Explaining Black-Boxes in Federated Learning

Luca Corbucci[1](✉), Riccardo Guidotti[1,2], and Anna Monreale[1]

[1] University of Pisa, Pisa, Italy
{luca.corbucci,anna.monreale}@phd.unipi.it
[2] ISTI-CNR, Pisa, Italy
riccardo.guidotti@isti.cnr.it

Abstract. Federated Learning has witnessed increasing popularity in the past few years for its ability to train Machine Learning models in critical contexts, using private data without moving them. Most of the work in the literature proposes algorithms and architectures for training neural networks, which although they present high performance in different predicting tasks and are easy to be learned with a cooperative mechanism, their predictive reasoning is obscure. Therefore, in this paper, we propose a variant of SHAP, one of the most widely used explanation methods, tailored to Horizontal server-based Federated Learning. The basic idea is having the possibility to explain an instance's prediction performed by the trained Machine Leaning model as an aggregation of the explanations provided by the clients participating in the cooperation. We empirically test our proposal on two different tabular datasets, and we observe interesting and encouraging preliminary results.

Keywords: Explainable AI · Federated Learning · Features Importance

1 Introduction

Federated Learning (FL) [14] has become a popular approach to training Machine Learning (ML) models on distributed data sources. This approach was originally proposed to preserve data privacy since the users involved do not have to share their training datasets with a central server. Usually, the models trained with FL are deep learning models and therefore their transparency remains a challenge [8,12]. Indeed, although the trained ML models present very excellent performance in different tasks, their drawback lies in their complexity, which makes them black-boxes and causes the non-interpretability of the internal decision process for humans [5]. However, when it comes to making high-stakes decisions, such as clinical diagnosis, the explanation aspect of the models used by Artificial Intelligence (AI) systems becomes a critical building block of a trustworthy interaction between the machine and human experts. Meaningful explanations [16] of predictive models would augment the cognitive ability of

L. Longo (Ed.): xAI 2023, CCIS 1902, pp. 151–163, 2023.
https://doi.org/10.1007/978-3-031-44067-0_8

domain experts, such as physicians, to make informed and accurate decisions and to better support responsibility in decision-making.

In the last years, the scientific community posed much attention to the design of explainable AI (XAI) techniques [1,4,8,12] but a relatively limited effort has been spent in the study of interpretability issues in FL [2,6,18,19]. Most of the studies of interpretability in FL are focused on the Vertical FL and exploit method based on feature importance.

In this paper, we address the problem of interpretability by proposing an alternative way to employ the explainer SHAP [13] in the context of FL. In particular, our proposal enables the explanation of an instance's prediction performed by the trained global ML model by aggregating the explanation of the clients participating in the federation. The proposed approach is based on the requirements that in order to produce the explanation of the global model is not necessary to access any information on the training data used by the clients. We propose an analytical methodology that enables a comparison to determine the approximation introduced by our approach with respect to a scenario where we simulate a server which can access the training data. Preliminary experiments conducted on two tabular datasets show that the approximation introduced by our proposal is negligible and that our SHAP explanation tends to agree with the explanation provided by the server in terms of the importance of each feature.

The remaining of the paper is organized as follows. Section 2 discusses the literature on XAI for FL. Section 3 provides an overview on FL and XAI and Sect. 4 presents our proposal and the analytical methodology adopted to validate it. In Sect. 5 we discuss the preliminary experimental results, while Sect. 6 discusses our findings and contributions to the field of XAI. Lastly, Sect. 7 concludes the paper and discusses future research directions.

