Privacy-Preserving Federated Interpretability

Abtahi Fahliani, Azra; Aminifar, Amin; Aminifar, Amir

Published in:
Proceedings - 2024 IEEE International Conference on Big Data, BigData 2024

DOI:
10.1109/BigData62323.2024.10825590

2024

Document Version:
Peer reviewed version (aka post-print)

Privacy-Preserving Federated Interpretability

Azra Abtahi
Electrical and Information Technology
Lund University
Lund, Sweden
azra.abtahi_fahliani@eit.lth.se

Amin Aminifar
Institute of Computer Engineering
Heidelberg University
Heidelberg, Germany
amin.aminifar@ziti.uni-heidelberg.de

Amir Aminifar
Electrical and Information Technology
Lund University
Lund, Sweden
amir.aminifar@eit.lth.se

***Abstract*—Interpretability has become a crucial component in the Machine Learning (ML) domain. This is particularly important in the context of medical and health applications, where the underlying reasons behind how an ML model makes a certain decision are as important as the decision itself for the experts. However, interpreting an ML model based on limited local data may potentially lead to inaccurate conclusions. On the other hand, centralized decision making and interpretability, by transferring the data to a centralized server, may raise privacy concerns due to the sensitivity of personal/medical data in such applications.**

In this paper, we propose a federated interpretability scheme based on SHAP (SHapley Additive exPlanations) value and DeepLIFT (Deep Learning Important FeaTures) to interpret ML models, without sharing sensitive data and in a privacy-preserving fashion. Our proposed federated interpretability scheme is a decentralized framework for interpreting ML models, where data remains on local devices, and only values that do not directly describe the raw data are aggregated in a privacy-preserving fashion to interpret the model.

***Index Terms*—explainable machine learning, privacy-preserving, federated learning, epilepsy, seizure prediction, seizure detection, EEG, ECG.**

I. Introduction

For a Machine Learning (ML) model, interpretability is crucial because it enables users to understand how the model operates. This interpretability is especially important in fields where decisions made by ML models have major consequences, for instance, in the healthcare and medical domain. In these domains, it is as important for the experts to understand how an ML model makes a certain decision, as the decision itself.

One of the main challenges, however, in these domains is that interpreting an ML model based on limited data available locally may potentially lead to drawing inaccurate conclusions. At the same time, transferring the data to a centralized server for centralized decision making and interpretation, may raise privacy concerns due to the sensitivity of personal/medical data in such applications. For instance, in the healthcare domain, medical data such as patient records are highly sensitive. Many countries have stringent regulations, such as GDPR in Europe [1] and HIPAA in the USA [2], governing the use and transfer of personal data. This sensitivity poses a challenge to the interpretability of ML models, as the data required to understand model behavior is often dispersed across multiple institutions and geographic locations.

Federated Learning (FL) is a machine learning framework where multiple clients, such as mobile devices or organizations, collaboratively train a model under a central server's coordination while keeping the training data decentralized [3]–[16]. To date, several studies on the interpretability of FL have been conducted. Such studies target Interpretable Client Selection [17]–[24], Interpretable Sample Selection [20], [24]–[28], Interpretable Feature Selection [29]–[33], Interpretable Model Optimization [34]–[37], and Interpretable Contribution Evaluation [38]–[46]. These works focus on interpreting FL at various stages. However, the collaborative, privacy-preserving interpretation of an ML model in a federated setting has not been addressed to date.

In this paper, we propose the first privacy-preserving federated interpretability framework, to the best of our knowledge, inspired by FL. Our framework focuses mainly on SHAP value [47], [48] and DeepLIFT [49] to thoroughly interpret an ML model based on the decentralized data of different parties, such as health centers, without explicitly sharing the raw data. Federated interpretability is a decentralized approach for interpreting ML models, where data remains on local devices, and only values that do not directly describe the raw data are shared and aggregated to interpret the model. However, adopting federated interpretability does not fully address the privacy concerns associated with decentralized computing. Therefore, in this work, we ensure that all data shared among the parties involved in the federated interpretability remains secure by adopting state-of-the-art privacy-preservation schemes. Our main contributions are summarized as follows:

- We propose a federated framework to perform interpretability considering the local data available on all participating parties/clients. This is required because relying only on the local data at each party to interpret the ML model, as its distribution may be different from the overall data, can result in inaccuracies in the interpretation.
- We extend this framework by introducing a privacy-preserving scheme to enable federated interpretability without privacy concerns. This is ensured not only be-

This research has been partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP), Swedish Research Council (VR), ELLIIT Strategic Research Environment, Swedish Foundation for Strategic Research (SSF), and European Union (EU) Interreg Program. The computations and data handling were enabled by resources provided by the Swedish National Infrastructure for Computing (SNIC) partially funded by the Swedish Research Council (VR) through grant agreement no. 2018-05973.

cause the local data on each device remains local in federated interpretability, but also thanks to the secure-multi-party-computation scheme adopted.
- We evaluate our proposed framework considering two well-established interpretability schemes, namely, SHAP value [47], [48] and DeepLIFT [49], based on two real-world medical applications for mobile devices, namely, epilepsy seizure prediction using EPILEPSIAE electrocardiogram (ECG) dataset [50] and epilepsy seizure detection using CHB-MIT Scalp electroencephalogram (EEG) dataset [51].

This paper is structured as follows: In Section II, we introduce federated interpretability based on SHAP Value and DeepLIFT. Then, in Section III, we extend our proposed scheme and present the privacy-preserving federated interpretability framework. In Section IV-A, we discuss the scenarios considered for evaluating the proposed scheme, along with their corresponding experimental setups. The experimental results are presented in Section IV. Finally, Section V provides the conclusion of this paper.

II. Federated Interpretability

In this section, we illustrate our federated interpretability framework, focusing mainly on SHAP value [47], [48] and DeepLIFT [49]. The abstract overview of our privacy-preserving federated interpretability procedure is illustrated in Fig. 1.

A. Federated Interpretability Based on SHAP Value

Let us first consider the well-established framework of SHAP [47], [48], [52]. The concept of SHAP values originates from this game theory question: *How should we divide up the payoff among the players with different skills in a coalition?* To address this, the marginal contribution of each player is calculated by adding them to the coalition set, and these contributions are then averaged across all possible sets in which the player could have joined.

This approach has been adapted to evaluate the importance of different features in ML models. In this context, the SHAP value represents the average contribution of a feature value to the model's output, considering all possible sets in which it could be integrated. Therefore, it indicates the feature value's importance.

Additionally, SHAP values offer insights into how each feature affects the model's output and how decisions are made by the model. Specifically, they allow us to determine when a feature significantly influences the output and whether its impact is positive or negative. Analyzing the SHAP values enables us to potentially pinpoint the feature values that are highly probable to result in a specific classification outcome or prediction value for a feature with a significant impact on the output.

