyan (f=5), with MultiKrum (20) standing out slightly. As we have shown, although Bulyan is presented as an improvement of MultiKrum in combination with trimmed-mean, if the pre-selected clients are benign clients, this truncation is not necessary and even superfluous. On the other hand, the more basic defences such as median and trimmed-mean show good enough behaviour in some experiments, even outperforming MultiKrum and Bulyan with some parameters.

This superiority of the most basic defences over MultiKrum and Bulyan with specific parameter values evidences the high dependence of these defences on the values of the input parameters. This behaviour matches with the assertion of the authors of MultiKrum and Bulyan that they are the most robust defences with the optimal value of the input parameters. This dependency on the values of the input parameters represents an obstacle for the use of these defences, since the value of some parameters is difficult to know, e.g. the number of adversarial clients. A clear example of this problem is Bulyan (f=1) in the random weights Byzantine model-poisoning attack, whose results are comparable to using no defence at all by filtering out too few adversarial clients.

To conclude, untargeted attacks are highly effective, especially those based on model-poisoning, which achieve random behaviour in the federated model. The defences proposed in the literature perform reasonably well, substantially improving the effect of the attacks, even the simplest ones. However, none of them manage to completely dissipate the attack, and the best-performing ones are highly dependent on configuration parameters, so there is still room for improvement in designing defences against Byzantine attacks.

5.1.2 Experimental study of targeted attacks

In order to make a sufficiently broad experimental study, in this section we consider backdoor attacks from the two main groups presented: (1) Input-instance-key strategies and (2) pattern-key strategies. With respect to attacks implementing input-instance-key strategies, we perform a single attack where the target samples correspond to some samples belonging to the adversarial clients for each dataset and associate them with a specific target label. However, with respect to the pattern-key attacks, we choose a different static for each dataset, single and accessory injection pattern.

We chose the state of the art against Backdoor attacks as baselines. In particular, we use the following ones (see Section 5.1.1):

- Median and Trimmed-mean [84].
- Norm-clipping [42].

- Weak Differential Privacy (Weak DP) Sun et al. [42].
- Robust Learning Rate (RLR) Ozdayi et al. [18].

For these defences based on clipping and noise addition, we use M and σ to specify both the clip factor and the noise added, respectively. For the experiments, we choose the values recommended by the authors.

Study of Input-instance-key attacks In Table 4 we show the results obtained after testing the input-instance-key attack and the different defences. For the implementation, we randomly select some samples of the adversarial clients and associate them with the target label "0". We evaluate the effectiveness of the attack, showing both the original and backdoor performances. We measure the original performance using the mean accuracy in the original test dataset and the backdoor performance by means of the mean accuracy in the set of selected samples for the attack.

			Input-instance backdoor attack					
			Federated MNIST		**Fashion MNIST**		**CIFAR-10**	
	M	σ	**Original**	**Backdoor**	**Original**	**Backdoor**	**Original**	**Backdoor**
No attack	0	0	**0.965**	-	0.871	-	0.835	-
FedAvg	0	0	0.866	0.823	0.804	0.944	0.612	0.903
Median	0	0	0.944	0.030	**0.875**	0.032	0.861	0.140
Trim.-mean	0	0	0.952	0.025	0.872	0.016	**0.863**	0.133
NormClip	3	0	**0.960**	0.876	0.863	0.144	0.843	0.115
Weak DP	3	2.5e-3	0.937	0.157	0.843	0.119	0.823	0.093
RLR	0.5/0.5/1	1e-4	0.954	**0.012**	0.863	**0.002**	0.853	**0.014**

Table 4: Mean results for the input-instance backdoor attack in terms of accuracy. We also show, in the first row, the expected accuracy with *FedAvg* but without any attack.

If we first analyse the effectiveness of the attack (see row of *FedAvg* and *Backdoor* columns) we find the attack is relatively effective, with the result in Fashion MNIST standing out, and always being higher than 0.82 accuracy. However, if we focus on the stealthiness we note that this type of attack lacks this valuable quality, even affecting the performance on the original task (see row of *FedAvg* and *Original* columns) in 22 points of accuracy (CIFAR-10).

Regarding the performance of the defences, we find that every one of the defences leads to a substantial improvement, both increasing the original task accuracy and reducing the backdoor task accuracy. In addition, we would like to highlight the good performance of the simpler defences, such as trimmed-mean, which achieves very competitive results. If we analyse the state-of-the-art defences (Weak DP and RLR), we found the results to be appropriate, but perhaps a mite disappointing on a complexity-performance trade-off compared to the other defences. Moreover, there are likely to be other M and σ parameters that optimize the results of these defences, but they are not known in advance, which is the main weakness of such parameter-dependent defences.

To conclude, input-instance-key backdoor attacks are considerably powerful, performing better in the backdoor task than in the original one, but being too eye-catching and detrimental to the original task. Moreover, although the defences mitigate the effect of the attack, none of them completely dissipate it, so there is still plenty of scope for further research.

Study of the pattern-key attacks Table 5 shows the results obtained after testing different pattern-key attacks with the considered defences. We implement the attacks by randomly selecting the adversarial clients and poisoning some of their samples with different patterns. In particular, we use the following patterns of different levels of difficulty according to the number of pixels: (1) one single black pixel for Federated MNIST, (2) a red cross of length 4 for Fashion MNIST (8-pixel pattern) and (3) a white pixel in each of the corners of the image (4-pixel pattern) for CIFAR-10. We evaluate both the effectiveness and the stealthiness of the attack. We measure the stealthiness of the attack by means of the mean accuracy obtained in the original task (Original). We also evaluate the effectiveness of the attack in terms of two additional tests: (1) Backdoor, which contains the poisoned samples of the adversarial clients and (2) Test, which represents the test of the backdoor task and is composed of test samples poisoned following the specific pattern. Therefore, an attack will be more effective the higher the performance it obtains in both the original and the backdoor task, while a defence will be better if it manages to maintain the performance in the original task while decreasing the performance in the backdoor task as much as possible.

Pattern-key backdoor attack											
			Federated MNIST			**Fashion MNIST**			**CIFAR-10**		
	M	σ	**Original**	**Backdoor**	**Test**	**Original**	**Backdoor**	**Test**	**Original**	**Backdoor**	**Test**
No attack	0	0	0.965	-	-	0.871	-	-	0.835	-	-
FedAvg	0	0	0.974	1.0	1.0	0.843	0.999	0.944	0.413	1.0	0.99
Median	0	0	0.954	0.009	0.015	**0.873**	0.067	0.053	0.854	0.193	0.183
Trim.-mean	0	0	0.966	0.011	0.014	0.872	0.052	0.065	0.853	0.194	0.170
NormClip	1	0	**0.968**	0.055	0.053	0.843	0.143	0.164	0.834	0.143	0.131
Weak DP	1	2.5e-3	0.935	0.093	0.0175	0.869	0.053	0.074	**0.859**	0.144	0.170
RLR	1	1e-4	0.962	**0.008**	**0.008**	0.870	**0.020**	**0.031**	0.856	**0.073**	**0.061**

Table 5: Mean results for the *pattern-key backdoor attack* in terms of accuracy. We also show, in the first row, the expected accuracy with *FedAvg* without any attack. The best result for each of the test sets is highlighted in bold.

Regarding the effectiveness of the attack without any defence (see row of the *FedAvg* and *Backdoor* and *Test* columns), it reaches a performance of 100% or close to it, which shows it harmfulness. However, if we analyse the stealthiness of the attack (see row of the *FedAvg* and *Original* columns), the conclusions depend on the dataset. While in Federated MNIST and Fashion MNIST the performance in the original task is maintained or even improved, the performance in the original task in CIFAR-10 is reduced by up to half.

Regarding the behaviour of the defences, we also obtain a substantial improvement with respect to the scenario without any defence with all of them. As in the untargeted attacks, the simplest defences obtain competitive results, even outperforming the most complex defences in some situations. In general, deciding which defence is superior is not a trivial task.

Since it is a matter of achieving the best trade-off between performance in the original task and dissipation of the backdoor attack. For example, RLR achieves in Federated EMNIST the best defence against the attack, but it is more detrimental to performance on the original task. However, in general, we can affirm that it is the best performing defence, standing out particularly in CIFAR-10.

To conclude, pattern-key backdoor attacks are highly threatening attacks, as they achieve almost 100% success in the backdoor task, without, in most cases, harming the performance of the original task. Defences manage to dissipate the effect of the attack in the backdoor task, but in most cases impair performance in the original task. Therefore, in this case, the key is to find the trade-off between mitigating the attack and not harming the performance of the model.

5.2 Privacy attacks

Even tough there is a wide range of privacy attacks, in this section we study those which meet the following requirements:

1. The attack is performed while the federated model is being trained. As a consequence, most defences are aimed to make the training secure from privacy attacks. Alternatively, the defences mask or perturb the shared information to make it less vulnerable.
2. The description of the attack and its setup in its publication is enough to implement it or an implementation which matches the publication is publicly available. The same applies for defences.

The found privacy attacks that matched our requirements allowed us to divide this section into the following two subsections, corresponding to Membership inference attacks (see Section 5.2.1) and Feature inference attacks (see Section 5.2.2), restricted to HFL scenarios.

5.2.1 Experimental study of Membership inference attacks

We choose to implement the federated white-box Membership inference attack from Nasr et al. [40] using the source code publicly available for the white-box centralized setting[6] as there is no public implementation of the federated version. Both clients and server can be the attacker. On the one hand, when the attacker is the client, her objective is to infer the membership of data points belonging to other clients. On the other hand, when the attacker is the server every client is attacked individually, thus the objective is to infer the membership of data points for each client. We mainly focus on their server side attack or global attacker as it is the most powerful, that is, it poses the highest threat to privacy.

We make our federated scenarios the same as the ones proposed in Nasr et al. [40], which represents a small population of clients with big amounts of sensitive data such as banks or hospitals, willing to jointly train a privacy preserving deep learning model. As each client owns

[6] https://github.com/privacytrustlab/ml_privacy_meter

great quantities of data, some records can be duplicated among them, that is, the dataset owned by each client is sampled uniformly with replacement from the following datasets: EMNIST, Fashion MNIST and CIFAR-10. Consequently, each of them is divided between 4 clients and each client owns a sample of half the size of the entire dataset, sampled with replacement.

Each federated task is run for 300 rounds where each client shares her local model after each local training epoch and the attacker observes the rounds: 50, 100, 200, 250 and 300. The attack is trained for 100 epochs and the model with best testing accuracy is selected. The attacker training dataset is made of 4000 random samples belonging to each attacked client, 4000 random samples which do not belong to any client and each one is labelled according to its membership to the attacked client. For all the experiments, the batch size is set to 32. We highlight that this federated setup is taken from Nasr et al. [40].

We report the average accuracy and AUC of the global attacker in the described settings in Table 6. Note that, the membership inference attack is performed by a binary classifier, therefore the choice of the classification threshold is key to separate between member and non member instances. An attacker with background knowledge may have the ability of selecting a classification threshold that maximizes the separation between member and non members, leading to a greater privacy leakage [162]. While the authors of the attack focus on reporting accuracy, we have found in our experiments that the AUC metric better shows the capabilities of the attacker, due to the fact that AUC is independent of the classification threshold used to perform the inference. This decision is also driven by the fact that a single classification threshold only represents a possible attacker, therefore we need a way of evaluating every possible attacker, including those with great amounts of background knowledge. We can observe that in our experiments the attack is barely effective, as both accuracy and AUC are close to 0.5. We also highlight that the Gradient Ascent technique does not bring significant performance improvements, probably because it is hard to calibrate. While in the MNIST dataset we see that the attack is not successful, in the other the membership of some instances is revealed, so there is a privacy leak, although it is very small.

We also report the success of the attack with the state-of-the-art defence Local DP in Table 6. The privacy budget in each client of the Local DP is $\epsilon = 3, \delta = 10^{-5}$, which is considered in the literature to be a high privacy budget. We employ the AutoDP framework[7] to calibrate the differentially private Gaussian noise to the privacy budget using Renyi DP [163]. We can observe that this defence is quite successful as it avoids leaking any membership information, thus making the attack classifier behave randomly.

In Table 7, we can see the accuracy of the federated task with the attack. As noted before, the Gradient Ascent technique degrades the accuracy of the federated task, mainly due to the fact that some of the instances which were absent belong to the federated test set. While this is true for the MNIST and Fashion MNIST datasets, it is not true for the CIFAR-10 dataset. It might be because of the transfer learning approach used for this dataset being more resilient to gradient direction changes. As expected, DP based defences reduce the accuracy

[7] `https://github.com/yuxiangw/autodp`

Membership Inference attack								
	Without Local DP defence				With Local DP defence			
	Client Isolation		Client Isolation + Gradient Ascent		Client Isolation		Client Isolation + Gradient Ascent	
	Accuracy	AUC	Accuracy	AUC	Accuracy	AUC	Accuracy	AUC
MNIST	0.501	0.502	0.489	0.502	0.500	0.497	0.496	0.500
Fashion MNIST	0.513	0.546	0.511	0.516	0.500	0.499	0.497	0.500
CIFAR-10	0.540	0.551	0.500	0.528	0.500	0.500	0.500	0.500

Table 6: Accuracy and AUC of the global federated attack from Nasr et al. [40] with and without Local DP defence.

of the federated task. The smallest reduction of federated task accuracy is achieved with the CIFAR-10 dataset, which confirms that the transfer learning approach is more resilient to gradient changes, moreover the Gradient Ascent technique does not change significantly the accuracy when applied.

Membership inference attack				
	Without Local DP defence		With Local DP defence	
	Client Isolation	Client Isolation + Gradient Ascent	Client Isolation	Client Isolation + Gradient Ascent
MNIST	0.990	0.100	0.672	0.100
Fashion MNIST	0.910	0.100	0.579	0.100
CIFAR-10	0.862	0.862	0.686	0.668

Table 7: Federated task's accuracy while the global federated attack from Nasr et al. [40] is performed with and without Local DP defence.

In this experimental study, we have explored the performance of a Membership inference attack on a federated setting of few clients with big amounts of data. We have found that the success of the attack is small, even though the membership of some instances was revealed. The DP based defence stopped these privacy leaks, at the cost of a considerable reduction of the federated task accuracy. Additionally, we have found that using a transfer learning approach might reduce the impact of DP in the federated task accuracy while also being resilient to the Gradient Ascent technique which has drastically reduced the federated task accuracy with the other datasets and deep learning approaches.

5.2.2 Experimental study of Feature inference attacks

We study multiple gradient based Feature inference attacks, which use stolen gradients from the federated training procedure. Particularly we focus on the attacks described in Zhu and Han [15], Geiping et al. [68], Wei et al. [70]. In order to do it, we use the code provided with each publication, which is publicly available.[8,9,10]

[8] https://github.com/mit-han-lab/dlg

[9] https://github.com/JonasGeiping/invertinggradients

[10] https://github.com/git-disl/CPL_attack

The federated scenario which fits these attacks is the following: clients with very little data, such as IoT devices or smartphones, which run a federated task where they share gradients from small batches. We study under which circumstances we can reconstruct images from gradients. Our study focuses on three aspects to evaluate the success of these attacks:

- **Success rate**. The approximate probability of convergence of each attack. The majority of the attacks studied in this section are known to have stability issues, that is, their convergence greatly depends on the initialization seed used to bootstrap them. For Wei et al. [70] and Zhu and Han [15], we choose as initialization a geometric pattern which improves both convergence rate and speed. It consists in covering a small portion of the initialization space with a random image and duplicate it to fill the feature space. In our experiments, we choose 1/4 of the feature space as in Wei et al. [70]. For the attack of Geiping et al. [68], we choose random initialization, as it does not seem to be affected by the choice of the initialization pattern.

- **Training stage of the local model at which the attack can succeed**. Most of the studied attacks consider an untrained model as they claim that the attack can run at any point of the training procedure, however this claim does not seem to have a lot of experimental support. As a consequence, we want to confirm such claims and find whether the stage of training of the local model is relevant to the success of the attack.

- **Success of the defences against Feature inference attacks**. We study the performance of two state-of-the-art defences: gradient compression and the addition of Gaussian noise, which are known to thwart the effectiveness of the attack from Zhu and Jin [164], so we evaluate whether these defences are also applicable to the other attacks.

We begin our study analysing the success rate of each attack, as they are known to suffer from stability issues [70]. We run each attack with gradients from an untrained simple convolutional model *LeNet* [165] as in [70, 15] with a batch size of 1. Each attack is run until one of the following conditions is satisfied:

- Success condition: for the attacks [70, 15], we consider that the attack is successful if the Mean Square Error (MSE) with respect to the target image to reconstruct is smaller than 0.5 and the Multi-Scale Structural Similarity (SSIM) [166] is greater than 0.5. The purpose of these criteria is twofold, the former ensures that the reconstructed image is close enough to the target image and the latter ensures that the reconstructed image is perceptibly similar to the target image.

- Failure condition: if the maximum number of epochs set for the attack is reached without satisfying any of the conditions stated above, then we consider the attack is marked as a failure. In other words, the attack failed to converge.

Additionally, we want to study whether the training stage at which the attack is performed is relevant. To do so, we run each attack at different moments of the local training process: before any training, after 1, 5 and 10 rounds of training. Each attack is going to try to reconstruct an image that belongs to their training set, but it has not been used to train the model previously. We report the success rate of each attack across 25 runs, using the same end conditions specified before.

The experimental results of the study of **the success rate** and **the training stage of the local model at which the attack can succeed** are shown in Tables 8, 9 and 10. First, we highlight the results from Table 10 that show that the attack from Geiping et al. [68] is independent of the considered training stage of the local model. The same is not true for the results in Tables 8 and 9. In its first column of results, we can see that the attacks have almost no issues to converge when the local model is not trained, so we can conclude that if the appropriate initialization method is chosen, the attacks are almost 100% guaranteed to converge. If we observe the remaining columns of the Tables 8 and 9, the results change considerably. The attack from Wei et al. [70] (Table 9) has slightly better convergence rates than the attack from Zhu and Han [15] (Table 8), both show a similar trend: the more trained the model is, the harder it is for the attacks to achieve success.

The complexity of the dataset has an important role in the success of the attacks from [70] and Zhu and Han [15]. Both EMNIST and Fashion-MNIST are considered easier datasets, as there are many works that achieve high training accuracy after few epochs of training [167, 168]. The same is not true for CIFAR-10, as more complex models are required to achieve a reasonable accuracy [169, 170]. EMNIST and Fashion-MNIST images are hard

Feature inference attack				
Dataset	Before training	After 1 training epoch	After 5 training epochs	After 10 training epochs
EMNIST	1	0	0.04	0
Fashion-MNIST	1	0.28	0	0.08
CIFAR-10	0.96	0.80	0.60	0.68

Table 8: Success rate of 25 trials of reconstructing an image from a shared gradient of a local model with the attack from Zhu and Han [15]. We run the attack at different stages of training of the local model. *Before training* means that the local model has not been trained at all.

Feature inference attack				
Dataset	Before training	After 1 training epoch	After 5 training epochs	After 10 training epochs
EMNIST	1	0	0	0
Fashion-MNIST	1	0.32	0.12	0.24
CIFAR-10	1	0.96	0.84	0.80

Table 9: Success rate of 25 trials of reconstructing an image from a shared gradient of a local model with the attack from Wei et al. [70]. We run the attack at different stages of training of the local model. *Before training* means that the local model has not been trained at all.

	Feature inference attack			
Dataset	Before training	After 1 training epoch	After 5 training epochs	After 10 training epochs
EMNIST	1	1	1	1
Fashion-MNIST	1	1	1	1
CIFAR-10	1	1	1	1

Table 10: Success rate of 25 trials of reconstructing an image from a shared gradient of a local model with the attack from Geiping et al. [68]. We run the attack at different stages of training of the local model. *Before training* means that the local model has not been trained at all.

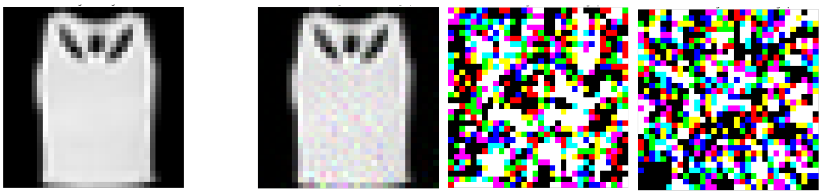


Figure 11: From left to right, reconstruction using the attack of Wei et al. [70] of an image with label 0 from Fashion-MNIST dataset which correspond to the t-shirt/top category, after 1, 5 and 10 epochs of local training.

to reconstruct after 1 training epoch, that is, the gradients after 1 training epoch leak little information about the datasets. An example of such difficulties is shown in Figure 11. In contrast, in CIFAR-10 the training model takes longer to converge and gradients leak a lot of information, even after 10 epochs of training. The main reason that allows us to understand this behaviour is the fact that both attacks try to mimic the structure and content of the shared gradient (that is, minimizing the MSE between the shared gradient and the reconstructed image), so the more information is stored in the gradient, the easier the reconstruction process is. In other words, gradients that more significantly change the weights of the model make the reconstruction process easier. This is not true for the attack from [68], as its objective is to minimize the cosine similarity between gradient vectors.

To end our study, we study the performance of two state-of-the-art defences:

- Gradient compression with 20% sparsity.
- The addition of Gaussian noise with variance of 10^{-2}.

We run each attack with defences 25 times with a batch size of 1, with the model untrained and report the success rate of each attack.

In Table 11, we can observe the stunning performance of both defences as they completely stop the attacks of [15, 70] from achieving success. While the addition of Gaussian noise of this magnitude is known to reduce the performance of the task [15], the gradient compression defence can handle higher compression rates without significantly hurting performance [171]. When it comes to the attack of Geiping et al. [68], we find that the Gaussian

	Feature inference attack					
	Attack of Zhu and Han [15]		Attack of Wei et al. [70]		Attack of Geiping et al. [68]	
Dataset	Gaussian noise	Gradient compression	Gaussian noise	Gradient compression	Gaussian noise	Gradient compression
EMNIST	0	0	0	0	0	0.04
Fashion-MNIST	0	0	0	0	0	0.48
CIFAR-10	0	0	0	0	0	0.12

Table 11: Success rate of 25 trials of the reconstruction attacks from [15], [70] and Geiping et al. [68] with Gaussian noise and Gradient compression defences.

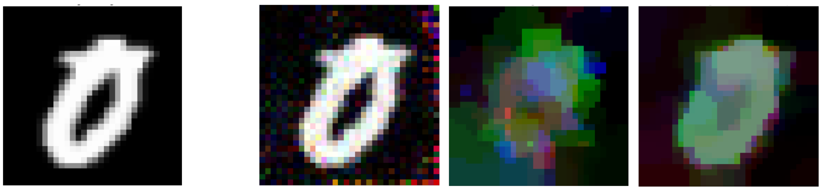


Figure 12: From left to right, reconstruction using the attack of Geiping et al. [68] of an image with label 0 from EMNIST dataset, without any defence, with Gaussian noise defence and with Gradient compression defence.

noise defence is as effective as in the other attacks. This might be due to the differentially private properties of the Gaussian noise. However, the Gradient compression defence fails to completely stop the attack of Geiping et al. [68]. Specially for the Fashion-MNIST dataset, where almost half of the times the attack succeeded. An example of a reconstruction trial with and without defences is shown in Figure 12, and gives a hint of the behaviour of the attack. Gradient compression is the worst performing defence, probably due to the fact that compressing the gradient does not affect the task of minimizing the cosine similarity between the shared and the reconstructed image gradient.

In conclusion, the Feature inference attacks studied in this section pose a high risk to privacy, as there are many attacks that succeed at extracting private information from gradients. Luckily, there are defences that can thwart the success rate of the attacks and provide privacy without significantly changing performance of the trained model. However, this is not true for all the analysed attacks, there is still room for improvement as the attack from Geiping et al. [68] seems to be able to escape them in some situations. Additionally, this threat is not only related to FedSGD scenarios, it is also related to federated scenarios where parameters are exchanged between clients.

6 Guidelines for the application of defences against adversarial attacks

Due to the large number of attacks identified, and the wide variety of defences proposed in the literature, it can be difficult to choose which type of defence is appropriate for each situation. Moreover, most defences are designed with the objective of defending against a particular adversarial attack. However, as a collateral benefit, they can prevent the success of other types of adversarial attacks.

In this section we provide some guidelines in terms of a summary of which categories of defences work to defend the identified categories of attacks, specifying the degree to which they do so.

In Table 12 we summarize which categories of defences are able to defend against attacks to the model and privacy attacks, respectively. For the sake of clarity, we represent the relationship between categories of attacks and categories of defences. Hence, when we affirm that a category of defence is able to defend against a category of attacks, this means that there is at least one defence belonging to that category which is able to defend against them.

In this line, we differentiate between:

- Complete defence 🟢: the defence category is able to stop the attacks of the attack category.
- Partial defence 🟠: the defence category is able to significantly reduce the performance of the attacks of the attack category but not stop them.
- No defence 🔴: the performance of the attacks of the attack category is not affected significantly by the defence category.
- Unknown defence ●: there is neither enough experimentation available nor theoretical support to assess the behaviour of the attacks of the attack category with the defence category.

The summary of the state of the art provided in the Table 12 allows us to draw the following conclusions:

1. In general, defences based on DP, which are designed to defend against privacy attacks, partially defend against attacks to the model, specially those based on DP, but not the other way around.
2. Broadly speaking, the defence against attacks to the model is more settled than the defences against privacy attacks. In particular, for property inference attacks, there is no defence considered as complete.
3. There is still a long way to go in designing defences to prevent attacks in FL and, crucially, to find a defence that prevents all types of attacks at the same time.

		Attacks to the federated model		Privacy attacks		
		Untargeted	Targeted	Property	Membership	Feature
Server defences	Mod. of learning rate	🟢	🟢	🔴	🔴	🔴
	Robust agg.	🟢	🟢	🔴	🔴	🔴
	Anomaly detection	🟢	🟢	🔴	🔴	🔴
	Based on DP	🟡	🟡	🟡	🟢	🟢
	Less is more	🟢	🟡	🔴	🔴	🔴
Client defences	Based on DP	🟡	🟡	🟡	🟢	🟢
	Optimized training	🟢	🟡	⚪	⚪	⚪
	Perturbations methods	⚪	⚪	⚪	🟢	🟢
Comm. channel	Blockchain	🟡	🟡	🟡	🟡	🟡
	SMPC	🟡	🟡	🟡	🟡	🟡

Table 12: Most recommendable defence methods to attacks to the federated model and privacy attacks, respectively.

7 Lessons learned

Based on the extensive research and analysis of the available works, we have built up the taxonomy proposed in this paper. However, what has been learned goes beyond this. To sum up, the lessons learned are:

1. The identification of vulnerabilities in the form of adversarial attacks and the proposal of defences against them in FL is a field of research in continuous development. The volume to date of scientific work covering these challenges is growing and is not likely to diminish in the coming years.

2. Attacks to the federated model are easier to defend against than privacy attacks. However, they have shown much greater effectiveness, with even the simplest attacks being detrimental to the model.

