2025 International Joint Conference on Neural Networks (IJCNN) | 979-8-3315-1042-8/25/$31.00 ©2025 IEEE | DOI: 10.1109/IJCNN64981.2025.11227553

Differentially Private FastSHAP for Federated Learning Model Explainability

Valerio Bonsignori
KDDLab
Scuola Normale Superiore
Pisa, Italy
valerio.bonsignori@sns.it

Luca Corbucci
KDDLab
University of Pisa
Pisa, Italy
luca.corbucci@phd.unipi.it

Francesca Naretto
KDDLab
University of Pisa
Pisa, Italy
francesca.naretto@unipi.it

Anna Monreale
KDDLab
University of Pisa
Pisa, Italy
anna.monreale@unipi.it

Abstract—**Explaining the reasoning behind black-box model predictions while preserving user privacy is a significant challenge. This becomes even more complex in Federated Learning, where legal constraints restrict the data that clients can share with external entities. In this paper, we introduce `FastSHAP`$^{++}$, a method that adapts `FastSHAP` to explain Federated Learning trained models. Unlike existing approaches, `FastSHAP`$^{++}$ mitigates client privacy risks by incorporating Differential Privacy into the explanation process and preventing the exchange of sensitive information between clients and external entities. We evaluate the effectiveness of `FastSHAP`$^{++}$ testing it on three different datasets, and comparing the explanations with those produced by a centralized explainer with access to clients' training data. Lastly, we study the impact of varying levels of Differential Privacy to analyse the trade-offs between privacy and the quality of the explanations.**

Index Terms—**Federated Learning, Explainable AI, Privacy-Preserving Machine Learning**

I. Introduction

The growing adoption of Machine Learning (ML) models across various domains has raised concerns regarding their black-box nature and the challenges in interpreting their decision-making processes. The field of Explainable Artificial Intelligence (XAI) has emerged to address these issues, developing methods that improve the transparency of ML models, provide insights into their predictions and enhance user trust and accountability [8]. At the same time, the rising concern for privacy has led to the adoption of Federated Learning (FL) [29], which trains ML models without sharing raw data. While FL improves data privacy by decentralizing model training, it still produces black-box models, which remain difficult to interpret. This introduces new challenges in ensuring explainability without compromising privacy. Regulations, such as the EU General Data Protection Regulation (GDPR) [39], emphasize the necessity of both interpretability and privacy protection, further motivating the need for solutions that balance these two aspects also in FL.

Initially, researchers adapted methods designed to explain models trained in classic centralized learning scenarios to use them in FL. However, since the methods were originally thought for centralised learning, their adaptation to FL requires finding a trade-off between the right to explanation and the client's privacy. Most of the existing methods to explain models trained with FL require sharing some information about clients' data during either training or testing. This practice raises significant privacy concerns for the users involved in the training process. Moreover, recent studies highlighted the privacy risks associated with traditional XAI methods [33, 34], demonstrating how Membership Inference Attack [38, 32] can be adapted to target these explainers.

In this paper, we introduce FastSHAP^{++} an extension to FL of the FastSHAP explainer which guarantees both the black-box explanation and privacy protection. Unlike existing approaches in the literature [10, 14], FastSHAP^{++} preserves client privacy in two ways: by preventing the sharing of sensitive information between clients and the server, and by incorporating Differential Privacy (DP) through its neural network (NN) based FastSHAP explainer.

We summarize our contribution in the following:

- We introduce FastSHAP^{++}, the first fully federated and private explainer that needs clients neither to share a reference background dataset to build the explainer nor to access samples requiring explanation at inference time;
- FastSHAP^{++} trains a single neural network-based explainer that can be used to explain samples at inference time. Once training is complete, clients are not required to participate in the explanations computation;
- FastSHAP^{++} uses DP to ensure clients' training dataset privacy protection;
- We evaluate FastSHAP^{++} on three different tabular datasets, comparing the explanations generated by our FL-trained explainer against those produced by a centralized FastSHAP explainer.
- We test FastSHAP^{++} with five different levels of privacy protection, measuring their impact on the generated explanation quality using six metrics;
- FastSHAP^{++} source code and datasets used in the experiments are publicly available to ensure reproducibility[1].

The rest of the paper is organized as follows. Section II discusses the related work addressing explainable AI in FL. Section III introduces important notions and concepts useful for understanding our proposal while Section IV describes the details of FastSHAP^{++} method. Sections V and VI present the

[1] FastSHAP^{++} Source Code: https://github.com/lucacorbucci/fastshap_plusplus

details of our experiments and their results. Lastly, Section VII concludes the paper and discusses future work.

II. Related Work

The widespread adoption of ML across various fields has significantly impacted modern society. As a result, and with the introduction of several AI regulations [37, 7, 19, 18], interest in the XAI research area has grown, driven by the need to open and explain black-box models [8]. While numerous approaches have been developed for traditional centralized learning scenarios [27, 21, 12], there are only a few solutions designed to explain models trained with FL [5, 22].

For NN trained using FL, most of the approaches presented in the literature are based on the popular SHAP explainers [27]. A first work in this direction [10] introduced a SHAP-based approach for feature importance explanations. In this methodology, each client maintains its explainer. For each sample requiring explanation, clients generate individual explanations and share them with the server that is responsible for averaging them. This approach preserves the privacy of the clients' training data, as each client trains its local explainer without sharing it with the server. However, at test time, the sample requiring explanation must be accessible to all clients. A different SHAP-based approach was proposed in [14], utilizing Federated Fuzzy C-Means clustering [4] to establish a reference set for server-side SHAP explanation. The applicability of this method is restricted to cross-silo scenarios. Moreover, [14] requires the clients to share the reference sets with the server, leading to potential privacy risks. Another solution based on SHAP is presented in [20], in which the goal is not only to extract the feature importances but also to preserve the privacy of the clients by exploiting secure multi-party computation [40] for explanation aggregation. The associated communication overhead limits its practical use.