2 Related Work

Machine learning has become more and more pervasive in our lives. ML models are used nowadays in many different contexts and can impact our lives. Alongside the development of novel ML techniques, there was a very active development of techniques to explain the reasoning of black box models [1,4,7,12]. Explainable Artificial Intelligence (XAI) is the research field that studies the interpretability of AI models [8]. This research field aims to develop methods that can be helpful to "open" these complex and not interpretable models and to explain their predictions. To this end, a lot of approaches have been developed in the past few years. Explanation methods can be categorized with respect to two aspects [8]. One contrasts *model-specific vs model-agnostic* approaches, depending on whether the explanation method exploits knowledge about the internals of the black-box or not. The other contrasts *local vs global* approaches, depending on whether the explanation is provided for any specific instance classified by the black-box or for the logic of the black-box as a whole. Finally, we can distinguish *post-hoc vs ante-hoc* methods if they are designed to explain a pre-trained approach or if they are interpretable by design. While the explanation of ML models has been widely addressed in recent years [1,4,8,12], quite surprisingly, the use of XAI in FL has not gained much attention. A review of the current approaches used to explain models trained with FL is presented in [2]. Most of the approaches provide post-hoc explanation by feature importance [6,18,19]. Wang et al. [19] exploits Shapley values to explain models trained with FL. In particular, they adopt SHAP [13] to compute the feature importance and measure the contributions of different parties in Vertical FL [20], where the users that participate in the training have the same sample space but different feature space. The choice to use Shapley values in FL is justified by the possible privacy risks that could arise from classical feature importance that may reveal some aspect of the private local data. Since cooperative learning explanations could reveal the underlined feature data from other users, it becomes essential to guarantee model privacy. Therefore, in [18], Wang proposes a method to explain models based on SHAP values able to balance interpretability with privacy. The main idea is to reveal detailed feature importance for owned features and a unified feature importance for the features from the other parties. In [6], Fiosina studies the interpretability issues in Horizontal FL [20]. In particular, they adopt a Federated deep learning model to predict taxi trip duration within the Brunswick region through the FedAvg algorithm [14]. In order to explain the trained model, the authors derive feature importance exploiting Integrated Gradients [10]. Explainable AI techniques have also been used to explain the misbehaviour of models trained using FL. Haffar et al. [9], focus on the wrong predictions of an FL model because these predictions could be signs of an attack. In order to observe changes in the model behaviour, the nodes involved during the computation explain the trained model at each epoch. An attacker's presence could be revealed by changes in feature importance between two consecutive epochs greater than a predetermined threshold. To the best of our knowledge, no previous work addressed the problem of interpretability in horizontal FL by designing a SHAP variant while adhering to participants' privacy.

3 Background

We keep this paper self-contained by summarizing the key concepts necessary to comprehend our proposal.

Federated Learning. FL [14] aims to train an ML model by exploiting the cooperation of different parties while protecting user data. The main idea is that participants in the federation do not have to share their data among themselves or with a server. Each participant first trains a *local* model using their own data. Then, it sends the gradient or weights of the model to a central server or to the other participants to the end of learning a *global* and common model[1].

Depending on how many clients are involved in the training of the model and their nature, we can have two different types of FL: cross-silo and cross-

[1] We underline that the meaning of local and global in the context of FL is entirely different from the meaning in the context of XAI.

device [11]. In the cross-silo scenario, we only have a few clients (10–50) that should always be available during the training.

On the contrary, in the cross-device scenario, we can have millions of devices involved in the computation that can only train the model under certain conditions.

The most widely used architecture is the *server-based* one, where a central server orchestrates the communication between the clients and the server itself.

In this paper, we consider a cross-silos scenario with a server-based architecture. In particular, we adopt the *Federated Averaging (FedAvg)* aggregation algorithm [14]. In each round of this algorithm, the updated local models of the parties are transferred to the server, which then further aggregates the local models to update the global model. FedAvg works as follows. We suppose to have a central server S, which orchestrates the work of a federation of C clients. The goal is to train a neural network $\mathcal{N}$ by executing a given set of Federated rounds. The procedure starts with the server that randomly initializes the neural network parameters w_0 and then it executes the specified training rounds. We refer to them as *global iterations* to distinguish them from the training rounds executed on the client side, also called *local iterations.* A generic global iteration j can be split into four main phases: sending, local training, aggregation and evaluation phase. In the *sending* phase, the server samples a subset C_i of k clients and sends them w^j, that is the current global model's parameters, through the dedicated communication channels. Every client $c \in C_i$, after having received w^j, starts training it for E epochs on its private dataset, applying one classic optimizer, like SGD, Adam or RMSProp. The number of local epochs and the optimizer are user-defined parameters. Finally, the client c sends back to the server the updated model parameters w_c^{j+1}, ending the local training phase of the algorithm. When the server ends gathering all the results from the clients, it performs the *aggregation* phase, where it computes the new global model parameters, w^{j+1} as $w^{j+1} = w^j + \sum_{c \in C_i} \frac{n_c}{n} \Delta w_c^{j+1}$, where n_c is the number of records in the client c's training set and $n = \sum_{c \in C_i} n_c$. Therefore, in the last phase, the *evaluation* one, the server evaluates the new global model w^{j+1} according to the chosen metrics.

Feature Importance Explanations. Feature importance is one of the most popular types of explanation returned by local explanation methods [4,8]. For feature importance-based explanation methods, the explainer assigns to each feature an importance value which represents how much that particular feature is important for the prediction under analysis. Given a record x, an explainer $f(\cdot)$ models a feature importance explanation as a vector $e = \{e_1, e_2, \dots, e_f\}$, in which the value $e_i \in e$ is the importance of the i^{th} feature for the decision made by the black-box model $b(x)$. For understanding the contribution of each feature, the sign and the magnitude of each value e_i are considered. W.r.t. the sign, if $e_i < 0$, it means that the feature contributes negatively to the outcome y; otherwise, if $e_i > 0$, the feature contributes positively. The magnitude, instead, represents how great the contribution of the feature is to the final prediction y. In particular, the greater the value of $|e_i|$, the greater its contribution. Hence, when $e_i = 0$, it means that the i^{th} feature is showing no contribution. An example of a feature based explanation is $e = \{(age, 0.8), (income, 0.0), (education, -0.2)\}, y = deny$. In this case, *age* is the most important feature for the decision *deny*, *income* is not affecting the outcome, and *education* has a small negative contribution.