3. Privacy attacks require very peculiar settings to achieve a reasonable success, that is, most of the assumptions required to perform them are very hard to achieve in real FL scenarios. Such scenarios are usually bounded by the lack of the following resources: data, raw computing power and time.

4. Most defences against inference attacks, although designed for them, dissipate the performance of attacks against the federated model, but not the other way around. Therefore, the use of DP-inspired mechanisms will be crucial if we want to defend against generic category attacks.

5. The implementation of defences based on DP and based on perturbation methods require extensive fine-tuning in order to provide a nice trade-off between privacy and

performance. Most of the defences require access to big computational resources, or they are too slow to apply. Therefore, such defences might not be suitable for FL settings with low power devices. Additionally, to our knowledge, there is not a consensus about how to measure the trade-off between privacy and performance.

To finish, a fundamental lesson learned is that the field of adversarial attacks and defences in FL is a research area in steady development, which is not expected to change in the forthcoming years. There are still many vulnerabilities which need to be faced in order to ensure a truly secure and privacy-preserving learning environment.

8 Challenges of addressing federated learning threats

Regarding the previous lessons learnt, we identified the following challenges that the field will have to face up in the next years.

Defences vs. attacks An identified trend is that for each defence proposed, it is possible to identify a vulnerability that can be turned into an adversarial attack, and vice versa. Therefore, one of the biggest challenges is to find both: (1) all vulnerabilities present in a FL scenario that an attacker could exploit, and (2) a defence effective enough to defend against any attack. For the time being, this seems a long way off, as the different perspectives from which both problems have been approached are ad hoc to the type of attack to be identified or defended. From our point of view, the study of defences is crucial, since the final goal is to achieve a secure, robust and private learning environment. Along this vein, the optimal defence will be the one that combines the best proposals in each of the categories, in such a way that it manages to defend against all types of attacks while maintaining performance in the original task. There are existing approaches that combine client's filtering with noise addition [18], although there is still a long way to go.

Trade-off in defences In most defences, we find that it is difficult to strike a trade-off between preventing the model from attacking and not impairing performance in the original task. For example, in those based on DP, we find that in order to ensure data privacy, a large amount of noise has to be added, which significantly impairs the performance of the model [103]. Therefore one of the main challenges would be the development of more efficient DP methods, and the extension of DP to defences against all adversarial attacks. This situation also occurs in defences based on client filtering when more clients than necessary are filtered out, thus losing information in the aggregation process. In short, it is difficult to strike a trade-off between preventing an attack and not losing or poisoning the information received by clients.

Non-IID assumptions The non-IID nature of the training data distributed among clients often makes it difficult to differentiate between adversarial clients and those who have had a very different from the rest, but still valuable, learning process. One common approach is to use anomaly detection algorithms suitable for non-IID distributions [172] or approaches which do not rely on data distribution [107], however, there are still problems in differentiating between clients with a highly skewed distribution and adversarial clients in most cases.

Generalised FL The vast majority of adversarial attacks have been identified for HFL. In particular, the adversarial attacks to the federated model. Although there is already existing work on privacy attacks in VFL [81], there is still a long way to go in identifying and analysing the vulnerabilities in terms of leakage of information of attacker possibilities in other categories of FL wich are becoming widely used such as VFL or FTL [173]. Therefore, we believe that in the coming years, work on identifying attacks for VFL and FTL and the research in defences against them will take centre stage.

Combination with other trends While ensuring data privacy is one of the main goals of FL, there are other desirable features. For example, some of the most popular trends are Personalized FL [174] or fairness in FL [175]. We believe that, at the end of the day, data security and privacy must be a requirement in all other approaches. Therefore, several future works will address this issue as a cross-cutting objective while developing proposals for more concrete desirable features. For example, a method of personalized FL that is secure against adversarial attacks.

9 Conclusions

FL emerges as a solution to the computational costs and privacy-preserving demands of the most groundbreaking ML. However, this new learning paradigm brings new challenges, particularly in terms of adversarial attacks and defending against them. Hence, several proposals of new adversarial attacks or adaptations of centralised ones as well as defences ad hoc to these attacks have been proposed in the recent years.

We proposed several taxonomies according to different criteria that eases the knowledge of the wide field of FL threats. In addition, we conducted an extensive experimental study which leads us to propose guidelines for the application of defences against adversarial attacks, and to highlight a set of lessons learned and challenges related to FL threats.

We conclude that the study of FL threats is an ongoing field of research, due to its importance in ensuring FL as a robust machine learning paradigm that safeguards data privacy. There are still several challenges to be faced and directions to be studied in order to identify additional threats (or vulnerabilities) of FL, as well as the appropriate mechanisms to make it a resilient and robust learning paradigm against those threats.

Acknowledgments

This research work is partially supported by the R&D&I, Spain grants PID2020-119478GB-I00, PID2020-116118GA-I00 and EQC2018-005084-P funded by MCIN/AEI/10.13039/501100011033. Nuria Rodríguez Barroso were supported by the grant FPU18/04475 funded by MCIN/AEI/10.13039/501100011033 and by “ESF Investing in your future”, Spain.

References

[1] Saif Al-Kuwari. *Multiple Perspectives on Artificial Intelligence in Healthcare: Opportunities and Challenges*, chapter Privacy-Preserving AI in Healthcare, pages 65–77. Springer International Publishing, 2021.

[2] Madhura Joshi, Ankit Pal, and Malaikannan Sankarasubbu. Federated learning for healthcare domain - pipeline, applications and challenges. *ACM Trans. Comput. Healthcare*, apr 2022.

[3] Frederic Boissay, Torsten Ehlers, Leonardo Gambacorta, and Hyun Song Shin. *The Palgrave Handbook of Technological Finance*, chapter Big Techs in Finance: On the New Nexus Between Data Privacy and Competition, pages 855–875. Springer International Publishing, 2021.

[4] 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.

[5] Oihane Gómez-Carmona, Diego Casado-Mansilla, Frank Alexander Kraemer, Diego López de Ipiña, and Javier García-Zubia. Exploring the computational cost of machine learning at the edge for human-centric internet of things. *Future Generation Computer Systems*, 112:670–683, 2020.

[6] Jing Zhang and Dacheng Tao. Empowering things with intelligence: A survey of the progress, challenges, and opportunities in artificial intelligence of things. *IEEE Internet of Things Journal*, 8(10):7789–7817, 2021.

[7] Tanweer Alam and Ruchi Gupta. Federated learning and its role in the privacy preservation of iot devices. *Future Internet*, 14(9), 2022.

[8] C. Ma, J. Konečný, M. Jaggi, V. Smith, M.I. Jordan, P. Richtárik, and M. Takáč. Distributed optimization with arbitrary local solvers. *Optimization Methods and Software*, 32(4):813–848, 2017.

[9] Qiang Yang, Yang Liu, Yong Cheng, Yan Kang, Tianjian Chen, and Han Yu. *Federated Learning*. Synthesis Lectures on Artificial Intelligence and Machine Learning. 2019.

[10] Ling Huang, Anthony D. Joseph, Blaine Nelson, Benjamin I.P. Rubinstein, and J. D. Tygar. Adversarial machine learning. *Proceedings of the 4th ACM Workshop on Security and Artificial Intelligence*, page 43–58, 2011.

[11] Luciano Baresi, Giovanni Quattrocchi, and Nicholas Rasi. Open challenges in federated machine learning. *IEEE Internet Computing*, pages 1–12, 2022.

[12] Nilesh Dalvi, Pedro Domingos, Mausam, Sumit Sanghai, and Deepak Verma. Adversarial classification. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, page 99–108, 2004.

[13] 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, 2020.

[14] Minghong Fang, Xiaoyu Cao, Jinyuan Jia, and Neil Zhenqiang Gong. Local model poisoning attacks to byzantine-robust federated learning. In *USENIX Security Symposium*, pages 1605–1622, 2020.

[15] Ligeng Zhu and Song Han. Deep leakage from gradients. In *Federated learning*, pages 17–31. Springer, 2020.

[16] Wanrong Zhang, Shruti Tople, and Olga Ohrimenko. Leakage of dataset properties in Multi-Party machine learning. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2687–2704, 2021.

[17] Jungwuk Park, Dong-Jun Han, Minseok Choi, and Jaekyun Moon. Sageflow: Robust federated learning against both stragglers and adversaries. In *Thirty-fifth Conference on Neural Information Processing Systems (NeurIPS)*, 2021.

[18] Mustafa Safa Ozdayi, Murat Kantarcioglu, and Yulia R. Gel. Defending against backdoors in federated learning with robust learning rate. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 9268–9276, 2021.

[19] Jingwei Sun, Ang Li, Binghui Wang, Huanrui Yang, Hai Li, and Yiran Chen. Soteria: Provable Defense Against Privacy Leakage in Federated Learning From Representation Perspective. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9311–9319, 2021.

[20] Jingwei Sun, Ang Li, Louis DiValentin, Amin Hassanzadeh, Yiran Chen, and Hai Li. FL-WBC: enhancing robustness against model poisoning attacks in federated learning from a client perspective. In *Thirty-fifth Conference on Neural Information Processing Systems (NeurIPS)*, 2021.

[21] A.N. Bhagoji, S. Chakraborty, P. Mittal, and S. Calo. Analyzing federated learning through an adversarial lens. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, volume 97, pages 634–643, 2019.

[22] Yann Fraboni, Richard Vidal, and Marco Lorenzi. Free-rider attacks on model aggregation in federated learning. In *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130, pages 1846–1854, 2021.

[23] Virat Shejwalkar, Amir Houmansadr, Peter Kairouz, and Daniel Ramage. Back to the drawing board: A critical evaluation of poisoning attacks on federated learning. In *43rd IEEE Symposium on Security and Privacy*, 2022.

[24] David Enthoven and Zaid Al-Ars. An Overview of Federated Deep Learning Privacy Attacks and Defensive Strategies. *Studies in Computational Intelligence*, 965:173–196, 2021.

[25] Muhammad Asad, Ahmed Moustafa, and Chao Yu. A critical evaluation of privacy and security threats in federated learning. *Sensors (Switzerland)*, 20(24):1–15, 2020.

[26] 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.

[27] L. Lyu, Han Yu, Xingjun Ma, Lichao Sun, Jun Zhao, Qiang Yang, and Philip S. Yu. Privacy and Robustness in Federated Learning: Attacks and Defenses. *CoRR*, abs/2012.06337, 2020.

[28] Battista Biggio, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim Šrndić, Pavel Laskov, Giorgio Giacinto, and Fabio Roli. Evasion attacks against machine learning at test time. In *Machine Learning and Knowledge Discovery in Databases*, pages 387–402, 2013.

[29] Lingjuan Lyu, Han Yu, Jun Zhao, and Qiang Yang. Threats to Federated Learning. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 12500 LNCS:3–16, 2020.

[30] Malhar S. Jere, Tyler Farnan, and Farinaz Koushanfar. A Taxonomy of Attacks on Federated Learning. *IEEE Security and Privacy*, 19(2):20–28, 2021.

[31] Nader Bouacida and Prasant Mohapatra. Vulnerabilities in Federated Learning. *IEEE Access*, 9:63229–63249, 2021.

[32] Elena Fedorchenko, Evgenia Novikova, and Anton Shulepov. Comparative review of the intrusion detection systems based on federated learning: Advantages and open challenges. *Algorithms*, 15(7), 2022.

[33] Q. Yang, Y. Liu, T. Chen, and Y. Tong. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology*, 10(2):12:1–12:19, 2019.

[34] Nuria Rodríguez-Barroso, Goran Stipcich, Daniel Jiménez-López, José Antonio Ruiz-Millán, Eugenio Martínez-Cámara, Gerardo González-Seco, M. Victoria Luzón, Miguel Ángel Veganzones, and Francisco Herrera. Federated learning and differential privacy: Software tools analysis, the Sherpa.ai FL framework and methodological guidelines for preserving data privacy. *Information Fusion*, 64:270 – 292, 2020.

[35] Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.

[36] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography*, pages 265–284, 2006.

[37] Stacey Truex, Ling Liu, Mehmet Emre Gursoy, Lei Yu, and Wenqi Wei. Demystifying membership inference attacks in machine learning as a service. *IEEE Transactions on Services Computing*, 2019.

[38] Clement Fung, Chris J. M. Yoon, and Ivan Beschastnikh. The limitations of federated learning in sybil settings. In *23rd International Symposium on Research in Attacks, Intrusions and Defenses (RAID 2020)*, pages 301–316, 2020.

[39] Arjun Nitin Bhagoji, Supriyo Chakraborty, Prateek Mittal, and Seraphin Calo. Model poisoning attacks in federated learning. In *In Workshop on Security in Machine Learning (SecML), collocated with the 32nd Conference on Neural Information Processing Systems (NeurIPS'18)*, 2018.

[40] Milad Nasr, Reza Shokri, and Amir Houmansadr. Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. *IEEE Symposium on Security and Privacy (SP)*, pages 739–753, 2019.

[41] Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and D. Song. Targeted backdoor attacks on deep learning systems using data poisoning. *CoRR*, abs/1712.05526, 2017.

[42] Ziteng Sun, Peter Kairouz, Ananda Theertha Suresh, and H. Brendan McMahan. Can you really backdoor federated learning? *CoRR*, abs/1911.07963, 2019.

[43] Arjun Nitin Bhagoji, Supriyo Chakraborty, Prateek Mittal, and Seraphin Calo. Analyzing federated learning through an adversarial lens. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, volume 97, pages 634–643, 2019.

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

[45] Gan Sun, Yang Cong, Jiahua Dong, Qiang Wang, Lingjuan Lyu, and Ji Liu. Data poisoning attacks on federated machine learning. *IEEE Internet of Things Journal*, PP:1–1, 2021.

[46] Chulin Xie, Keli Huang, Pin-Yu Chen, and Bo Li. Dba: Distributed backdoor attacks against federated learning. In *International Conference on Learning Representations*, 2020.

[47] Ahmed Salem, Rui Wen, Michael Backes, Shiqing Ma, and Yang Zhang. Dynamic backdoor attacks against deep neural networks, 2021.

[48] Yang Liu, Zhihao Yi, and Tianjian Chen. Backdoor attacks and defenses in feature-partitioned collaborative learning. *CoRR*, abs/2007.03608, 2020.

[49] Minghong Fang, Xiaoyu Cao, Jinyuan Jia, and Neil Zhenqiang Gong. Local model poisoning attacks to byzantine-robust federated learning. *29th USENIX Security Symposium*, 2020.

[50] Peva Blanchard, El Mahdi El Mhamdi, Rachid Guerraoui, and Julien Stainer. Machine learning with adversaries: Byzantine tolerant gradient descent. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 119–129. Curran Associates, Inc., 2017.

[51] Leslie Lamport, Robert Shostak, and Marshall Pease. The byzantine generals problem. *ACM Trans. Program. Lang. Syst.*, 4(3):382–401, 1982.

[52] Shengshan Hu, Jianrong Lu, Wei Wan, and Leo Yu Zhang. Challenges and approaches for mitigating byzantine attacks in federated learning. *CoRR*, abs/2112.14468, 2021.

[53] Yann Fraboni, Richard Vidal, and Marco Lorenzi. Free-rider attacks on model aggregation in federated learning. In *Proceedings of the 24th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 1846–1854, 2021.

[54] Vale Tolpegin, Stacey Truex, Mehmet Gursoy, and Ling Liu. *Data Poisoning Attacks Against Federated Learning Systems*, pages 480–501. 2020.

[55] Di Cao, Shan Chang, Zhijian Lin, Guohua Liu, and Donghong Sun. Understanding distributed poisoning attack in federated learning. In *2019 IEEE 25th International Conference on Parallel and Distributed Systems (ICPADS)*, pages 233–239, 2019.

[56] Xingyu Li, Zhe Qu, Shangqing Zhao, Bo Tang, Zhuo Lu, and Yao Liu. LoMar: A local defense against poisoning attack on federated learning. *IEEE Transactions on Dependable and Secure Computing*, pages 1–1, 2021.

[57] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, volume 27, pages 2672–2680, 2014.

[58] Jiale Zhang, Bing Chen, Xiang Cheng, Huynh Thi Thanh Binh, and Shui Yu. Poisongan: Generative poisoning attacks against federated learning in edge computing systems. *IEEE Internet of Things Journal*, 8(5):3310–3322, 2021.

[59] Jiale Zhang, Junjun Chen, Di Wu, Bing Chen, and Shui Yu. Poisoning attack in federated learning using generative adversarial nets. In *2019 18th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/13th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*, pages 374–380, 2019.

[60] Stanislav Fort, Jie Ren, and Balaji Lakshminarayanan. Exploring the limits of out-of-distribution detection. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021.

[61] Pang Wei Koh, Jacob Steinhardt, and Percy Liang. Stronger data poisoning attacks break data sanitization defenses. *Mach. Learn.*, 111(1):1–47, 2022.

[62] Xiaoyun Xu, Jingzheng Wu, Mutian Yang, Tianyue Luo, Xu Duan, Weiheng Li, Yanjun Wu, and Bin Wu. Information leakage by model weights on federated learning. In *Proceedings of the 2020 Workshop on Privacy-Preserving Machine Learning in Practice*, page 31–36. Association for Computing Machinery, 2020.

[63] Gabriele Costa, Fabio Pinelli, Simone Soderi, and Gabriele Tolomei. Covert channel attack to federated learning systems. *CoRR*, abs/2104.10561, 2021.

[64] J. Konečný, H.B. McMahan, F.X. Yu, P. Richtarik, A.T. Suresh, and D. Bacon. Federated learning: Strategies for improving communication efficiency. In *NIPS Workshop on Private Multi-Party Machine Learning*, 2016.

[65] Sébastien Andreina, Giorgia Azzurra Marson, Helen Möllering, and Ghassan O. Karame. Baffle: Backdoor detection via feedback-based federated learning. In *2021 IEEE 41st International Conference on Distributed Computing Systems (ICDCS)*, pages 852–863, 2021.

[66] Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. idlg: Improved deep leakage from gradients. *CoRR*, abs/2001.02610, 2020.

[67] Zhaorui Li, Zhicong Huang, Chaochao Chen, and Cheng Hong. Quantification of the leakage in federated learning. *CoRR*, abs/1910.05467, 2019.

[68] Jonas Geiping, Hartmut Bauermeister, Hannah Dröge, and Michael Moeller. Inverting Gradients - How easy is it to break privacy in federated learning? In *Advances in Neural Information Processing Systems*, volume 33, pages 16937–16947, 2020.

[69] Hanchi Ren, Jingjing Deng, and Xianghua Xie. Grnn: Generative regression neural network—a data leakage attack for federated learning. *ACM Trans. Intell. Syst. Technol.*, 13(4), 2022.

[70] Wenqi Wei, Ling Liu, Margaret Loper, Ka-Ho Chow, Mehmet Emre Gursoy, Stacey Truex, and Yanzhao Wu. A framework for evaluating client privacy leakages in federated learning. In *European Symposium on Research in Computer Security*, pages 545–566, 2020.

[71] Xiao Jin, Pin-Yu Chen, Chia-Yi Hsu, Chia-Mu Yu, and Tianyi Chen. Cafe: Catastrophic data leakage in vertical federated learning. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 994–1006. Curran Associates, Inc., 2021.

[72] Briland Hitaj, Giuseppe Ateniese, and Fernando Perez-Cruz. Deep models under the GAN: information leakage from collaborative deep learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 603–618, 2017.

[73] Zhibo Wang, Mengkai Song, Zhifei Zhang, Yang Song, Qian Wang, and Hairong Qi. Beyond inferring class representatives: User-level privacy leakage from federated learning. In *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*, pages 2512–2520, 2019.

[74] Xiaoyong Yuan, Xiyao Ma, Lan Zhang, Yuguang Fang, and Dapeng Wu. Beyond Class-Level Privacy Leakage: Breaking Record-Level Privacy in Federated Learning. *IEEE Internet Things J.*, 4662(c):1–11, 2021.

[75] Xinjian Luo, Yuncheng Wu, Xiaokui Xiao, and Beng Chin Ooi. Feature inference attack on model predictions in vertical federated learning. In *Proc. - Int. Conf. Data Eng.*, volume 2021-April, pages 181–192, 2021.

[76] Haiqin Weng, Juntao Zhang, Feng Xue, Tao Wei, Shouling Ji, and Zhiyuan Zong. Privacy leakage of real-world vertical federated learning. *CoRR*, abs/2011.09290, 2020.

[77] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 3–18, 2017.

[78] Yaoru Mao, Xiaoyan Zhu, Wenbin Zheng, Danni Yuan, and Jianfeng Ma. A novel user membership leakage attack in collaborative deep learning. In *2019 11th International Conference on Wireless Communications and Signal Processing (WCSP)*, pages 1–6, 2019.

[79] Jingwen Zhang, Jiale Zhang, Junjun Chen, and Shui Yu. Gan enhanced membership inference: A passive local attack in federated learning. In *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, pages 1–6, 2020.

[80] Jiale Chen, Jiale Zhang, Yanchao Zhao, Hao Han, Kun Zhu, and Bing Chen. Beyond model-level membership privacy leakage: an adversarial approach in federated learning. In *2020 29th International Conference on Computer Communications and Networks (ICCCN)*, pages 1–9, 2020.

[81] Oscar Li, Jiankai Sun, Xin Yang, Weihao Gao, Hongyi Zhang, Junyuan Xie, Virginia Smith, and Chong Wang. Label Leakage and Protection in Two-party Split Learning. *CoRR*, abs/2102.08504, 2021.

[82] Lixu Wang, Shichao Xu, Xiao Wang, and Qi Zhu. Eavesdrop the Composition Proportion of Training Labels in Federated Learning. *CoRR*, abs/1910.06044, 2019.

[83] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54, pages 1273–1282, 2017.

[84] Dong Yin, Yudong Chen, Ramchandran Kannan, and Peter Bartlett. Byzantine-robust distributed learning: Towards optimal statistical rates. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, volume 80, pages 5650–5659, 2018.

[85] Zhaoxian Wu, Qing Ling, Tianyi Chen, and Georgios B. Giannakis. Federated variance-reduced stochastic gradient descent with robustness to byzantine attacks. *IEEE Transactions on Signal Processing*, 68:4583–4596, 2020.

[86] Krishna Pillutla, Sham M. Kakade, and Zaid Harchaoui. Robust aggregation for federated learning. *IEEE Transactions on Signal Processing*, 70:1142–1154, 2022.

[87] 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*, 30:119–129, 2017.

[88] El Mahdi El Mhamdi, Rachid Guerraoui, and Sébastien Rouault. The hidden vulnerability of distributed learning in Byzantium. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pages 3521–3530, 2018.

[89] Luis Muñoz-González, Kenneth T. Co, and Emil C. Lupu. Byzantine-robust federated machine learning through adaptive model averaging. *CoRR*, abs/1909.05125, 2019.

[90] Shuhao Fu, Chulin Xie, Bo Li, and Qifeng Chen. Attack-resistant federated learning with residual-based reweighting. *CoRR*, abs/1912.11464, 2021.

[91] E. Tahanian, M. Amouei, H. Fateh, and M. Rezvani. A game-theoretic approach for robust federated learning. *International Journal of Engineering*, 34(4):832–842, 2021.

[92] John F. Nash. Equilibrium points in n-person games. *Proceedings of the National Academy of Sciences*, 36(1):48–49, 1950.

[93] Shiqi Shen, S. Tople, and P. Saxena. Auror: defending against poisoning attacks in collaborative deep learning systems. In *Proceedings of the 32nd Annual Conference on Computer Security Applications*, pages 508–519, 2016.

[94] Felix Sattler, Klaus-Robert Müller, Thomas Wiegand, and Wojciech Samek. On the byzantine robustness of clustered federated learning. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8861–8865, 2020.

[95] 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), 2018.

[96] Xinhong Hei, Xinyue Yin, Yichuan Wang, Ju Ren, and Lei Zhu. A trusted feature aggregator federated learning for distributed malicious attack detection. *Computers & Security*, 99:102033, 2020.

[97] Shahar Azulay, Lior Raz, Amir Globerson, Tomer Koren, and Yehuda Afek. Holdout sgd: Byzantine tolerant federated learning. *CoRR*, abs/2008.04612, 2020.

[98] 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, 2019.

[99] Ying Zhao, Junjun Chen, Jiale Zhang, Di Wu, Jian Teng, and Shui Yu. Pdgan: A novel poisoning defense method in federated learning using generative adversarial network. In *ICA3PP*, 2019.

[100] Suyi Li, Yong Cheng, W. Wang, Y. Liu, and Tianjian Chen. Learning to detect malicious clients for robust federated learning. *CoRR*, abs/2002.00211, 2020.

[101] Mohammad Naseri, Jamie Hayes, and Emiliano De Cristofaro. Local and central differential privacy for robustness and privacy in federated learning. In *Proceedings of the 29th Network and Distributed System Security Symposium (NDSS)*, 2022.

[102] Xiang Wu, Yongting Zhang, Minyu Shi, Pei Li, Ruirui Li, and Neal N. Xiong. An adaptive federated learning scheme with differential privacy preserving. *Future Generation Computer Systems*, 127:362–372, 2022.

[103] Eugene Bagdasaryan, Omid Poursaeed, and Vitaly Shmatikov. Differential privacy has disparate impact on model accuracy. *Advances in Neural Information Processing Systems*, 32:15479–15488, 2019.

[104] Peter Kairouz, H. Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Keith Bonawitz, Zachary B. Charles, Graham Cormode, Rachel Cummings, Rafael G. L. D'Oliveira, Salim Y. El Rouayheb, David Evans, Josh Gardner, Zachary Garrett, Adrià Gascón, Badih Ghazi, Phillip B. Gibbons, Marco Gruteser, Zaïd Harchaoui, Chaoyang He, Lie He, Zhouyuan Huo, Ben Hutchinson, Justin Hsu, Martin Jaggi, Tara Javidi, Gauri Joshi, Mikhail Khodak, Jakub Konecný, Aleksandra Korolova, Farinaz Koushanfar, Oluwasanmi Koyejo, Tancrède Lepoint, Yang Liu, Prateek Mittal, Mehryar Mohri, Richard Nock, Ayfer Özgür, R. Pagh, Mariana Raykova, Hang Qi, Daniel Ramage, Ramesh Raskar, Dawn Xiaodong Song, Weikang Song, Sebastian U. Stich, Ziteng Sun, Ananda Theertha Suresh, Florian Tramèr, Praneeth Vepakomma, Jianyu Wang, Li Xiong, Zheng Xu, Qiang Yang, Felix X. Yu, Han Yu, and Sen Zhao. Advances and open problems in federated learning. *Found. Trends Mach. Learn.*, 14:1–210, 2021.

[105] Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. Learning differentially private recurrent language models. In *International Conference on Learning Representations (ICLR)*, 2018.

[106] Jeremy Bernstein, Yu-Xiang Wang, Kamyar Azizzadenesheli, and Animashree Anandkumar. signSGD: Compressed optimisation for non-convex problems. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, volume 80, pages 560–569, 2018.

[107] Chen Wu, Xian Yang, Sencun Zhu, and Prasenjit Mitra. Mitigating backdoor attacks in federated learning. *CoRR*, abs/2011.01767, 2020.

[108] Amit Portnoy, Yoav Tirosh, and Danny Hendler. Towards federated learning with byzantine-robust client weighting. *Applied Sciences*, 12(17), 2022.