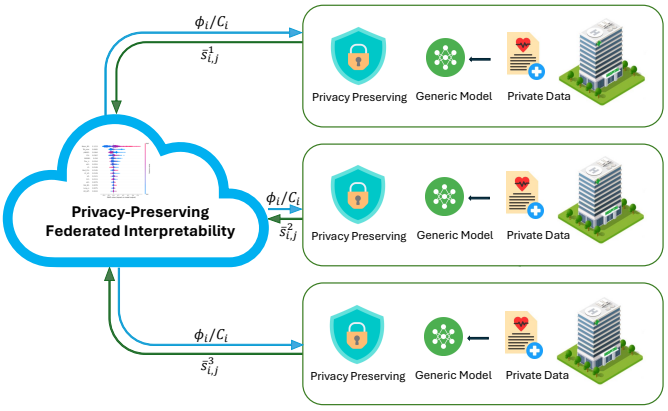


Fig. 1: Privacy-Preserving Federated Interpretability

Let us consider that we have a model with n features. For a given sample $\boldsymbol{x}$ with feature values $x_1, x_2, \ldots, x_n$, the SHAP value decomposition for the sample $\boldsymbol{x}$ is expressed as follows:

$$f(\boldsymbol{x}) = \mathbb{E}[f(\boldsymbol{x})] + \sum_{i=1}^{n} \phi_i(\boldsymbol{x}), \tag{1}$$

where $f(\boldsymbol{x})$ is the model's output for the sample $\boldsymbol{x}$; $\phi_i(\boldsymbol{x})$ is the SHAP value for feature i, representing the contribution of feature x_i to the output; and $\mathbb{E}[f(\boldsymbol{x})]$ is the expected value of the model's output over the entire dataset, serving as the reference.

For a given sample $\boldsymbol{x}$, the SHAP value for feature i is calculated as follows [47], [53]:

$$\phi_i(\boldsymbol{x}) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!\,(|N| - |S| - 1)!}{|N|!} \left(f(S \cup \{i\}) - f(S)\right), \tag{2}$$

where N is the set of all features; S is a subset of features excluding feature i; $|S|$ is the size of subset S; $|N| = n$ is the total number of features. Furthermore, $f(S)$ is the model's output using only the features in subset S, and $f(S \cup \{i\})$ is the model's output when feature i is added to subset S. The term $\frac{|S|!\,(|N|-|S|-1)!}{|N|!}$ is the weight given to each marginal contribution, ensuring that the contributions are averaged over all possible subsets of features.

We assume that we have m samples, denoted by $\boldsymbol{x}^j$ for $j = 1, \ldots, m$, for the interpretation of the model. In general, the importance of feature i is defined as follows:

$$\phi_i = \frac{\sum_{j=1}^{m} |\phi_i(\boldsymbol{x}^j)|}{m}. \tag{3}$$

Now, let us consider that the overall data is distributed over different parties/clients, such as health centers or wearable devices. Due to privacy concerns, we cannot share the data for interpretation. If we calculate the SHAP value for a specific feature value considering the same generic ML model at each party, based on Equation (2), the SHAP value remains unchanged compared to the case of having all data for interpretability. In other words, the SHAP values for a specific

input sample will remain the same regardless of whether the entire dataset is used or just part of it. This is because the SHAP values are computed based on the sample's feature values and the model, not the set of samples being considered. However, the lack of sufficient data can lead to inaccuracies in determining the significance of each feature, their impact on the output, and the interactions between features.

In federated interpretability, if we have K parties, the importance of feature i corresponding to party u for $u \in \{1, 2, \dots K\}$, denoted by ϕ_i^u, can be calculated privately and then be shared. Each party also shares the number of exploited samples for this calculation, denoted by m_i^u. Then, the overall importance of feature i is calculated as follows:

$$\phi_i = \frac{\sum_{u=1}^{K} m_i^u \cdot \phi_i^u}{\sum_{u=1}^{K} m_i^u}. \tag{4}$$

Besides determining feature importance, by collectively looking into the individual SHAP values received from different parties, we can thoroughly interpret the model. This enables us to understand how each feature affects the output, including whether a feature has a positive or negative impact on the outcome, or if it favors one class in classification tasks. With this information, we can further optimize the model to improve its performance.

B. Federated Interpretability Based on DeepLIFT

DeepLIFT [49] is a technique for attributing a neural network's output to its input features, offering a way to interpret model decisions. It addresses the limitations in the traditional attribution methods, such as gradient-based approaches [54], [55], which can be noisy and prone to saturation.

DeepLIFT assigns attribution scores that explain how each feature's deviation from a reference impacts the output. It is particularly effective when the reference represents a neutral or typical input. Using customized backpropagation, DeepLIFT computes each feature's contribution relative to a reference by propagating the difference between actual and reference outputs.

Compared to SHAP, DeepLIFT is more computationally efficient because it does not require evaluating all possible feature combinations. Its backpropagation-based approach scales well with deep networks, making it faster and more efficient, especially when a clear reference exists. DeepLIFT also has extensions like DeepLIFT-SHAP, which approximates Shapley values using its methodology.

In DeepLIFT, the *reference* is essential for interpreting the contributions of each input feature. Let x_i^r represent the value of the i-th input feature in the reference. The contribution of the deviation in the i-th input feature, i.e., $\Delta x_i = x_i - x_i^r$, to the change in output $\Delta f(\boldsymbol{x}) = f(\boldsymbol{x}) - f(\boldsymbol{x}^r)$, is denoted as $C_{\Delta x_i \to \Delta f(\boldsymbol{x})}$. DeepLIFT computes this attribution score using *multiplier* $M_{\Delta x_i \to \Delta f(\boldsymbol{x})}$:

$$C_{\Delta x_i \to \Delta f(\boldsymbol{x})} = M_{\Delta x_i \to \Delta f(\boldsymbol{x})} \cdot \Delta x_i.$$

The multiplier $M_{\Delta x_i \to \Delta f(\boldsymbol{x})}$ is computed using the *chain rule*, which propagates contributions through multiple layers in the network. If we assume an input layer with neurons x_i and a hidden layer with neurons y_j, the multiplier $M_{\Delta x_i \to \Delta f}$ can be obtained based on $M_{\Delta x_i \to \Delta y_j}$ and $M_{\Delta y_j \to \Delta f}$ as follows,

$$M_{\Delta x_i \to \Delta f} = \sum_j M_{\Delta x_i \to \Delta y_j} \cdot M_{\Delta y_j \to \Delta f}. \tag{5}$$

The chain rule ensures that the contribution of the input feature x_i is propagated layer by layer, through both linear and non-linear transformations, using appropriate multipliers at each layer. For linear layers, the multiplier is proportional to the weights between the input and the hidden neurons. For non-linear layers (e.g., ReLU), the multiplier reflects the difference in activations between the actual and reference inputs [49].

By recursively applying the chain rule across all layers, DeepLIFT ensures that the sum of all feature contributions equals the total change in the model's output:

$$\Delta f(\boldsymbol{x}) = \sum_i C_{\Delta x_i \to \Delta f(\boldsymbol{x})}.$$

For interpreting the model across m samples, denoted by $\boldsymbol{x}^j$ for $j = 1, \dots, m$, the importance of feature i for the model output can be obtained by taking the absolute value of each feature's attribution score $C_{\Delta x_i^j \to \Delta f(\boldsymbol{x}^j)}$ and averaging them:

$$C_i = \frac{\sum_{j=1}^{m} |C_{\Delta x_i^j \to \Delta f(\boldsymbol{x}^j)}|}{m} \tag{6}$$

This provides a robust measure of feature importance by aggregating attribution scores across multiple outputs and instances.

Now, let us assume that the entire dataset is spread across several parties/clients, such as health centers or wearable devices. Due to privacy concerns, sharing the data for the interpretation is not feasible. Hence, we propose to use federated interpretability, where instead of sharing the local data, the mean absolute attribution scores of different features are shared. In federated interpretability, if we have K parties, the importance of feature i based on the data available at party u, denoted by C_i^u, can be calculated privately and then be shared. Each party also shares the number of exploited samples, i.e., m_i^u, in this process. Finally, the overall importance of feature i is calculated as follows:

$$C_i = \frac{\sum_{u=1}^{K} m_i^u C_i^u}{\sum_{u=1}^{K} m_i^u}. \tag{7}$$

III. Privacy-Preserving Federated Interpretability

The objective of this section is to build on our proposed approach in the previous section and enable interpretability among several parties each with local data in a federated fashion, while ensuring privacy. We present the proposed scheme step by step, as presented in the following.