All the methods based on SHAP share the common drawback of being slow due to the combinatorial computation required to calculate the Shapley values. Our current work addresses these limitations proposing a solution based on FastSHAP [23], a popular method that approximates the Shapley values computed by SHAP by using NN. Moreover, we enhance the privacy protection of the explainer by applying DP [15], a technique that allows the release of information derived from private data while quantifying potential user information leakage. We provide an overview of FastSHAP in Section III-B and of DP in Section III-C.

III. Background

A. Federated Learning

FL [29] is a technique introduced in 2016, in which C clients collaboratively train an ML model while keeping the training dataset distributed. Usually, this process is orchestrated by a server S that selects a subset of available clients in each training round $r \in [0, R]$, alternatives include fully distributed and peer-to-peer systems [9]. S is also responsible for aggregating the models received by the clients. Once the clients are selected, the server sends them a model γ_r. The weights of the model γ_r are random in the first round when $r = 0$. The behaviour of the server S depends on the algorithm used for aggregating the trained models. One of the most popular aggregation algorithms is Federated Average (FedAvg) [29] which limits the communication cost by allowing the clients to perform multiple gradient updates before sharing the updated model with the server. In this case, once a client c is selected, it can start the training of the model, received by the server, using its local training dataset D_c. The model is trained for E local steps. At the end of the E steps, each client c shares the updated model γ_c with the server that performs the aggregation $\gamma_{r+1} \leftarrow \sum_{c=1 \in C} \frac{n_c}{n} \gamma_c$ where n_c is the amount of training sample of client c while $n = \sum_{c \in C} n_c$.

B. FastSHAP

SHAP [27] is a widely used XAI method based on game theory. The SHAP value $\phi_{shap-i}(v)$ provides the feature attribution of feature i with value v by measuring the average marginal contribution across all possible feature coalitions. While theoretically sound, due to its combinatorial nature, SHAP often encounters significant computational challenges, especially for high dimensional inputs. To address this, KernelShap [27] was introduced as a practical approximation of Shapley values through weighted least squares regression, offering a more practical solution while preserving the theoretical properties of Shapley values.

Despite its improvements, KernelShap remains computationally demanding for many real-world applications. In contrast, FastSHAP [23] is a novel neural approach for efficiently computing Shapley values. It provides a significant speed-up while ensuring high fidelity to the original Shapley value computations. The key intuition behind FastSHAP lies in its use of a neural solver to approximate Shapley values. Although the approach requires an initial training phase, it eliminates the need for explicit Shapley value computation by framing it as an optimization problem. This makes FastSHAP particularly efficient, as it avoids the computational burden of calculating actual Shapley values during training. The optimization problem leverages the output of the FastSHAP explainer to be as close as possible to SHAP.

Like SHAP, FastSHAP keeps the property of being an agnostic explainer. This is possible due to a dual neural architecture composed of a surrogate and an explainer model. The surrogate model $\hat{\gamma}$ is trained to mimic the black-box model to explain: $\gamma : \mathcal{X} \to \Delta^{K-1}$ where $x \in \mathcal{X}$ is the input vector and the output is a distribution over the set of classes $y \in \mathcal{Y} = (1, .., K)$. The surrogate $\hat{\gamma}(\mathbf{y}|m(x, b); \beta)$ takes as input a vector of masked features $m(x, b)$, where the masking function $m(x, b)$ removes features from the input x according to the binary mask b, with $b_i = 0$ indicating that feature i should be masked out. This masking mechanism allows the surrogate to simulate predictions with only subsets of the original features. The explainer $\phi_{fast} : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^d$, once trained, generates explanations in a single forward pass, providing significant speed-up over the methods that approximate SHAP. As the author illustrates in [23], the surrogate's training process aims to minimize:

$$\mathcal{L}(\beta) = \mathop{\mathbb{E}}_{p(x)} \mathop{\mathbb{E}}_{p(b)} \left[D_{KL}(\gamma(x;\theta) || \hat{\gamma}(y|m(x,b);\beta) \right] \tag{1}$$

Once the surrogate model is trained, it is exploited to train the FastSHAP explainer $\phi_{fast}(x,y;\eta)$. Rather than using a dataset of ground truth Shapley values for training, the FastSHAP explainer is trained by minimizing:

$$\mathcal{L}(\eta) = \mathop{\mathbb{E}}_{p(x)} \mathop{\mathbb{E}}_{\text{Unif}(y)} \mathop{\mathbb{E}}_{p(b)} \left[\left(v_{x,y}(b) - v_{x,y}(0) - b^T \phi_{fast}(x,y;\eta) \right)^2 \right] \tag{2}$$

where Unif(y) represent a uniform distribution over classes. The surrogate model is needed since black-box models γ usually do not allow inference with a partial set of input features, hence it is necessary for a model able to simulate the effects on the prediction using only a subset of features. The authors of [23] suggest that the choice of a sufficiently expressive model for ϕ_{fast} and a large dataset allows the outputs of the global optimum $\phi_{fast}(x,y;\eta^*)$ to match the theoretical Shapley values.

To keep the notation light, in the remaining sections we will use ϕ to indicate the FastSHAP explanation function, γ to indicate the black-box and $\hat{\gamma}$ to denote the surrogate.