[109] Martin Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 308–318, 2016.

[110] Qinqing Zheng, Shuxiao Chen, Qi Long, and Weijie Su. Federated f-differential privacy. In *International Conference on Artificial Intelligence and Statistics*, pages 2251–2259, 2021.

[111] Zhiqi Bu, Jinshuo Dong, Qi Long, and Weijie J Su. Deep learning with gaussian differential privacy. *Harvard data science review*, 2020(23), 2020.

[112] Xiaoyu Cao, Jinyuan Jia, and Neil Zhenqiang Gong. Data poisoning attacks to local differential privacy protocols. In *30th {USENIX} Security Symposium ({USENIX} Security 21)*, pages 947–964, 2021.

[113] Krishna Yadav, Brij B Gupta, Kwok Tai Chui, and Konstantinos Psannis. Differential privacy approach to solve gradient leakage attack in a federated machine learning environment. In *International Conference on Computational Data and Social Networks*, pages 378–385, 2020.

[114] Meng Hao, Hongwei Li, Guowen Xu, Sen Liu, and Haomiao Yang. Towards efficient and privacy-preserving federated deep learning. In *ICC 2019-2019 IEEE International Conference on Communications (ICC)*, pages 1–6, 2019.

[115] Wenqi Wei, Ling Liu, Yanzhao Wut, Gong Su, and Arun Iyengar. Gradient-leakage resilient federated learning. In *2021 IEEE 41st International Conference on Distributed Computing Systems (ICDCS)*, pages 797–807, 2021.

[116] Nicolas Papernot, Martín Abadi, Úlfar Erlingsson, Ian J. Goodfellow, and Kunal Talwar. Semi-supervised knowledge transfer for deep learning from private training data. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017.

[117] Nicolas Papernot, Shuang Song, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Úlfar Erlingsson. Scalable private learning with PATE. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, 2018.

[118] Yuqing Zhu, Xiang Yu, Manmohan Chandraker, and Yu-Xiang Wang. Private-knn: Practical differential privacy for computer vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11854–11862, 2020.

[119] Yuqing Zhu, Xiang Yu, Yi-Hsuan Tsai, Francesco Pittaluga, Masoud Faraki, Yu-Xiang Wang, et al. Voting-based Approaches For Differentially Private Federated Learning. *CoRR*, abs/2010.04851, 2020.

[120] Chang Wang, Jian Liang, Mingkai Huang, Bing Bai, Kun Bai, and Hao Li. Hybrid differentially private federated learning on vertically partitioned data. *CoRR*, abs/2009.02763, 2020.

[121] Abhishek Bhowmick, John Duchi, Julien Freudiger, Gaurav Kapoor, and Ryan Rogers. Protection against reconstruction and its applications in private federated learning. *CoRR*, abs/1812.00984, 2018.

[122] Hongkyu Lee, Jeehyeong Kim, Seyoung Ahn, Rasheed Hussain, Sunghyun Cho, and Junggab Son. Digestive Neural Networks: A Novel Defense Strategy Against Inference Attacks in Federated Learning. *Computers & Security*, 2021.

[123] Lixin Fan, Kam Woh Ng, Ce Ju, Tianyu Zhang, Chang Liu, Chee Seng Chan, and Qiang Yang. Rethinking privacy preserving deep learning: How to evaluate and thwart privacy attacks. In *Federated Learning*, pages 32–50. Springer, 2020.

[124] Mengjiao Zhang and Shusen Wang. Matrix Sketching for Secure Collaborative Machine Learning. In *International Conference on Machine Learning (ICML)*, pages 12589–12599, 2021.

[125] David P. Woodruff. *Sketching as a Tool for Numerical Linear Algebra*. 2014.

[126] Xue Yang, Yan Feng, Weijun Fang, Jun Shao, Xiaohu Tang, Shu-Tao Xia, and Rongxing Lu. An accuracy-lossless perturbation method for defending privacy attacks in federated learning. In *Proceedings of the ACM Web Conference 2022*, page 732–742. Association for Computing Machinery, 2022.

[127] Chien-Lun Chen, L. Golubchik, and Marco Paolieri. Backdoor attacks on federated meta-learning. *CoRR*, abs/2006.07026, 2020.

[128] Jiale Zhang, Di Wu, Chengyong Liu, and Bing Chen. Defending poisoning attacks in federated learning via adversarial training method. In *Frontiers in Cyber Security*, pages 83–94. Springer Singapore, 2020.

[129] Huafei Zhu. On the relationship between (secure) multi-party computation and (secure) federated learning. *CoRR*, abs/2008.02609, 2020.

[130] Yehuda Lindell. How to simulate it–a tutorial on the simulation proof technique. *Tutorials on the Foundations of Cryptography*, pages 277–346, 2017.

[131] Oded Goldreich. *The Foundations of Cryptography - Volume 1, Basic Techniques.* 2001.

[132] Oded Goldreich. *The Foundations of Cryptography - Volume 2, Basic Applications.* 2004.

[133] Amos Beimel. Secret-sharing schemes: A survey. In *International conference on coding and cryptology*, pages 11–46, 2011.

[134] Oded Goldreich and Yair Oren. Definitions and properties of zero-knowledge proof systems. *Journal of Cryptology*, 7(1):1–32, 1994.

[135] Mihir Bellare, Viet Tung Hoang, and Phillip Rogaway. Foundations of garbled circuits. In *Proceedings of the 2012 ACM conference on Computer and communications security*, pages 784–796, 2012.

[136] Jing Ma, Si-Ahmed Naas, Stephan Sigg, and Xixiang Lyu. Privacy-preserving Federated Learning based on Multi-key Homomorphic Encryption. *CoRR*, abs/2104.06824, 2021.

[137] Zoe L Jiang, Hui Guo, Yijian Pan, Yang Liu, Xuan Wang, and Jun Zhang. Secure Neural Network in Federated Learning with Model Aggregation under Multiple Keys. In *2021 8th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/2021 7th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom)*, pages 47–52, 2021.

[138] Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. Practical secure aggregation for privacy-preserving machine learning. In *proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 1175–1191, 2017.

[139] Keith Bonawitz, Fariborz Salehi, Jakub Konečný, Brendan McMahan, and Marco Gruteser. Federated learning with autotuned communication-efficient secure aggregation. In *2019 53rd Asilomar Conference on Signals, Systems, and Computers*, pages 1222–1226, 2019.

[140] Dan Meng, Hongyu Li, Fan Zhu, and Xiaolin Li. FedMONN: Meta Operation Neural Network for Secure Federated Aggregation. In *2020 IEEE 22nd International Conference on High Performance Computing and Communications; IEEE 18th International Conference on Smart City; IEEE 6th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*, pages 579–584, 2020.

[141] Swanand Kadhe, Nived Rajaraman, O Ozan Koyluoglu, and Kannan Ramchandran. Fastsecagg: Scalable secure aggregation for privacy-preserving federated learning. *CoRR*, abs/2009.11248, 2020.

[142] Thomas Sandholm, Sayandev Mukherjee, and Bernardo A Huberman. SAFE: Secure Aggregation with Failover and Encryption. *CoRR*, abs/2108.05475, 2021.

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

[144] Muhammad Asad, Ahmed Moustafa, and Takayuki Ito. Fedopt: towards communication efficiency and privacy preservation in federated learning. *Applied Sciences*, 10 (8):2864, 2020.

[145] Yong Li, Yipeng Zhou, Alireza Jolfaei, Dongjin Yu, Gaochao Xu, and Xi Zheng. Privacy-preserving federated learning framework based on chained secure multiparty computing. *IEEE Internet of Things Journal*, 8:6178–6186, 2021.

[146] Nhan Khanh Le, Yang Liu, Quang Minh Nguyen, Qingchen Liu, Fangzhou Liu, Quanwei Cai, and Sandra Hirche. Fedxgboost: Privacy-preserving xgboost for federated learning. *CoRR*, abs/2106.10662, 2021.

[147] Yiwei Li, Tsung-Hui Chang, and Chong-Yung Chi. Secure federated averaging algorithm with differential privacy. In *2020 IEEE 30th International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6, 2020.

[148] César Sabater, Aurélien Bellet, and Jan Ramon. Distributed differentially private averaging with improved utility and robustness to malicious parties. *CoRR*, abs/2006.07218, 2020.

[149] Badih Ghazi, Rasmus Pagh, and Ameya Velingker. Scalable and differentially private distributed aggregation in the shuffled model. *CoRR*, abs/1906.08320, 2019.

[150] Peter Kairouz, Ziyu Liu, and Thomas Steinke. The distributed discrete gaussian mechanism for federated learning with secure aggregation. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 5201–5212. PMLR, 18–24 Jul 2021.

[151] Jiasi Weng, Jian Weng, Jilian Zhang, Ming Li, Yue Zhang, and Weiqi Luo. Deepchain: Auditable and privacy-preserving deep learning with blockchain-based incentive. *IEEE Transactions on Dependable and Secure Computing*, 2019.

[152] 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.

[153] Xiaoqi Li, Peng Jiang, Ting Chen, Xiapu Luo, and Qiaoyan Wen. A survey on the security of blockchain systems. *Future Generation Computer Systems*, 107:841–853, 2020.

[154] Shengling Wang, Chenyu Wang, and Qin Hu. Corking by forking: Vulnerability analysis of blockchain. In *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*, pages 829–837, 2019.

[155] Shijie Zhang and Jong-Hyouk Lee. Mitigations on sybil-based double-spend attacks in bitcoin. *IEEE Consumer Electronics Magazine*, 2020.

[156] Rui Qin, Yong Yuan, Shuai Wang, and Fei-Yue Wang. Economic issues in bitcoin mining and blockchain research. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 268–273, 2018.

[157] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, volume 86, pages 2278–2324, 1998.

[158] Gregory Cohen, Saeed Afshar, Jonathan Tapson, and André van Schaik. Emnist: Extending mnist to handwritten letters. In *International Joint Conference on Neural Networks (IJCNN)*, pages 2921–2926, 2017.

[159] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *CoRR*, abs/1708.07747, 2017.

[160] Antonio Torralba, Rob Fergus, and William T. Freeman. 80 million tiny images: A large data set for nonparametric object and scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(11):1958–1970, 2008.

[161] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning (ICML)*, pages 6105–6114, 2019.

[162] Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. Privacy risk in machine learning: Analyzing the connection to overfitting. In *2018 IEEE 31st Computer Security Foundations Symposium (CSF)*, pages 268–282, 2018.

[163] Yu-Xiang Wang, Borja Balle, and Shiva Prasad Kasiviswanathan. Subsampled rényi differential privacy and analytical moments accountant. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1226–1235, 2019.

[164] H. Zhu and Y. Jin. Multi-objective evolutionary federated learning. *IEEE Transactions on Neural Networks and Learning Systems*, 31(4):1310–1322, 2020.

[165] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

[166] Zhou Wang, Eero P Simoncelli, and Alan C Bovik. Multiscale structural similarity for image quality assessment. In *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, volume 2, pages 1398–1402, 2003.

[167] Li Wan, Matthew Zeiler, Sixin Zhang, Yann L. Cun, and Rob Fergus. Regularization of neural networks using dropconnect. In *Proceedings of the 30th International Conference on Machine Learning(ICML - 13)*, pages 1058–1066, 2013.

[168] Alexander Novikov, Dmitrii Podoprikhin, Anton Osokin, and Dmitry P Vetrov. Tensorizing neural networks. In *Advances in Neural Information Processing Systems*, volume 28, pages 442–450, 2015.

[169] Mingxing Tan and Quoc Le. Efficientnetv2: Smaller models and faster training. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, volume 139, pages 10096–10106, 2021.

[170] Ben Graham, Alaaeldin El-Nouby, Hugo Touvron, Pierre Stock, Armand Joulin, Hervé Jégou, and Matthijs Douze. Levit: a vision transformer in convnet's clothing for faster inference. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 12239–12249, 2021.

[171] Yujun Lin, Song Han, Huizi Mao, Yu Wang, and William Dally. Deep gradient compression: Reducing the communication bandwidth for distributed training. In *6th International Conference on Learning Representations*, 2018.

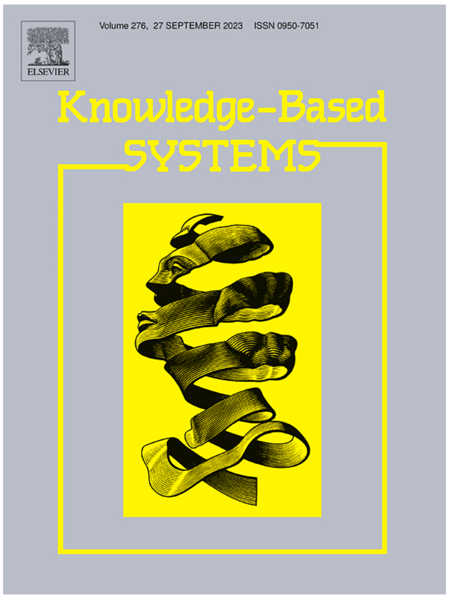
[172] Guansong Pang, Longbing Cao, and Ling Chen. Homophily outlier detection in non-iid categorical data. *Data Mining and Knowledge Discovery*, pages 1–62, 2021.

[173] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. Federated machine learning: Concept and applications. *ACM Trans. Intell. Syst. Technol.*, 10(2), 2019.

[174] Alysa Ziying Tan, Han Yu, Lizhen Cui, and Qiang Yang. Towards personalized federated learning. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–17, 2022.

[175] Yahya H. Ezzeldin, Shen Yan, Chaoyang He, Emilio Ferrara, and Salman Avestimehr. Fairfed: Enabling group fairness in federated learning. In *Workshop on New Frontiers in Federated Learning: Privacy, Fairness, Robustness, Personalization and Data Ownership (NeurIPS 2021)*, 2021.

2 Backdoor Attacks-Resilient Aggregation based on Robust Filtering of Outliers in Federated Learning for image classification



- **Journal:** Knowledge-based Systems
- **JCR Impact Factor:** 8.139
- **Rank:** 29/287
- **Quartile:** Q1
- **Category:** Computer Science, Artificial Intelligence
- **Status:** Published

Ref.: Rodríguez-Barroso, N., Martínez-Cámara, E., Luzón, M. V., & Herrera, F. (2022). Backdoor attacks-resilient aggregation based on Robust Filtering of Outliers in federated learning for image classification. *Knowledge-Based Systems*, 245, 108588. DOI: `https://doi.org/10.1016/j.knosys.2022.108588`.

Backdoor Attacks-Resilient Aggregation based on Robust Filtering of Outliers in Federated Learning for image classification

Nuria Rodríguez-Barroso [*,a] **Eugenio Martínez-Cámara** [a]

M. Victoria Luzón [b] **Francisco Herrera** [a]

[a] *Department of Computer Science and Artificial Intelligence, Andalusian Research Institute in Data Science and Computational Intelligence (DaSCI), University of Granada, Spain*
[b] *Department of Software Engineering, Andalusian Research Institute in Data Science and Computational Intelligence (DaSCI), University of Granada, Spain*

Abstract

Federated Learning is a distributed machine learning paradigm vulnerable to different kind of adversarial attacks, since its distributed nature and the inaccessibility of the data by the central server. In this work, we focus on model-poisoning backdoor attacks, because they are characterized by their stealth and effectiveness. We claim that the model updates of the clients of a federated learning setting follow a Gaussian distribution, and those ones with an outlier behavior in that distribution are likely to be adversarial clients. We propose a new federated aggregation operator called Robust Filtering of one-dimensional Outliers (RFOut-1d), which works as a resilient defensive mechanism to model-poisoning backdoor attacks. RFOut-1d is based on an univariate outlier detection method that filters out the model updates of the adversarial clients. The results on three federated image classification dataset show that RFOut-1d dissipates the impact of the backdoor attacks to almost nullifying them throughout all the learning rounds, as well as it keeps the performance of the federated learning model and it outperforms that state-of-the-art defenses against backdoor attacks.

Keywords Federated Learning · Backdoor Attacks · Resilient Aggregation · Robust Filtering of Outliers.

* Corresponding Author

Email addresses: rbnuria@ugr.es (Nuria Rodríguez-Barroso), emcamara@decsai.ugr.es (Eugenio Martínez-Cámara), luzon@ugr.es (M. Victoria Luzón), herrera@decsai.ugr.es (Francisco Herrera)

1 Introduction

Federated learning (FL) is a nascent learning paradigm based on the distributed training of a learning model among a set of clients under the orchestration of a central server, while keeping the training data sequestered in those clients [1, 2, 3]. FL is vulnerable to adversarial attacks as machine learning systems are [4], but the distributed nature of FL and the inaccessibility of the data hinder the defense against those malicious attacks [5, 6, 7]. Since the capacity of the adversarial clients of misleading the behavior of the FL model, injecting a backdoor attack or breaking the data privacy, the development of robust and resilient FL aggregation operators to those adversarial clients is a real need [8].

The aim of an adversarial attack may be to poison the FL model [9], or to infer any properties of the training data as in the inference attacks [10]. Likewise, the attacks to the FL model may have an specific target [11], or they may only focus on hindering the performance of the FL model without any particular target, as the Byzantine attacks do [12, 13]. The attacks to the FL model can be performed by corrupting the learning model (model-poisoning attacks) or the data (data-poisoning attacks). The latter ones pursue the perversely alteration of the data for provoking their misclassification, *e.g.* the dirty-label poisoning attack [14]. However, this kind of attack is mitigated by the distributed nature of FL and the usual reduced size of adversarial clients, since the aggregation of the local models dissipates the influence of the manipulated data points [5]. In contrast, the model-poisoning attacks may adulterate the FL model without a predefined target or by injecting a backdoor task, which tricks the model in favor of a specific target while keeping good performance on the main task [15]. Also, the backdoor task can be built upon the exploitation of data poisoning to alter the parameter updates. A broad review of each adversarial attack can be read in [16].

In this paper, we focus on the model-poisoning attacks based on data-poisoning and boosting of the model updates of the adversarial clients, specifically on the input-instance and two instances of pattern backdoor attacks, namely pattern-key backdoor attacks [17] and distributed backdoor attacks [18]. Since they are grounded in subtle alterations of the data on the clients, which are inaccessible, and the performance of the main task is not affected, they represent a high risk for FL. We claim that the model updates of the clients in a FL setting follow a Gaussian distribution, and those ones that have an outlier behavior in that distribution may be adversarial clients.

We propose the federated aggregation operator Robust Filtering of one-dimensional Outliers (RFOut-1d), which is able to filter out those clients whose model update represents an outlier in the Gaussian distribution of the model updates of the clients, thereby becoming a defense against model-poisoning attacks based on data-poisoning. The RFOut-1d federated aggregation operator performs the Standard Deviation Method on each dimension of the model clients updates for identifying univariate outliers [19], and it replaces them with the mean of the one-dimensional vector for dispensing with their participation in the aggregation. Since RFOut-1d filters out the adversarial clients, or outliers in our setting, the FL model converges faster and its performance is enhanced. Moreover, RFOut-1d can be combined with other FL defenses against backdoor attacks, as norm threshold of updates or weak differential privacy [17], enlarging its utility for preventing FL from backdoor attacks.

We evaluate the federated aggregation operator RFOut-1d on two settings of model-poisoning attacks, the input-instance and pattern backdoor attacks. The input-instance attack is based on modifying the label of some data points with a target label. Likewise, we define three difficulty levels of the pattern attack by modifying the pattern setting for both the pattern-key backdoor attack and the distributed backdoor attack. We conduct the evaluation on federated datasets, *i.e.* the distribution among the clients is predefined in the datasets. We compare RFOut-1d with FL aggregation operators like the classical FedAvg [20], and classical and state-of-the-art defenses against backdoor attacks in FL such as Median [21], Trimmed-mean [22], Norm Clipping and Weak Differential Privacy [17] and Robust Learning Rate [23].

The results show that RFOut-1d is the defense that highly minimizes the performance of the backdoor attacks in both attack settings. Moreover, RFOut-1d allows to reach the highest performance on the main task, and in some cases meets and even improves the performance of the FL model in a scenario without any adversarial client, which means that the defense of RFOut-1d does not hinder the learning of the FL model. Therefore, the consideration of adversarial clients as outliers on a Gaussian distribution allows (1) to minimize the influence of backdoor adversarial clients, and (2) to keep or even improve the performance of the FL model.

The rest of the work is organized as follows: Section 2 sumps up the related works about adversarial attacks and defenses in FL; Section 3 presents the proposed federated aggregation operator based on the robust filtering of outliers, which works as a defense mechanism; Section 4 details the experimental set-up carried out; Section 5 analyzes the performance of the proposal and; finally, we expound the conclusions of the work in Section 6.

2 Adversarial attacks and Defenses in Federated Learning

Machine learning is highly susceptible to adversarial attacks [24], and most of the defensive approaches are based on [25]: (1) game theory [4], (2) data sanitation [26] and (3) resilient and robust learning models, which assume that a fraction of the training data may be manipulated and consider it as outliers [27]. The first approach cannot be directly applied in FL, since the federated aggregation operator is usually agnostic in relation to the amount of adversarial client and to which one is adversarial. Likewise, the second approach is not feasible in a FL setting, since the data is inaccessible and kept in the clients. Hence, the most plausible defense approach is developing resilient and robust federated aggregation operators able to mitigate the malicious intention of the attacker. Accordingly, we introduce below a taxonomy of adversarial attacks in FL, some outstanding defenses against them and the backdoor attacks types and properties.

2.1 Taxonomy of adversarial attacks

According to [28], there are two types of adversarial attacks: (1) *Inference attacks* [29], which aim at inferring information from the training data; and (2) *poisoning attacks* [30], which pursue to compromise the global learning model. Concerning inference attacks, there are different types of them depending on the information being inferred. The most important ones are the property and membership inference attacks, which respectively seek to infer certain properties of the data and the membership of specific samples in the training set. Due to their nature, the defenses proposed in the literature are based on Differential Privacy [31].

Concerning model attacks, we identify two taxonomies:

1. Depending on which part of the FL schema is attacked, we differentiate between *model-poisoning* [32] and *data-poisoning attacks* [33]. In practice, both are almost equivalent, since a poisoning of the data results in a poisoned model. However, data-poisoning attacks and some of the model-poisoning attacks fail to be effective since the attack dissipates in the aggregation of many clients. For that reason, these attacks are usually combined with *model-replacement* [5] techniques, which boosts the adversarial model (or models) in order to replace the global model in the aggregation.

2. Depending on the purpose of the attack, we distinguish between *untargeted or byzantine attacks* [34], which seek to affect the model's performance, and *targeted or backdoor attacks* [5], which aim at injecting a secondary (or backdoor) task into the global model by stealth. The second ones may be more harmful, since they may be jeopardizing the integrity of the global model without been detected. Moreover, as adversarial client models optimize both the original and the adversarial task, they are also more difficult to detect in the aggregation process. Accordingly, in this paper we focus on backdoor attacks.

2.2 Defenses against adversarial attacks

The research into defenses mechanisms against adversarial attacks in FL is a booming field, and therefore many works have been published in recent years. The literature provides multiple solutions to both byzantine and backdoor attacks in classical machine learning. The vast majority of these defenses are based on data inspection methods, such as removing outliers from the training data in centralized learning [35] or, in a distributed setting, removing outliers from participant's training data or models [36, 37]. In both cases, the available defenses require data inspection, which is not possible in FL. Therefore, defenses against backdoor attacks in FL must be designed ad hoc.

Regarding the state-of-the-art defenses designed to be applied in federated settings, they are based on the modification of the aggregation operator, because the attack is usually carried out by the clients. The first proposed defenses are based on a more robust aggregation of the updates such as the *Byzantine-robust aggregation rules* [38]: coordinate-wise aggregations (trimmed mean or median) [39], Krum [40] or Bulyan [41]. However, these defenses are not effective enough against backdoor attacks due to the stealthy nature of backdoor attacks [15], which stresses the need of ad hoc defenses to mitigate them.

We find some specific defenses against backdoor attacks. The most simple ones are based on the need to apply boosting, such as model-replacement, to these attacks in order to be effective. Therefore, these defenses consist of applying norm bounding of the updates (*Norm Clipping*) with the aim of weakening the effect of the most influencing clients (presumably the attacker) [17]. Moreover, these defenses can be combined with Differential Privacy [31] to get a more generalizable aggregation protection from attacks. More specific defenses are nowadays being proposed, which are based on the assumption that the attackers' updates will have different features than the rest. Some of the most influential examples are: *signSGD* [42] or Robust Learning Rate [23].

2.3 Backdoor attacks: types and properties

We subsequently introduce the backdoor attacks conducted for assessing the defensive capacity of RFOut-1d, as well as their properties. In particular, we perform three backdoor attacks based on the manipulation of the data for replacing the global model. Those attacks differ on how the data is poisoned [43], and specifically they are:

Input-instance backdoor attacks. The objective of the attack is to lead the FL model to misclassify some particular samples of the input distribution in favor of a certain target. For example, in a facial recognition system to access a room allowing access to someone (specific input) who originally did not have it.

Pattern backdoor attacks. The aim is to misclassify some modified samples according to a certain pattern in favor to a specific target. For instance, in the same facial recognition system allowing access to all people wearing purple glasses (certain pattern). The pattern can be known by all the adversarial clients or partially distributed among them, so each client fractionally knows it.

The previous backdoor attacks have a set of *hyper-parameters* or properties for configuring out their behavior. We introduce those properties that support the definition of the backdoor attack setting of the evaluation, as in [17]:

Number of backdoor tasks. In the input-instance backdoor attacks, due to the differences between clients' distributions, we consider the samples of each client as a specific backdoor task, so the number of backdoor tasks corresponds to the number of clients from which we select samples for the backdoored dataset, which we call $D_{backdoor}$. In the pattern backdoor attacks this term is not necessary, since the attack should be generalizable and it may be thus considered to be addressed by just one backdoor task (one pattern).

Number of adversarial clients. Number of clients compromised and coordinated in order to perform the backdoor tasks. The local training dataset of each adversarial client i is composed by the union of its original training dataset $D^i_{original}$ and the backdoored dataset $D_{backdoor}$. That is, $D^i_{adv} = D^i_{original} \bigcup D_{backdoor}$. In the input-instance backdoor attack, $D_{backdoor}$ will correspond to the set of samples from every backdoor task. Regarding the pattern backdoor attack, the $D_{backdoor}$ is composed of all the samples perversely altered according to a certain pattern.

Sampling of adversarial clients and frequency. The frequency of appearance of adversarial clients in the subset of clients selected for each aggregation is a key factor. In [17], the authors discuss between fixed-frequency appearance or random sampling. They conclude that the fraction of adversarial clients required for the attack to be effective is too high and unrealistic when using random sampling. Hence, we focus on the fixed-frequency attacks, in which we determine the number of adversarial clients participating in each aggregation.