A. Privacy-preserving calculation of the distribution of feature contributions

In this section, we focus on the distributions of data, because both SHAP or DeepLIFT work with distributions. SHAP extracts the distribution of feature contribution across all samples. Similarly, DeepLIFT extracts the contribution of each feature.

1) Setup for privacy-preserving federated interpretability scheme: The first step in our proposed scheme is the setup of pair-wise private keys. Our approach here is based on the scheme proposed in [56]. We consider K data-holder parties, and the set of parties is represented by $\mathcal{U}$. In our method, all participating parties must establish pair-wise private keys to generate random masks for secure aggregation. To facilitate this, the Diffie-Hellman key exchange protocol [57] is adopted by the parties.

In this scheme, each party, denoted by p^u for $u = 1, \ldots, K$, shares its public key with the other $K-1$ parties via the server. The private pairwise key between p^u and p^v is generated on each party and denoted by $\text{DHKey}_{u,v}$.

2) Distribution of feature contributions: Next, we discuss how to obtain the distribution of each feature contribution in a privacy-preserving fashion, as follows:

- Step 1: Each party p^u locally calculates the distribution of SHAP values or DeepLIFT attribution scores for each feature i across all local data points.
- Step 2: Each party p^u divides the distribution of SHAP values or DeepLIFT attribution scores for each feature into discrete intervals or bins (e.g., based on quantiles or fixed ranges). Then, calculate the Binned Contribution Scores for each bin j of each feature i on party p^u, denoted as $b^u_{i,j}$.[1]
- Step 3: Each party p^u multiplies $b^u_{i,j}$ by $m^u_{i,j}$, the number of SHAP values or attribution scores in bin j (where $\sum_j m^u_{i,j} = m^u_i$), to obtain the secret value $s^u_{i,j}$. This secret value will be securely aggregated with the secret values from other parties.
- Step 4: Each party p^u generates random masks using the Diffie-Hellman pair-wise keys:
  - $\text{Mask}^u_{\text{hide}}$: Generate and aggregate masks based on $\text{DHKey}_{u,v}$, for all $v \in \mathcal{U} : u < v$.
  - $\text{Mask}^u_{\text{reveal}}$: Generate and aggregate masks based on $\text{DHKey}_{u,v}$, for all $v \in \mathcal{U} : u > v$.
- Step 5: The masked secret value is calculated as $\overline{s}^u_{i,j} = s^u_{i,j} + \text{Mask}^u_{\text{hide}} - \text{Mask}^u_{\text{reveal}}$.[2]
- Step 6: The masked results are sent to the central server, which aggregates them and divides by the total number of samples, $\sum_{u=1}^K m^u_i$, to calculate the global contribution score $b_{i,j}$.

3) Mean absolute attributions: Finally, we discuss how to obtain other quantitative values, e.g., mean absolute attribution ϕ_i in SHAP, from the distribution of each feature contribution in a privacy-preserving fashion. The following briefly outlines the process for securely obtaining the mean absolute attribution scores, given the distribution of each feature contribution, i.e., $b_{i,j}$. Let us assume that each bin captures the occurrence of feature contribution in the interval $[I_{i,j}, I_{i,j+1})$, i.e., $h_{i,j}$. Then, the mean absolute attributions can be estimated as follows: $\phi_i \approx (\sum_{\forall j} h_{i,j})^{-1} \cdot (\sum_{\forall j} h_{i,j} \cdot \frac{I_{i,j}+I_{i,j+1}}{2})$. Note that, the mean absolute attribution ϕ_i can be obtained with an arbitrary precision, by using fine-grain bins in the histogram. Alternatively, given each party has computed the local mean absolute attribution ϕ^u_i, the proposed scheme can be adjusted to exactly obtain the value of $\phi_i = (\sum_{u=1}^K m^u_i)^{-1} \cdot (\sum_{u=1}^K s^u_{i,1})$, if we assume that our distribution has only one bin.

4) Privacy and correctness: As discussed, $\text{Mask}_{\text{hide}}$ and $\text{Mask}_{\text{reveal}}$ are the results of aggregating several other masks generated based on the pairwise Diffie-Hellman keys. A mask generated based on $\text{DHKey}_{u,v}$ is denoted by $\mathcal{M}_{u,v}$. Note that $\mathcal{M}_{u,v} = \mathcal{M}_{v,u}$ since $\text{DHKey}_{u,v} = \text{DHKey}_{v,u}$.

Privacy: To determine the secret value of p^u, we need both $\text{Mask}^u_{\text{hide}}$ and $\text{Mask}^u_{\text{reveal}}$. Calculating these masks requires all pairwise Diffie-Hellman keys between p^u and the other parties to generate all relevant $\mathcal{M}$ values associated with p^u.

Correctness: Here, we show that the sum of the masked secret values equals the sum of the original secret values:

$$\begin{aligned}\sum_{u=1}^K \overline{s}^u_{i,j} &= \sum_{u=1}^K s^u_{i,j} + \sum_{u=1}^K \text{Mask}^u_{\text{hide}} - \sum_{u=1}^K \text{Mask}^u_{\text{reveal}} \\ &= \sum_{u=1}^K s^u_{i,j} + \sum_{u=1}^K \sum_{v\in\mathcal{U}:u<v} \mathcal{M}_{u,v} - \sum_{u=1}^K \sum_{v\in\mathcal{U}:v<u} \mathcal{M}_{u,v} \\ &= \sum_{u=1}^K s^u_{i,j}, \qquad \pmod R.\end{aligned} \tag{8}$$

To prove that the above equation holds, we show that the set of pairs summed over in the two expressions $\sum_{u=1}^K \sum_{v\in\mathcal{U}:v<u} \mathcal{M}_{u,v}$ and $\sum_{u=1}^K \sum_{v\in\mathcal{U}:u<v} \mathcal{M}_{u,v}$ are equal, i.e.,

$$\sum_{u=1}^K \sum_{v\in\mathcal{U}:u<v} \mathcal{M}_{u,v} - \sum_{u=1}^K \sum_{v\in\mathcal{U}:v<u} \mathcal{M}_{u,v} = 0, \qquad \pmod R. \tag{9}$$

Let us consider the two sets $A = \{(u,v) \in \mathcal{U}\times\mathcal{U} \mid u < v\}$ and $B = \{(u,v) \in \mathcal{U}\times\mathcal{U} \mid v < u\}$. The first sum is over pairs in A, and the second sum is over pairs in B. We define a bijection $\psi : A \to B$ by swapping the elements of each pair: $\psi(u,v) = (v,u)$. For any $(u,v) \in A$, since $u < v$, it follows that $(v,u) \in B$. Similarly, for any $(v,u) \in B$, swapping gives $(u,v) \in A$. Therefore, ψ is a bijection between A and B.