In this paper, we adopted SHapley Additive exPlanations (SHAP) [13] a local post-hoc model-agnostic explanation method computing features importance by means of Shapley values[2], a concept from cooperative game theory. SHAP is one of the most widely used explanation methods returning explanations in terms of feature importance. The explanations returned by SHAP are *additive feature attributions* and respect the following definition: $g(z') = \phi_0 + \sum_{i=1}^{F} \phi_i z'_i$, where z' is a record similar to x obtained as a copy of x where some features and values are replaced with some real values observed from the training set or from a reference set X, while $\phi_i \in \mathbb{R}$ are effects assigned to each feature, and F is the number of simplified input features. SHAP retains three properties: *(i) local accuracy*, meaning that $g(x)$ matches $b(x)$; *(ii) missingness*, which allows for features $x_i = 0$ to have no attributed impact on the SHAP values; *(iii) stability*, meaning that if a model changes so that the marginal contribution of a feature value increases (or stays the same), the SHAP value also increases (or stays the same) [15]. The construction of the SHAP values allows us to employ them both *locally*, in which each observation gets its own set of SHAP values, and *globally*, by exploiting collective SHAP values. We highlight that SHAP can be realized through different explanation models that differ in how they approximate the computation of the SHAP values. In our experiments, we adopted *KernelExplainer*, i.e., the completely model-agnostic version.

4 SHAP Explanations in Horizontal FL

Our proposal is to exploit SHAP [13] to explain the ML model learned by the FedAvg algorithm [14], in the case of Horizontal FL architecture. We recall that SHAP requires access to the training set D_{tr}, or to a "reference set" which is similar to the training set used by the model to explain, to create records z' to study the impact of each feature value in the final prediction. Sometimes, to speed up the explanation process, a medoid of the dataset is used or a small set of centroids [17] describing D_{tr} with a few records capturing the main characteristics, i.e. feature-values [15]. As a consequence, in server-based FL, in order to explain the learned global model, it is necessary that the server may gain access to the complete set of training data of its clients or has the possibility of computing the centroids of the dataset resulting from the union of the training sets of all the clients. Since the basic idea of FL is to avoid data sharing, in this setting we propose to have an explanation of the global model as the result of the aggregation of local (client-side) explanations.

Let $C = \{c_1, \ldots, c_m\}$ the set of m clients participating to the cooperation. After the FL algorithm, each client $c_i \in C$ has its ML model M_i received by

[2] We refer the interested reader to: https://christophm.github.io/interpretable-ml-book/shapley.html.

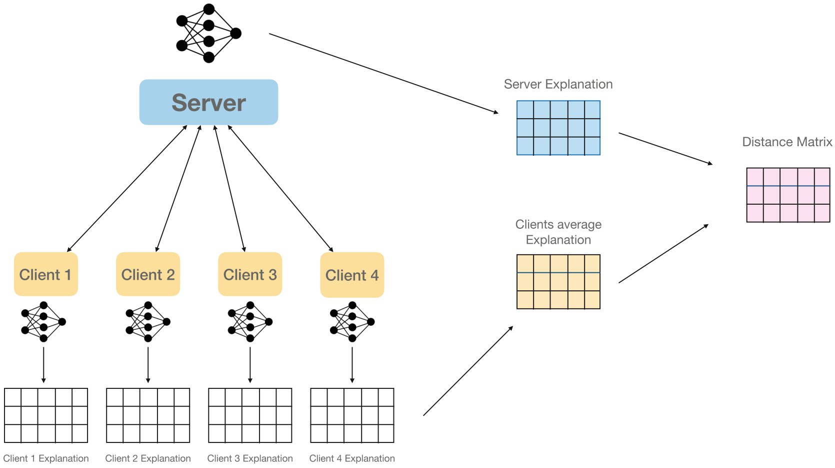

Fig. 1. Overview of our methodology. The server and all clients explain the model obtaining a matrix of SHAP values. The clients compute the mean of these matrices. To understand the difference between the explanations, we subtract the client's average explanation from the server explanation matrix.

the server. We denote with M_S the model on the server side resulting from the weights averaging. Each client c_i can derive a SHAP explainer ψ_i of its own model M_i which strongly depends on its training data. We propose to exploit the additive property of the SHAP values to generate explanations of the model M_S as an aggregation of explanations of the models belonging to M. More formally, given an instance x to be explained, the explanation of the prediction performed by the model M_S is obtained by $\psi_S(x) = \frac{1}{|C|} \sum_{c_i \in C} \psi_{c_i}(x)$. Specifically, the server's explanation $\psi_S(x)$ is composed by $|x|$ values resulting from the average of SHAP values of m clients, meaning that for each x_j we have $v_j = \frac{1}{|C|} \sum_{c_i \in C} \psi_{c_i}(x_j)$, where we assume that $\psi_{c_i}(x_j)$ returns the SHAP value associated by the client c_i to the feature x_j (Fig. 1).