3 Defense against Model-poisoning backdoor attacks based on Robust Filtering of Outliers

We consider the notations and definitions of FL as defined in [5] in order to describe the attacks discussed in this work. In particular, let G^t and L^t_i be the global model and local model of client i-th at the learning round t respectively, n the total number of clients selected for each aggregation and η the server learning rate. Accordingly, the update of the global model in the learning round t is performed as follows in Equation 1:

$$G^t = G^{t-1} + \frac{\eta}{n}\sum_{i=1}^{n}(L^t_i - G^{t-1}). \tag{1}$$

In this context, we define the backdoor attack scenario as one or several clients which are coordinated to inject a secondary or backdoor task into the global model. Typically, these attacks do not negatively affect the original task performance, which makes them harder to identify. Since the distributed character of the learning process, the high number of clients participating in each aggregation and the assumption that the proportion of adversarial clients will be significantly lower than of benign clients, the influence of the adversarial clients would be dissipated among the rest of the clients and no effective attack would take place. For that reason, we focus on model-poisoning backdoor attacks based on the model-replacement paradigm proposed in [5, 15, 17], which is based on boosting the influence of the adversarial attack for avoiding its dissipation among the large size of benign clients.

As we consider that only one adversarial client is selected in the learning round t, its aim is to replace the global model G^t with its backdoored model L^t_{adv}, which optimizes both original and backdoor tasks by sending to the FL server

$$\hat{L}^t_{adv} = \beta(L^t_{adv} - G^{t-1}), \tag{2}$$

where $\beta = \frac{n}{\eta}$ is the boost factor required to conduct model-replacement [5]. Then, replacing Equation 2 in Equation 1 we have[1]

$$G^t = G^{t-1} + \frac{\eta}{n}\frac{n}{\eta}(L^t_{adv} - G^{t-1}) + \frac{\eta}{n}\sum_{i=2}^{n}(L^t_i - G^{t-1}). \tag{3}$$

According to the definition of FL [20], eventually the FL model will converge to a solution, so we can assume that $L^t_i - G^{t-1} \approx 0$ for benign clients. Hence, we rewrite Equation 3 as follows

$$G^t \approx G^{t-1} + \frac{\eta}{n}\frac{n}{\eta}(L^t_{adv} - G^{t-1}) = L^t_{adv}, \tag{4}$$

which replaces the global model with the *backdoored* model. If multiple adversarial clients participate in the same learning round, we assume that they can coordinate for the attack by dividing the boosting factor between the attackers. In the rest of the paper, we consider $\eta = 1$.

We consider the attack scenario described below, in which the model updates of benign clients minimizes the global task loss, while the model updates of adversarial clients optimize the global and backdoor task loss. We base our proposal on the following two assumptions:

1. The model updates of the clients follow a Gaussian distribution from a certain learning round, since the global aggregated model tends to converge to a common solution. This is intuitively proven based on the Central Limit Theorem [44], which states that the sum of independent random variables closely approaches to a Gaussian distribution. Let the clients local weight's distributions be each of the random variables, then, linear combinations of them approach closely to a Gaussian distribution. Therefore, aggregation over aggregation, the result will converge to a Gaussian distribution. In particular, the data distribution for each of the dimensions of the updates converges to an univariate Gaussian distribution.

2. Since the model update of adversarial clients has a twofold target, we assume that it represents an outlier in the distribution of client updates for a specific learning round.

Regarding the previous assumptions, we propose the **RFOut-1d** (**R**obust **F**ilterig of **1-d**imensional **Out**liers) federated aggregation operator based on filtering out the outliers in the distribution of client model updates with the objective of producing a more robust aggregation in each learning round t. Since the high dimensionality of the updates (usually from neural networks), and with the aim of avoiding the loss of information by applying dimensionality reduction techniques, we perform an univariate anomaly detection for each dimension of

[1] For the sake of clarity, we assume that the adversarial client is client 1.

the model updates. Therefore, for each of dimension $i \in \{1, \dots, m\}$, where m is the dimension of the vectors of the model updates, we consider the vector formed by the local model update of each client in that dimension $L_i = (L_1^t[i], L_2^t[i], \dots, L_n^t[i])$, where n is the number of clients participating in the aggregation, and we apply the Standard Deviation Method for identifying univariate outliers in Gaussian distributions. Hence, it filters out those that verify that the difference between the value and the mean is greater or equal than δ times the standard deviation. Formally, we replace by μ_i those that verify

$$abs(L_j^t[i] - \mu_i) \geq \delta \sigma_i, \tag{5}$$

where μ_i and σ_i are the mean and standard deviation of L_i, respectively, $L_j^t[i]$ is the parameter of the dimension i of the model update at the round of learning t of the client j and $\delta = 3$ according to an experimental result of [19]. We use the mean estimator since, as the model updates of the clients are subsequently aggregated, it filters out the participation of the outliers in the aggregation.

At the end, the federated aggregation RFOut-1d consists of the 1-dimensional mean of the non-filtered out parameters. Formally, the resulting aggregated model G_t in each dimension i of the parameters is

$$G_t[i] = \frac{1}{n} \sum_{i=1}^{n} \hat{L}_j^t[i] \quad \forall i \in \{1, \dots, m\}, \quad \text{where} \tag{6}$$

$$\hat{L}_j^t[i] = \begin{cases} \mu_i & \text{if } abs(L_j^t[i] - \mu_i) \geq \delta \sigma_i \\ L_j^t[i], & \text{otherwise} \end{cases}, \forall j \in \{1, \dots, n\} \tag{7}$$

where $\hat{L}_j^t[i]$ the resulting vector after applying Equation 5 criteria to $L_j^t[i]$. Algorithm 1 depicts the proposed aggregation operator.

Note that RFOut-1d, in addition to filtering out those clients that are presumably attackers, optimizes the learning process by favoring a faster convergence towards a common solution. Moreover, it can be combined with other aggregation mechanisms proposed as defenses, such as norm threshold of updates or weak Differential Privacy [17].

4 Experimental set-up

We subsequently detail the experimental framework for assessing the RFOut-1d federated aggregation operator. We describe the datasets used in the evaluation, the configuration of the backdoor attacks and the evaluation measures. We follow the guidelines of [45] for conducting the experiments.[2]

[2] We provide the source code of RFOut-1d at this GitHub Repository.

Algorithm 1 RFOut-1d

```
Input: local updates $\{L_1^t, L_2^t, \dots, L_n^t\}$
$num\_dimensions = length(L_1^t)$
Initialize $G^t$
$\delta = 3$
for $i = 0$ to $num\_dimensions$ do
    $\hat{L}_i = (L_1^t[i], L_2^t[i], \dots, L_n^t[i])$
    $\mu_i = mean(\hat{L}_i)$
    $\sigma_i = std(\hat{L}_i)$
    for $j = 1$ to $n$ do
        if $abs(L_j[i] - \mu_i) \geq \delta\sigma_i$ then
            $L_j[i] \leftarrow \mu_i$
        end if
    end for
    $G_t[i] = mean(\hat{L}_i)$
end for
Return $G_t$
```

4.1 Datasets

The few availability of non-simulated federated datasets is one of the difficulties for evaluating FL models. It is possible to use classical machine learning datasets and distribute them among clients according to different data distributions. However, although the non-IID character of data distribution can be simulated [8], it is quite complex to simulate the customization of data among clients, so that they represent their individual features. For that reason, we decided to use datasets that are by definition federated. We focus on the following image classification datasets included in the LEAF benchmark:

1. *Digits FEMNIST*:[3] The digits dataset of the federated version of EMNIST, where each client corresponds to an original writer.

2. *CelebA*:[4] An image classification dataset composed by famous face images with 40 binary attributes annotations per image, where we associate each famous with a client. We use it as a binary image classification dataset, selecting a specific attribute as target, in particular, *Smiling* (*CelebA-S*) and *Attractive* (*CelebA-A*).

The use of federeated datasets may result in some clients with insufficient amount of data. Accordingly, we set the minimum number of samples per client k and discard the clients that do not satisfy this condition. For *CelebA* datasets, we use $k = 30$, specified as the best option in [46], and for *FEMNIST* we set $k = 8$, as it is the minimum number of samples per client. Table 1 shows the statistics per dataset.

[3] https://www.nist.gov/itl/products-and-services/emnist-dataset

[4] http://mmlab.ie.cuhk.edu.hk/projects/CelebA.html

	FEMNIST	CelebA-S	CelebA-A
Clients	3579	1878	1878
k	8	30	30
Number of labels	10	2	2
Training samples	240000	56364	56364
Samples per client (mean)	67.05	30.01	30.01
Samples per client (std)	11.17	0.19	0.19
Testing samples	40000	19962	19962

Table 1: Description of the FEMNIST, CelebA-S and CelebA-A datasets.

4.2 Backdoor attacks set-up

According to the definition of backdoor attacks, the design of such attacks has a wide range of options as the backdoor task depends on the aim of the injected task. We define an input-instance and the two pattern backdoor attacks settings to assess RFOut-1d in each dataset.

4.2.1 Input-instance backdoor attacks set-up

We set a target label and a set of samples ($D_{backdoor}$) from clients which belong to another class (original label). The attack consists in classifying the highest amount of these samples with the target label without modifying any sample. Due to the particularity of each client, we set that the number of backdoor tasks corresponds to the number of clients from whom samples have been taken for the backdoored dataset $D_{backdoor}$. We set the number of adversarial clients as the number of clients who have the backdoored dataset among their data and the frequency of attack. Based on these parameters, we define these attacks in Table 2.

	FEMNIST	CelebA-S	CelebA-A
Backdoor tasks	30	30	10
$\lvert D_{backdoor} \rvert$	213	247	228
Adversarial clients	11	20	15
Frequency of attack	1	1	1
Origin label	7	No	No
Target label	1	Yes	Yes

Table 2: Definition of input-instance backdoor attacks set-up for the FEMNIST, CelebA-S and CelebA-A datasets.

4.2.2 Pattern backdoor attacks set-up

We evaluate RFOut-1d in two types of pattern backdoor attack: (1) Pattern-key backdoor attack, in which all the clients know the complete pattern and use it in their training process and (2) Distributed backdoor attack [18], in which each client knows the pattern partially and the aim is to coordinate to inject the complete pattern.

Pattern-key backdoor attacks We set a target label and a pattern-key. Thus, the attack consists in classifying any sample poisoned with the pattern-key as the target label. In this case, the number of backdoor tasks corresponds to the number of adversarial clients, because only the adversarial clients poison some of their samples with the pattern-key. In order to show that the behavior of RFOut-1d is agnostic of the pattern-key, we use three patterns of different levels of difficulty expressed in numbers of pixels (see Figure 1): (1) one single black pixel, (2) a red cross of length 4 and (3) a yellow square of side 5x5. Analogously, we define these attacks in Table 3, and we show the patterns used for poisoning implemented in Figure 2. When the pattern-key is small in comparison with the original image we add a zoom of the pattern in the corner.

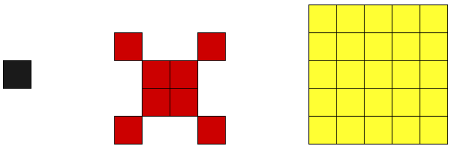


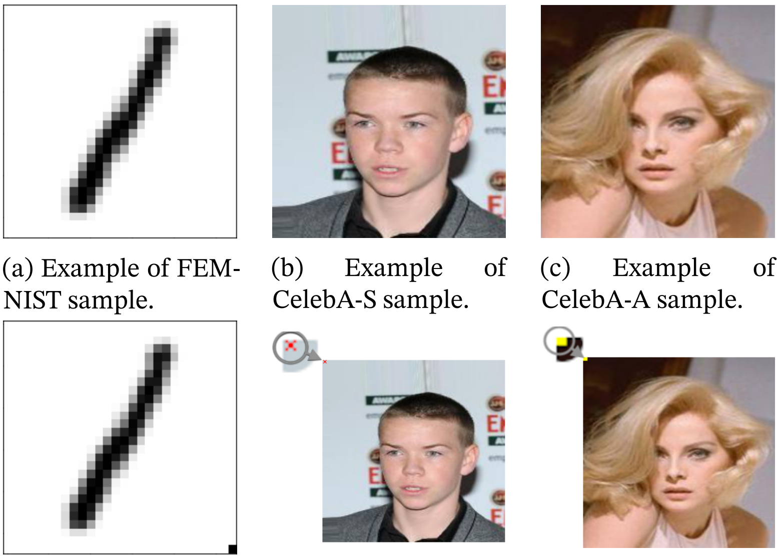
Figure 1: Representation of the pattern-key employed. From left to right: the 1-pixel pattern, the 8-pixel pattern (a red cross of length 4) and the 25-pixel pattern (a 5x5 yellow square).

	FEMNIST	CELEBA-S	CELEBA-A
ADVERSARIAL CLIENTS	30	15	15
FREQUENCY OF ATTACK	1	1	1
TARGET LABEL	0	YES	YES
PIXELS OF THE PATTERN	1	8	25

Table 3: Definition of pattern-key backdoor attacks set-up for the FEMNIST, CelebA-S and CelebA-A datasets.

Distributed backdoor attack We set the target label, the complete pattern and the partial pattern of each adversarial client. Clearly, the attack consists in classifying each sample poisoned with the complete pattern as the target label, not the partial ones. For that reason, in each aggregation participates one adversarial client from each partial pattern, thus involving multiple adversarial clients in each learning round. In order to show that the behavior of RFOut-1d is agnostic of the pattern, we use different patterns for each database (see Figure 3):

1. Black corners. Four single black pixels distributed among the four corners of the image for FEMNIST. We distribute the pattern by setting 4 adversarial clients and assigning each corner to one of them.
2. Monocolor cross. A cross of length 5 in the upper left corner red for CelebA-S and blue for CelebA-A. We distribute the pattern by setting 2 adversarial clients and assigning each diagonal of the cross to one of them.



(a) Example of FEMNIST sample.

(b) Example of CelebA-S sample.

(c) Example of CelebA-A sample.

(d) Backdoored FEMNIST sample (1-pixel pattern).

(e) Backdoored CelebA-S sample (8-pixel pattern).

(f) Backdoored CelebA-A sample (25-pixel pattern).

Figure 2: Examples of original (a, b and c) and backdoored samples (d, e and f) of each dataset.

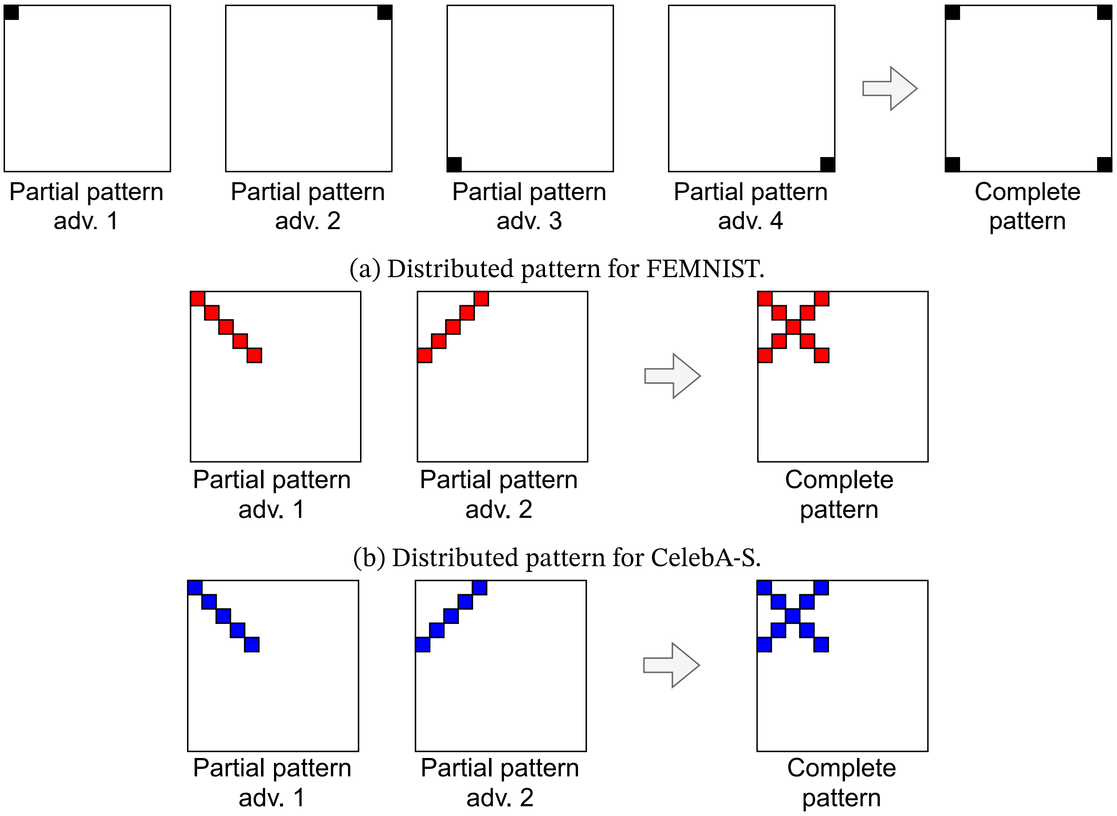
	FEMNIST	CelebA-S	CelebA-A
Adversarial clients	4	2	2
Frequency of attack	4	2	2
Target label	0	Yes	YES
Pixels of the complete pattern	4	10	10
Pixels of each partial pattern	1	5	5

Table 4: Definition of distributed backdoor attacks set-up for the FEMNIST, CelebA-S and CelebA-A datasets.

4.3 Evaluations metrics and baselines

The task of defending against backdoor attacks is a twofold task, and its evaluation thus requires of measuring the prevention against the attack and the performance of the resulting model in the original task. The aim of the defense mechanism is to reduce the effects of the attack as much as possible without compromising the performance of the model in the original task. We consider two test datasets:

- *Original task test*. The original test of the dataset used for measuring the performance in terms of accuracy in the original task.
- *Backdoor task test*. Dataset which represents the attack in order to measure the performance in terms of accuracy in the backdoor task. Regarding the input-instance



(a) Distributed pattern for FEMNIST.

(b) Distributed pattern for CelebA-S.

(c) Distributed pattern for CelebA-A.

Figure 3: Representation of the patterns of the distributed backdoor attacks. We specify the partial pattern of each adversarial client on the left and the complete pattern on the right. Note that the proportion between the pattern and the image size is not real, we enlarge the pattern to make the image illustrative.

backdoor attacks, we consider the backdoored dataset $D_{backdoor}$ as in [17]. Concerning the pattern backdoor attacks, we consider two test datasets [43]: (1) *Backdoor task test* as in the input-instance backdoor attack situation to measure the effectiveness of the attack; and (2) *Global backdoor task test*, consisting of the test instances not originally belonging to the target class, but poisoned using the pattern in order to measure the capability of generalization of the attack.

Since the results can be highly heterogeneous in each of the learning rounds depending on the defense mechanism and in order to show robust results, we use the average of each of these measures throughout the last ten learning rounds.

We compare RFOut-1d with the following federated aggregation operators and backdoor defense mechanisms, which represent the classical baselines and the state-of-the-art in defenses against backdoor attacks:

1. *Federated Averaging (FedAvg)* [20]. It is based on the (weighted) averaging of the local models. We use this aggregation operator as the simplest baseline due to it represents the *no-defense* situation.

2. *Median* [21]. It is one of the Byzantine-robust aggregation rules which is based on replacing the mean with the median in the aggregation method. We use it as a baseline, due to the higher robustness of the median with respect to the mean in the presence of extreme values.

3. *Trimmed-mean* [22]. It represents another Byzantine-robust aggregation rule. It relies on using a more robust version of the mean that consists in eliminating a fixed percentage of extreme values both below and above the data distribution.

4. *Norm Clipping of updates* [17]. Since model-poisoning backdoor attacks produce updates with large norms because of the boosting factor, norm clipping of updates is widely used as a simple defense mechanism. It consists in clipping the update by dividing it with the appropriate scalar if it exceeds a fixed threshold M, as in Equation 8, where $\Delta L_i^t = L_i^{t+1} - G^t$.

$$G^{t+1} = G^t + \frac{\eta}{n} \sum_{i=1}^{n} \frac{\Delta L_i^t}{\max(1, ||\Delta L_i^t||_2/\mathbf{M})} \tag{8}$$

5. *Weak Differential Privacy (WDP)* [17]. This defense is based on Differential Privacy [31], which is commonly used to defend against backdoor attacks [47]. This mechanism consists of applying norm techniques combined with a little amount of Gaussian noise as a function of σ according to Equation 9.

$$G^{t+1} = G^t + \frac{\eta}{n} \sum_{i=1}^{n} \frac{\Delta L_i^t}{\max(1, ||\Delta L_i^t||_2/\mathbf{M})} + \mathcal{N}(0, \frac{\boxed{?}\mathbf{M}}{n}) \tag{9}$$

6. *Robust Learning Rate (RLR)* [23]. They determine the direction of the update, for each dimension, in form of signs of the updates using a threshold parameter θ. Hence, if the sum of the signs of the updates is less than θ, they change the direction of the update by multiplying by -1. They assert that this defense can be combined with the two previous ones, by means applying the norm clipping and noise addition specified in Equation 9 to the modified models' updates, producing a better performance.

We did not compare RFOut-1d with the defenses Krum [40] and Bulyan [41], because they are design against Byzantine attacks, while RFOut-1d works against backdoor attacks.

We use the configuration values specified in [17, 23] in our experiments. In addition, we evaluate the use of *norm clipping* and *noise addition* in RFOut-1d following the Equation 9 with the same parameter values as RLR. Table 5 shows these parameters, where M is the

	M	σ	θ
NORM CLIPPING	3/1	0	-
WDP	3/1	0.0025	-
RLR	0.5/1	0.0001	7
RFOUT-1D	0.5/1	0.0001	-

Table 5: Parameters used in our experiments according to the parameters recommended by the authors.

threshold for the updates norm, σ the Gaussian noise parameter and θ the threshold for RLR.

Since the main aim of this work is to propose a robust federated aggregation operator to defense against backdoor attacks, we use an standard CNN-based image classification model composed of two CNN layers followed by its corresponding max-pooling layers, a dense layer and the output layer with a softmax activation function. In particular, we use the models and the hyperparameters included in the LEAF[5] benchmark for each dataset. All details concerning hyperparameters, number of epochs or batch size can be found in the GitHub repository.

5 Analysis of the results

We evaluate RFOut-1d on the datasets described in Section 4.1, and in the two backdoor attack settings described in Section 4.2 during 100 rounds of learning. Subsequently, we expose the assessment in each backdoor attack, and we analyze the capacity of RFOut-1d of enhancing the FL model convergence.

5.1 Analysis of the performance against Input-instance Backdoor Attacks

Table 6 shows that RFOut-1d outperforms all the baselines in the twofold goal of minimizing the backdoor task performance and maximizing the performance of the original task (image classification), which means that filtering out the parameters that represent outliers in the distribution of updates mitigates these attacks.

Generally, as we use a more complex defense, the results obtained improve notably in favor of the defense. In particular, RLR is the most powerful baseline (especially the norm clipping and noise version), namely as far as the accuracy of the original task is concerned.

The highest result in all test sets is always achieved by RFOut-1d. On the one hand, the ability to mitigate the attack is shown, achieving a null effect of the attack (0.0 of backdoor accuracy) in two of the three datasets. On the other hand, we show that it does not compromise the performance in the original task, even improving the result of the task without attack in the case of *FEMNIST* and *CelebA-A*, which proves that it also filters out low-value information. This suggest that it may not be only filtering out adversarial clients, but those

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

			FEMNIST		CelebA-S		CelebA-A	
	M	σ	Original	Backdoor	Original	Backdoor	Original	Backdoor
No attack	0	0	0.9657	-	**0.7900**	-	0.7973	-
FedAvg	0	0	0.8661	0.8230	0.3630	0.9738	0.5140	0.5194
Median	0	0	0.9448	0.0306	0.7881	0.0457	0.7961	0.0152
Trimmed-mean	0	0	0.9526	0.0256	0.7852	0.0423	0.7961	0.0221
NormClip	3	0	0.9606	0.6373	0.6852	0.1431	0.6078	0.2558
WDP	3	0.0025	0.9374	0.1578	0.7204	0.1195	0.6119	0.2399
RLR	0	0	0.8404	0.0288	0.6539	0.0457	0.7877	0.0451
RLR†	0.5/0.5/1	0.0001	0.9546	0.0128	0.7852	0.0388	0.7934	0.0043
RFOut-1d	0	0	0.9629	**0.0048**	0.7883	0.0046	0.7973	**0.0.**
RFOut-1d†	0.5/0.5/1	0.0001	**0.9670**	0.0054	0.7892	**0.0**	**0.7975**	**0.0**

Table 6: Mean results for the input-instance backdoor attack in terms of accuracy. The symbol † denotes the combination of a defense with norm clipping and noise addition. We also show, in the first row, the expected accuracy with *FedAvg* but without any attack. The best result for each of the test sets is highlighted in bold.

clients who have such poor training that they confuse the model rather than contributing to its convergence towards a global solution.

Regarding to the combination of the defenses with *norm clipping* and *noise*, both RLR and RFOut-1d can be combined. However, for RLR it seems to be a necessity as the results improve markedly while RFOut-1d obtains strong results on its own, which confirms the robustness of our proposal.

Therefore, the results show that RFOut-1d is a robust federated aggregation operator against input-instance backdoor attack, and it does not need any additional operation to preserve the FL model from this kind of adversarial attack.

5.2 Analysis of the performance against Pattern Backdoor Attacks

We analyze the behavior of RFOut-1d in two different pattern backdoor attacks: (1) the analysis of the performance of the pattern-key backdoor attacks, in which only one adversarial client participates in each aggregation process; and (2) the analysis of the distributed backdoor attacks, in which participate as many clients as different partial patterns defined in each aggregation process.

Pattern-key backdoor attacks Analogously, the results in Tables 7, 8, 9 show the higher performance of RFOut-1d compared to the baselines in FEMNIST, CelebA-A and CelebA-S respectively, which proves that our claim is also confirmed for pattern-key backdoor attacks and, moreover, for patterns of different level of difficulty.

If we compare the effectiveness of these pattern-key attacks without any defense (*FedAvg*) with the same condition as in the input-instance attacks, we find that the first ones are, generally, more effective. This is due to the alteration of images with a pattern is a more

				FEMNIST	
	M	σ	Original	Backdoor	Test
No attack	0	0	0.9657	-	-
FedAvg	0	0	0.9741	1.0	1.0
Median	0	0	0.9540	0.0091	0.0154
Trimmed-mean	0	0	0.9664	0.0114	0.0148
NormClip	1	0	0.9687	0.0553	0.0538
WDP	1	0.0025	0.9357	0.0938	0.0175
RLR	0	0	0.9039	0.0407	0.0575
RLR†	0.5/1	0.0001	0.9265	0.0089	0.0085
RFOut-1d	0	0	0.9741	**0.0043**	0.0072
RFOut-1d†	0.5/1	0.0001	**0.9753**	0.0059	**0.0051**

Table 7: Mean results for the pattern-key backdoor attack in terms of accuracy in FEMNIST. The symbol † denotes the combination of a defense with norm clipping and noise addition. We also show, in the first row, the expected accuracy with *FedAvg* but without any attack. The best result for each of the test sets is highlighted in bold.