C. Differential Privacy and Privacy-Preserving ML

DP is a privacy-enhancing technology introduced by Cyntia Dwork [15]. DP ensures that running the same algorithm $\mathcal{M}$ on two neighbouring datasets produces indistinguishable outputs up to an upper bound ε called privacy budget. More formally,

***Definition** 1: An algorithm $\mathcal{M}$ satisfies (ε, δ)-DP, where $\varepsilon > 0$ and $\delta \in [0,1]$, if for any pair of "neighbouring" datasets D, $D^{'}$ differing in exactly one entry, and for any set of outputs O the following condition holds:*

$$P[\mathcal{M}(D) \in O] \leq e^{\varepsilon} P[\mathcal{M}(D^{'}) \in O] + \delta$$

In this paper, we rely on the "add/remove neighbourhood" definition. More formally, D and $D^{'}$ are considered neighbourhoods if $D^{'}$ can be derived from D just by adding or removing a sample. To manage the accounting of the expended privacy budget we leverage the accountant provided by Opacus [41], a well-known library used to train Differentially Private models. Opacus' accountant uses Rényi Differential Privacy (RDP) [31], a relaxation of the original DP definition based on the Rényi divergence.

IV. Methodology

The proposed method, FastSHAP^{++}, provides explanations in the form of *feature importance* for black-box models trained through FL, while ensuring both faithful explanations and privacy for the data of clients involved in the training process. To the best of our knowledge, all existing methods for providing explanations in FL rely on sharing data, aggregated or synthetic, to compose the reference set for SHAP, raising inherent privacy concerns and potentially compromising the principles of FL. In contrast, FastSHAP^{++} represents a novel approach as it is the first explainer to eliminate data sharing between parties. By transmitting only the explainer model to the server S, the clients involved in the FL training preserve their data privacy. In particular, the proposed approach is based on FastSHAP [23], which is a neural network-based methodology. We redefine FastSHAP to collaboratively train both the surrogate and the explainer neural networks in an FL setting, enabling the sharing of only the model weights between clients and server and ensuring that no raw data is transmitted. Leveraging this collaborative training approach, FastSHAP^{++} avoids the limitations of post-hoc explanation aggregation. In fact, as highlighted in [26], aggregating explanations can lead to lower fidelity and less coherent results, particularly in non-IID data settings.

Regarding privacy, we enhance protection by applying DP [16] during the training of the network-based modules of FastSHAP^{++}. This enrichment mitigates privacy risks and enable us to quantify the level of protection through defined privacy bounds.

In the following sections, we first define the federated version of FastSHAP in Section IV-A. Then, in Section IV-B, we describe the privacy protection model that we employ to obtain FastSHAP^{++}.

A. Federated FastSHAP

In this section, we present our implementation of FastSHAP in FL. As introduced in Section III-B, FastSHAP is an **agnostic post-hoc explanation method** based on a NN structure, originally designed for centralized settings. However, its NN architecture makes it well-suited for training in FL scenarios. Leveraging this, our method integrates FastSHAP into FL to explain any kind of already trained FL black-box model γ. Similarly to FastSHAP, FastSHAP^{++} consists of two stages. In the first stage, FL is used to collaboratively train a surrogate model $\hat{\gamma}$ that mimics the behaviour of the black-box γ. The objective is to create a surrogate model to enable the masking step of SHAP, where the importance of each feature is calculated considering different combinations of features. To measure the goodness of a surrogate $\hat{\gamma}$ we use the *fidelity*, which quantifies how often the surrogate predictions match the ones of the black-box.

At round r, the server S sends the surrogate model $\hat{\gamma}_r$ to the clients. Upon receiving $\hat{\gamma}_r$, each client performs a local training on its dataset to minimize the loss function in Equation 1 (Section III-B). Once the local training is completed, each client c shares its updated surrogate model $\hat{\gamma}_c$ with the server. The server aggregates the updates to produce a new global surrogate model $\hat{\gamma}_{r+1}$, which is then used in subsequent training rounds. This stage ends after R training rounds, obtaining a federated surrogate model $\hat{\gamma}_F$.

At this point, FastSHAP^{++} performs its second stage, in which it trains the explainer ϕ_F that outputs Shapley values, exploiting $\hat{\gamma}_F$. Similarly to the first stage, we exploit the FL training procedure to minimize Equation 2, as detailed in Section III-B. In this way, we adhere to the key feature of

FastSHAP, which avoids the explicit computation of explanations effectively adapting to the FL framework. Once the explainer training is completed, the outcome of FastSHAP^{++} is a globally shared explainer ϕ_F accessible to all clients of the federation. This is achieved without sharing raw training nor synthetic data but only model updates.

B. Differentially Private FastSHAP^{++}

Avoiding the sharing of the raw data to train the federated surrogate $\hat{\gamma}_F$ and explainer ϕ_F does not provide any formal privacy guarantees. To ensure robust privacy protection with bounded guarantees, we propose to adopt DP [16] during the model training. Specifically, we exploit the sample-level definition [16] of (ε, δ)-DP, which protects the privacy of individual samples within each client's training dataset. To achieve (ε, δ)-DP guarantees, we train the models using the DP-SGD [1] algorithm implemented in Opacus [41] and apply the Gaussian Mechanism [17] to introduce the necessary noise for privacy preservation. Poisson sampling is used during the training to select the mini-batches. To account for the total privacy budget consumed during the training of each model we use the RDP accountant [31] available in Opacus [41].

Since all three models involved in the FL process (i.e. the black-box γ_F, the surrogate $\hat{\gamma}_F$ and the explainer ϕ_F) use clients' sensitive data, DP-SGD must be applied during the training of each model to ensure (ε, δ)-DP guarantee.

In particular, for each of these models we define:

- A privacy budget (ε_1, δ_1) for the training of the black-box. Based on this budget, we can compute the noise σ_1 required to guarantee (ε_1, δ_1) for the black-box training, obtaining γ_F^P.
- A privacy budget (ε_2, δ_2) for the training of the surrogate model. A corresponding noise σ_2 is then used to guarantee DP, obtaining $\hat{\gamma}_F^P$.
- A privacy budget (ε_3, δ_3) for the training of the explainer model. A corresponding noise σ_3 is used to meet DP requirements, obtaining ϕ_F^P.