Using the bijection ψ and the equality of $\mathcal{M}_{u,v}$ and $\mathcal{M}_{v,u}$, we can relate the sums over A and B:

$$\sum_{\forall(u,v)\in A} \mathcal{M}_{u,v} = \sum_{(v,u)=\psi(u,v),\forall(u,v)\in A} \mathcal{M}_{v,u} = \sum_{\forall(v,u)\in B} \mathcal{M}_{v,u}.$$

Therefore, the sum over A is equal to the sum over B: $\sum_{(u,v)\in A} \mathcal{M}_{u,v} = \sum_{(u,v)\in B} \mathcal{M}_{u,v}$. This means that the difference of the two sums in equation (9) is zero.

[1] Note that b_i corresponds to ϕ_i or C_i in the previous section.

[2] All calculations are modulo R, i.e., (mod R).

B. Handling dropped parties

The proposed protocol is designed assuming that all parties reliably communicate with the server during the entire process. In certain scenarios, e.g., in real-world applications such as Internet of Things (IoT) settings, however, several parties may fail to communicate their results to the server. This failure disrupts the proposed scheme's process because the masks corresponding to the dropped parties are introduced by other available parties but are not canceled by the missing masks from the unavailable parties.

To address this issue, we implement a k-out-of-n threshold secret sharing scheme, namely the Shamir secret sharing method [58]. The objective of this scheme is to divide a secret into n pieces such that any k out of those n parties can collaborate to reconstruct the original secret when necessary.

Using this approach, each party divides its private Diffie-Hellman key into K pieces and distributes each piece to a different party. In the event that party p^u fails to communicate its masked data to the server, the server can request the other parties to share their respective pieces of p^u's private Diffie-Hellman key. The server can then reconstruct p^u's private Diffie-Hellman key and generate the corresponding masks related to p^u, allowing it to remove those masks from the result.

Next, we discuss the privacy and correctness aspects of our approach discussed above.

Privacy: Since the private Diffie-Hellman key of each party is divided and shared using a k-out-of-n threshold secret sharing scheme, it can be reconstructed through the collaboration of k parties. Reconstructing this key allows the regeneration of the masks and their removal from the masked secret value, thereby revealing the original secret value. The privacy challenge associated with this approach is discussed in Section III-C. Specifically, if the masked secret values of a party, which was assumed to be unavailable, are delayed and later delivered to the server, even an honest-but-curious server could potentially identify the secret value. This vulnerability is addressed in Section III-C by introducing individual masks for each party.

Correctness: If a party becomes unavailable, its masks are regenerated by the server. Therefore, for all parties, regardless of their availability, we have $\text{Mask}_{\text{hide}}$ and $\text{Mask}_{\text{reveal}}$. Consequently, in Equation 8, since $s_{i,j}$ for the unavailable parties is zero (a neutral value for aggregation), then the equality $\sum_{u=1}^{K} \overline{s}_{i,j}^{u} = \sum_{u=1}^{K} s_{i,j}^{u}$ still holds.

C. Addressing the privacy issue for handling dropped parties

In certain cases, network delays may cause the input from a party, which was initially assumed to be dropped, to arrive late at the server. In such a situation, the private Diffie-Hellman key of that party might have already been reconstructed under the assumption that the party was unavailable. This creates a privacy risk, even in scenarios where the server is honest-but-curious, since the server could use the regenerated masks to infer the party's secret information.

To mitigate this risk, each party must aggregate its masked secret with an additional individual mask. These individual masks do not cancel out with each other. To ensure that the individual masks can be removed from the final aggregation result, each party splits its individual mask into K pieces and shares each piece with another party using the k-out-of-n threshold secret sharing scheme.

Once the server has received all the inputs from the available parties, it will request the shares of the individual masks from the available parties. Based on any k of these pieces, the server can reconstruct the individual masks and remove them from the aggregation result. It is important to note that to maintain privacy, the server must not have access to both the individual mask value and the mask generated based on the Diffie-Hellman key. Therefore, after all the inputs from available parties have been received, the server will: (i) For parties that were available and whose inputs were received: request the pieces of their individual masks. (ii) For parties that were dropped and whose inputs were not received: request the pieces of their private Diffie-Hellman keys.

Now, we discuss the privacy and correctness aspects of our approach presented above.

Privacy: Each party's individual mask is divided and shared using a k-out-of-n threshold secret sharing scheme, allowing for reconstruction through the collaboration of k parties. In our scheme, the server can request either the shares of a party's private Diffie-Hellman key or its individual mask. However, the server cannot simultaneously remove both the individual masks and the pairwise masks from a party's masked secret values.

Correctness: For all parties, $\overline{s}_{i,j}$ includes their individual mask. If a party is available, its individual mask will be reconstructed from the k pieces received from k parties. This mask will then be subtracted from $\overline{s}_{i,j}$, leaving the result as the secret value masked by pairwise masks. If a party is unavailable, its individual mask can be considered as zero. Therefore, as shown in Equation 8, the sum of the masked secret values (after removing the individual masks) will be equal to the sum of the secret values.

D. Communication overhead of our scheme

In the setup phase, i.e., Section III-A, all parties share their public Diffie-Hellman keys through the server. This means that each party sends and receives one message. Moreover, regarding the sharing of the splits of private Diffie-Hellman keys and individual masks using the Shamir scheme, as discussed in Sections III-B and III-C, each party encrypts each split based on the pairwise Diffie-Hellman key and shares the splits through the server. This also involves each party sending and receiving one message, but this occurs sequentially after the previous step since the pairwise Diffie-Hellman keys are needed for this step.

In the main secure aggregation phase, each party sends one message to the server to share their masked secret values. The server then checks the availability of parties, and based on that, it sends a message to all parties instructing them to either

share a split of their private Diffie-Hellman key (if the party was unavailable) or a split of their individual mask (if the party was available).

IV. Evaluation

A. Experimental Setup

We consider two different applications of federated interpretability: federated interpretability for epileptic seizure prediction based on ECG and federated interpretability for epileptic seizure detection based on EEG.

1) Federated Interpretability for Epileptic Seizure Prediction Based on ECG: Epilepsy is a brain disorder causing recurrent seizures. The prediction of epileptic seizures can improve epileptic patients' lives by preventing or reducing the severity of the seizures by administering drugs, providing first aid in time, and preventing accidents caused due to seizures. In this paper, first, we focus on using federated interpretability for interpreting the epileptic seizure prediction done by a generic ML model. This interpretation can be done to collect important information to update the generic model, while the data remains at the local wearable devices.

ECG Dataset and Data Preparation: We utilize the ECG data from the public EPILEPSIAE dataset [50], which is among the largest epilepsy datasets manually annotated by medical experts for the purposes of seizure detection and prediction. The recordings are conducted in a routine clinical setting, meaning they may contain various non-seizure activities and artifacts such as head/body movements, chewing, blinking, early stages of sleep, and electrode pops/movement. The dataset includes complex partial (CP), simple partial (SP), and secondarily generalized (GS) seizures, with no restrictions on seizure types.

The EPILEPSIAE ECG data is collected from 30 patients, covering 4603 hours of recordings segmented into one-hour files, with 277 seizures recorded. The recordings are sampled at 256 Hz with 16-bit resolution. The number of seizures per patient ranges from 5 to 23, with an average of 9.23 seizures per patient, and the average seizure duration is 75.81 seconds. The total recording time per patient varies between 92.90 and 266.36 hours, with an average of 153.43 hours.

For the pre-ictal signals, we select 14 minutes from the one-hour segments containing seizures, ensuring the selected period is at least one hour away from the previous seizure and one minute before the next. For inter-ictal signals (those far from seizures), we also select 14 minutes, ensuring the signal is at least one hour away from both the previous and next seizures. These pre-ictal and inter-ictal signals are then windowed into 60-second segments with a 30-second overlap. Any windowed signals with no information, e.g., signals equal to zero, are discarded.