				CelebA-S	
	M	σ	Original	Backdoor	Test
No attack	0	0	0.7900	-	-
FedAvg	0	0	0.6858	1.0	0.9999
Median	0	0	0.6978	0.0678	0.0532
Trimmed-mean	0	0	0.7013	0.0521	0.0654
NormClip	1	0	0.6798	0.1433	0.1647
WDP	1	0.0025	0.7413	0.0538	0.0743
RLR	0	0	0.7132	0.0574	0.0469
RLR†	0.5/1	0.0001	0.7714	0.0205	0.0316
RFOut-1d	0	0	**0.7900**	**0.0**	**0.0**
RFOut-1d†	0.5/1	0.0001	0.7896	**0.0**	0.0010

Table 8: Mean results for the pattern-key backdoor attack in terms of accuracy in CelebA-S. The symbol † denotes the combination of a defense with norm clipping and noise addition. We also show, in the first row, the expected accuracy with *FedAvg* but without any attack. The best result for each of the test sets is highlighted in bold.

sophisticated attack, and it allows to reach its aims with a higher success than the input-instance backdoor attack.

Despite being more powerful attacks, the defenses, the baselines and RFOut-1d, show similar behavior, improving as we use a more complex defense. In particular, the defense that outperforms in both tasks of maximizing the performance of the global task and minimizing the performance of the backdoor task is, again, RFOut-1d. Therefore, RFOut-1d outperforms all the baselines in the target of defending the FL model against the pattern-key backdoor task, which means that our claim holds in this kind of backdoor attack. In this case, it also outperforms the results without any attack, which confirms its proper performance as a federated aggregation operator even without the presence of adversarial clients.

			CelebA-A		
	M	σ	Original	Backdoor	Test
No attack	0	0	0.7973	-	-
FedAvg	0	0	0.7375	1.0	0.99
Median	0	0	0.7452	0.0163	0.0189
Trimmed-mean	0	0	0.7498	0.0092	0.0101
NormClip	1	0	0.7126	0.1433	0.1316
WDP	1	0.0025	0.6609	0.1440	0.1707
RLR	0	0	0.6657	0.0280	0.0286
RLR†	0.5/1	0.0001	0.7923	0.0031	0.0016
RFOut-1d	0	0	**0.7967**	**0.0023**	**0.0015**
RFOut-1d†	0.5/1	0.0001	0.7874	0.0054	0.0124

Table 9: Mean results for the pattern-key backdoor attack in terms of accuracy in CelebA-A. The symbol † denotes the combination of a defense with norm clipping and noise addition. We also show, in the first row, the expected accuracy with *FedAvg* but without any attack. The best result for each of the test sets is highlighted in bold.

			FEMNIST		
	M	σ	Original	Backdoor	Test
No attack	0	0	0.9657	-	-
FedAvg	0	0	0.9678	0.8556	0.1649
Median	0	0	0.9437	0.0114	0.0053
Trimmed-mean	0	0	0.9649	0.0102	0.0046
NormClip	1	0	0.9731	0.3289	0.0526
WDP	1	0.0025	0.9729	0.3342	0.0211
RLR	0	0	0.9518	0.7821	0.0263
RLR†	0.5/1	0.0001	0.9614	0.0107	0.0062
RFOut-1d	0	0	0.9721	0.2130	0.0089
RFOut-1d†	0.5/1	0.0001	**0.9737**	**0.0000**	**0.0032**

Table 10: Mean results for the distributed backdoor attack in terms of accuracy in FEMNIST. The symbol † denotes the combination of a defense with norm clipping and noise addition. We also show, in the first row, the expected accuracy with *FedAvg* but without any attack. The best result for each of the test sets is highlighted in bold.

Distributed backdoor attacks The results of Tables 10, 11 and 12 show the outperforming of RFOut-1d compared with the baselines in FEMNIST, CelebA-S and CelebA-A respectively. It is worth mentioning that the backdoor set is less significant in this case as it represents the effectiveness of the partial patterns, while we are interested in the effectiveness of the complete pattern.

Regarding the effectiveness of the attack, we find that distributed backdoor attack has achieved a lower performance on the backdoor task in FEMNIST than the pattern-key backdoor attack. However, the behavior in both partitions of CelebA is comparable. We attribute this phenomenon to the fact that the distributed backdoor attack is more complicated to be successful, being too challenging to carry it out in a multi-class problem as FEMNIST. How-

			CelebA-S		
	M	σ	Original	Backdoor	Test
No attack	0	0	0.7900	-	-
FedAvg	0	0	0.6793	0.9772	0.9944
Median	0	0	0.3701	0.8636	0.8178
Trimmed-mean	0	0	0.7831	**0.0000**	0.0014
NormClip	1	0	0.7604	**0.0000**	0.0499
WDP	1	0.0025	0.7896	**0.0000**	0.0031
RLR	0	0	0.2276	0.9454	0.9704
RLR†	0.5/1	0.0001	0.2686	0.8878	0.9351
RFOut-1d	0	0	0.7602	0.0021	0.0076
RFOut-1d†	0.5/1	0.0001	**0.7897**	**0.0000**	**0.0000**

Table 11: Mean results for the distributed backdoor attack in terms of accuracy in CelebA-S. The symbol † denotes the combination of a defense with norm clipping and noise addition. We also show, in the first row, the expected accuracy with *FedAvg* but without any attack. The best result for each of the test sets is highlighted in bold.

			CelebA-A		
	M	σ	Original	Backdoor	Test
No attack	0	0	0.7973	-	-
FedAvg	0	0	0.5796	0.9643	0.9871
Median	0	0	0.7759	0.0363	0.0525
Trimmed-mean	0	0	0.7888	0.0714	0.0166
NormClip	1	0	0,7954	0.0666	0.0079
WDP	1	0.0025	0.7087	0.1764	0.1602
RLR	0	0	0.2113	0.9765	0.9523
RLR†	0.5/1	0.0001	0.4127	0.6154	0.6433
RFOut-1d	0	0	0.6223	0.0284	0.0367
RFOut-1d†	0.5/1	0.0001	**0.7997**	**0.0000**	**0.0013**

Table 12: Mean results for the distributed backdoor attack in terms of accuracy in CelebA-A. The symbol † denotes the combination of a defense with norm clipping and noise addition. We also show, in the first row, the expected accuracy with *FedAvg* but without any attack. The best result for each of the test sets is highlighted in bold.

ever, even in this case the presence of the defenses is notable, significantly diminishing the effectiveness of the backdoor attacks in test.

Concerning the evaluation of the different defenses, both the proposal and baselines, the results further confirm the satisfactory performance of RFOut-1d in backdoor attacks. To conclude, it is worthy noting that RLR, which in the evaluation of the input-instance and pattern-key backdoor attacks had achieved quite successful results, is outperformed by the other simpler baselines in both partitions of CelebA. It shows that it may not be useful for this type of distributed backdoor attacks, or at least with the parameters used in the experimentation.

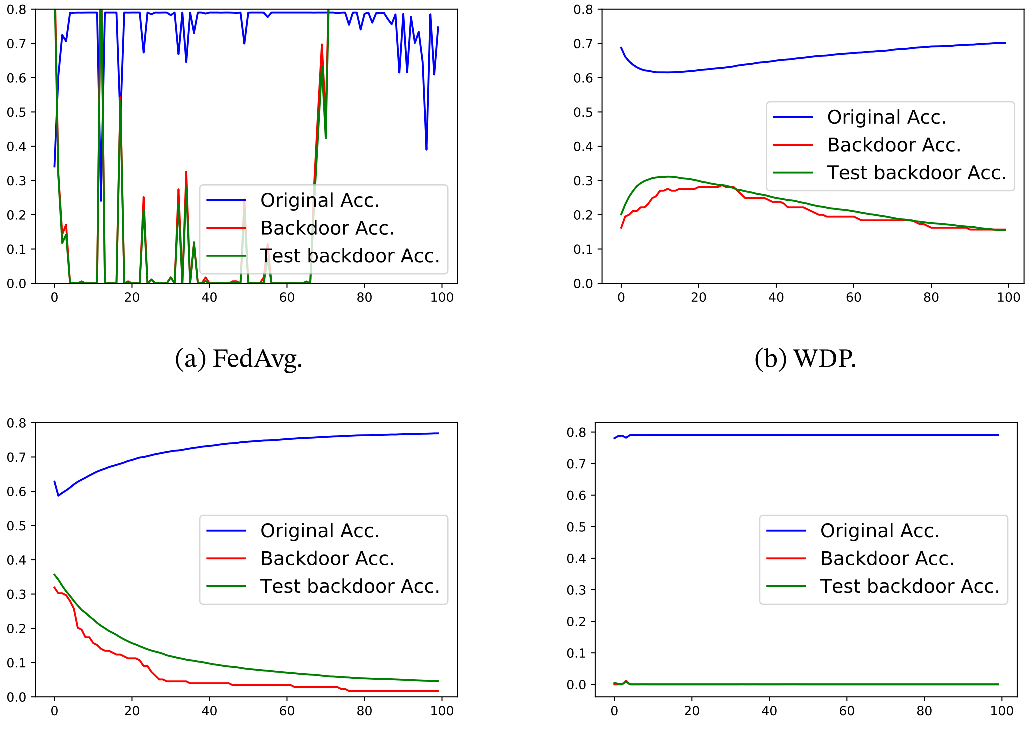
5.3 Analysis of the convergence with RFOut-1d

We claim that RFOut-1d, in addition to being an effective defense in FL, allows the global model to converge to a common solution in less rounds of learning, by means filtering out those parameters that deviate from the solution set by majority. We show it by analyzing the convergence of the models in both the original and the backdoor task throughout the learning rounds.

We choose the pattern-key backdoor attack on CelebA-S and show only two classical aggregation operators (*FedAvg* and *WDP*), *RLR* in its best version including *norm clipping* and *noise* and RFOut-1d in order to reduce the number of figures.

The convergence of the chosen models is presented in Figure 4. Clearly, FedAvg shows the worst performance while RFOut-1d outperforms all baselines in two ways:

1. Regarding the accuracy of the original task, RFOut-1d ensures that it is not compromised in any of the attack attempts, while in the rest of the baselines the performance



(a) FedAvg.

(b) WDP.

(c) RLR.

(d) RFOut-1d.

Figure 4: Convergence plots in the CelebA-S pattern-key backdoor attack experiment. We show both the convergence of the original task (Original task accuracy, in blue) and the backdoor task (Backdoor accuracy and Test backdoor accuracy, in red and green respectively).

is more unstable, becoming the global model's compromised in several rounds of learning.

2. Regarding the backdoor tasks, RFOut-1d demonstrates an outstanding performance and shows a clear improvement over the rest of the baselines. In fact, the attack is not successful in any learning round.

We stress out the relevance of this fact because, despite the acceptable results achieved by the rest of the defenses, the attack is relatively successful at certain learning rounds, which also compromise the integrity of the model. This fact, combined with the fast convergence provided by RFOut-1d, further highlights the success of this approach as an aggregation operator as well as a defense in FL.

6 Conclusions

We addressed the defense against model-poisoning backdoor attacks, which is a real challenge of FL. Based on the claim that the updates from adversarial clients would represent outliers in the Gaussian distribution of clients' updates, we propose RFOut-1d, a defense mechanism based on a robust filtering of one-dimensional outliers in the federated aggregation operator. After evaluating RFOut-1d in a variety of settings under different backdoor attacks, and comparing it with the state of the art defenses, the results shows that our claim holds. Therefore, we state that:

- RFOut-1d is a highly effective defense that dissipates the impact of the backdoor attacks to the point of (almost) nullifying them throughout all the learning rounds.
- In some scenarios, RFOut-1d outperforms the results achieved without any attack, which shows its capacity to filter out clients who are hindering the training process.
- In contrast to other defenses, it does not hinder the FL process by keeping (or even improving) the performance of the model in the original task.
- The convergence of the model to the common solution is accelerated and optimized by filtering out clients that diverge from this solution.

To conclude, we have shown that RFOut-1d is a high quality defense as well as a proper federated aggregation operator by effectively stopping the effect of attacks while favoring the learning of the global model.

Acknowledgments

This research work is supported by the R&D&I grants PID2020-119478GB-I00, PID2020-116118GA-I00 and EQC2018-005084-P funded by MCIN/AEI/10.13039/501100011033 and by "ERDF A way of making Europe". Nuria Rodríguez Barroso were supported by the grant

FPU18/04475 funded by MCIN/AEI/10.13039/501100011033 and by "ESF Investing in your future". Eugenio Martínez Cámara were supported by the grant IJC2018-036092-I funded by MCIN/AEI/10.13039/501100011033.

References

[1] Qiang Yang, Yang Liu, Yong Cheng, Yan Kang, Tianjian Chen, and Han Yu. Federated learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 13(3):1–207, 2019.

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

[3] Junjie Pang, Yan Huang, Zhenzhen Xie, Jianbo Li, and Zhipeng Cai. Collaborative city digital twin for the covid-19 pandemic: A federated learning solution. *Tsinghua Science and Technology*, 26(5):759–771, 2021.

[4] Nilesh Dalvi, Pedro Domingos, Mausam, Sumit Sanghai, and Deepak Verma. Adversarial classification. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, page 99–108, 2004.

[5] Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. How to backdoor federated learning. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108, pages 2938–2948, 2020.

[6] Zuobin Xiong, Zhipeng Cai, Daniel Takabi, and Wei Li. Privacy threat and defense for federated learning with non-i.i.d. data in AIoT. *IEEE Transactions on Industrial Informatics*, (Early Access), 2021.

[7] Yunfei Song, Tian Liu, Tongquan Wei, Xiangfeng Wang, Zhe Tao, and Mingsong Chen. Fda3: Federated defense against adversarial attacks for cloud-based IIoT applications. *IEEE Transactions on Industrial Informatics*, (Early Access), 2020.

[8] Peter Kairouz, H. Brendan McMahan, Brendan Avent, Aurélien Bellet, and Mehdi Bennis et al. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, 14(1–2):1–210, 2021.

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

[10] Milad Nasr, Reza Shokri, and Amir Houmansadr. Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. In *2019 IEEE Symposium on Security and Privacy*, pages 739–753, 2019.

[11] Fnu Suya, Saeed Mahloujifar, David Evans, and Yuan Tian. Model-targeted poisoning attacks: Provable convergence and certified bounds. *CoRR*, abs/2006.16469, 2020.

[12] Liping Li, Wei Xu, Tianyi Chen, Georgios B. Giannakis, and Qing Ling. RSA: Byzantine-robust stochastic aggregation methods for distributed learning from heterogeneous

datasets. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):1544–1551, 2019.

[13] J. So, B. Güler, and A. S. Avestimehr. Byzantine-resilient secure federated learning. *IEEE Journal on Selected Areas in Communications*, Early access, 2020.

[14] Clement Fung, Chris JM Yoon, and Ivan Beschastnikh. The limitations of federated learning in sybil settings. In *23rd International Symposium on Research in Attacks, Intrusions and Defenses*, pages 301–316, 2020.

[15] Arjun Nitin Bhagoji, Supriyo Chakraborty, Prateek Mittal, and Seraphin Calo. Analyzing federated learning through an adversarial lens. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pages 634–643, 2019.

[16] Lingjuan Lyu, Han Yu, Xingjun Ma, Lichao Sun, Jun Zhao, Qiang Yang, and Philip S. Yu. Privacy and robustness in federated learning: Attacks and defenses. *CoRR*, abs/2012.06337, 2020.

[17] Ziteng Sun, Peter Kairouz, Ananda Theertha Suresh, and H. Brendan McMahan. Can you really backdoor federated learning? *CoRR*, abs/1911.07963, 2019.

[18] Chulin Xie, Keli Huang, Pin-Yu Chen, and Bo Li. DBA: Distributed backdoor attacks against federated learning. In *International Conference on Learning Representations*, 2020.

[19] Ihab F. Ilyas and Xu Chu. Data cleaning. chapter 3: Outlier Detection. Association for Computing Machinery, 2019.

[20] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. Agüera y Arcas. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54, pages 1273–1282, 2017.

[21] Yudong Chen, Lili Su, and Jiaming Xu. Distributed statistical machine learning in adversarial settings: Byzantine gradient descent. *CoRR*, abs/1705.05491, 2017.

[22] Dong Yin, Yudong Chen, Kannan Ramchandran, and Peter L. Bartlett. Byzantine-robust distributed learning: Towards optimal statistical rates. *CoRR*, abs/1803.01498, 2018.

[23] Mustafa Safa Ozdayi, Murat Kantarcioglu, and Yulia R Gel. Defending against backdoors in federated learning with robust learning rate. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 9268–9276, 2021.

[24] Pavel Laskov and Richard Lippmann. Machine learning in adversarial environments. *Machine Learning*, 81:115–119, November 2010.

[25] Ling Huang, Anthony D. Joseph, Blaine Nelson, Benjamin I.P. Rubinstein, and J. D. Tygar. Adversarial machine learning. In *Proceedings of the 4th ACM Workshop on Security and Artificial Intelligence*, AISec '11, page 43–58, New York, NY, USA, 2011. Association for Computing Machinery.

[26] Blaine Nelson, Marco Barreno, Fuching Jack Chi, Anthony D. Joseph, Benjamin I. P. Rubinstein, Udam Saini, Charles Sutton, J. D. Tygar, and Kai Xia. *Misleading Learners: Co-opting Your Spam Filter*, pages 17–51. Springer US, Boston, MA, 2009.

[27] C. Croux, P. Filzmoser, and M.R. Oliveira. Algorithms for projection–pursuit robust principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, 87(2):218–225, 2007.

[28] Lingjuan Lyu, Han Yu, Xingjun Ma, Lichao Sun, Jun Zhao, Qiang Yang, and Philip S. Yu. Privacy and robustness in federated learning: Attacks and defenses. *CoRR*, abs/2012.06337, 2020.

[29] Milad Nasr, Reza Shokri, and Amir Houmansadr. Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. *2019 IEEE Symposium on Security and Privacy (SP)*, May 2019.

[30] Di Cao, Shan Chang, Zhijian Lin, Guohua Liu, and Donghong Sun. Understanding distributed poisoning attack in federated learning. In *2019 IEEE 25th International Conference on Parallel and Distributed Systems (ICPADS)*, pages 233–239, 2019.

[31] Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.

[32] Xingchen Zhou, Ming Xu, Yiming Wu, and Ning Zheng. Deep model poisoning attack on federated learning. *Future Internet*, 13(3), 2021.

[33] Gan Sun, Yang Cong, Jiahua Dong, Qiang Wang, and Ji Liu. Data poisoning attacks on federated machine learning. *CoRR*, abs/2004.10020, 2020.

[34] Leslie Lamport, Robert Shostak, and Marshall Pease. *The Byzantine Generals Problem*, page 203–226. Association for Computing Machinery, New York, NY, USA, 2019.

[35] Jacob Steinhardt, Pang Wei Koh, and Percy Liang. Certified defenses for data poisoning attacks. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, page 3520–3532, 2017.

[36] Muhammad Shayan, Clement Fung, Chris J. M. Yoon, and Ivan Beschastnikh. Biscotti: A ledger for private and secure peer-to-peer machine learning. *CoRR*, abs/1811.09904, 2018.

[37] Shiqi Shen, S. Tople, and P. Saxena. Auror: defending against poisoning attacks in collaborative deep learning systems. *Proceedings of the 32nd Annual Conference on Computer Security Applications*, pages 508–519, 2016.

[38] Xiaoyu Cao, Minghong Fang, Jia Liu, and Neil Zhenqiang Gong. FLTrust: Byzantine-robust federated learning via Trust Bootstrapping. *ISOC Network and Distributed System Security Symposium*, 2021.

[39] Dong Yin, Yudong Chen, Ramchandran Kannan, and Peter Bartlett. Byzantine-robust distributed learning: Towards optimal statistical rates. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pages 5650–5659, 2018.

[40] Peva Blanchard, El Mahdi El Mhamdi, Rachid Guerraoui, and Julien Stainer. Machine learning with adversaries: Byzantine tolerant gradient descent. In *Advances in Neural Information Processing Systems*, volume 30, pages 119–129, 2017.

[41] El Mahdi El Mhamdi, Rachid Guerraoui, and Sébastien Rouault. The hidden vulnerability of distributed learning in Byzantium. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 3521–3530. PMLR, 10–15 Jul 2018.

[42] Jeremy Bernstein, Jiawei Zhao, Kamyar Azizzadenesheli, and Anima Anandkumar. SignSGD with majority vote is communication efficient and fault tolerant. In *International Conference on Learning Representations*, 2019.

[43] Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. Targeted backdoor attacks on deep learning systems using data poisoning. *CoRR*, abs/1712.05526, 2017.

[44] Byron P. Roe. *Central Limit Theorem*, pages 66–68. Springer New York, New York, NY, 2008.

[45] Nuria Rodríguez-Barroso, Goran Stipcich, Daniel Jiménez-López, José Antonio Ruiz-Millán, Eugenio Martínez-Cámara, Gerardo González-Seco, M. Victoria Luzón, Miguel Angel Veganzones, and Francisco Herrera. Federated learning and differential privacy: Software tools analysis, the sherpa.ai fl framework and methodological guidelines for preserving data privacy. *Information Fusion*, 64:270 – 292, 2020.

[46] Sebastian Caldas, Peter Wu, Tian Li, Jakub Konecný, H. Brendan McMahan, Virginia Smith, and Ameet Talwalkar. LEAF: A benchmark for federated settings. *CoRR*, abs/1812.01097, 2018.

[47] Yuzhe Ma, Xiaojin Zhu, and Justin Hsu. Data poisoning against differentially-private learners: Attacks and defenses. In *International Joint Conferences on Artificial Intelligence Organization*, pages 4732–4738, 7 2019.

3 Dynamic defence against byzantine poisoning attacks in federated learning.

- **Journal**: Future Generation Computer Systems
- **JCR Impact Factor**: 7.307
- **Rank**: 28/373
- **Quartile**: Q1
- **Category**: Computer Science, Computer Networks and Communications
- **Status**: Published

Ref.: Rodríguez-Barroso, N., Martínez-Cámara, E., Luzón, M. V., & Herrera, F. (2022). Dynamic defence against byzantine poisoning attacks in federated learning. *Future Generation Computer Systems*, 133, 1-9. DOI: `https://doi.org/10.1016/j.future.2022.03.003`.

Dynamic Defense Against Byzantine Poisoning Attacks in Federated Learning

Nuria Rodríguez-Barroso [*,a] **Eugenio Martínez-Cámara** [a]

M. Victoria Luzón [b] **Francisco Herrera** [a]

[a] *Department of Computer Science and Artificial Intelligence, Andalusian Research Institute in Data Science and Computational Intelligence (DaSCI), University of Granada, Spain*
[b] *Department of Software Engineering, Andalusian Research Institute in Data Science and Computational Intelligence (DaSCI), University of Granada, Spain*

Abstract

Federated learning, as a distributed learning that conducts the training on the local devices without accessing to the training data, is vulnerable to Byzatine poisoning adversarial attacks. We argue that the federated learning model has to avoid those kind of adversarial attacks through filtering out the adversarial clients by means of the federated aggregation operator. We propose a dynamic federated aggregation operator that dynamically discards those adversarial clients and allows to prevent the corruption of the global learning model. We assess it as a defense against adversarial attacks deploying a deep learning classification model in a federated learning setting on the Fed-EMNIST Digits, Fashion MNIST and CIFAR-10 image datasets. The results show that the dynamic selection of the clients to aggregate enhances the performance of the global learning model and discards the adversarial and poor (with low quality models) clients.

***K*eywords** Federated Learning · Deep Learning · Adversarial Attacks · Byzantine Attacks · Dynamic Aggregation Operator.

[*] Corresponding Author
Email addresses: `rbnuria@ugr.es` (Nuria Rodríguez-Barroso), `emcamara@decsai.ugr.es` (Eugenio Martínez-Cámara), `luzon@ugr.es` (M. Victoria Luzón), `herrera@decsai.ugr.es` (Francisco Herrera)

1 Introduction

The standard machine learning approach is built upon an algorithm that learns from a centralized data source. Distributed machine learning proposes the distribution of the data and elements of a learning model among several nodes as a solution for the unceasing growing of learning model complexity and the size of training data [1, 2]. However, the distributed machine learning solution is neither valid for the data privacy challenge, nor for an scenario with a large number of clients and a non homogeneous data distribution [3, 4].

Federated learning (FL) is a machine learning approach in which the algorithms learn from sequestered data [3, 5]. The FL model is mainly composed of two components: a global server that owns the global learning model and a set of clients storing the local learning models and the local training datasets. Likewise, FL consists in: (1) training the local learning models in each data source, (2) distilling the parameters of the local learning models into a central server, (3) aggregating the parameters of the local models in the global learning model and (4) updating the local learning models with the aggregated federated global learning model after the aggregation. This specific setting supports its main feature, which is the prevention of data leakage and the protection of data privacy, since the data do not abandon its local storage and they are not shared with any other client or third party. Since FL is a user privacy-preserving approach designed to decentralized scenarios, an Artificial Intelligence of Things (AIoT) setting is a natural way to use it, for both the distributed nature and the privacy needed in IoT (Internet of Things) devices [6].

Machine learning is vulnerable to malicious manipulations on the input data or the learning model to cause incorrect classification [7]. This vulnerability becomes harder to address in FL due to most of the defensive approaches are based data inspection techniques. Among the different kind of adversarial attacks in the literature [8], in this paper we focus on byzantine poisoning attacks [9], which are based on the arbitrary manipulation of the training data (data poisoning attack [10, 11]) or the client model updates (model poisoning attacks [12]) with the aim of hindering the performance of the FL model.

We argue in this paper that the FL model has to be able to dynamically avoid adversarial clients to preserve the learning model from byzantines poisoning attacks, which is usually performed on the server by the federated aggregation operator. In the literature there are a number of federated aggregation operators, but they do not prevent the federated model from this kind of attacks [13, 14, 15], or they do it following some assumptions about the nature of the adversarial clients [16] or prove to be insufficiently effective [17].

We propose the Dynamic Defense Against Byzantine Attacks (DDaBA), which is a dynamic aggregation operator that dynamically selects the clients to be aggregated and discards those ones considered as adversarial, and it features agnostic about the number and nature of the adversarial clients. This dynamic defense is built upon an Induced Ordered Weighted Averaging (IOWA) operator [18], which aggregates the clients on a weighted basis according to an induced-ordered function and a linguistic quantifier. We use as induced-ordered function the performance of the local learning models on a validation set stored in the server. The linguistic quantifier addresses the weighting of the clients, which usually depends on the knowledge of the problem and predefined parameters. We design an agnostic linguistic quantifier on the nature of the problem, which is based on: (1) considering the distribution of data resulting from measuring the performance variation between local learning models on the validation set, (2) assuming that the resulting distribution follows an exponential distribution, and (3) using the properties of that distribution to set the parameters of the linguistic quantifier in order to discard the adversarial clients that correspond to outliers in the exponential distribution according to the Tukey criteria.

We evaluate the DDaBA as a defense in a FL model for image classification. For that purpose, we leverage the benchmark image classification datasets Fed-EMNIST[1] Digits [19], Fashion MNIST[2] [20] and CIFAR-10,[3] and we distribute the data over the clients following a non independent and identically distributed (non-IID) distribution. We compare the DDaBA with the classical federated aggregation operator FedAvg [13] with no defense and the state-of-the-art defenses against three different byzantine attacks: label-flipping [21], out-of-distribution [22] and random weights [23] attacks. We show that the DDaBA is able to identify the adversarial and poor clients, filter them out and enhance the performance of the global learning model.