The total privacy budget consumed for the entire pipeline (i.e. training of γ_F^P, $\hat{\gamma}_F^P$ and ϕ_F^P) is computed using the basic composition theorem [16]. Therefore FastSHAP^{++}, starting from a (ε_1, δ_1) γ guarantees a final (ε, δ)-DP where ε=$\varepsilon_1 + \varepsilon_2 + \varepsilon_3$ and δ=$\delta_1 + \delta_2 + \delta_3$.

V. Experimental setup

In this section, we validate FastSHAP^{++} on three popular tabular datasets in a cross-device setting, already used in FL research [11]: **Dutch** [25], ACSIncome (**ACSI**) [13], and ACSEmployment (**ACSE**) [13]. The cross-device setting is motivated by the widespread adoption of this approach and its inherent difficulty, which makes it a challenging evaluation scenario. Each dataset has a binary target variable. **Dutch** consists of 60,420 samples and provides demographic and economic information about individuals. The task is to classify whether an individual's salary exceeds $50,000 using 12 numerical socio-demographic characteristics. **ACSI** is derived from the American Community Survey (ACS) and includes data from all 50 U.S. states and Puerto Rico described using 11 features, numeric and categorical. The classification task is to determine whether a person's salary is above $50,000. **ACSE**, also based on ACS data, differs from **ACSI** as it focuses on predicting an individual's occupation status based on 17 socio-demographic features (numeric and categorical).

The natural partition of **ACSI** and **ACSE** into 51 states, makes them a natural choice for FL experimentation allowing us to consider a naturally distributed dataset. In the case of **Dutch** instead, we split the dataset into 50 clients using the Latent Dirichlet Allocation with concentration parameter $\alpha = 5$, generating balanced sample size partitions while preserving statistical heterogeneity across clients. To ensure a fair comparison of the generated explanations, we compute all metrics on independent test sets: 200,000 samples for **ACSE**, 166,933 for **ACSI**, and 13,074 for **Dutch**. These test sets remain separate from the training phase, ensuring an unbiased assessment of explanation quality across the data distribution.

For handling the categorical features of **ACSI** and **ACSE**, we employ Target Encoders [30], which replaces categorical variables with their corresponding target mean values. This encoding technique captures the relationship between categories and the target variable by computing smooth estimates of the target mean per category.

In the case of **ACSI** and **ACSE** we pre-process the dataset implementing a data capping strategy[2].

As described in Section IV, the FastSHAP^{++} pipeline is designed to explain a target black-box model γ_F^P by training a surrogate model $\hat{\gamma}_F^P$ and an explainer ϕ_F^P. In our experiments, γ_F^P is a NN consisting of two fully connected layers. $\hat{\gamma}_F^P$ is a NN composed of a custom masking layer, which selectively masks input features, followed by three fully connected layers and ReLU activations in between. ϕ_F^P is a NN composed of three linear layers interconnected with ReLU activations. We implement the NN using Pytorch [35] and simulate a cross-device FL scenario[3] using Flower [6]. In our experiments, we use 80% of the clients for training and the remaining 20% exclusively to evaluate the quality of the models and their explanations. As a consequence, the same training data are used for training the γ, the $\hat{\gamma}$ and the ϕ, in line with the setting used in the original FastSHAP paper [23]. Given our setting, in each training round, S select the 15% of the training clients in the experiments with **Dutch** and 17% in the experiments with **ACSI** and **ACSE**. For the hyperparameter tuning phase, we divide the training clients into two groups: training and validation. This process aims to identify the optimal hyperparameters for the black-box γ, the surrogate model $\hat{\gamma}$ and the explainer ϕ.

For the black-box γ, we optimize hyperparameters to maximize validation accuracy. The surrogate model is optimized to minimize the loss in Equation 1, while the explainer is tuned

[2] We limit the maximum number of samples used for local training to 20,000 instances per round. This was necessary to reduce the computational power required to perform all the tests in all the settings we wanted to consider.

[3] It is worth noticing that the same pipeline system can be applied in the same fashion in the case of cross-silo scenario.

	Accuracy BB	**Fidelity Surrogate**
Dutch	0.83±0.01	0.97±0.01
Dutch (DP)	0.82±0.01	0.91±0.01
ACSI	0.77±0.01	0.97±0.01
ACSI (DP)	0.77±0.02	0.95±0.01
ACSE	0.80±0.01	0.95±0.01
ACSE (DP)	0.74±0.03	0.90±0.01

TABLE I: Performance of the black-box federated models γ_F and γ_F^P and the surrogate $\hat{\gamma}_F$ and $\hat{\gamma}_F^P$. For the black-boxs we report the the accuracy of the private (DP) and non-private models. For the surrogates we report the fidelity. The results reported are the average and standard deviation on three runs.

to minimize the loss in Equation 2. Given the FL environment, we compute these metrics by aggregating the values reported by each client to the server using a weighted average.

To apply DP-SGD [1] fairly, we impose a limit on the maximum amount of times each client can be selected for FL training. This is needed in the evaluation to prevent certain clients from being excessively selected, which implies the need to introduce more noise to maintain the (ε, δ) guarantee. Based on this information, on the required privacy budget and the number of local training epochs, each client computes the noise parameter σ to be used during DP training. In particular, we guarantee: (ε=1, δ=10^{-3})-DP for both the training of γ_F^P and $\hat{\gamma}_F^P$ with **Dutch** dataset, (ε=1, δ=3×10^{-4})-DP for **ACSI** and (ε=1, δ=5×10^{-5})-DP for **ACSE**. For the training of the ϕ_F^P, we test five ε in the range $[0.1, 5]$ while the δ for the three datasets are the same used during γ and $\hat{\gamma}$ training.