Thus, according to our proposal, any client can derive its explanation for the instance x exploiting its own training data without the need to share them with the server, while the server only needs to receive the clients' explanations.

Analytical Methodology. In our experiments, we aim at comparing the proposed variant of SHAP explanations tailored for FL with the explanations obtained by the server. Hence, we propose an analytical methodology for validating our proposal based on the comparison of two settings: *(i)* the server gains access to training data of its clients i.e., $D_{tr} = \cup_{c_i \in C} D_{tr}^{c_i}$; *(ii)* the server cannot access training data and thus can only receive the clients' explanation for each prediction to be explained. In order to conduct our analysis given a test set D_{te} the following analytical methodology is applied:

- Each client c_i computes the SHAP explanation for each $x \in D_{te}$, i.e., it gets $\psi_{c_i}(x)$. Thus, each client produces a $k \times f$ matrix E^{c_i} where k is the number of records in D_{te} and f is the number of the features.

- A global explanation for each $x \in D_{te}$ is computed by averaging the clients' explanations as described above. Therefore, given the matrices $\{E^{c_1}, \dots, E^{c_m}\}$ we can compute the matrix $\hat{E}$ where each element $e_{ij} = \frac{1}{|C|} \sum_{c \in C} e^c_{ij}$. We call this explanation *clients-based explanation.*
- A *server-based explanation* is computed by simulating the server's access to the client's training data. Accessing training data, the server can obtain the SHAP explainer ψ_S which applies to each $x \in D_{te}$ and the $k \times f$ matrix E_S.
- Finally, given the two matrices E_S and $\hat{E}$ we analyze the differences to understand the degree of approximation introduced by our approach which does not assume data access. We perform this analysis by computing: *(i)* a difference matrix $\Delta = E_S - \hat{E}$; *(ii)* the average importance for each feature j produced by the two methods in the dataset D_{te} and then, how the two methods differ, this means computing a vector having for each feature j a value $\delta_j = \frac{1}{|k|} \sum_{i \in [1 \dots k]} \delta ij$.

5 Experiments

This section presents the experimental results obtained by applying the analytical methodology described in the previous section. We use the `CoverType` and `Adult` tabular datasets available in the UCI Machine Learning Repository[3]. `CoverType` contains $581,012$ records, 54 attributes and 7 different class labels. The attributes represent cartographic variables, and the class labels represent forest cover types. The classification task involves recognizing forest cover types. On the other hand, `Adult` is composed of $48,842$ records and 13 variables (after discarding "fnlwgt" and "education-num"), both numerical and categorical. Information such as age, job, capital loss, capital gain, and marital status can be found in it. The labels have values $<= 50K$ or $> 50K$, indicating whether the person will earn more or less than $50K$ in a fiscal year.

We defined ML models using Keras. In particular, for `CoverType`, we developed a model with three dense layers consisting of 1024, 512, and 256 units and a final output layer, while for `Adult`, we used a model with three dense layers with ten units. In both models, we used Relu as an activation function in each layer except for the output layer, where we applied softmax. After each layer, we used Dropout to prevent overfitting. We employed Flower [3] to simulate FL training. The experiments were performed on a server with an Intel Core i9-10980XE processor with 36 cores and 1 GPU Quadro RTX 6000.

In our experiments, we tested architectures with a different number of clients $m \in \{8, 16, 32\}$ involved in the computation. Indeed, one of the objectives of our analysis is to understand how this parameter impacts the aggregated explanation. In this preliminary experimentation, we considered a scenario where the clients have IID data which we distribute by stratified sampling. Also, each client has the same amount of samples. We are perfectly aware that this scenario is unlikely in real applications, and indeed we plan to perform further experiments on non-IID data. Nevertheless, the experimented configuration allows us to analyze FL impact on SHAP explanations without excessive variability.

[3] https://archive.ics.uci.edu/ml/index.php.