We analyze the behavior of the DDaBA in an scenario with a extreme proportion of adversarial clients, and we see that the performance of the federated global model is hindered. Although this is a very unlikely scenario, we also introduce the static version of DDaBA, Static Defense Against Byzantine Attacks (SDaBA), which predefine the parameters of the linguistic quantifier of the IOWA operator for discarding the susceptible adversarial clients. The SDaBA, as well as the DDaBA, outperforms all the baselines in the three adversarial attacks developed for the evaluation.

The rest of the work is organized as follows: the following section summarizes the background related to FL, adversarial attacks in FL and defenses against them. Section 3 is focused on the description of the dynamic FL model for identifying adversarial clients. We detail the experimental set-up in Section 4 and evaluate and analyze the results of the FL models in Section 5. Finally, conclusions are described in Section 6.

2 Background

We expound in this section some relevant concepts and related works. We introduce FL in Section 2.1, we describe the main types of adversarial attacks in FL in Section 2.2, and we detail the proposed defenses against byzantine attacks in Section 2.3.

2.1 Federated Learning

[1] https://www.nist.gov/node/1298471/emnist-dataset

[2] https://github.com/zalandoresearch/fashion-mnist

[3] https://www.cs.toronto.edu/~kriz/cifar.html

FL is a learning approach pushed by the need of overcoming the limitations of distributed learning for preserving data privacy and for processing large number of clients following a non homogeneous data distribution [24]. FL proposes a new training approach of learning algorithms that consists in the iterative training of the model in the devices that own the data, the aggregation of those models in the federated model, and the updating of the local models with the federated model. Hence, FL prevents from data leakage and preserves data privacy, since the data do not leave the electronic device.

Formally, FL is a distributed machine learning paradigm consisting of a set of clients $\{C_1, \dots, C_n\}$ with their respective local training data $\{D_1, \dots, D_n\}$. Each of these clients C_i has a local learning model named as L_i represented by the parameters $\{L_1, \dots, L_n\}$. FL aims at learning a global learning model represented by G, using the scattered data across clients through an iterative learning process known as *round of learning*. For that purpose, in each round of learning t, each client trains its local learning model over their local training data D_i^t, which updates the local parameters L_i^t to $\hat{L}_i^t$. Subsequently, the global parameters G^t are computed aggregating the trained local parameters $\{\hat{L}_1^t, \dots, \hat{L}_n^t\}$ using an specific federated aggregation operator Δ, and the local learning models are updated with the aggregated parameters:

$$\begin{aligned} G^t &= \Delta(\hat{L}_1^t, \hat{L}_2^t, \dots, \hat{L}_n^t) \\ L_i^{t+1} &\leftarrow G^t, \quad \forall i \in \{1, \dots, n\} \end{aligned} \tag{1}$$

The updates among the clients and the server are repeated as much as needed for the learning process. Thus, the final value of G will sum up the knowledge sequestered in the clients.

2.2 Related works about adversarial attacks

Machine learning is highly susceptible to adversarial attacks [25], and the vast majority of the defensive approaches are based on three approaches [8]: (1) game theory [26], (2) data sanitation [27] and (3) resilient and robust learning models, which assume that a fraction of the training data may be manipulated and consider it as outliers [28]. Due to the federated aggregation operator is agnostic in relation with adversarial clients information, the first approach can not be applied in FL. Likewise, since the training data in FL is inaccessible by the server, the second approach is also not feasible in FL. Therefore, the most promising defense approach is developing resilient and robust federated aggregation operators with the ability to safeguard the model from the effect of attacks.

According to [29], there are two types of adversarial attacks in FL: (1) *Inference attacks* [30], which aim at inferring information from the training data; and (2) *poisoning attacks* [31], which pursue to compromise the global learning model. Concerning inference attacks, there are different types of them depending on the information being inferred. The most important ones are the property and membership inference attacks, which respectively seek to infer certain properties of the data and the membership of specific samples in the training set. Because of their nature, the defenses proposed in the literature are based on applications derived from or inspired by the Differential Privacy [32]. Regarding poisoning attacks, we identify two taxonomies:

1. Depending on which part of the FL model is attacked, we differentiate between *model-poisoning* [33] and *data-poisoning attacks* [34]. In practice, both are almost equivalent, since a poisoning of the data results in a poisoned model. However, data-poisoning attacks and some of the model-poisoning attacks fail to be effective since the attack dissipates in the aggregation of many clients. Hence, these attacks are combined with *model-replacement* [17] techniques, which boosts the adversarial model (or models) in order to replace the global model.

2. Depending on the purpose of the attack, we distinguish between *untargeted or byzantine attacks* [35], which seek to affect the model's performance, and *targeted or backdoor attacks* [17], which aim at injecting a secondary or backdoor task into the global model by stealth.

2.3 Defenses against adversarial attacks

The literature provides multiple solutions to both byzantine and backdoor attacks in classical machine learning. The vast majority of these defenses are based on data inspection methods, such as removing outliers from the training data in centralized learning [36] or, in a distributed setting, removing outliers from participant's training data or models [37, 38]. In both cases, the available defenses require data inspection, which is not possible in FL. Therefore, defenses against adversarial attacks in FL must be designed ad hoc.

Regarding the state-of-the-art defenses designed to be applied in federated settings, they are based on the modification of the aggregation operator, because the attack is usually carried out by the clients. The most important defenses against byzantine attacks are based on a more robust aggregation of the updates and they are called *byzantine-robust aggregation rules*. We highlight the following ones:

- Coordinate-wise aggregations [39], which replaces the mean of the classical aggregation operator FedAvg [13] with more robust statistics to outliers or anomalous data. The main ones are the trimmed-mean and the median.

- Krum (and MultiKrum) [40], which is based on using geometric properties to determine the most central model updates vectors. This defense requires a k hyperparameter that determines the number of clients remaining in the aggregation.

- Bulyan [41] which is the state of the art. It is built as a combination of Krum and trimmed-mean. Accordingly, the model updates vectors are sorted according to their geometrical centrality and are aggregated through a trimmed-mean with a m parameter, which discards a total of $2m$ clients.

Additionally, differential privacy [32] methods are an important safeguard for the information shared during the communication between the server and the clients. Therefore, the defensive challenges of the FL should focus on client attacks.

The main weakness of the defenses proposed in the literature is that they are highly dependent on parameters, which beforehand are difficult to set without information about the number or nature of the adversary clients. Thus, we propose in this paper a defense mechanism against poisoning attacks, which dynamically selects the clients that are not adversarial and filters out the adversarial or the poor ones (clients with low quality models) without the requirement to set any parameters.

3 Dynamic Defense Against Poisoning Attacks

FL is featured by its restriction to access to the training data, which is sequestered in the clients. Accordingly, poisoning attacks, both data and (local) model poisoning [10, 11], grounded in the malicious manipulation of the training data or the local model updates, can corrupt the FL model, which cannot inspect the training to defend itself against this kind of adversarial attacks.

We propose a defense against byzantine poisoning attacks built upon a federated aggregation operator based on a Induced Ordered Weighted Averaging (IOWA) [18] that dynamically selects the clients to be aggregated, and filters out the adversarial ones. We call it Dynamic Defense Against Byzantine Attacks (DDaBA).

The IOWA operators, and more generally the Ordered Weighted Averaging (OWA) ones [42], are functions for weighting the contribution of a set of clients in a aggregation process, as it is the aggregation of the parameters of the local learning models in FL. We mathematically introduce OWA and IOWA operators in Appendix A, and according to the definition the IOWA operator is composed of (1) an order-inducing function to set the weighting assignation order, and (2) a linguistic quantifier to calculate the weight contribution value. We define the induced-order function used in DDaBA in Section 3.1, and the linguistic quantifier that dynamically adapts the weighting value calculation during the FL training in Section 3.2. Finally, we sum up DDaBA in Section 3.3.

3.1 Accuracy-based induced ordering function for clients model updates

The aim of byzantine poisoning adversarial attacks is hindering the performance of a FL model through altering the training data or directly the model updates. Since FL is grounded in the aggregation of the L_i, those maliciously altered ones would perform lower than the non-altered ones. Hence, the validation of the L_i before the aggregation may help to identify the suspicious adversarial clients.

We propose the Local Accuracy Function, f_{LA}, to measure the performance of each L_i before its aggregation. The f_{LA} function is based on the availability of a validation set shared among the clients. The viability of this validation set is justified by its reduced size compared to the size required for training, and the possibility of making it up through expert or prior knowledge. We define the f_{LA} function in Definition 3.1.

Definition 3.1 (Local Accuracy Function (f_{LA})) *it measures the performance of a local learning model L_i using a fixed validation dataset named as VD. For that, it computes the accuracy of L_i over VD:*

$$f_{LA}(L_i) = accuracy(L_i, VD) \tag{2}$$

where $accuracy(L_i, VD)$ refers to the standard accuracy evaluation measure of the local learning model L_i in the dataset VD.

Once the clients model updates are sorted according to this sorting function, we expect that the benign client's models will converge to a common solution, while the adversarial client's models will not, but they will converge to a worse solution for the original problem. Therefore, if we define the random variable resulting from the differences in accuracy among all clients with the client that scored the highest accuracy as follows:

$$\mathbb{X}_i^{f_{LA}} = \max_i \{f_{LA}(L_i)\} - f_{LA}(L_i). \tag{3}$$

We assume that this random variable $\mathbb{X}$ will approximate an Exponential Distribution, since there will be many values close to zero (and always positive), and very few far from zero.

3.2 Dynamic linguistic quantifier for weighting the contribution of clients

The non-IID data distribution of most of the FL settings make impossible to know beforehand the nature of the clients, and hence it is impossible to know the amount of adversarial clients. Therefore, the selection of the FL clients by its weighted contribution has to be dynamically calculated for adapting to the nature of the clients.

The dynamic selection of the DDaBA model is based on a IOWA linguistic quantifier that some of its parameters values depend on the resulting exponential distribution after ordering the clients model updates $\mathbb{X}_i^{f_{LA}}$. Before the definition of the linguistic quantifier of DDaBA, we first define the IOWA linguistic quantifier in Definition 3.2.

Definition 3.2 (Linguistic quantifier) *It is a function $Q : [0,1] \to [0,1]$ verifying $Q(0) = 0$, $Q(1) = 1$ and $Q(x) \geq Q(y)$ for $x > y$. Equation 4 defines how the function Q computes the weighting values where w_i represents the weighting associated to the position i of a vector of dimension n, and Equation 5 defines the behaviour of the function Q.*

$$w_i^{(a,b)} = Q_{a,b}\left(\frac{i}{n}\right) - Q_{a,b}\left(\frac{i-1}{n}\right) \tag{4}$$

$$Q_{a,b}(x) = \begin{cases} 0 & 0 \leq x \leq a \\ \dfrac{x-a}{b-a} & a \leq x \leq b \\ 1 & b \leq x \leq 1 \end{cases} \tag{5}$$

where $a, b \in [0,1]$ satisfying $0 \leq a \leq b \leq 1$, and they set the intervals for calculating the contribution weight of each L_i. For the sake of clarification, those x values in the same interval will have the same weighting value.

$$Q_{a,b,c,y_b}(x) = \begin{cases} 0 & 0 \leq x \leq a \\ \dfrac{x-a}{b-a} \cdot y_b & a \leq x \leq b \\ \dfrac{x-b}{c-b} \cdot (1-y_b) + y_b & b \leq x \leq c \\ 1 & c \leq x \leq 1 \end{cases} \tag{6}$$

We redefine the function $Q_{a,b}$ for providing it a dynamic behaviour and a higher weighting of top clients, which depends on the random variable $\mathbb{X}_i^{f_{LA}}$. Accordingly, we propose Q_{a,b,c,y_b} that is defined in Equations 6, and incorporates two new parameters to the model (c and y_b), in addition to the two existing ones. The definition of each of the parameters is as follows:

1. Parameter a. This parameter represents the proportion of clients to which null weighing is assigned. Since we do not want to filter out those clients which stand out ”at the top”, i.e. those that obtain the best accuracy, we set the value to 0.

2. Parameter b. It sets the portion of clients we consider as top clients and we want to weight higher. The choice of this parameter is done dynamically, so that the top clients correspond to the first decile of the distribution of $\mathbb{X}_i^{f_{LA}}$. Formally, b is the portion of clients that verify
$$\mathbb{X}_i^{f_{LA}} \leq \frac{\ln(10/9)}{\lambda}, \tag{7}$$
where $\lambda = \frac{1}{\mu_{\mathbb{X}_i^{f_{LA}}}}$ and $\mu_{\mathbb{X}_i^{f_{LA}}}$ the mean of $\mathbb{X}_i^{f_{LA}}$.

3. The dynamic behavior of the parameter c. This parameter represents the portion of clients that we do not discard. For example, a value of $c = 0.8$ means that the 20% of the clients will be discarded. With the aim of dynamically adapt it in each aggregation, we identify the problem of filtering out adversarial clients as a problem of outlier detection in $\mathbb{X}_i^{f_{LA}}$. We thus employ the Tukey criteria [43, 44] for anomalies in exponential probability distribution functions and set $c = 1 - \hat{c}$ where $\hat{c}$ is the portion of clients that verify
$$\mathbb{X}_i^{f_{LA}} \geq Q_3 + 1.5IQR = \frac{\ln(4)}{\lambda} + 1.5\frac{\ln(3)}{\lambda}, \tag{8}$$

where $\lambda = \frac{1}{\mu_{\mathbb{X}_i^{f_{LA}}}}$ and $\mu_{\mathbb{X}_i^{f_{LA}}}$ the mean of $\mathbb{X}_i^{f_{LA}}$.

4. Parameter y_b. It provides the weighting of the top clients together with b. In particular, it represents the portion of the total weight assigned to these clients. In order to weight the top clients with double the importance of the rest of the clients participating in the aggregation, we set

$$y_b = \frac{2|Top|}{2|Top| - |Rest|}, \tag{9}$$

where $|Top| = b \times n$ and $|Rest| = (c - b) \times n$.

Analogously to Equation 4, we obtain the weighting of each client from the Q_{a,b,c,y_b} function according to Equation .

$$w_i^{(a,b,c,y_b)} = Q_{a,b,c,y_b}\left(\frac{i}{n}\right) - Q_{a,b,c,y_b}\left(\frac{i-1}{n}\right) \tag{10}$$

3.3 Defense based on the federated aggregation

Finally, using the equations defined above and the definitions of FL (Equation 1), we define DDaBA as a defense consisting of the following aggregation operator:

$$DDaBA(\{\hat{L}_1^t, \hat{L}_2^t, \dots, \hat{L}_n^t\}, VD) = \sum_{i=1}^{n} w_i^{(a,b,c,y_b)} \hat{L}_i^t \tag{11}$$

where $w_i^{(a,b,c,y_b)}$ is defined in Equation 10 and $\hat{L_i^t}$ the local model update of the client i for $i \in \{1, \dots, n\}$. Algorithm 1 depicts the DDaBA pseudo-code.

4 Experimental set-up

The evaluation of DDaBA is performed by means of the accuracy of the resulting FL model in three datasets arranged for FL, and we describe them in Section 4.1. Also, we deployed an image classification deep learning model in the FL setting. Since the main aim of this work is to propose a dynamic defense against byzantine attacks, we use an standard CNN-based image classification model composed of two CNN layers followed by its corresponding max-pooling layers, a dense layer and the output layer with a softmax activation function for the Fed-EMNIST and Fashion MNIST datasets and a pre-tained model based on EfficientNet [45] for the CIFAR-10 dataset. Finally, the federated aggregation operators used as baselines are introduced in Section 4.2 and the covered attacks in Section 4.3.

Algorithm 1 DDaBA

```
  **Input:** local updates $\{\hat{L}_1^t, \hat{L}_2^t, \dots, \hat{L}_n^t\}$ and $VD$
  Initialize $G^t$
  **for** $i = 0$ **to** $n$ **do**
      $f_{LA}(L_i) = \text{accuracy}(L_i, VD)$
  **end for**
  **for** $i = 0$ **to** $n$ **do**
      $\mathbb{X}_i^{f_{LA}} = \max_i\{f_{LA}(L_i)\} - f_{LA}(L_i)$
  **end for**
  $a = 0$
  $b = |\mathbb{X}_i^{f_{LA}} \leq \frac{\ln(10/9)}{\lambda}|$
  $c = |\mathbb{X}_i^{f_{LA}} \geq \frac{\ln(4)}{\lambda} + 1.5\frac{\ln(3)}{\lambda}|$
  $y_b = \frac{2|b \times n|}{2|b \times n| - |(c-b) \times n|}$
  **for** $i = 0$ **to** $n$ **do**
      $w_i = w_i^{(a,b,c,y_b)}$ according to Equation 10.
  **end for**
  $G_t = \sum_{i=0}^{n} w_i \hat{L}_i^t$
  **Return** $G_t$
```

4.1 Evaluation datasets

Since the DDaBA needs a validation set for dynamically discarding adversarial clients, we create it from the test subsets of the three datasets, by assigning 20% of the sample in the test dataset to the validation set. The three datasets used in the evaluation are described as what follows:

1. The Fed-EMNIST (Federated Extended Modified NIST) dataset, which was presented in 2017 in [19] as an extension of the MNIST dataset [46]. The EMNIST Digits class contains a balanced subset of the digits dataset containing 28,000 samples of each digit. The dataset consists of 280,000 samples, which 240,000 are training samples and 40,000 test samples. We use its federated version by identifying each client with an original writer.

2. The Fashion MNIST [20] aims to be a more challenging replacement for the original MNSIT dataset. It contains a balanced subset of the 10 different classes containing 7,000 samples of each class. Hence, the dataset consists of 70,000 samples, which 60,000 are training samples and 10,000 test samples. We set the number of clients to 500.

3. The CIFAR-10 dataset is a labeled subset of the 80 million tiny images dataset [47]. It consists of 60000 32x32 color images in 10 classes, with 6000 images per class. There are 50000 training images and 10000 test images, which correspond to 1000 images of each class. We set the number of clients to 100.

In summary, the datasets, after appropriate modifications to prepare the validation sets, follow the data distributions shown in Table 1.

Table 1: Size of the training, validation and test sets of Fed-EMNIST, Fashion MNIST and CIFAR-10 datasets.

	Training	Validation	Test
Fed-EMNIST	240,000	8,000	32,000
Fashion MNIST	60,000	2,000	8,000
CIFAR-10	60,000	2,000	8,000

With the aim of adapting both Fashion MNIST and CIFAR-10 datasets to a federated environment, the training data is distributed among the clients following a non-IID distribution. Accordingly, we randomly assign instances of a reduced number of labels to each client simulating a scenario in which each client contains partial information.

4.2 Baselines based on federated aggregation operators

We compare the DDaBA defense with the classical federated aggregation operator FedAvg [48] and the following state-of-the-art defenses against byzantine poisoning attacks:

- *Median* [49]. It is one of the byzantine-robust aggregation rules which is based on replacing the mean with the median in the aggregation method, which is more robust against extreme values.

- *Trimmed-mean* [50]. It represents another byzantine-robust aggregation rule. It relies on using a more robust version of the mean that consists in eliminating a fixed percentage (k) of extreme values both below and above the data distribution.

- *Krum and Multikrum* [40]. It sorts the clients according to the geometric distances of their model updates distributions and chooses the one closest to the majority as the aggregated model. Multikrum incorporates an d parameter, which specifies the number of clients to be aggregated (the first d after being sorted) resulting in the aggregated model.

- *Bulyan* [41]. It represents the state-of-the-art combining Krum and the thrimmed-mean. Hence, it sorts the clients according to their geometric distances and, according to an f parameter, filters out the $2f$ clients of the tails of the sorted distribution of clients and aggregates the rest of them.

The main weakness of Multikrum and Bulyan is that they strongly depend on a parameter given by the user. Both are optimal if the number of adversarial clients is known, which is usually not the case.

4.3 Byzantine Data and Model Poisoning Attacks

There are multitude of byzantine adversarial attacks both data and model poisoning. Due to the high number of clients participating in the aggregation and the low proportion of clients that will be adversarial in a reasonable configuration, poisoning attacks are very ineffective as their effect dissipates in the aggregation. For that reason, poisoning attacks are combined with model-replacement [17] techniques, which weight the contribution of adversarial clients in the aggregation according to a boosting parameter that is distributed among the adversarial clients.

The adversarial attacks covered in this work are the following:

- *Label-flipping attack* [21]. It is a data poisoning attack consisting of randomly flipping the labels of the adversarial attacks. This way, the adversarial clients learn incorrect information that send to the server.
- *Out-of-distribution attack* [22]. It is another data poisoning attack consisting of introducing into the adversarial clients' training dataset some samples out of the training distribution. In practice, the most frequent approaches are to introduce samples from another dataset with the same features (e.g. EMNIST and Fashion MNIST) or to introduce randomly generated samples. We adopt the second approach in the experimentation.
- *Random weights* [23]. It is a model poisoning attack based on randomly generate the model updates of each adversarial client.

Table 2: Mean results for the label-flipping byzantine attack in terms of accuracy. We also show, in the first row, the expected accuracy with *FedAvg* but without any attack. The best result for each of the scenarios is highlighted in bold.

	Federated EMNIST			**Fashion MNIST**			**CIFAR-10**		
	1-out-of-30	**5-out-of-30**	**10-out-of-50**	**1-out-of-30**	**5-out-of-30**	**10-out-of-50**	**1-out-of-30**	**5-out-of-30**	**10-out-of-50**
No attack	0,9657	0,9657	0,9629	0,8719	0,8719	0,8697	0,8357	0,8357	0,8231
FedAvg	0,1591	0,4212	0,4007	0,1917	0,3665	0,4322	0,1184	0,1436	0,2448
Trim.-mean	0,9428	0,8739	0,8370	0,8672	0,8325	0,861	0,8239	0,7346	0,8220
Median	0,9313	0,9161	0,9097	0,8671	0,8473	0,8585	0,8287	0,8090	0,8289
Krum	0,8917	0,8706	0,8634	0,7264	0,7197	0,7473	0,7479	0,7610	0,7698
MultiKrum (5)	0,9132	0,9270	0,9189	0,8403	0,8433	0,8255	0,8164	0,8232	0,8114
MultiKrum (20)	0,9563	0,9571	0,9504	0,8727	0,8724	0,8680	0,8439	0,8479	0,8518
Bulyan (f=1)	0,9523	0,7813	0,5809	0,8689	0,7830	0,7875	0,8265	0,6595	0,6454
Bulyan (f=5)	0,9365	0,9421	0,9516	0,8617	0,8652	0,8726	0,8492	0,8451	0,8540
DDaBA	**0,9657**	**0,9663**	**0,9643**	**0,8817**	**0,8783**	**0,8807**	**0,8633**	**0,8503**	**0,8557**

We experiment with four different settings of adversarial clients for each of the previously described attacks:

- *1-out-of-30 attack scenario*. Consisting of 1 adversarial clients of a total of 30 clients participating in each aggregation, which represents 1/30 of adversarial clients.

Table 3: Mean results for the out-of-distribution byzantine attack in terms of accuracy. We also show, in the first row, the expected accuracy with *FedAvg* but without any attack. The best result for each of the scenarios is highlighted in bold.

	Federated EMNIST			**Fashion MNIST**			**CIFAR-10**		
	1-out-of-30	**5-out-of-30**	**10-out-of-50**	**1-out-of-30**	**5-out-of-30**	**10-out-of-50**	**1-out-of-30**	**5-out-of-30**	**10-out-of-50**
No attack	0,9657	0,9657	0,9629	0,8719	0,8719	0,8697	0,8357	0,8357	0,8231
FedAvg	0,4093	0,4404	0,4350	0,2041	0,3667	0,4657	0,1468	0,1922	0,3419
Trim.-mean	0,9456	0,8602	0,8531	0,8652	0,8348	0,8310	0,8202	0,7441	0,7400
Median	0,9345	0,9200	0,9144	0,8662	0,8465	0,8454	0,8223	0,8019	0,8073
Krum	0,8693	0,8668	0,8621	0,7361	0,7062	0,7281	0,7202	0,7310	0,7408
MultiKrum (5)	0,9169	0,9330	0,9198	0,8493	0,8430	0,8345	0,8305	0,8191	0,8023
MultiKrum (20)	0,9545	0,9544	0,9506	0,8747	0,8719	0,8733	0,8607	0,8519	0,8521
Bulyan (f=1)	0,9507	0,7872	0,5812	0,8704	0,7601	0,6930	0,8319	0,6862	0,5551
Bulyan (f=5)	0,9353	0,9383	0,9502	0,8713	0,8654	0,8757	0,8440	0,8498	0,8481
DDaBA	**0,9652**	**0,9620**	**0,9654**	**0,8761**	**0,8841**	**0,8783**	**0,8626**	**0,8599**	**0,8632**

Table 4: Mean results for the random weights byzantine attack in terms of accuracy. We also show, in the first row, the expected accuracy with *FedAvg* but without any attack. The best result for each of the scenarios is highlighted in bold.

	Federated EMNIST			**Fashion MNIST**			**CIFAR-10**		
	1-out-of-30	**5-out-of-30**	**10-out-of-50**	**1-out-of-30**	**5-out-of-30**	**10-out-of-50**	**1-out-of-30**	**5-out-of-30**	**10-out-of-50**
No attack	0,9657	0,9657	0,9629	0,8719	0,8719	0,8697	0,8357	0,8357	0,8231
FedAvg	0,0997	0,0994	0,1001	0,1006	0,1016	0,0997	0,0998	0,0994	0,1005
Trim.-mean	0,9537	0,1039	0,0990	0,8751	0,1004	0,0999	0,8608	0,0992	0,0998
Median	0,9367	0,9354	0,9342	0,8654	0,8618	0,8554	0,8499	0,8664	0,8646
Krum	0,8314	0,8652	0,8541	0,7156	0,7459	0,7342	0,7184	0,7164	0,7994
MultiKrum (5)	0,9325	0,9228	0,9191	0,8348	0,8343	0,8278	0,8164	0,8115	0,8167
MultiKrum (20)	0,9565	0,9577	0,9510	0,8764	0,8751	0,8676	0,8488	0,8488	0,8531
Bulyan (f=1)	0,9598	0,0997	0,0998	0,0990	0,1001	0,0990	0,8529	0,0996	0,0993
Bulyan (f=5)	0,9379	0,9377	0,9514	0,8746	0,8690	0,8746	0,8502	0,8411	0,8519
DDaBA	**0,9653**	**0,9645**	**0,9622**	**0,8801**	**0,8778**	**0,8777**	**0,8656**	**0,8624**	**0,8626**

- *5-out-of-30 attack scenario.* Consisting of 5 adversarial clients of a total of 30 clients participating in each aggregation, which represents 1/6 of adversarial clients.
- *10-out-of-50 attack scenario.* Consisting of 5 adversarial clients of a total of 50 clients participating in each aggregation, which represents 1/5 of adversarial clients.

In each of the scenarios described, the boosting factor is divided by the number of adversarial clients in order to carry out the model-replacement.