VI. Experimental results

Comparison with Centralised FastSHAP To evaluate the impact of adapting FastSHAP to an FL scenario, we compare the explanations generated by the FastSHAP^{++} explainer ϕ_F with those obtained using the centralized FastSHAP ϕ, trained by aggregating all clients' training data. We emphasize that, in this comparison, we do not consider DP to ensure a fair evaluation focused only on the impact of FL. Table I presents the performance of the black-box model γ_F and the surrogate model $\hat{\gamma}_F$ trained in FL, highlighting their strong performance.

To evaluate the similarity of the explanations we use two distance metrics based on [28], three agreement metrics based on [24] and a faithfulness metric [3]. The metrics are:

- ℓ_2 **Distance**: Euclidean distance between two explanations using the ℓ_2 norm;
- **Cosine Similarity**: the Cosine of the angle between the two explanations;
- **Feature Agreement**: the fraction of common features in the *top-k* sets of two explanations;
- **Sign Agreement**: the fraction of common features in the *top-k* features sets that also share the same sign[4];
- **Rank Correlation**: Spearman's rank correlation coefficient to evaluate the alignment between feature rankings;
- Δ **faithfulness**: the absolute value of the difference of the faithfulness [3] of the explanations with ϕ and ϕ_F.

In Table II we report the average and standard deviation of these metrics on the entire test set. In terms of distance metrics, the ℓ_2 values remain low across all datasets, indicating a small difference between the explanations. Instead, the Cosine Similarity values are consistently high, suggesting strong alignment between the explanations. Thus, we can conclude that ϕ_F closely approximates ϕ in terms of Shapley values.

In terms of agreement metrics, the **Dutch** dataset achieves the best results, indicating strong overlap in both the *top-k* selected features and their direction. In contrast, **ACSI** and **ACSE** show a slight decrease in agreement metrics, suggesting divergence in *top-k* feature ranking and direction.

For the Δ Faithfulness, values remain close to 0 for the **Dutch** dataset, indicating minimal differences and high consistency between the explanations generated by ϕ and ϕ_F. However, **ACSI** and **ACSE** exhibit a higher average difference and standard deviation in Δ Faithfulness, likely due to their complexity and greater heterogeneity in data distributions.

Impact of DP on the explanations: As introduced in Section IV-B, FastSHAP^{++} formally bounds the privacy risk of the clients involved in the training using DP. To ensure that FastSHAP^{++} respect (ε, δ)-DP guarantees, DP-SGD is applied throughout the entire pipeline. Performance of γ_F^P and $\hat{\gamma}_F^P$ using an FL setting and DP can be found in Table I. We evaluate the impact of the DP integration considering three different experimental settings. We indicate by: *(i)* w/o privacy the setting in which explanations are produced by a non-private FastSHAP^{++} explainer ϕ_F, using a non-private surrogate $\hat{\gamma}_F$; *(ii)* semi private the setting in which explanations are computed by a non-private FastSHAP^{++} explainer ϕ_F, using a private surrogate $\hat{\gamma}_F^P$; and *(iii)* full private the setting in which both the explainer ϕ_F^P and the surrogate $\hat{\gamma}_F^P$ are trained using DP-SGD. In every setting, we always consider a differential private black-box model.

Table III reports the results on the three datasets evaluated with the same six metrics as in previous experiments. In Figure 1 we graphically report the same results only for **Dutch** (due to space constraints) to better analyse the impact of both ε values on the explanation quality and applying a full private explainer with respect to a semi private explainer. In our experiments, ε ranges $[0.1, 5]$, where the privacy guarantee decreases as ε increases; while δ is computed as $\max_{c=1}^{C} \frac{1}{|D_c|}$, following [36].

In particular, in our experiments we analyse: (1) w/o privacy vs. semi private setting, which in Figure 1 is represented by a horizontal dashed line, and which corresponds to the mean of the comparisons for these settings and the coloured area around it is the standard deviation across the test set. The representation is solely a line to represent the single combination of a surrogate $\hat{\gamma}_F^P$ with $\varepsilon = 1$ and an explainer trained without DP; (2) w/o privacy vs. full private setting, represented by green circle markers in Figure 1.

As expected, we observe that for each explanation quality metric, we have an improvement while reducing the privacy

[4] In our experiments $k = 5$ for Feature Agreement and Sign Agreement

	ℓ_2 **Dist.** (↓)	**Cosine Sim.** (↑)	**Feat. Agr.** (↑)	**Sign Agr.** (↑)	**Rank Corr.** (↑)	**Δ Faith** (↓)
Dutch	0.03±0.02	0.99±0.01	0.87±0.13	0.87±0.13	0.84±0.11	0.02±0.03
ACSI	0.09±0.05	0.81±0.22	0.77±0.14	0.70±0.17	0.70±0.17	0.13±0.10
ACSE	0.10±0.05	0.80±0.19	0.66±0.16	0.64±0.17	0.65±0.16	0.21±0.18

TABLE II: Differences in the metrics when explaining γ_F, comparing a centralized explainer ϕ (and $\hat{\gamma}$) in `FastSHAP` to a federated on ϕ_F (and $\hat{\gamma}_F$) in `FastSHAP`$^{++}$.