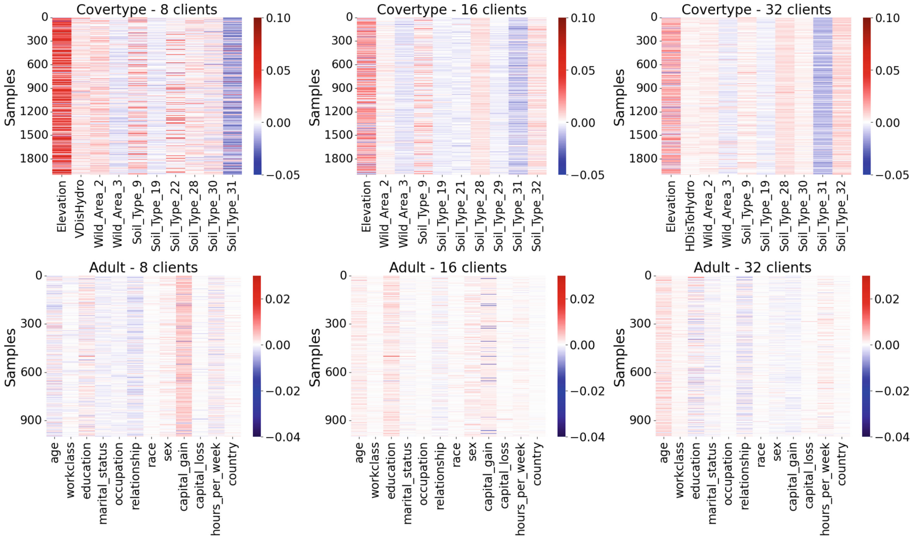

Fig. 2. Heatmaps showing the magnitude of the difference between server-based explanations and clients-based explanations for each sample. The first row shows the results for `CoverType` while the second one shows the results for `Adult`.

Results. In this section, we analyze the differences among the explanations with respect to two different aggregation criteria. Indeed, our goal is to investigate both the differences in the explanations from the point of view of the *features* and from the point of view of the *clients*.

In Fig. 2, we show through heatmaps, for each sample of the test set, the differences between the SHAP values of the server-based explanations and the ones of the clients-based explanations. These heatmaps are a graphical representation of the matrix Δ introduced in Sect. 4. To guarantee the readability of our results, in the plots of `CoverType`, we report only 10 features over 54, i.e., the features that, on average, have the highest discrepancy between the server-based explanations and clients-based explanations. As expected, the differences are negligible. For `CoverType` the features *“Soil_Type_31”* and *“Elevation”* have a greater divergence from 0. In particular, the clients-based explanation tends to overestimate the SHAP values of *“Elevation”* and underestimate the SHAP values of *“Soil_Type_31”*. We highlight that these two features present the highest divergence regardless of the number of clients involved in the training process. However, as we increase the number of clients, their divergence decreases. In `Adult`, we observe even smoother results in terms of divergence between server-based and clients-based explanations since the divergence varies in a smaller range with respect to `CoverType`. We can notice that, in any setting, we have only a couple of features having a magnitude of the difference more prominent with respect to the others. For example, in the setting with $m = 8$, clients *“capital_gain”* and *“realtionship”* present higher divergence.