To divide the data into training and test sets, we group all windowed signals from each one-hour file into one of the sets. This approach prevents any overlap across the training and test sets. 70% of the windowed signals are allocated to training and 30% for testing. Next, we balance the data so that each set contains an equal number of pre-ictal and inter-ictal windowed signals.

Seizure Prediction Model: For epileptic seizure prediction, we use a Random Forest classifier [59] to train a prediction model based on the EPILEPSIAE dataset. The model incorporates well-established Heart Rate Variability (HRV) features, which have been previously utilized for seizure prediction [60]–[62], in addition to ECG Lorenz features, primarily employed for seizure detection [63]–[67]. We assume the model is trained using the entire training set. Subsequently, all clients utilize this trained model.

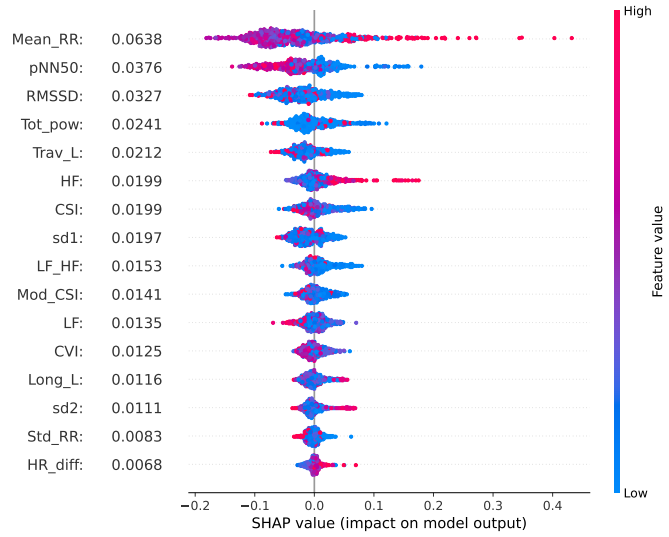
2) Federated Interpretability for Epileptic Seizure Detection Based on EEG: Besides epileptic seizure prediction, which aims to forecast seizures before they occur, it is important to consider seizure detection. Detection focuses on identifying seizures as they happen in real-time, allowing for immediate intervention to ensure patient safety. Seizure detection is critical for recognizing and responding to ongoing events, helping to mitigate risks and manage seizures effectively. In this paper, we also consider using federated interpretability based on DeepLIFT for interpreting the epileptic seizure detection, done by a generic Deep Neural Networks (DNN) model.

EEG Dataset and Data Preparation: We use CHB-MIT Scalp EEG Dataset [51]. This dataset contains 23 cases from 22 patients (5 males and 17 females) with epilepsy. Only two channels (T7F7 and T8F8) are considered to maintain consistency with wearable IoT devices for real-time seizure monitoring [68]–[70]. Patients 6, 14, and 16 are excluded due to very short-lasting seizures.

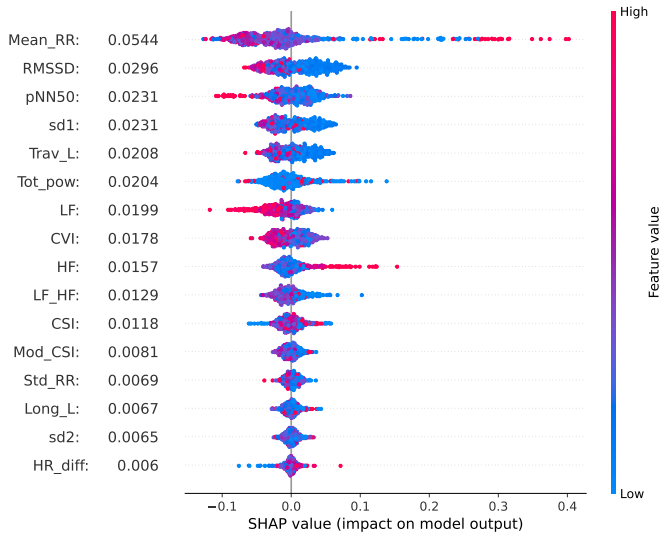
A bandpass filter with a passband of 1–30 Hz is applied to the raw EEG signals. The filtered signal is segmented with a window length of 4 seconds (i.e., 1024 samples) and standardized using Z-score normalization [71]. The FFT is computed for the windowed signal, and the FFTs of the two channels are concatenated into 2048 dimensions, serving as input for the DNNs. The dataset includes seizure and non-seizure data, where 70% of each is used for training. The 30% remaining data is split randomly in such a way that 70% is allocated to validation and 30% to testing.

Seizure Detection Model: For epileptic seizure detection, we use an end-to-end model, a 1-Dimensional Convolutional Neural Network (1D-CNN), exploiting the FFT of the windowed EEG signals. This 1D-CNN consists of two convolutional layers with ReLU activation for feature extraction, followed by max-pooling layers to reduce dimensionality. A dropout layer with a 20% rate is included to prevent overfitting. The extracted features are passed through two fully connected layers, with the final layer producing an output for binary classification. We assume the model is trained using the entire training set. Subsequently, all clients utilize this trained model.

3) Implementation Details: In this study, we trained, validated, tested, and interpreted our models in Python, leveraging the SHAP and DeepLIFT packages for interpretation. All experiments conducted in this study were performed on a system characterized by an 11th Gen Intel(R) Core(TM) i7-11800H



(a) SHAP value summary plot from one of the clients/wearable devices.



(b) SHAP value summary plot by federated interpretability.

Fig. 2: SHAP value summary plots from an individual client/wearable device and federated interpretability.

@ 2.30GHz, 2304 Mhz, 8 Core(s), 16 Logical Processor(s), and a physical memory (RAM) capacity of 16.0 GB.

B. Experimental Results

In this section, we examine the performance of proposed federated interpretability. First, we focus on SHAP-based federated interpretability for epileptic seizure prediction using ECG features. Next, we evaluate the performance of DeepLIFT-based federated interpretability for epileptic seizure detection, utilizing an end-to-end model based on EEG data.

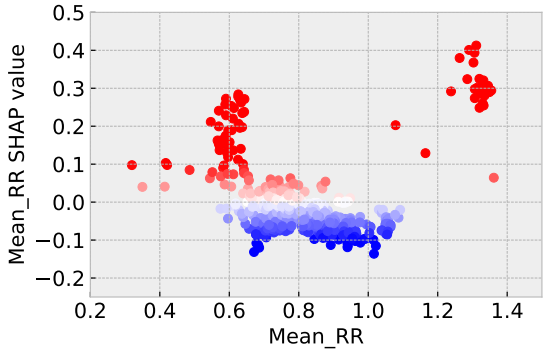
1) Evaluation of SAHP-Based Federated Interpretability: The summary plots of the SHAP values for the generic seizure prediction model, based on test data from a single wearable device/patient and leveraging federated interpretability, are shown in Fig. 2. In a SHAP value summary plot, the interpretation for each feature value is represented by a single dot on each feature row. In this plot, feature importance decreases from top to bottom, determined by the mean absolute SHAP values across all feature values. In a summary plot, the horizontal axis indicates whether a feature's effect is associated with a higher output (favoring the pre-ictal class) or a lower output (favoring the inter-ictal class). The color represents feature values, with high values shown in red and low values in blue.