	Setting	ε Explainer	ℓ_2 **Dist.** (↓)	**Cosine Sim.** (↑)	**Feature Agr.** (↑)	**Sign Agr.** (↑)	**Rank Corr.** (↑)	**Δ Faithfulness** (↓)
Dutch	`w/o` vs `full`	ε=0.1	0.33±0.12	0.24±0.40	0.44±0.16	0.23±0.16	0.00±0.28	0.41±0.30
		ε=0.5	0.31±0.12	0.36±0.38	0.53±0.13	0.37±0.14	0.09±0.20	0.27±0.21
		ε=1	0.23±0.11	0.67±0.33	0.60±0.13	0.45±0.13	0.27±0.17	0.17±0.13
		ε=2	0.23±0.11	0.66±0.36	0.62±0.12	0.47±0.13	0.32±0.17	0.14±0.11
		ε=5	0.18±0.09	0.76±0.32	0.61±0.13	0.50±0.12	0.33±0.19	0.13±0.10
	`w/o` vs `semi`	NO `DP`	0.18±0.09	0.77±0.31	0.61±0.13	0.48±0.13	0.32±0.19	0.13±0.09
ACSI	`w/o` vs `full`	ε=0.1	0.14±0.07	0.73±0.15	0.66±0.16	0.56±0.15	0.42±0.24	0.19±0.19
		ε=0.5	0.11±0.05	0.88±0.08	0.68±0.16	0.64±0.15	0.52±0.32	0.11±0.16
		ε=1	0.05±0.02	0.96±0.02	0.76±0.15	0.74±0.16	0.69±0.18	0.07±0.13
		ε=2	0.06±0.02	0.96±0.02	0.78±0.11	0.78±0.11	0.70±0.20	0.09±0.10
		ε=5	0.05±0.01	0.97±0.02	0.80±0.15	0.80±0.15	0.74±0.21	0.05±0.12
	`w/o` vs `semi`	NO `DP`	0.05±0.02	0.97±0.02	0.82±0.11	0.82±0.11	0.76±0.10	0.03±0.13
ACSE	`w/o` vs `full`	ε=0.1	0.12±0.05	0.48±0.44	0.41±0.18	0.36±0.19	0.21±0.25	0.27±0.19
		ε=0.5	0.07±0.03	0.84±0.17	0.67±0.17	0.66±0.18	0.58±0.17	0.16±0.15
		ε=1	0.07±0.03	0.83±0.18	0.73±0.19	0.72±0.19	0.66±0.14	0.17±0.14
		ε=2	0.07±0.02	0.82±0.19	0.70±0.19	0.70±0.20	0.64±0.15	0.17±0.13
		ε=5	0.08±0.04	0.73±0.28	0.66±0.20	0.63±0.22	0.56±0.17	0.22±0.18
	`w/o` vs `semi`	NO `DP`	0.07±0.03	0.80±0.20	0.69±0.20	0.67±0.21	0.60±0.16	0.23±0.17

TABLE III: Results obtained with **ACSI**, **ACSE** and **Dutch** datasets. For each dataset, we compare `w/o privacy` with `full private` and `semi private` with `full private`. "NO-DP" in the ε Explainer column means we do not apply `DP` on the explainer. When `DP` is applied δ=10^{-3} for **Dutch**, δ=3×10^{-4} for **ACSI** and δ=5×10^{-5} for **ACSE**.

protection (i.e., ε increases). This trend is confirmed for each dataset as reported in Table III and is evident from Figure 1.

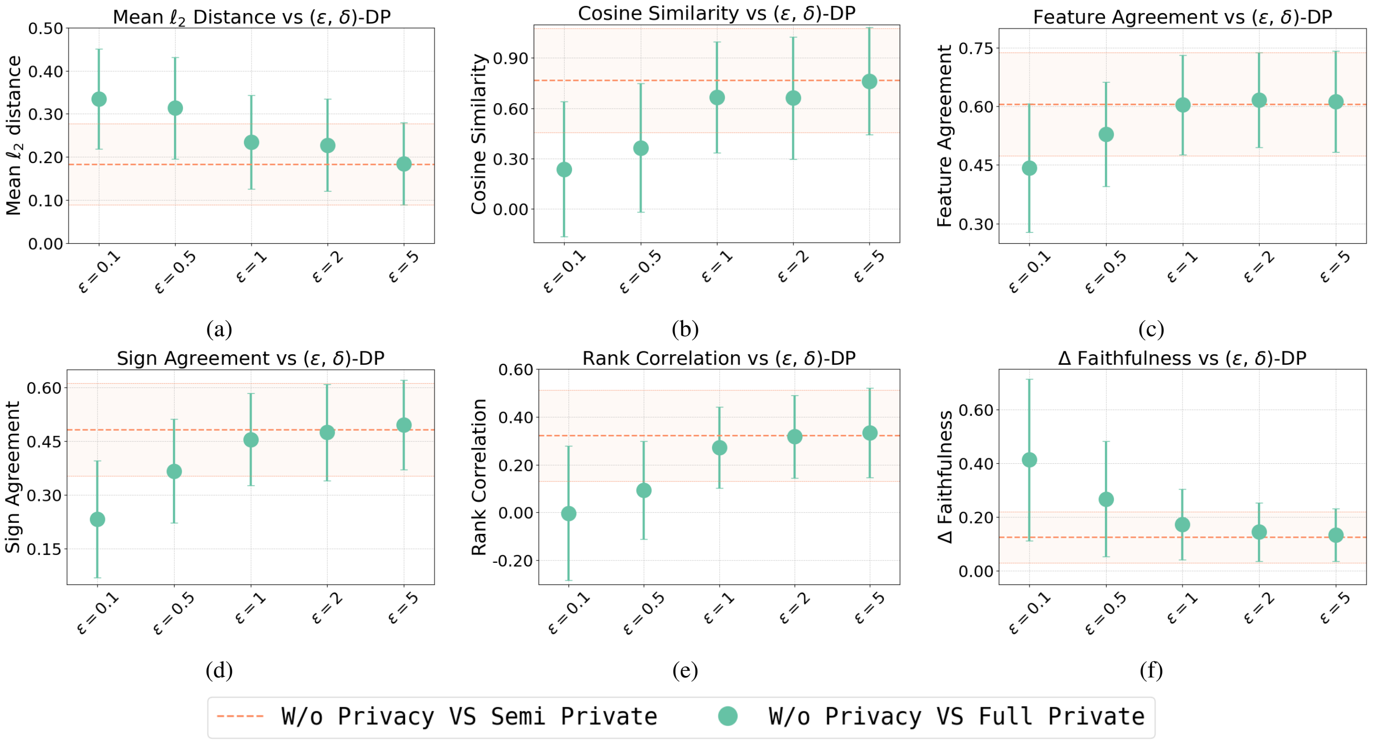
Moreover, we observe that for all the datasets the values of Δ faithfulness are very good. This means that, while guaranteeing strong privacy protection, the faithfulness of the private explanations is very similar to the faithfulness of the explanations without any privacy mitigation. We can also observe good performance in terms of ℓ_2 Distance and Cosine Similarity which shows high consistency between private and non-private explanations, especially for **ACSI** and **ACSE**. The metrics evaluating feature agreement, sign agreement and ranking becomes acceptable for values of $\varepsilon \geq 1$. In general, our results highlight lower performance for **Dutch** due to the higher negative impact of the `DP` application to the neural models. This is likely due to the lower amount of data available to the different clients in the case of the **Dutch** dataset [1]. Analysing the graphical presentation of the results in Figure 1, we can easily verify that, in terms of explanation quality, settings `semi private` and `full private` become equivalent. In particular, when the green circle marker (`w/o privacy` vs `full private`) intersects the horizontal dashed line (`w/o privacy` vs `semi private`), we can say that applying the privacy protection to the explainer does not degrade the overall explanation's quality compared with the setting in which `DP` is only applied to black-box and surrogate. In **Dutch**, we can observe that for values of $\varepsilon \geq 1$ results suggest negligible performance degradation of `full private` with respect to `semi private` setting.