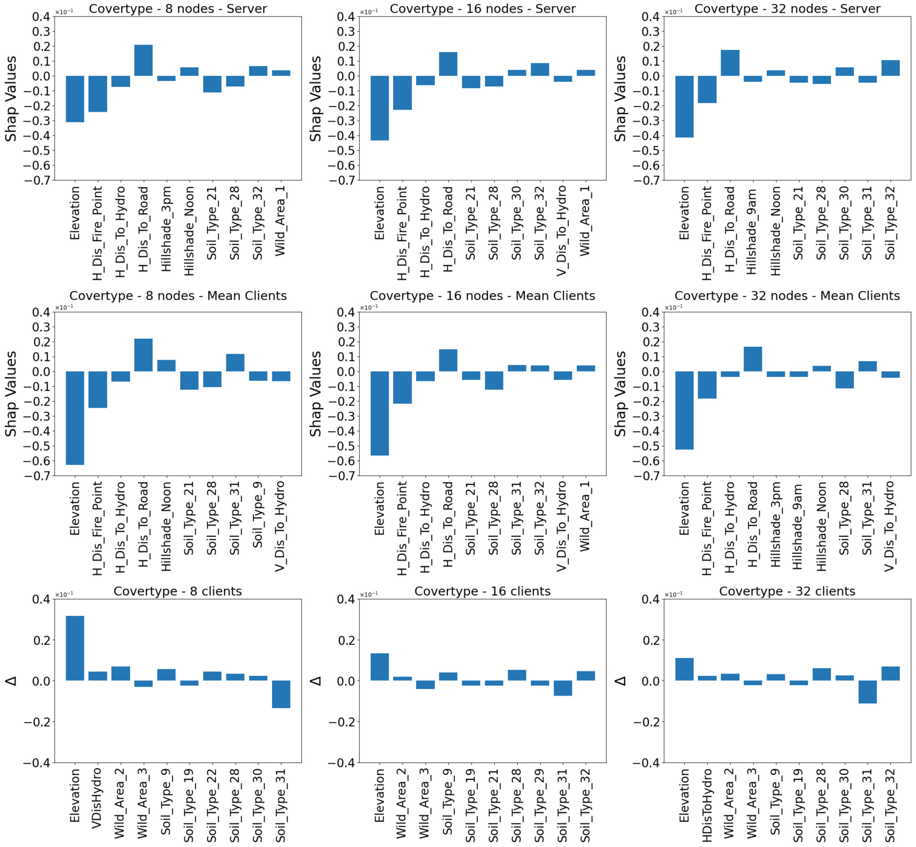

Fig. 3. SHAP values for `CoverType`. Top: calculated by the server. Middle: calculated by the clients. Bottom: Difference between SHAP values obtained by the server and those obtained by the clients.

We also conducted a more detailed analysis focused on the features. We report the results for `CoverType` in Fig. 3. The three plots in the first row depict the average SHAP values per feature of the server-based explanations, while the three in the middle row depict the average SHAP values per feature computed by clients-based explanations. As expected, these plots indicate that the two explanations almost always agree. The plots in the bottom row, instead, show the mean of the SHAP values for the top 10 features we selected for `CoverType`. They confirm our discussion based on the above heatmaps. Moreover, we observe that with an increasing number of clients m, some picks disappear, and the differences per feature vary in a smaller range of values.

Figure 4 shows the same analysis for `Adult`. As for `CoverType`, the two types of explanations almost always agree. Looking at the third row of the figure, we notice that, in general, the magnitude of the differences between the two

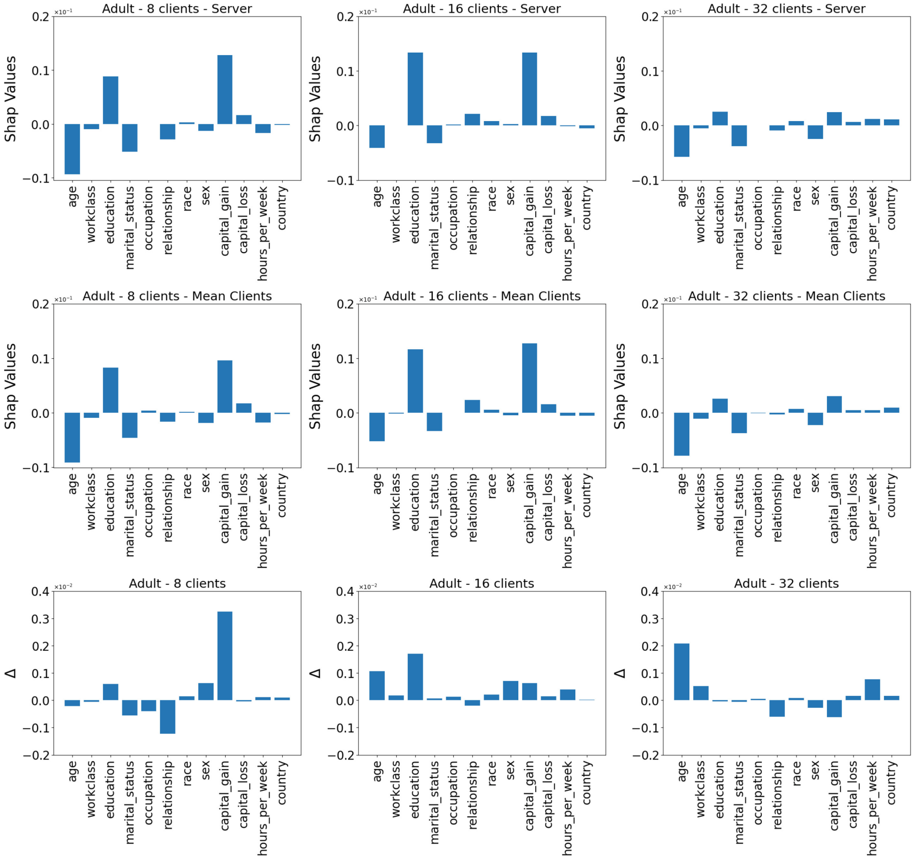

Fig. 4. SHAP values for `Adult`. Top: calculated by the server. Middle: calculated by the clients. Bottom: Difference between SHAP values obtained by the server and those obtained by the clients.

types of explanation decreases because, also in this case, some relevant picks disappear. As an example, the pick we have with the feature *"capital_gain"* in the experiment with $m = 8$ clients disappears as the number of clients increases.