According to Fig. 2b, exploiting federated interpretability, the four most important features are "Mean_RR," "RMSSD," "pNN50," and "sd1," respectively, as they appear on top of the summary plot. However, if we have access to only one client's or patient's test data, the most important features in this model are identified as "Mean_RR," "pNN50," "RMSSD," and "Tot_pow," respectively.

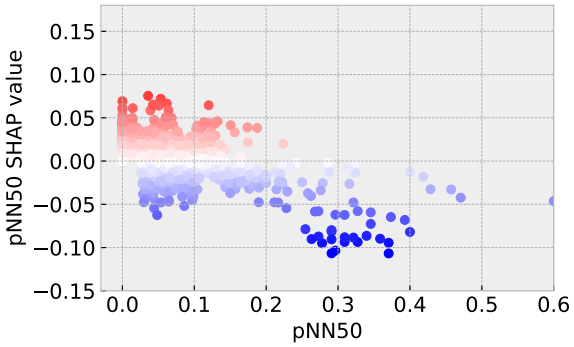
Fig. 3 presents SHAP dependence plots for "Mean_RR" and "pNN50," based on test data from one of the wearable devices/patients and utilizing federated interpretability. These plots show the SHAP values corresponding to the feature values. SHAP dependence plots can be used to gain deeper insights into the influence of individual features on the output of an ML model. In this figure, we observe that for federated interpretability when "Mean_RR" values are more than 1.12, the SHAP values are consistently positive and also not negligible, implying that samples with "Mean_RR" values more than 1.12 are likely to be classified as pre-ictal samples by the prediction model. However, if we rely on the data from only one of the wearable devices, we cannot reach this conclusion due to the lack of sufficient test samples.

Furthermore, if we only consider the test data from one wearable device, the "pNN50" dependence plot (Fig. 3b) may suggest that samples with "pNN50" values between 0.34 and 0.39 consistently have significant negative SHAP values, implying that these samples are likely to be classified as inter-ictal by the prediction model. However, when looking at Fig. 3d, which accounts for SHAP values across all clients, we can see this conclusion is not accurate. There is a considerable number of SHAP values close to zero, indicating that in this interval, other features also play a significant role in decision-making.

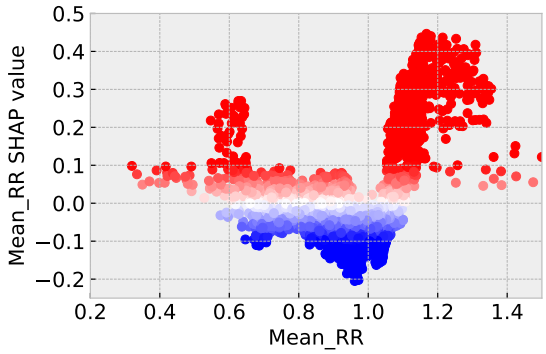
2) Evaluation of DeepLIFT-Based Federated Interpret ability: Frequency components of EEG signals are valuable for detecting epileptic seizures [72]. By analyzing the DeepLIFT mean absolute attribution scores, it is possible to identify which components or frequency bands play a more significant role in this detection. The DeepLIFT mean absolute attribution scores for the FFT of the windowed EEG signals in our DNN seizure detection model, based on the available test data at one of the wearable devices and by utilizing federated interpretability, are shown in Fig. 4.



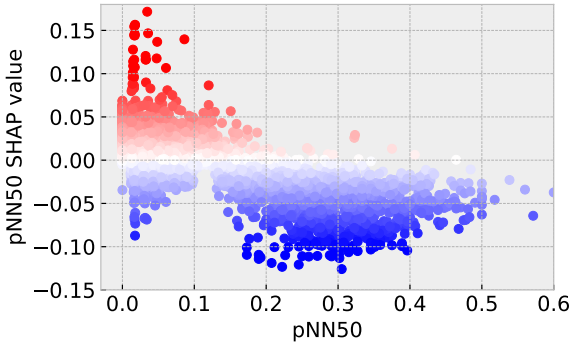
(a) Mean_RR Dependence plot from one of the clients/wearable devices.



(b) pNN50 Dependence plot from one of the clients/wearable devices.



(c) Mean_RR Dependence plot by federated interpretability.



(d) pNN50 Dependence plot by federated interpretability.

Fig. 3: Dependence plots from an individual client/wearable device and federated interpretability.

As it can be seen from Fig. 4, interpreting the generic ML model using local data from a single client yields a different understanding of important frequency components compared to federated interpretability, which takes into account the overall data distribution without requiring data sharing across clients. Hence, any generic model debugging or updating based solely on the available data from only one client, in order to be applicable across all different clients, can be inaccurate.

V. Conclusions

In this paper, we address the collaborative, privacy-preserving interpretation of a generic ML model, which we refer to as *federated interpretability*, inspired by FL. We propose federated interpretability based on SHAP values and DeepLIFT to interpret a generic ML model deployed across different parties/clients, such as health centers, without sharing their data. Federated interpretability is a decentralized approach for interpreting ML models, where data remains on local devices, and only values that do not directly describe the raw data are shared and aggregated to interpret the model. We extend our framework by introducing a privacy-preserving scheme to enable federated interpretability without privacy concerns. This is ensured not only because the local data on each device remains local, but also thanks to the secure-multi-party-computation scheme adopted.

We have evaluated the performance of our proposed federated interpretability framework considering two medical applications. First, we evaluate SHAP-based federated interpretability for epileptic seizure prediction using ECG features. Then, we assess the performance of DeepLIFT-based federated interpretability for epileptic seizure detection, utilizing an end-to-end DNN model based on EEG data. Our experimental results confirm the importance of federated interpretability in enabling the model interpretation, with data distributed among different parties/clients.

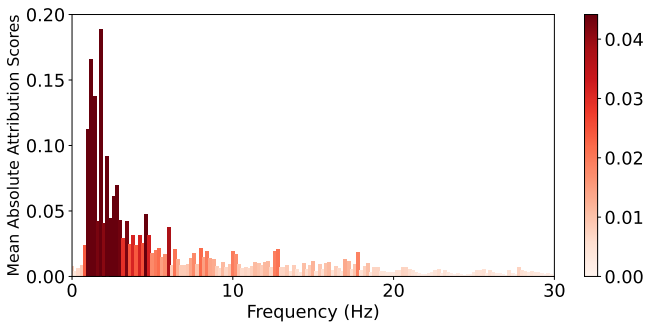
References

[1] European Parliament and Council of the European Union, "Regulation (eu) 2016/679 of the european parliament and of the council of 27 april 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data," *Official Journal of the European Union*, 2016. [Online]. Available: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A32016R0679

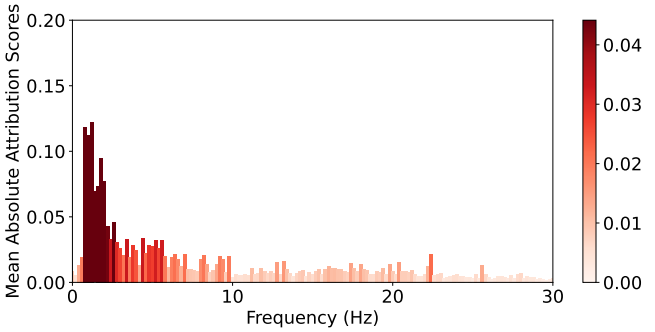
[2] U.S. Department of Health and Human Services, "Health insurance portability and accountability act of 1996," 1996. [Online]. Available: https://www.hhs.gov/hipaa/for-professionals/privacy/index.html

[3] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*. PMLR, 2017.