We remark that the value of ε reported in the plot refers specifically to the privacy guarantee of the trained explainer. Applying the composition theorem [16], we can combine the explainer privacy budget with the (ε=1, δ=10^{-3})-`DP` guarantee used for both the model γ_F^P and the surrogate $\hat{\gamma}_F^P$. The overall pipeline guarantees a bound on the privacy risk ranging between (ε=2.1, δ=10^{-3})-`DP` and (ε=7, δ=10^{-3})-`DP`, depending on the chosen explainer: these values are valid and acceptable in the context of `ML` model training [36, 2].

A. *Key Takeaways*

We summarize our experimental findings below:

- *Centralized vs Federated*: We empirically prove that `FastSHAP`$^{++}$ can achieve comparable explanation quality to the centralised `FastSHAP`, without requiring clients to share raw training data with a centralised server;
- *Impact of `DP`*: We evaluate the impact of incorporating `DP` into the `FastSHAP`$^{++}$ pipeline by comparing three configurations. Our results show a good trade-off between the explanation quality and privacy protection guarantee;
- *Robustness*: We validate `FastSHAP`$^{++}$'s robustness and performance across three datasets and scenarios, proving



(a) (b) (c) (d) (e) (f)

Fig. 1: Results of the experiments with the **Dutch** Dataset with the explainer with a varying range of $\varepsilon \in [0.1, 5]$ and $\delta{=}10^{-3}$.

that our method maintains high-fidelity explanations even under strong privacy constraints. These properties make `FastSHAP`$^{++}$ a practical and effective solution for real-world applications.

VII. Conclusions

In this paper, we introduced `FastSHAP`$^{++}$, the first fully federated and private explainer, showing that it achieves centralized-level explanation quality while preserving clients' data privacy. Our evaluation, across three real-world datasets, showed that `FastSHAP`$^{++}$ closely matches the performance of a centralised explainer while enabling the integration of `DP` into the training to formally bound clients' privacy risk.

We observed that incorporating `DP` into the `FastSHAP`$^{++}$ pipeline with privacy protection levels in the range $\varepsilon \in [0.1, 5]$ can guarantee privacy without degrading the explanation quality too much, particularly when $\varepsilon \geq 1$. Both the distance, agreement and faithfulness metrics used in our evaluations confirmed that the explanations generated while guaranteeing `DP` remain coherent with the one computed with the non-private explainers, demonstrating the practical applicability of `DP` for the explanation generation process in an `FL` scenario.

`FastSHAP`$^{++}$ represents a significant step forward for explaining black-box models trained in `FL` settings, especially considering the regulations that protect the right to explanation and ensure user privacy [19, 18]. By enabling federated explanations and preserving data privacy, our approach supports the wider adoption of `FastSHAP`$^{++}$ in regulated environments. Future works will focus on evaluating `FastSHAP`$^{++}$ on different kinds of data, especially images, as they represent both sensitive and challenging data types to explain and protect.

VIII. Acknowledgements

This research was partially supported by: The European Commission under the NextGeneration EU programme – National Recovery and Resilience Plan (PNRR), under agreements: PNRR - M4C2 - Investimento 1.3, Partenariato Esteso PE00000013 - ”FAIR - Future Artificial Intelligence Research” - Spoke 1 ”Human-centered AI”, and SoBigData.it – ”Strengthening the Italian RI for Social Mining and Big Data Analytics” – Prot. IR0000013 – Avviso n. 3264 del 28/12/2021. The European Union Horizon 2020 program under grant agreement No. 101120763 (TANGO) and the EU H2020 under grant agreement 834756 (XAI). Views and opinions expressed are, however those of the author(s) only and do not necessarily reflect those of the European Union or the European Health and Digital Executive Agency (HaDEA). Neither the European Union nor the granting authority can be held responsible for them.

References

[1] M. Abadi et al. “Deep Learning with Differential Privacy”. In: *ACM SIGSAC Conference on Computer and Communications Security*. 2016.

[2] M. Aerni et al. “Evaluations of Machine Learning Privacy Defenses are Misleading”. In: *SIGSAC Conference on Computer and Communications Security*. 2024.

[3] D. Alvarez et al. “Towards Robust Interpretability with Self-Explaining Neural Networks”. In: *NeurIPS*. 2018.