4.4 Implementation details

We provide the code of DDaBA federated aggregtion operator[4] in order to ensure the reproducibility of the experiments. Due to the large number of existing FL frameworks [51] and with the aim of showing that DDaBA is independent of the framework, we have selected two of them:

[4] `https://github.com/ari-dasci/S-DDaBA`

- The Sherpa.ai FL [51] framework.
- The Flower [52] framework.

For each framework, we include Jupyter notebooks to visualise how the aggregation operator works and to facilitate its understanding.

5 Experimental results

We evaluate the performance of DDaBA as a defense to the byzantine attacks described in Section 2.2 in two ways: (1) In Section 5.1, we compare the behavior of DDaBA in terms of the performance of the resulting FL model with the baselines described in Section 4.2 and, (2) In Section 5.2 we analyze DDaBD in a scenario with a high number of adversarial clients, and we propose a modification of it for this particular scenario.

5.1 Analysis of the results

Tables 2, 3 and 4 show the results obtained in label-flipping, out-of-distribution and random weights attacks. Regarding the strength of the attacks, we find that all three are sufficiently powerful to pose a challenge to defenses. In fact, notice that the attack is slightly more effective when there are fewer adversarial clients since the boosting factor is divided among fewer clients. The out-of-distribution attack is slightly less damaging while the random weights attack achieves the lowest performance without defense, ranking as the most challenging. The results obtained both in the different types of attacks and in the considered datasets confirm common conclusions, so we discuss all the results as a whole.

When evaluating the performance of the baselines we hereby confirm that MultiKrum and Bulyan do indeed represent the state of the art. However, they are highly dependent of the d and f parameters since they set the number of clients to keep or discard, respectively, in the aggregation. For example, in the 10-out-of-50 scenario and Bulyan with $f = 1$ we verify this weakness, since only $2f = 2$ clients would be discarded from the aggregation, which is not enough to defend the model in the presence of 10 adversarial clients. A possible solution would be to set this value always to high, but this is also a limitation because in the case of having fewer adversarial clients than $2f$ the quality of the model decreases (e.g., 1-out-of-30 using Bulyan with $f = 5$). Finally, MultiKrum and Bulyan promise optimal performance in the case of knowing the number of adversarial clients, which is not the case. This enhances the need for a defense that dynamically estimates how many clients to filter in the aggregation.

In contrast, the outperformance of DDaBA is confirmed in all the attack settings considered enhancing its success regardless of the type of attack and the proportion of adversarial clients. Moreover, DDaBA achieves better results than the no attack situation in the vast majority of the scenarios. This is because the dynamic filtering of clients not only discards those that are adversarial but also those that perform too poorly to contribute to improving the global learning model.

5.2 Extreme attack scenarios - A static version of DDaBA

It has been proven that discarding clients based on whether or not they are outliers in a distribution formed from performance on a common validation set overcomes the defenses of the state of the art. However, this approach based on data distributions has a weakness stemmed from the fact that the distribution we use to search outliers is configured with the same data that we subsequently evaluate. Therefore, with a very high presence of adversarial clients, the resulting distribution will be highly skewed by this data, resulting in no outlier. Although we recognize this weakness, we point out that it is not a major one, since it is highly unlikely for the percentage of adversarial clients in an FL scenario to be so high as to cause the defense to fail.

To overcome this weakness, we propose a static version of DDaBA called Static Defense Against Byzantine Attacks (SDaBA), which incorporates the only difference that the proportion of clients to be discarded from the aggregation is computed using a fixed parameter α. In particular, instead of eliminating those clients that represent outliers in the distribution $\mathbb{X}_i^{f_{LA}}$, we eliminate those clients whose distance to the best accuracy is greater than α times the maximum of the distances. In other words, using $\mathbb{X}_i^{f_{LA}}$, we set $c = 1 - \beta$ where β is the portion of clients verifying that

$$\mathbb{X}_i^{f_{LA}} \geq \alpha \mathbb{X}_n^{f_{LA}} \quad \forall i \in \{1, \dots, n\} \tag{12}$$

in Equations 6 and 10. Analogously, we set $b = 0.2$ in order to consider as top clients the top 20% clients.

With the aim of evaluating SDaBA we set $\alpha = 1/4$ and the 10-out-of-30 attack scenario consisting of 10 adversarial clients of a total of 30 clients participating in each aggregation, which represents 1/3 of adversarial clients, which is an unusual high proportion of them. Table 5 shows the results of DDaBA and SDaBA in comparison with the baselines in this extreme attack scenario in Federated EMNIST.

The results show how this extreme scenario highly affects to DDaBA, but also Bulyan (f=1). With respect to the baselines, in this case it is MultiKrum with $d = 20$ that achieves the best results by setting the d parameter to its optimal value. Finally, we highlight the appropriate performance of SDaBA, outperforming the rest of the defenses and solving the problem of extreme scenarios.

6 Conclusion and future work

We addressed the problem of defending against byzantine attacks in FL, which is a real challenge since the existing defenses are not enough. Using the exponential distribution resulting of the differences between the best model and the rest of them in terms of accuracy over a central validation set, we consider that those clients that represent outliers in that distribution are likely to be adversarial ones. Hence, we propose DDaBA, a defense against byzantine attacks which dynamically filters out the adversarial and poor clients.

Table 5: Mean results for the extreme scenario (10-out-of-30) in Federated EMNIST in terms of accuracy. We also show, in the first row, the expected accuracy with *FedAvg* but without any attack. The best result for each of the scenarios is highlighted in bold.

	Label-flipping	**Out-of-dist.**	**Random weights**
No attack	0,9657	0,9657	0,9657
FedAvg	0,3561	0,4394	0,0994
Trimmed-mean	0,6256	0,5778	0,1002
Median	0,8595	0,8347	0,9355
Krum	0,8801	0,8678	0,8633
MultiKrum (5)	0,9336	0,9366	0,9349
MultiKrum (20)	0,9623	0,9617	0,8595
MultiKrum (25)	0,9623	0,9617	0,8595
Bulyan (f=1)	0,4755	0,5005	0,1000
Bulyan (f=5)	0,9485	0,9475	0,9455
DDaBA	0,4235	0,4819	0,0997
SDaBA (1/4)	**0,9654**	**0,9653**	**0,9629**

We evaluated the DDaBA in three different byzantine attacks, in three datasets and using three different settings. In addition, we proposed a static version of the defense approach in order to use it in scenarios with an extremely high proportion of adversarial clients. Both the experiments corroborate the following conclusions:

- DDaBA is a highly effective defense against byzantine attacks in real attack scenarios.
- It properly filters out adversarial and poor clients improving the performance of the global model in scenarios with adversarial clients, even outperforming the performance in the original task.
- The static version SDaBA is an effective solution for extreme attack scenarios.

To conclude, we have proven that DDaBA is a high quality defense against byzantine attacks, and it can act as a proper federated aggregation operator, since it defends the global model against the effect of the attacks while improving the learning of the global model.

A Ordered weighted model averaging

Group decision making is the AI task focused on finding out a consensus decision from a set of experts by summing up their individual evaluations. Yager proposed in [42] the Ordered Weighted Averaging (OWA) operators with the aim of modelling the fuzzy opinion majority [53] in group decision making. Yager and Filev generalised the OWA operator definition in [18], where they defined the OWA operator with an order-induced vector for ordering the argument variable. They called this generalisation of OWA operators with a specific semantic in the aggregation process as Induced Ordered Weighted Averaging (IOWA). The OWA and IOWA operators are weighted aggregation functions that are mathematically defined as what follows:

Definition A.1 (OWA Operator [42]) *An OWA operator of dimension n is a function $\Phi : \mathbb{R}^n \rightarrow \mathbb{R}$ that has an associated set of weights or weighting vector $W = (w_1, \dots, w_n)$ so that $w_i \in [0,1]$ and $\sum_{i=1}^{n} w_i = 1$, and it is defined to aggregate a list of real values $\{c_1, \dots, c_n\}$ according to the Equation 13:*

$$\Phi(c_1, \dots, c_n) = \sum_{i=1}^{n} w_i c_{\sigma(i)} \tag{13}$$

being $\sigma : \{1, \dots, n\} \rightarrow \{1, \dots, n\}$ a permutation function such that $c_{\sigma(i)} \geq c_{\sigma(i+1)}$, $\forall i = \{1, \dots, n-1\}$.

Definition A.2 (IOWA Operator [18]) *An IOWA operator of dimension n is a mapping $\Psi : (\mathbb{R} \times \mathbb{R})^n \rightarrow \mathbb{R}$ which has an associated set of weights $W = (w_1, \dots, w_n)$ so that $w_i \in [0,1]$ and $\sum_{i=1}^{n} w_i = 1$, and it is defined to aggregate the second arguments of a 2-tuple list $\{\langle u_1, c_1 \rangle, \dots, \langle u_n, c_n \rangle\}$ according to the following expression:*

$$\Psi(\langle u_1, c_1 \rangle, \dots, \langle u_n, c_n \rangle) = \sum_{i=1}^{n} w_i c_{\sigma(i)} \tag{14}$$

being $\sigma : \{1, \dots, n\} \rightarrow \{1, \dots, n\}$ a permutation function such that $u_{\sigma(i)} \geq u_{\sigma(i+1)}$, $\forall i = \{1, \dots, n-1\}$. The vector of values $U = (u_1, \dots, u_n)$ is called the order-inducing vector and $(c_1, \dots, c_n)$ the values of the argument variable.

The OWA and IOWA operators are functions for weighting the contribution of experts for the global decision in the case of group decision making, and the contribution of a set of clients in an aggregation process in a general scenario. However, they need an additional function to calculate the values of the parameters, which in the context of group decision making means the grade of membership to a fuzzy concept. The weight value calculation function is known as linguistic quantifier [54], which is defined as a function $Q : [0,1] \rightarrow [0,1]$ such as $Q(0) = 0$, $Q(1) = 1$ and $Q(x) \geq Q(y)$ for $x > y$. Equation 15 defines how the function Q computes the weight values and Equation 16 defines the behaviour of the function Q.

$$w_i^{(a,b)} = Q_{a,b}\left(\frac{i}{n}\right) - Q_{a,b}\left(\frac{i-1}{n}\right) \tag{15}$$

$$Q_{a,b}(x) = \begin{cases} 0 & 0 \leq x \leq a \\ \dfrac{x-a}{b-a} & a \leq x \leq b \\ 1 & b \leq x \leq 1 \end{cases} \tag{16}$$

where $a, b \in [0,1]$ satisfying $0 \leq a \leq b \leq 1$.

The function Q in Equation 16 can be redefined in order to model different linguistic quantifiers. Since the definition of the notion quantifier guided aggregation [42, 54], other definitions of the function Q has been proposed to model different linguistic quantifiers like "most" or "at least" [53].

Acknowledgements

This research work is supported by the R&D&I grants PID2020-119478GB-I00, PID2020-116118GA-I00 and EQC2018-005084-P funded by MCIN/AEI/10.13039/501100011033 and by "ERDF A way of making Europe". Nuria Rodríguez Barroso were supported by the grant FPU18/04475 funded by MCIN/AEI/10.13039/501100011033 and by "ERDF A way of making Europe". Eugenio Martínez Cámara were supported by the grant IJC2018-036092-I funded by MCIN/AEI/10.13039/501100011033.

References

[1] J. Dean, G. Corrado, R. Monga, C. Kai, M. Devin, M. Mao, M. Ranzato, A. Senior, P. Tucker, K. Yang, Q.V. Le, and A.Y. Ng. Large scale distributed deep networks. *Advances in Neural Information Processing Systems 25*, pages 1223–1231, 2012.

[2] C. Ma, J. Konečný, M. Jaggi, V. Smith, M.I. Jordan, P. Richtárik, and M. Takáč. Distributed optimization with arbitrary local solvers. *Optimization Methods and Software*, 32(4):813–848, 2017.

[3] Peter Kairouz and H. Brendan McMahan. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, 14(1), 2021.

[4] Ming Chen, Bingcheng Mao, and Tianyi Ma. Fedsa: A staleness-aware asynchronous federated learning algorithm with non-iid data. *Future Generation Computer Systems*, 120:1–12, 2021.

[5] 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.

[6] Xinqian Zhang, Ming Hu, Jun Xia, Tongquan Wei, Mingsong Chen, and Shiyan Hu. Efficient federated learning for cloud-based AIoT applications. In *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2020.

[7] Nilesh Dalvi, Pedro Domingos, Mausam, Sumit Sanghai, and Deepak Verma. Adversarial classification. *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, page 99–108, 2004.

[8] Ling Huang, Anthony D. Joseph, Blaine Nelson, Benjamin I.P. Rubinstein, and J. D. Tygar. Adversarial machine learning. *Proceedings of the 4th ACM Workshop on Security and Artificial Intelligence*, page 43–58, 2011.

[9] Battista Biggio, Igino Corona, Blaine Nelson, Benjamin I. P. Rubinstein, Davide Maiorca, Giorgio Fumera, Giorgio Giacinto, and Fabio Roli. *Security Evaluation of Support Vector Machines in Adversarial Environments*, pages 105–153. 2014.

[10] T. Gu, K. Liu, B. Dolan-Gavitt, and S. Garg. Badnets: Evaluating backdooring attacks on deep neural networks. *IEEE Access*, 7:47230–47244, 2019.

[11] M. Jagielski, A. Oprea, B. Biggio, C. Liu, C. Nita-Rotaru, and B. Li. Manipulating machine learning: Poisoning attacks and countermeasures for regression learning. *IEEE Symposium on Security and Privacy (SP)*, pages 19–35, 2018.

[12] Arjun Nitin Bhagoji, Supriyo Chakraborty, Prateek Mittal, and Seraphin Calo. Model poisoning attacks in federated learning. In *In Workshop on Security in Machine Learning (SecML), collocated with the 32nd Conference on Neural Information Processing Systems (NeurIPS'18)*, 2018.

[13] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-Efficient Learning of Deep Networks from Decentralized Data. 54:1273–1282, 2017.

[14] J. Konečný, H.B. McMahan, D. Ramage, and P. Richtárik. Federated optimization: Distributed machine learning for on-device intelligence. *CoRR*, abs/1511.03575, 2016.

[15] Y. Wang. Co-op: Cooperative machine learning from mobile devices. Master's thesis, University of Alberta, 2017.

[16] C. Fung, C. J. M. Yoon, and I. Beschastnikh. Mitigating sybils in federated learning poisoning. *CoRR*, abs/1808.04866, 2018.

[17] Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. How to backdoor federated learning. *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, 108:2938–2948, 2020.

[18] R.R. Yager and D.P. Filev. Induced ordered weighted averaging operators. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 29(2):141–150, 1999.

[19] Gregory Cohen, Saeed Afshar, Jonathan Tapson, and André van Schaik. Emnist: Extending mnist to handwritten letters. *International Joint Conference on Neural Networks (IJCNN)*, pages 2921–2926, 2017.

[20] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *CoRR*, abs/1708.07747, 2017.

[21] Vale Tolpegin, Stacey Truex, Mehmet Emre Gursoy, and Ling Liu. Data poisoning attacks against federated learning systems. *CoRR*, abs/2007.08432, 2020.

[22] Stanislav Fort, Jie Ren, and Balaji Lakshminarayanan. Exploring the limits of out-of-distribution detection. *CoRR*, abs/2106.03004, 2021.

[23] Jierui Lin, Min Du, and Jian Liu. Free-riders in federated learning: Attacks and defenses. *CoRR*, abs/1911.12560, 2019.

[24] J. Konečný, H.B. McMahan, F. X. Yu, P. Richtárik, A.T. Suresh, and D. Bacon. Federated learning: Strategies for improving communication efficiency. *CoRR*, abs/1610.05492, 2016.

[25] Pavel Laskov and Richard Lippmann. Machine learning in adversarial environments. *Machine Learning*, 81:115–119, 2010.

[26] Nilesh Dalvi, Pedro Domingos, Mausam, Sumit Sanghai, and Deepak Verma. Adversarial classification. *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, page 99–108, 2004.

[27] Blaine Nelson, Marco Barreno, Fuching Jack Chi, Anthony D. Joseph, Benjamin I. P. Rubinstein, Udam Saini, Charles Sutton, J. D. Tygar, and Kai Xia. *Misleading Learners: Co-opting Your Spam Filter*, pages 17–51. 2009.

[28] C. Croux, P. Filzmoser, and M.R. Oliveira. Algorithms for projection–pursuit robust principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, 87(2):218–225, 2007.

[29] Lingjuan Lyu, Han Yu, Xingjun Ma, Lichao Sun, Jun Zhao, Qiang Yang, and Philip S. Yu. Privacy and robustness in federated learning: Attacks and defenses. *CoRR*, abs/2012.06337, 2020.

[30] Milad Nasr, Reza Shokri, and Amir Houmansadr. Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. *IEEE Symposium on Security and Privacy (SP)*, pages 739–753, 2019.

[31] Di Cao, Shan Chang, Zhijian Lin, Guohua Liu, and Donghong Sun. Understanding distributed poisoning attack in federated learning. *IEEE 25th International Conference on Parallel and Distributed Systems (ICPADS)*, pages 233–239, 2019.

[32] Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.

[33] Xingchen Zhou, Ming Xu, Yiming Wu, and Ning Zheng. Deep model poisoning attack on federated learning. *Future Internet*, 13(3), 2021.

[34] Gan Sun, Yang Cong, Jiahua Dong, Qiang Wang, and Ji Liu. Data poisoning attacks on federated machine learning. *CoRR*, abs/2004.10020, 2020.

[35] Leslie Lamport, Robert Shostak, and Marshall Pease. *The Byzantine Generals Problem*, page 203–226. Association for Computing Machinery, 2019.

[36] Jacob Steinhardt, Pang Wei Koh, and Percy Liang. Certified defenses for data poisoning attacks. *Proceedings of the 31st International Conference on Neural Information Processing Systems*, page 3520–3532, 2017.

[37] Muhammad Shayan, Clement Fung, Chris J. M. Yoon, and Ivan Beschastnikh. Biscotti: A ledger for private and secure peer-to-peer machine learning. *CoRR*, abs/1811.09904, 2018.

[38] Shiqi Shen, S. Tople, and P. Saxena. Auror: defending against poisoning attacks in collaborative deep learning systems. *Proceedings of the 32nd Annual Conference on Computer Security Applications*, pages 508–519, 2016.

[39] Dong Yin, Yudong Chen, Ramchandran Kannan, and Peter Bartlett. Byzantine-robust distributed learning: Towards optimal statistical rates. *Proceedings of the 35th International Conference on Machine Learning*, 80:5650–5659, 2018.

[40] 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*, 30:119–129, 2017.

[41] El Mahdi El Mhamdi, Rachid Guerraoui, and Sébastien Rouault. The hidden vulnerability of distributed learning in Byzantium. *Proceedings of the 35th International Conference on Machine Learning*, 80:3521–3530, 10–15 Jul 2018.

[42] R.R. Yager. On ordered weighted averaging aggregation operators in multicriteria decisionmaking. *IEEE Transactions on Systems, Man, and Cybernetics*, 18(1):183–190, 1988.

[43] John W. Tukey. *Exploratory Data Analysis*. Addison-Wesley, 1977.

[44] Kazumi Wada. Outliers in official statistics. *Japanese Journal of Statistics and Data Science*, 3, 10 2020.

[45] Mingxing Tan and Quoc Le. EfficientNet: Rethinking model scaling for convolutional neural networks. In *Proceedings of the 36th International journal on Machine Learning*, volume 97, pages 6105–6114, 2019.

[46] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[47] Antonio Torralba, Rob Fergus, and William T. Freeman. 80 million tiny images: A large data set for nonparametric object and scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(11):1958–1970, 2008.

[48] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. Agüera y Arcas. Communication-Efficient Learning of Deep Networks from Decentralized Data. volume 54, pages 1273–1282, 2017.

[49] Yudong Chen, Lili Su, and Jiaming Xu. Distributed statistical machine learning in adversarial settings: Byzantine gradient descent. volume 1, December 2017.

[50] Dong Yin, Yudong Chen, Kannan Ramchandran, and Peter L. Bartlett. Byzantine-robust distributed learning: Towards optimal statistical rates. *CoRR*, abs/1803.01498, 2018.

[51] Nuria Rodríguez-Barroso, Goran Stipcich, Daniel Jiménez-López, José Antonio Ruiz-Millán, Eugenio Martínez-Cámara, Gerardo González-Seco, M. Victoria Luzón, Miguel Ángel Veganzones, and Francisco Herrera. Federated learning and differential privacy: Software tools analysis, the Sherpa.ai FL framework and methodological guidelines for preserving data privacy. *Information Fusion*, 64:270 – 292, 2020.

[52] Daniel J Beutel, Taner Topal, Akhil Mathur, Xinchi Qiu, Titouan Parcollet, and Nicholas D Lane. Flower: A friendly federated learning research framework. *arXiv preprint arXiv:2007.14390*, 2020.

[53] G. Pasi and R.R. Yager. Modeling the concept of majority opinion in group decision making. *Information Science*, 176(4):390–414, 2006.

[54] R.R. Yager. Quantifier guided aggregation using owa operators. *International Journal of Intelligent Systems*, 11(1):49–73, 1996.

Chapter III

Trabajo actual:
Un enfoque más justo y explicable

«If you can’t give me poetry,
can’t you give me “poetical science”?».
– Ada Lovelace.

1 Introducción

En este último capítulo de la tesis, desarrollamos el trabajo actual que se está realizando. Lo hacemos a modo de trabajo científico para seguir con la estructura del anterior capítulo de publicaciones. En esta sección motivamos y justificamos el problema a abordar, que es la línea principal de la tesis, pero con un matiz en los objetivos. El resto de secciones se estructura de la siguiente manera. En la Sección 2 explicamos la propuesta en profundidad y el marco experimental. En la Sección 3 presentamos los resultados empíricos y el análisis de los mismos. Finalmente, en la Sección 4 destacamos las conclusiones.

1.1 Motivación

El uso de aplicaciones de AI en ámbitos cotidianos como la producción y transporte, energía o incluso educación, ha llegado para quedarse. Sin embargo, en la misma medida que crecen el número de aplicaciones de AI, mayor es el riesgo producido por estas [AIA21], sobre todo cuando su aplicación es en ámbitos delicados como la educación, la justicia, la cultura o la democracia. Por tanto, se hace imprescindible desarrollar sistemas de AIque aseguren resiliencia, seguridad, transparencia, equidad, respecto a la privacidad, autonomía, trazabilidad y auditabilidad, mientras que mantienen el buen rendimiento [Win21].

En este contexto, surgen soluciones como el ya presentado FL [YLC+19] motivado por la necesidad de mantener la privacidad de los datos al mismo tiempo que se permite el entrenamiento colaborativo de modelos de AI. Consiste en un paradigma de aprendizaje distribuido en el que varios nodos llamados clientes colaboran de forma que el entrenamiento se lleva a cabo en estos nodos, no siendo sus datos accesibles de ninguna forma. Este entrenamiento está coordinado por un servidor global que accede al conocimiento obtenido por cada cliente (nunca a los datos), agregando este conocimiento en un modelo global.

Sin embargo, como cualquier paradigma de aprendizaje automático, es vulnerable a ataques adversarios [RBJLL+23] que tienen como objetivo modificar el funcionamiento del modelo, o inferir información privada sobre los datos. Este tipo de ataques suponen un reto en FL dado que la mayoría de los mecanismos de defensa frente a este tipo de ataques en aprendizaje automático clásico que encontramos en la literatura se basan en la inspección de datos [CAD+21]. Al no ser esto posible en FL [LVN23], se tienen que diseñar mecanismos de defensa ad-hoc o adaptar los mecanismos de defensa ya existentes. En este trabajo nos centramos en proponer una defensa frente a ataques al modelo por envenenamiento de datos aleatorio, también conocidos como ataques bizantinos.

En la literatura existen multitud de propuestas de mecanismos de defensa que muestran resultados prometedores. Sin embargo, estas propuestas muestran varias debilidades entre las que destacamos:

- La mayoría de ellos no son capaces de distinguir a los clientes que son adversarios (están llevando a cabo un ataque por envenenamiento), de aquellos clientes denominados *pobres* (cuya distribución de datos está sesgada parcialmente). Esto puede ser un problema dado que rompe el principio de equidad, además de que se puede estar descartando información de interés.

- La mayoría de las propuestas están centradas en mejorar el rendimiento del modelo, mientras que no se centran en la transparencia o explicabilidad del filtrado de clientes.

1.2 Justificación

En este trabajo alegamos que es posible desarrollar un mecanismo de defensa capaz de proteger al sistema de FL asegurando así el buen funcionamiento y la preservación de la privacidad de los datos, al mismo tiempo que aporta otros requisitos deseables como equidad, transparencia y explicabilidad [BDM$^+$23, DSB$^+$23]. Para ello, nos basamos en nuestra anterior propuesta DDaBA [RBMCLH22], un mecanismo de defensa que ha demostrado tener un buen rendimiento frente a ataques bizantinos y modificamos las decisiones tomadas basadas en el rendimiento del modelo por decisiones basadas en medidas de explicabilidad. Para ello, usaremos modelos lineales de explicaciones locales (LLEs) [SGLH$^+$22, DK20] basados en la importancia de cada característica de los datos a la hora de tomar decisiones en la clasificación. Por este motivo, llamamos a la propuesta FTX-DDaBA (*Fair, Transparent and eXplainable DDaBA*).

Para testear el funcionamiento del modelo propuesto, utilizaremos como conjuntos de datos de clasificación de imágenes: Fed-EMNIST [LBBH98] y Fashion MNIST [XRV17]. Además, implementaremos dos tipos de ataques de envenenamiento de datos basados en intercambio aleatorio de etiquetas [TTGL20]. Además, nos compararemos con diferentes modelos base de la literatura. Para cada experimento mostraremos tanto medidas de rendimiento como un análisis más profundo de los datos en varios sentidos: (1) por un lado, mediremos la presencia de clientes pobres descartados durante las rondas de aprendizaje, y (2) por otro lado, analizaremos de forma visual las explicaciones del modelo para el filtrado de clientes adversarios.

2 Metodología y descripción de la propuesta

En esta sección desarrollamos formalmente la propuesta de un mecanismo de defensa frente a ataques bizantinos que sea justo con la participación de los clientes, que aporte una explicación del filtrado de los mismos y que mantenga el buen rendimiento del modelo en la Sección 2.1. Posteriormente, detallamos el entorno experimental en el que vamos a testear el funcionamiento de la propuesta en la Sección 2.2.

2.1 Propuesta de mecanismo de defensa justo y explicable: FX-DDaBA

Para ello, partimos del mecanismo de defensa frente a ataques bizantinos propuesto en la tesis DDaBA [RBMCLH22]. Este mecanismo de defensa consiste en un agregador que se basa en unos cuantificadores lingüísticos [Yag96] y operadores IOWA (*Induced Ordered Weighted Averaging*) [Yag88, YF99] que ponderan la participación de cada cliente, asignando 0 a los clientes que considera candidatos a descartar, y diferenciando entre clientes normales y clientes *top* en el resto de clientes. De esta forma habría tres tipos de ponderaciones:

- Ponderación nula para los clientes a descartar.
- Ponderación alta para los clientes considerados los mejores.
- Ponderación normal para el resto.