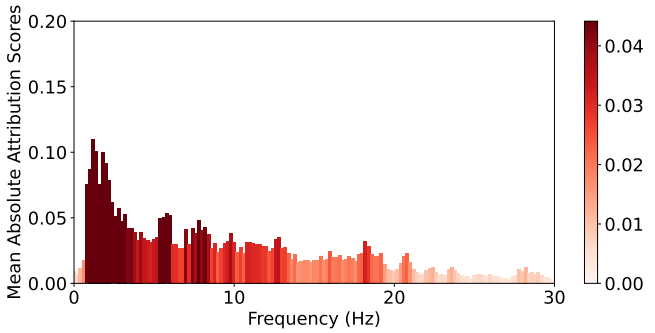
[4] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings *et al.*, "Advances and open problems in federated learning," *Foundations and trends® in machine learning*, 2021.



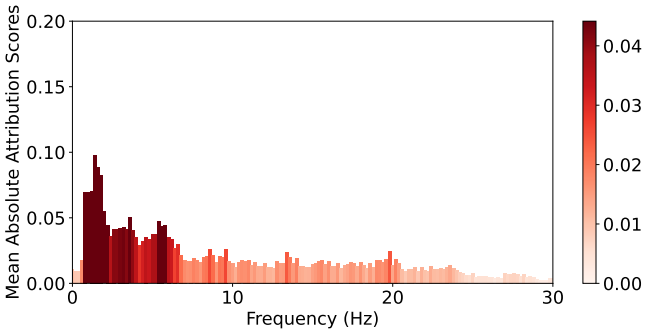
(a) CH1- one of the wearable devices



(b) CH2- one of the wearable devices



(c) CH1- Federated Interpretability



(d) CH2- Federated Interpretability

Fig. 4: Mean absolute DeepLIFT attribution score achieved from one of the wearable devices and by federated interpretability for EEG channels CH1 and CH2. (Top row: one of the wearable devices, Bottom row: federated interpretability)

[5] S. Baghersalimi, T. Teijeiro, D. Atienza, and A. Aminifar, "Personalized real-time federated learning for epileptic seizure detection," *IEEE J. Biomed. Health Inform.*, 2021.

[6] A. Aminifar, F. Rabbi, and Y. Lamo, "Scalable privacy-preserving distributed extremely randomized trees for structured data with multiple colluding parties," in *IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2021.

[7] A. Aminifar, F. Rabbi, K. I. Pun, and Y. Lamo, "Privacy preserving distributed extremely randomized trees," in *Proceedings of the 36th annual acm symposium on applied computing*, 2021.

[8] A. Aminifar, M. Shokri, F. Rabbi, V. K. I. Pun, and Y. Lamo, "Extremely randomized trees with privacy preservation for distributed structured health data," *IEEE Access*, 2022.

[9] S. Baghersalimi, T. Teijeiro, A. Aminifar, and D. Atienza, "Decentralized federated learning for epileptic seizures detection in low-power wearable systems," *IEEE Transactions on Mobile Computing*, 2023.

[10] A. Aminifar, M. Shokri, and A. Aminifar, "Privacy-preserving edge federated learning for intelligent mobile-health systems," *Future Generation Computer Systems*, 2024.

[11] P. Qi, D. Chiaro, A. Guzzo, M. Ianni, G. Fortino, and F. Piccialli, "Model aggregation techniques in federated learning: A comprehensive survey," *Future Generation Computer Systems*, 2024.

[12] D. Annunziata, M. Canzaniello, D. Chiaro, S. Izzo, M. Savoia, and F. Piccialli, "On the dynamics of non-iid data in federated learning and high-performance computing," in *IEEE 32nd Euromicro International Conference on Parallel, Distributed and Network-Based Processing)*, 2024.

[13] P. Qi, D. Chiaro, and F. Piccialli, "Small models, big impact: A review on the power of lightweight federated learning," *Future Generation Computer Systems*, 2024.

[14] M. M. Salim, D. Camacho, and J. H. Park, "Digital twin and federated learning enabled cyberthreat detection system for iot networks," *Future Generation Computer Systems*, 2024.

[15] A. Raza, A. Guzzo, and G. Fortino, "Federated learning for medical images analysis: A meta survey," in *IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress*. IEEE, 2023.

[16] J.-H. Syu and J. C.-W. Lin, "Heterogeneous federated learning for non-iid smartwatch data classification," *IEEE Internet of Things Journal*, 2024.

[17] G. Wang, C. X. Dang, and Z. Zhou, "Measure contribution of participants in federated learning," in *IEEE international conference on big data*, 2019.

[18] P. W. W. Koh, K.-S. Ang, H. Teo, and P. S. Liang, "On the accuracy of influence functions for measuring group effects," *Advances in neural information processing systems*, 2019.

[19] J. Goetz, K. Malik, D. Bui, S. Moon, H. Liu, and A. Kumar, "Active federated learning," *arXiv preprint arXiv:1909.12641*, 2019.

[20] A. Li, L. Zhang, J. Wang, J. Tan, F. Han, Y. Qin, N. M. Freris, and X.-Y. Li, "Efficient federated-learning model debugging," in *IEEE 37th International Conference on Data Engineering*, 2021.

[21] L. Zhang, L. Fan, Y. Luo, and L.-Y. Duan, "Intrinsic performance influence-based participant contribution estimation for horizontal federated learning," *ACM Transactions on Intelligent Systems and Technology*, 2022.

[22] A. Li, L. Zhang, J. Wang, F. Han, and X.-Y. Li, "Privacy-preserving efficient federated-learning model debugging," *IEEE Transactions on Parallel and Distributed Systems*, 2021.

[23] Y. Xue, C. Niu, Z. Zheng, S. Tang, C. Lyu, F. Wu, and G. Chen, "Toward understanding the influence of individual clients in federated learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021.

[24] A. Li, L. Zhang, J. Tan, Y. Qin, J. Wang, and X.-Y. Li, "Sample-level data selection for federated learning," in *IEEE Conference on Computer Communications*, 2021.

[25] A. Katharopoulos and F. Fleuret, "Not all samples are created equal: Deep learning with importance sampling," in *International conference on machine learning*. PMLR, 2018.