[4] J. L. C. Bárcena et al. "A federated fuzzy c-means clustering algorithm." In: *WILF*. 2021.

[5] J. L. C. Bárcena et al. "Fed-XAI: Federated Learning of Explainable Artificial Intelligence Models". In: *XAI.it@AI*IA*. 2022.

[6] D. J. Beutel et al. *Flower: A Friendly Federated Learning Research Framework*. 2020.

[7] J. R. Biden. "Executive Order on the Safe, Secure, and Trustworthy Development and Use of Artificial Intelligence". In: (2023).

[8] F. Bodria et al. "Benchmarking and survey of explanation methods for black box models". In: *Data Mining and Knowledge Discovery* (2023).

[9] L. Corbucci et al. "Enhancing Privacy and Utility in Federated Learning: A Hybrid P2P and Server-Based Approach with Differential Privacy Protection". In: *SECRYPT*. 2024.

[10] L. Corbucci et al. "Explaining Black-Boxes in Federated Learning". In: *Explainable Artificial Intelligence*. Springer, 2023.

[11] L. Corbucci et al. "PUFFLE: Balancing Privacy, Utility, and Fairness in Federated Learning". In: *27th European Conference on Artificial Intelligence*. IOS Press, 2024.

[12] L. Corbucci et al. "Semantic Enrichment of Explanations of AI Models for Healthcare". In: *International Conference on Discovery Science*. 2023.

[13] F. Ding et al. "Retiring Adult: New Datasets for Fair Machine Learning". In: *NeurIPS*. 2021.

[14] P. Ducange et al. "Consistent Post-Hoc Explainability in Federated Learning through Federated Fuzzy Clustering". In: *IEEE International Conference on Fuzzy Systems*. 2024.

[15] C. Dwork. "Differential privacy". In: *International colloquium on automata, languages, and programming*. Springer. 2006.

[16] C. Dwork et al. "Calibrating noise to sensitivity in private data analysis". In: *Theory of Cryptography*. Springer. 2006.

[17] C. Dwork et al. "Our data, ourselves: Privacy via distributed noise generation". In: *Advances in Cryptology-EUROCRYPT*. Springer. 2006.

[18] European Commission. *Artificial intelligence act*. 2021.

[19] European Commission. *Ethics guidelines for trustworthy AI*. Publications Office, 2019.

[20] A. A. Fahliani et al. "Privacy-Preserving Federated Interpretability". In: *IEEE International Conference on Big Data*. 2024.

[21] R. Guidotti et al. "Stable and actionable explanations of black-box models through factual and counterfactual rules". In: *Data Min. Knowl. Discov.* 38.5 (2024).

[22] R. Haffar et al. "GLOR-FLEX: Local to Global Rule-Based EXplanations for Federated Learning". In: *International Conference on Fuzzy Systems*. 2024.

[23] N. Jethani et al. "FastSHAP: Real-Time Shapley Value Estimation". In: *The Tenth International Conference on Learning Representations, ICLR*. 2022.

[24] S. Krishna et al. "The disagreement problem in explainable machine learning: A practitioner's perspective". In: *arXiv preprint arXiv:2202.01602* (2022).

[25] P. Van der Laan. "The 2001 census in the Netherlands: Integration of registers and surveys". In: *Conference at the Cathie Marsh Centre*. 2001.

[26] R. López-Blanco et al. "Federated Learning of Explainable Artificial Intelligence (FED-XAI): A Review". In: *Distributed Computing and Artificial Intelligence*. 2023.

[27] S. M. Lundberg et al. "A Unified Approach to Interpreting Model Predictions". In: *Advances in Neural Information Processing Systems*. 2017.

[28] E. Mariotti et al. "Beyond Prediction Similarity: ShapGAP for Evaluating Faithful Surrogate Models in XAI". In: *Explainable Artificial Intelligence*. Springer, 2023.

[29] B. McMahan et al. "Communication-Efficient Learning of Deep Networks from Decentralized Data". In: *International Conference on Artificial Intelligence and Statistics*. 2017.

[30] D. Micci-Barreca. "A preprocessing scheme for high-cardinality categorical attributes in classification and prediction problems". In: *SIGKDD Explor.* (2001).

[31] I. Mironov. "Rényi differential privacy". In: *2017 IEEE 30th computer security foundations symposium (CSF)*. IEEE. 2017.

[32] A. Monreale et al. "Agnostic Label-Only Membership Inference Attack". In: *17th International Conference on Network and System Security*. Springer, 2023.

[33] F. Naretto et al. "Evaluating the Privacy Exposure of Interpretable Global and Local Explainers". In: *Trans. Data Priv.* (2025).

[34] F. Naretto et al. "Evaluating the Privacy Exposure of Interpretable Global Explainers". In: *IEEE CogMI*. 2022.

[35] A. Paszke et al. "PyTorch: An Imperative Style, High-Performance Deep Learning Library". In: *NeurIPS*. 2019.

[36] N. Ponomareva et al. "How to DP-fy ML: A Practical Guide to Machine Learning with Differential Privacy". In: *Journal of Artificial Intelligence Research* 77 (2023).

[37] H. Roberts et al. "The Chinese approach to artificial intelligence: an analysis of policy, ethics, and regulation". In: *AI & society* (2021).

[38] R. Shokri et al. "Membership Inference Attacks Against Machine Learning Models". In: *2017 IEEE Symposium on Security and Privacy (SP)*. 2017.

[39] P. Voigt et al. *The EU General Data Protection Regulation (GDPR): A Practical Guide*. Springer Publishing Company, Incorporated, 2017.

[40] A. C. Yao. "Protocols for secure computations". In: *23rd Annual Symposium on Foundations of Computer Science*. 1982.

[41] Yousefpour et al. "Opacus: User-friendly differential privacy library in PyTorch". In: *ArXiv* abs/2109.12298 (2021).