Como el funcionamiento de este mecanismo de defensa está desarrollado en profundidad en el capítulo anterior, en esta sección no entraremos en detalles de su diseño, si no que nos limitaremos a destacar sus diferencias. El funcionamiento de este mecanismo de defensa basado en un agregador reside en la ordenación de los clientes según su rendimiento (en términos de *accuracy*) en un conjunto de validación situado en el servidor. En nuestra propuesta, nosotros cambiamos esta función de ordenación para cada cliente i cuyo modelo local está representado por los parámetros L_i por la siguiente función

$$f_{LE}(L_i, VD) = \frac{\sum_{v \in VD} \text{cosine\_similarity}(A^p_{i,v}, A^p_{j,v})}{|\sum_{v \in VD} \text{cosine\_similarity}(A^p_{i,v}, A^p_{j,v})|}, \quad \forall L_j \in C, \tag{10}$$

donde C es el conjunto de todas las actualizaciones de los clientes, $A^p_{i,v}$ es la LLE asociada al modelo L_i para cada muestra de v de VD y $|\cdot|$ la norma. Finalmente, la función de agregación del servidor quedaría de la siguiente forma:

$$\textbf{FX-DDaBA}(\{\hat{L}^t_1, \hat{L}^t_2, \ldots, \hat{L}^t_n\}, \text{VD}) = \sum_{i=1}^{n} w_i^{(a,b,c,y_b)} \hat{L}^t_i, \tag{11}$$

donde $w_i^{(a,b,c,y_b)}$ se corresponde con los pesos proporcionados con el cuantificador lingüístico descrito en [RBMCLH22], y $\hat{L_i^t}$ son los parámetros del modelo del cliente i para $i \in \{1, \dots, n\}$, y VD es un pequeño conjunto de validación en test.

De esta forma, nos beneficiamos del buen funcionamiento de partida del mecanismo de defensa DDaBA, al mismo tiempo que se garantiza que FTX-DDaBA es:

- **Justo**: dado que no va a eliminar a clientes pobres en función del mal rendimiento si no que inspecciona más allá centrando el foco en qué características se consideran importantes.
- **Explicable**: dado que al estar basado en explicaciones locales podemos inspeccionar las explicaciones generadas para los clientes descartados.

2.2 Entorno experimental

En esta sección detallamos el entorno experimental en el que se van a desarrollar las pruebas de la propuesta. Dado que el objetivo principal es comprobar su buen funcionamiento como mecanismo de defensa, nos centramos en problemas de clasificación de imágenes y usamos un modelo de aprendizaje profundo sencillo basado en dos capas CNNs (*Convolutional Neural Networks*) [KLSH21] seguidas de capas densas y capas de salida.

Conjuntos de datos Para la experimentación usamos dos conjuntos de datos de clasificación de imágenes ampliamente utilizados en la literatura:

- Fed-EMNIST [LBBH98] Digits. Que contiene conjuntos balanceados de imágenes de dígitos en blanco y negro. El conjunto está compuesto de 240.000 muestras de entrenamiento, y 40.000 de test (de las cuales 8.000 las destinamos al conjunto de validación en el servidor).
- Fashion MNSIT [XRV17], que contiene conjuntos balanceados de imágenes en blanco y negro de prendas de vestir. El conjunto está formado por 60.000 muestras de entrenamiento y 10.000 de test (de las cuales 2.000 las destinamos al conjunto de validación en el servidor).

Cuadro 1: Tamaño de los conjuntos de entrenamiento, validación y test en los conjuntos de datos Fed-EMNIST y Fashion MNIST.

	Entrenamiento	**Validación**	**Test**
Fed-EMNIST	240,000	8,000	32,000
Fashion MNIST	60,000	2,000	8,000

En el Cuadro 1 se muestran los tamaños de los conjuntos de entrenamiento, validación y test de los conjuntos de datos utilizados.

Modelos base Además del modelo de partida, DDaBA, como modelos base vamos a usar dos operadores de agregación más robustos que la media aritmética, así como otros mecanismos de defensa frente a ataques bizantinos ampliamente utilizados en la literatura:

- *Mediana* [CSX17]. Que es más robusto frente a datos muy extremos.
- *Media truncada* [YCRB18]. Que no se ve afectada por datos extremos pues se truncan antes de la agregación.
- *Multikrum* [BEMGS17]. Ordena los clientes en función de las distancias geométricas de las actualizaciones de sus modelos. Por lo tanto, emplea un parámetro de agregación d, que especifica el número de clientes que se agregarán (los primeros d después de ser ordenados).
- *Bulyan* [EMGR18]. Combina Multikrum y la media truncada. Por lo tanto, ordena los clientes en función de sus distancias geométricas y, según un parámetro f, filtra los $2f$ clientes de las colas de la distribución ordenada de clientes y agrega el resto de ellos.

Envenenamiento de los datos Existen diferentes tipos de ataques adversarios por envenenamiento de datos. En este caso, nos vamos a centrar en el más usado en la literatura por su facilidad de implementación y buenos resultados proporcionados. Este ataque es el intercambio aleatorio de etiquetas (*label-flipping*) [JDFSBJ22], y consiste básicamente en mezclar de forma aleatoria las etiquetas asociadas a los datos de entrenamiento de los clientes que queremos que sean adversarios, haciendo así que aprendan información errónea al entrenar sobre datos erróneos, y transmitan esta información errónea al servidor.

Clientes De cara a comprobar que FTX-DDaBA es capaz de actuar como mecanismo de defensa, al mismo tiempo que no elimina a clientes pobres, vamos a establecer que en la simulación haya:

- *Clientes adversarios*: que implementan el envenenamiento de datos descrito arriba. En cada agregación participan 5.
- *Clientes pobres*: que solo contienen la mitad de las etiquetas (en concreto, solo las etiquetas pares). En cada agregación participan 5.
- *Clientes normales*: En cada agregación participan 20.

3 Resultados

En esta sección mostramos los resultados experimentales que avalan nuestra propuesta. En un primer lugar mostraremos los resultados de rendimiento de nuestra propuesta junto con los modelos base seleccionados. Además, realizamos dos análisis, un primer análisis en el que se prueba la cualidad de equidad y justicia del agregador, y un segundo en el que mostramos las explicaciones proporcionadas por el modelo.

Resultados en términos de rendimiento En el Cuadro 2 mostramos los datos de rendimiento obtenidos por el modelo en términos de *accuracy*. Obtenemos que FTX-DDaBA, al igual que DDaBA supera con creces a los modelos base utilizados. Además, a pesar de no utilizar el *accuracy* como criterio para seleccionar los clientes (como hacía DDaBA) vemos que las pérdidas en rendimiento han sido mínimas.

Cuadro 2: Resultados en términos de *accuracy*. Se muestran los resultados medios de 3 ejecuciones. Marcamos en negrita el mejor resultado en cada caso.

	Fed-EMNIST	**Fashion MNIST**
Mediana	0.9298	0.8424
Media truncada	0.9428	0.8391
MultiKrum	0.9370	0.8433
Bulyan	0.9493	0.8665
DDaBA	**0.9663**	**0.8809**
FTX-DDaBA	0.9654	0.8798

Resultados en términos de equidad y justicia Dado que uno de los objetivos era obtener un mecanismo de defensa que no descarte a los clientes pobres por tener un rendimiento menor por no conocer determinadas clases, en los Cuadros 3 y 4 mostramos el número mínimo, máximo y medio de clientes tanto pobres como adversarios descartados a lo largo de las rondas de aprendizaje en Fed-EMNIST y Fashion MNIST, respectivamente. En ambos conjuntos de datos vemos que la tendencia es la misma:

1. Con respecto a los clientes adversarios, ambos mecanismos de defensa son muy buenos descartándolos, sin apenas diferencias significativas.
2. Con respecto a los clientes pobres, DDaBA filtra, en media, bastantes más clientes pobres. De esta forma FTX-DDaBA fomenta una participación más justa y equitativa de los clientes. De hecho, DDaBA descarta en media aproximadamente a un cliente pobre por ronda de aprendizaje.

Cuadro 3: Métricas sobre filtrado de clientes pobres y adversarios a lo largo de las rondas de aprendizaje (mínimo, máximo y número medio de clientes adversarios y pobres, respectivamente, filtrados) en Fed-EMNIST.

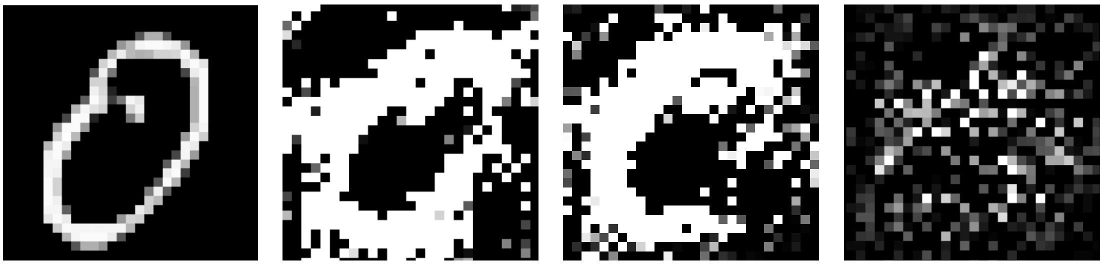
	Fed-EMNIST					
	Adversarios			**Pobres**		
	Min	**Max**	**Medio**	**Min**	**Max**	**Medio**
DDaBA	3	5	**4,92**	0	5	0,93
FTX-DDaBa	3	5	4,43	0	**2**	**0,12**

Cuadro 4: Métricas sobre filtrado de clientes pobres y adversarios a lo largo de las rondas de aprendizaje (mínimo, máximo y número medio de clientes adversarios y pobres, respectivamente, filtrados) en Fashion MNIST.

	Fashion MNIST					
	Adversarios			**Pobres**		
	Min	**Max**	**Medio**	**Min**	**Max**	**Medio**
DDaBA	2	5	4,87	0	5	1,18
FTX-DDaBa	**3**	5	**4,95**	0	3	**0,26**

Explicabilidad y transparencia de los resultados El eje central de FTX-DDaBA es que la selección de clientes se basa en las LLEs. Por ello, podemos obtener transparencia en el proceso y explicaciones sobre por qué un cliente ha sido descartado o no. Como los LLEs se basan en la importancia de las características, podemos representar en una imagen la importancia de cada característica y comprobar si el modelo presta atención a las zonas de la imagen que contienen información o no. Se trata de una justificación muy intuitiva de por qué se ha descartado o no a un cliente. Aunque esto puede obtenerse para cada imagen del conjunto de validación, en la Figura 1 se muestra un ejemplo de estas explicaciones.

Obtenemos que, aunque el cliente normal se ajusta más a las zonas relevantes (la circunferencia) que el cliente adversario, ambos se ajustan bastante. Sin embargo, si tenemos en cuenta las características que el modelo de cliente adversario considera importantes, observamos que son prácticamente aleatorias. De este modo, podemos explicar de forma visual para un ser humano, dado que se ha hecho de forma transparente, por qué se ha filtrado a cada cliente que se considera adversario.



(a) Imagen original. (b) Exp. cliente normal. (c) Exp. cliente pobre. (d) Exp. cliente adversario.

Figura 1: Ejemplo de una muestra (a), una explicación obtenida de un cliente normal (b) y una explicación obtenida de un cliente obre (c), y una explicación obtenida de un cliente adversario (d).

4 Conclusiones

En este trabajo se ha investigado la posibilidad de desarrollar un mecanismo de defensa frente ataques bizantinos por envenenamiento de datos en FL que, al mismo tiempo que mantiene el buen rendimiento como defensa, aporte otras cualidades como justicia con los clientes pobres, o explicabilidad de los resultados. Para ello, partimos de un mecanismo de defensa de la literatura DDaBA y modificamos la parte basada en ordenación de clientes por rendimiento por una ordenación de clientes basada en explicaciones locales obteniendo FTX-DDaBA. Las conclusiones obtenidas son:

- FTX-DDaBA es capaz de obtener resultados competitivos en cuanto a rendimiento con respecto a DDaBA y, sobre todo, con respecto al resto de modelos base, a pesar de no basar la ordenación de clientes en el rendimiento obtenido por los mismos. Esto muestra que se pueden obtener buenos resultados maximizando otras características deseables.

- FTX-DDaBA filtra, en media, muchos menos clientes pobres en comparación con DDaBA, mientras que el filtrado de clientes adversarios es equivalente. De esta forma se es justo con los clientes pobres, al mismo tiempo que no se descarta información potencialmente útil.

- FTX-DDaBA proporciona explicaciones visuales de por qué se ha filtrado o no a un cliente, pudiendo se posteriormente supervisado y auditado.

En conclusión, se ha propuesto un mecanismo de defensa que, a pesar de alejar el objetivo del rendimiento a otras cualidades deseables en un sistema de AI, sigue previniendo al esquema de FL de ataques adversarios al mismo tiempo que maximiza estas otras cualidades para ser un mecanismo de AI confiable.

Bibliography

[ABHKS17] Abouelmehdi K., Beni-Hssane A., Khaloufi H., and Saadi M. (2017) Big data security and privacy in healthcare: A review. *Procedia Computer Science* 113: 73–80.

[ADRDS+20] Arrieta A. B., Díaz-Rodríguez N., Del Ser J., Bennetot A., Tabik S., Barbado A., García S., Gil-López S., Molina D., Benjamins R., *et al.* (2020) Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information fusion* 58: 82–115.

[AIA21] Artificial Intelligence Act. (2021) European comision. Regulation of the European Parliament and of the Council laying down harmonised rules on AI and amending certain union legislative acts.

[ARP+21] Alazab M., RM S. P., Parimala M., Maddikunta P. K. R., Gadekallu T. R., and Pham Q.-V. (2021) Federated learning for cybersecurity: Concepts, challenges, and future directions. *IEEE Transactions on Industrial Informatics* 18(5): 3501–3509.

[BCM+18] Brisimi T. S., Chen R., Mela T., Olshevsky A., Paschalidis I. C., and Shi W. (2018) Federated learning of predictive models from federated electronic health records. *International Journal of Medical Informatics* 112: 59 – 67.

[BCMC19] Bhagoji A. N., Chakraborty S., Mittal P., and Calo S. (2019) Analyzing federated learning through an adversarial lens. In: *International Conference on Machine Learning*, pp. 634–643. Proceedings of Machine Learning Research.

[BDM+23] Bárcena J. L. C., Ducange P., Marcelloni F., Nardini G., Noferi A., Renda A., Ruffini F., Schiavo A., Stea G., and Virdis A. (2023) Enabling federated learning of explainable AI models within beyond-5G/6G networks. *Computer Communications.* 210: 356–375.

[BEMGS17] Blanchard P., El Mhamdi E. M., Guerraoui R., and Stainer J. (2017) Machine Learning with Adversaries: Byzantine Tolerant Gradient Descent. *Advances in Neural Information Processing Systems.* 30: 119–129.

[BVH+20] Bagdasaryan E., Veit A., Hua Y., Estrin D., and Shmatikov V. (2020) How to backdoor federated learning. In: *International conference on artificial intelligence and statistics*, pp. 2938–2948. Proceedings of Machine Learning Research.

[CAD+21] Chakraborty A., Alam M., Dey V., Chattopadhyay A., and Mukhopadhyay D. (2021) A survey on adversarial attacks and defences. *CAAI Transactions on Intelligence Technology* 6(1): 25–45.

[CCI+22] Criado M. F., Casado F. E., Iglesias R., Regueiro C. V., and Barro S. (2022) Non-IID data and Continual Learning processes in Federated Learning: A long road ahead. *Information Fusion* 88: 263–280.

[CLC+22] Casado F. E., Lema D., Criado M. F., Iglesias R., Regueiro C. V., and Barro S. (2022) Concept drift detection and adaptation for federated and continual learning. *Multimedia Tools and Applications* pp. 1–23.

[CSX17] Chen Y., Su L., and Xu J. (2017) Distributed Statistical Machine Learning in Adversarial Settings: Byzantine Gradient Descent. *Proc. Association for Computing Machinery Meas. Anal. Comput. Syst.* 1(2): 1–25.

[DK20] Dieber J. and Kirrane S. (2020) Why model why? Assessing the strengths and limitations of LIME. *CoRR* abs/2012.00093.

[DMNS06] Dwork C., McSherry F., Nissim K., and Smith A. (2006) Calibrating Noise to Sensitivity in Private Data Analysis. In: *Theory of Cryptography*, pp. 265–284.

[DRDSC+23] Díaz-Rodríguez N., Del Ser J., Coeckelbergh M., de Prado M. L., Herrera-Viedma E., and Herrera F. (2023) Connecting the dots in trustworthy Artificial Intelligence: From AI principles, ethics, and key requirements to responsible AI systems and regulation. *Information Fusion* page 101896.

[DSB+23] Daole M., Schiavo A., Bárcena J. L. C., Ducange P., Marcelloni F., and Renda A. (2023) OpenFL-XAI: Federated learning of explainable artificial intelligence models in Python. *SoftwareX,* 23: 101505.

[EMGR18] El Mhamdi E. M., Guerraoui R., and Rouault S. (2018) The hidden vulnerability of distributed learning in Byzantium. *Proceedings of the 35th International Conference on Machine Learning* 80: 3521–3530.

[FCJG20] Fang M., Cao X., Jia J., and Gong N. (2020) Local model poisoning attacks to Byzantine-Robust federated learning. In: *29th USENIX security symposium*, pp. 1605–1622.

[GOAZ23] Guendouzi B. S., Ouchani S., Assaad H. E., and Zaher M. E. (2023) A systematic review of federated learning: Challenges, aggregation methods, and development tools. *Journal of Network and Computer Applications* page 103714.

[Gol98] Goldreich O. (1998) Secure multi-party computation. *Manuscript. Preliminary version* 78(110).

[JDFSBJ22] Jebreel N. M., Domingo-Ferrer J., Sánchez D., and Blanco-Justicia A. (2022) Defending against the label-flipping attack in federated learning. *CoRR* abs/2207.01982.

[JSCS19] Jalalirad A., Scavuzzo M., Capota C., and Sprague M. R. (2019) A Simple and Efficient Federated Recommender System. In: , pp. 53–58.

[KLSH21] Kattenborn T., Leitloff J., Schiefer F., and Hinz S. (2021) Review on Convolutional Neural Networks (CNN) in vegetation remote sensing. *ISPRS journal of photogrammetry and remote sensing* 173: 24–49.

[KPN$^+$19] Kawa D., Punyani S., Nayak P., Karker1 A., and Jyotinagar V. (2019) Credit Risk Assessment from Combined Bank Records using Federated Learning. *International Research Journal of Engineering and Technology* 6(4): 1355–1358.

[LBBH98] LeCun Y., Bottou L., Bengio Y., and Haffner P. (1998) Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86(11): 2278–2324.

[LFTL20] Li L., Fan Y., Tse M., and Lin K.-Y. (2020) A review of applications in federated learning. *Computers & Industrial Engineering* 149: 106854.

[LL17] Lundberg S. M. and Lee S.-I. (2017) A unified approach to interpreting model predictions. *Advances in neural information processing systems* 30: 4768–4777.

[LLH$^+$20] Lim W. Y. B., Luong N. C., Hoang D. T., Jiao Y., Liang Y.-C., Yang Q., Niyato D., and Miao C. (2020) Federated learning in mobile edge networks: A comprehensive survey. *IEEE Communications Surveys & Tutorials* 22(3): 2031–2063.

[LLL$^+$22] Liu Y., Liu Y., Liu Z., Liang Y., Meng C., Zhang J., and Zheng Y. (2022) Federated Forest. *IEEE Transactions on Big Data* 8(3): 843–854.

[LVN23] Lewis C., Varadharajan V., and Noman N. (2023) Attacks against federated learning defense systems and their mitigation. *Journal of Machine Learning Research* 24(30): 1–50.

[LXW22] Liu P., Xu X., and Wang W. (2022) Threats, attacks and defenses to federated learning: issues, taxonomy and perspectives. *Cybersecurity* 5(1): 1–19.

[LZJ$^+$20] Li Y., Zhou Y., Jolfaei A., Yu D., Xu G., and Zheng X. (2020) Privacy-preserving federated learning framework based on chained secure multi-party computing. *IEEE Internet of Things Journal* 8(8): 6178–6186.

[McD09] McDonald G. C. (2009) Ridge regression. *Wiley Interdisciplinary Reviews: Computational Statistics* 1(1): 93–100.

[MMR$^+$17] McMahan B., Moore E., Ramage D., Hampson S., and y Arcas B. A. (2017) Communication-efficient learning of deep networks from decentralized data. In: *Artificial intelligence and statistics*, pp. 1273–1282. Proceedings of Machine Learning Research.

[MPP$^+$21] Mothukuri V., Parizi R. M., Pouriyeh S., Huang Y., Dehghantanha A., and Srivastava G. (2021) A survey on security and privacy of federated learning. *Future Generation Computer Systems* 115: 619–640.

[NDP$^+$21] Nguyen D. C., Ding M., Pathirana P. N., Seneviratne A., Li J., and Poor H. V. (2021) Federated learning for internet of things: A comprehensive survey. *IEEE Communications Surveys & Tutorials* 23(3): 1622–1658.

[PKH22] Pillutla K., Kakade S. M., and Harchaoui Z. (2022) Robust aggregation for federated learning. *IEEE Transactions on Signal Processing* 70: 1142–1154.

[PY06] Pasi G. and Yager R. (2006) Modeling the Concept of Majority Opinion in Group Decision Making. *Information Science* 176(4): 390–414.

[RBJLL$^+$23] Rodríguez-Barroso N., Jiménez-López D., Luzón M. V., Herrera F., and Martínez-Cámara E. (2023) Survey on federated learning threats: Concepts, taxonomy on attacks and defences, experimental study and challenges. *Information Fusion* 90: 148–173.

[RBMCLH22] Rodríguez-Barroso N., Martínez-Cámara E., Luzón M. V., and Herrera F. (2022) Dynamic defense against byzantine poisoning attacks in federated learning. *Future Generation Computer Systems* 133: 1–9.

[RBSJL$^+$20] Rodríguez-Barroso N., Stipcich G., Jiménez-López D., Ruiz-Millán J. A., Martínez-Cámara E., González-Seco G., Luzón M. V., Veganzones M. A., and Herrera F. (2020) Federated Learning and Differential Privacy: Software tools analysis, the Sherpa. ai FL framework and methodological guidelines for preserving data privacy. *Information Fusion* 64: 270–292.

[RHL$^+$20] Rieke N., Hancox J., Li W., Milletari F., Roth H. R., Albarqouni S., Bakas S., Galtier M. N., Landman B. A., Maier-Hein K., *et al.* (2020) The future of digital health with federated learning. *NPJ digital medicine* 3(1): 119.

[RSG16] Ribeiro M. T., Singh S., and Guestrin C. (2016) ”Why should i trust you?” Explaining the predictions of any classifier. In: *Proceedings of the 22nd Association for Computing Machinery SIGKDD international conference on knowledge discovery and data mining*, pp. 1135–1144.

[SGB+20] Soliman A., Girdzijauskas S., Bouguelia M.-R., Pashami S., and Nowaczyk S. (2020) Decentralized and Adaptive K -Means Clustering for Non-IID Data using HyperLogLog Counters. In: , pp. 343–355.

[SGLH+22] Sevillano-García I., Luengo J., Herrera F., *et al.* (2022) REVEL Framework to Measure Local Linear Explanations for Black-Box Models: Deep Learning Image Classification Case Study. *International Journal of Intelligent Systems* 2023: 8068569.

[SS23] Saranya A. and Subhashini R. (2023) A systematic review of Explainable Artificial Intelligence models and applications: Recent developments and future trends. *Decision analytics journal* page 100230.

[SWMS19] Sattler F., Wiedemann S., Müller K.-R., and Samek W. (2019) Robust and communication-efficient federated learning from non-iid data. *IEEE transactions on neural networks and learning systems* 31(9): 3400–3413.

[TBA+19] Truex S., Baracaldo N., Anwar A., Steinke T., Ludwig H., Zhang R., and Zhou Y. (2019) A Hybrid Approach to Privacy-Preserving Federated Learning. In: *Proceedings of the 12th Association for Computing Machinery Workshop on Artificial Intelligence and Security*, page 1–11. Association for Computing Machinery.

[TBZ+19] Tran N. H., Bao W., Zomaya A., Nguyen M. N. H., and Hong C. S. (2019) Federated Learning over Wireless Networks: Optimization Model Design and Analysis. In: *IEEE Conference on Computer Communications*, pp. 1387–1395.

[TLS21] Thiebes S., Lins S., and Sunyaev A. (2021) Trustworthy artificial intelligence. *Electronic Markets* 31: 447–464.

[TTGL20] Tolpegin V., Truex S., Gursoy M. E., and Liu L. (2020) Data poisoning attacks against federated learning systems. In: *25th European Symposium on Research in Computer Security.*, pp. 480–501. Springer.

[TYCY22] Tan A. Z., Yu H., Cui L., and Yang Q. (2022) Towards personalized federated learning. *IEEE Transactions on Neural Networks and Learning Systems* pp. 1–17.

[VWK+20] Verbraeken J., Wolting M., Katzy J., Kloppenburg J., Verbelen T., and Rellermeyer J. S. (2020) A survey on distributed machine learning. *Acm computing surveys* 53(2): 1–33.

[Win21] Wing J. M. (2021) Trustworthy AI. *Communications of the ACM* 64(10): 64–71.

[WZFY20] Wang T., Zhang X., Feng J., and Yang X. (2020) A Comprehensive Survey on Local Differential Privacy toward Data Statistics and Analysis. *Sensors* 20(24): 7030.

[XRV17] Xiao H., Rasul K., and Vollgraf R. (2017) Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms. *CoRR* abs/1708.07747.

[Yag88] Yager R. (1988) On ordered weighted averaging aggregation operators in multicriteria decisionmaking. *IEEE Transactions on Systems, Man, and Cybernetics* 18(1): 183–190.

[Yag96] Yager R. (1996) Quantifier guided aggregation using OWA operators. *International Journal of Intelligent Systems* 11(1): 49–73.

[YCRB18] Yin D., Chen Y., Ramchandran K., and Bartlett P. L. (2018) Byzantine-Robust Distributed Learning: Towards Optimal Statistical Rates. *CoRR* abs/1803.01498.

[YF99] Yager R. and Filev D. (1999) Induced ordered weighted averaging operators. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 29(2): 141–150.

[YLC$^+$19] Yang Q., Liu Y., Cheng Y., Kang Y., Chen T., and Yu H. (2019) *Federated Learning*, volumen 13 of *Synthesis Lectures on Artificial Intelligence and Machine Learning*. Morgan & Claypool.

[YLCT19] Yang Q., Liu Y., Chen T., and Tong Y. (2019) Federated Machine Learning: Concept and Applications. *Association for Computing Machinery Transactions on Intelligent Systems and Technology* 10(2): 12:1–12:19.

[ZXLJ21] Zhu H., Xu J., Liu S., and Jin Y. (2021) Federated learning on non-IID data: A survey. *Neurocomputing* 465: 371–390.