Besides considering the differences in terms of SHAP values of *features*, we investigated the differences between the server-based explanation and the one performed on each client. This gives us the opportunity to understand if there are clients contributing more to the divergences between the two types of explanations. We report the results in Fig. 5. In `CoverType`, we observe that in the case of $m = 8$ clients, the difference with respect to the server is equal for all the clients. As the number of clients increases, we notice different behaviour among the various participants. Moreover, in the case with $m = 32$ clients, we observe an increase in the divergences with respect to the setting with $m = 16$ clients. This result is evident also in the last plot of the first row in Fig. 5.

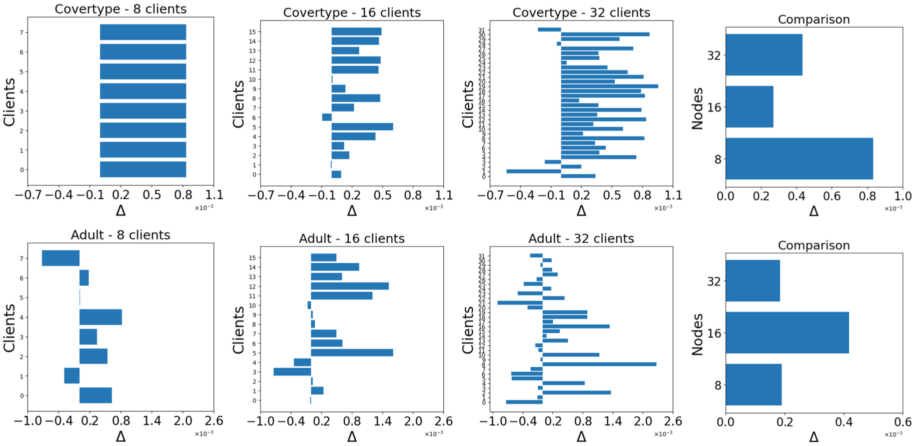

Fig. 5. Divergence of the server-based explanation w.r.t. clients-based. The first row reports results for `CoverType` and the second for `Adult`. For each setting, we plot the mean of the differences in the last plot of each row.

In `Adult`, we observe a different behaviour. The distance between the server and the various clients is different even when we use only 8 clients. As we increase the number of clients to 16, the distance increases, i.e., there are only a few clients with very low divergence and more clients with higher divergence. However, differently from the experiment with `CoverType`, when we increase the number of clients to 32, the overall difference decreases again (see the last plot of the second row in Fig. 5) because we have more clients with very low divergence.

In a nutshell, our results show that the clients-based explanation introduces a negligible approximation to SHAP values, proving that our method is promising.

6 Discussion of Findings

By aggregating local explanations, the proposed methodology investigates whether it is possible to derive an explanation for a model trained using Federated Learning. To achieve this goal, we exploited SHAP values' additive property. To be more specific, we aggregated the explanations computed by the individual clients to obtain a model explanation. We then compared this explanation with that of the server. The results obtained from the two datasets we considered support our initial guesses. Indeed, the differences between the aggregated explanation and the server explanation are minimal. Therefore, the explainer trained by the server and the one trained by the clients produce the same results. This means that they are both suitable for explaining a Federated Learning model. However, the explainer trained by the server requires some data to be transferred from the clients to the server to be trained. This is against the definition of Federated Learning [14]. By successfully showcasing the viability of aggregating local explanations, we proved that clients do not need to transmit their data to a central server. This ensures confidentiality and mitigates potential privacy risks due to data sharing.

By moving the explainers training from the server to the clients, we can also reduce the computation overhead on the server side. This is because it has only to perform the SHAP values aggregation. In addition, this approach could easily be extended and adapted to a peer-to-peer Federated Learning setting, where we would not have a server that could train an explainer. Instead, using our clients-based explanations, each client could first compute the explanations and then, after exchanging their SHAP values, aggregate them to derive the final explanation without sharing any data.

7 Conclusion

In this paper, we have presented a method for providing SHAP explanations in horizontal server-based Federated Learning systems. The basic idea is explaining an instance's prediction performed by the trained ML model by aggregating the explanation of the clients participating in the federation. Consequently, the proposed approach satisfies the strong requirements of a Federated Learning system by avoiding sharing clients' data with the server. We have presented empirical evidence that our proposal introduces an acceptable approximation to the SHAP explanations. In turn, it can be interpreted as a reasonable trade-off between privacy and utility. In future work, we intend to analyze the impact of adopting our method in a scenario with non-I.I.D. data distribution and in a peer-to-peer Federated learning setting where we do not have a central server. Moreover, we would also like to study the impact of a larger number of clients involved in the training. Lastly, we would also like to investigate the impact of privacy mitigation on explanation quality.

Acknowledgment. This work is partially supported by the EU NextGenerationEU programme under the funding schemes PNRR-PE-AI FAIR (Future Artificial Intelligence Research), "SoBigData.it - Strengthening the Italian RI for Social Mining and Big Data Analytics" - Prot. IR0000013, H2020-INFRAIA-2019-1: Res. Infr. G.A. 871042 *SoBigData++*, G.A. 761758 *Humane AI*, G.A. 952215 *TAILOR*, ERC-2018-ADG G.A. 834756 *XAI*, and CHIST-ERA-19-XAI-010 SAI, by MUR (N. not yet available), FWF (N. I 5205), EPSRC (N. EP/V055712/1), NCN (N. 2020/02/Y/ST6/00064), ETAg (N. SLTAT21096), BNSF (N. KP-06-AOO2/5).

References

1. Adadi, A., Berrada, M.: Peeking inside the black-box: a survey on explainable artificial intelligence (XAI). IEEE Access **6**, 52138–52160 (2018)
2. Bárcena, J.L.C., et al.: Fed-XAI: federated learning of explainable artificial intelligence models. In: XAI.it@AI*IA, CEUR Workshop Proceedings (2022)
3. Beutel, D.J., et al.: Flower: A friendly federated learning research framework (2020)
4. Bodria, F., Giannotti, F., Guidotti, R., Naretto, F., Pedreschi, D., Rinzivillo, S.: Benchmarking and survey of explanation methods for black box models. ArXiv: preprint, abs/2102.13076 (2021)

5. Doshi-Velez, F., Kim,B.: A roadmap for a rigorous science of interpretability. CoRR, abs/1702.08608 (2017)
6. Fiosina, J.: Explainable federated learning for taxi travel time prediction. In: VEHITS. SCITEPRESS (2021)
7. Freitas, A.A.: Comprehensible classification models: a position paper. SIGKDD Explor. **15**(1), 1–10 (2013)
8. Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., Pedreschi, D.: A survey of methods for explaining black box models. ACM Comput. Surv. **51**(5), 1–42 (2019)
9. Haffar, R., Sánchez, D., Domingo-Ferrer, J.: Explaining predictions and attacks in federated learning via random forests. Appl. Intell. , 1–17 (2022). https://doi.org/10.1007/s10489-022-03435-1
10. Janzing, D., Minorics, L., Blöbaum, P.: Feature relevance quantification in explainable AI: a causal problem. In: Chiappa,S., Calandra, R., (eds.) The 23rd International Conference on Artificial Intelligence and Statistics, AISTATS 2020, 26–28 August 2020, [Palermo, Sicily, Italy], volume 108 of Proceedings of Machine Learning Research, pp. 2907–2916. PMLR (2020)
11. Li, Q., Wen, Z., Wu, Z., Hu, S., Wang, N., He, B.: A survey on federated learning systems: Vision, hype and reality for data privacy and protection. arXiv e-prints (2019)
12. Longo, L., Goebel, R., Lecue, F., Kieseberg, P., Holzinger, A.: Explainable artificial intelligence: concepts, applications, research challenges and visions. In: Holzinger, A., Kieseberg, P., Tjoa, A.M., Weippl, E. (eds.) CD-MAKE 2020. LNCS, vol. 12279, pp. 1–16. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-57321-8_1
13. Lundberg, S.M., Lee, S.: A unified approach to interpreting model predictions. In: Guyon, I., et al., (eds.) Advances in Neural Information Processing Systems, vol. 30: Annual Conference on Neural Information Processing Systems 2017, 4–9 December 2017, Long Beach, CA, USA, pp. 4765–4774 (2017)
14. McMahan, B., Moore, E., Ramage, D., Hampson, S., Arcas, B.A.Y.: Communication-efficient learning of deep networks from decentralized data. In: Singh, A., Zhu, X.J., (eds.) Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017, 20–22 April 2017, Fort Lauderdale, FL, USA, volume 54 of Proceedings of Machine Learning Research, pp. 1273–1282. PMLR (2017)
15. Molnar, C.: Interpretable machine learning. Lulu. com (2020)
16. Pedreschi, D., Giannotti, F., Guidotti, R., Monreale, A., Ruggieri, S., Turini, F.: Meaningful explanations of black box AI decision systems. In: The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, 27 January–1 February 2019, pp. 9780–9784. AAAI Press (2019)
17. Tan, P., Steinbach, M.S., Kumar, V.: Introduction to Data Mining. Addison-Wesley, Boston (2005)
18. Wang, G.: Interpret federated learning with shapley values. ArXiv preprint, abs/1905.04519 (2019)
19. Wang, G., Dang, C.X., Zhou, Z.: Measure contribution of participants in federated learning. In: 2019 IEEE International Conference on Big Data (Big Data), Los Angeles, CA, USA, 9–12 December 2019, pp. 2597–2604. IEEE (2019)
20. Yang, Q., Liu, Y., Chen, T., Tong, Y.: Federated machine learning: concept and applications (2019)