[26] T. Tuor, S. Wang, B. J. Ko, C. Liu, and K. K. Leung, "Overcoming noisy

and irrelevant data in federated learning," in *International Conference on Pattern Recognition*. IEEE, 2021.
[27] J. Shin, Y. Li, Y. Liu, and S.-J. Lee, "Fedbalancer: Data and pace control for efficient federated learning on heterogeneous clients," in *Proceedings of the 20th Annual International Conference on Mobile Systems, Applications and Services*, 2022.
[28] P. W. Koh and P. Liang, "Understanding black-box predictions via influence functions," in *International conference on machine learning*. PMLR, 2017.
[29] P. Cassará, A. Gotta, and L. Valerio, "Federated feature selection for cyber-physical systems of systems," *IEEE Transactions on Vehicular Technology*, 2022.
[30] X. Li, R. Dowsley, and M. De Cock, "Privacy-preserving feature selection with secure multiparty computation," in *International Conference on Machine Learning*. PMLR, 2021.
[31] X. Zhang, A. Mavromatis, A. Vafeas, R. Nejabati, and D. Simeonidou, "Federated feature selection for horizontal federated learning in iot networks," *IEEE Internet of Things Journal*, 2023.
[32] S. Feng, "Vertical federated learning-based feature selection with non-overlapping sample utilization," *Expert Systems with Applications*, 2022.
[33] A. Li, H. Peng, L. Zhang, J. Huang, Q. Guo, H. Yu, and Y. Liu, "Fedsdg-fs: Efficient and secure feature selection for vertical federated learning," in *IEEE Conference on Computer Communications*, 2023.
[34] A. Imakura, H. Inaba, Y. Okada, and T. Sakurai, "Interpretable collaborative data analysis on distributed data," *Expert Systems with Applications*, 2021.
[35] K. Cheng, T. Fan, Y. Jin *et al.*, "Secureboost: A lossless federated learning framework," *IEEE intelligent systems*, 2021.
[36] S. R. Pandey, N. H. Tran, M. Bennis, Y. K. Tun, A. Manzoor, and C. S. Hong, "A crowdsourcing framework for on-device federated learning," *IEEE Transactions on Wireless Communications*, 2020.
[37] Y. Zhan, P. Li, Z. Qu, D. Zeng, and S. Guo, "A learning-based incentive mechanism for federated learning," *IEEE Internet of Things Journal*, 2020.
[38] T. Song, Y. Tong, and S. Wei, "Profit allocation for federated learning," in *IEEE International Conference on Big Data*, 2019.
[39] S. Wei, Y. Tong, Z. Zhou, and T. Song, "Efficient and fair data valuation for horizontal federated learning," *Federated Learning: Privacy and Incentive*, 2020.
[40] Z. Fan, H. Fang, Z. Zhou, J. Pei, M. P. Friedlander, and Y. Zhang, "Fair and efficient contribution valuation for vertical federated learning," *arXiv preprint arXiv:2201.02658*, 2022.
[41] S. Gollapudi, K. Kollias, D. Panigrahi, and V. Pliatsika, "Profit sharing and efficiency in utility games," in *25th Annual European Symposium on Algorithms*. Schloss-Dagstuhl-Leibniz Zentrum für Informatik, 2017.
[42] T. Nishio, R. Shinkuma, and N. B. Mandayam, "Estimation of individual device contributions for incentivizing federated learning," in *IEEE Globecom Workshops*, 2020.
[43] J. Wang, L. Zhang, A. Li, X. You, and H. Cheng, "Efficient participant contribution evaluation for horizontal and vertical federated learning," in *IEEE 38th International Conference on Data Engineering*, 2022.
[44] Z. Liu, Y. Chen, Y. Zhao, H. Yu, Y. Liu, R. Bao, J. Jiang, Z. Nie, Q. Xu, and Q. Yang, "Contribution-aware federated learning for smart healthcare," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022.
[45] R. Younis, Z. Ahmadi, A. Hakmeh, and M. Fisichella, "Flames2graph: An interpretable federated multivariate time series classification framework," in *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2023.
[46] Y. Chen, Y. Ning, Z. Chai, and H. Rangwala, "Federated multi-task learning with hierarchical attention for sensor data analytics," in *International Joint Conference on Neural Networks*. IEEE, 2020.
[47] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," *Advances in Neural Information Processing Systems*, 2017.
[48] M. Sundararajan and A. Najmi, "The many shapley values for model explanation," in *Proceedings of the International Conference on Machine Learning*. PMLR, 2020.
[49] A. Shrikumar, P. Greenside, and A. Kundaje, "Learning important features through propagating activation differences," in *International conference on machine learning*. PMlR, 2017.
[50] M. Ihle, H. Feldwisch-Drentrup, C. A. Teixeira, A. Witon, B. Schelter, J. Timmer, and A. Schulze-Bonhage, "Epilepsiae–a european epilepsy database," *Computer Methods and Programs in Biomedicine*, 2012.
[51] A. H. Shoeb, "Application of machine learning to epileptic seizure onset detection and treatment," Ph.D. dissertation, Massachusetts Institute of Technology, 2009.
[52] M. Christoph, *Interpretable machine learning: A guide for making black box models explainable*. Leanpub, 2020.
[53] S. M. Lundberg, G. G. Erion, and S.-I. Lee, "Consistent individualized feature attribution for tree ensembles," *arXiv preprint arXiv:1802.03888*, 2018.
[54] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," in *Workshop at International Conference on Learning Representations*, 2014.
[55] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic attribution for deep networks," in *International conference on machine learning*. PMLR, 2017.
[56] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth, "Practical secure aggregation for privacy-preserving machine learning," in *proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 2017.
[57] W. Diffie and M. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, 1976.
[58] A. Shamir, "How to share a secret," *Communications of the ACM*, 1979.
[59] L. Breiman, "Random forests," *Machine learning*, 2001.
[60] K. Fujiwara, M. Miyajima, T. Yamakawa, E. Abe, Y. Suzuki, Y. Sawada, M. Kano, T. Maehara, K. Ohta, T. Sasai-Sakuma *et al.*, "Epileptic seizure prediction based on multivariate statistical process control of heart rate variability features," *IEEE Transactions on Biomedical Engineering*, 2015.
[61] L. Billeci, D. Marino, L. Insana, G. Vatti, and M. Varanini, "Patient-specific seizure prediction based on heart rate variability and recurrence quantification analysis," *PloS one*, 2018.
[62] T. Yamakawa, M. Miyajima, K. Fujiwara, M. Kano, Y. Suzuki, Y. Watanabe, S. Watanabe, T. Hoshida, M. Inaji, and T. Maehara, "Wearable epileptic seizure prediction system with machine-learning-based anomaly detection of heart rate variability," *Sensors*, 2020.
[63] J. Jeppesen, S. Beniczky, P. Johansen, P. Sidenius, and A. Fuglsang-Frederiksen, "Using lorenz plot and cardiac sympathetic index of heart rate variability for detecting seizures for patients with epilepsy," in *Proceedings of the 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2014.
[64] ——, "Detection of epileptic seizures with a modified heart rate variability algorithm based on lorenz plot," *Seizure*, 2015.
[65] F. Forooghifar, A. Aminifar, L. Cammoun, I. Wisniewski, C. Ciumas, P. Ryvlin, and D. Atienza, "A self-aware epilepsy monitoring system for real-time epileptic seizure detection," *Mobile Networks and Applications*, 2019.
[66] F. Forooghifar, A. Aminifar, and D. Atienza, "Resource-aware distributed epilepsy monitoring using self-awareness from edge to cloud," *IEEE Transactions on Biomedical Circuits and Systems*, 2019.
[67] F. Forooghifar, A. Aminifar, T. Teijeiro, A. Aminifar, J. Jeppesen, S. Beniczky, and D. Atienza, "Self-aware anomaly-detection for epilepsy monitoring on low-power wearable electrocardiographic devices," in *IEEE International Conference on Artificial Intelligence Circuits and Systems*, 2021.
[68] D. Sopic, A. Aminifar, and D. Atienza, "e-glass: A wearable system for real-time detection of epileptic seizures," in *IEEE International Symposium on Circuits and Systems*, 2018.
[69] B. Huang, R. Zanetti, A. Abtahi, D. Atienza, and A. Aminifar, "Epilepsynet: Interpretable self-supervised seizure detection for low-power wearable systems," in *Proceedings of the IEEE 5th International Conference on Artificial Intelligence Circuits and Systems*, 2023.
[70] B. Huang, A. Abtahi, and A. Aminifar, "Lightweight machine learning for seizure detection on wearable devices," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2023.
[71] D. G. Zill, *Advanced engineering mathematics*. Jones & Bartlett Learning, 2020.
[72] G. Yogarajan, N. Alsubaie, G. Rajasekaran, T. Revathi, M. S. Alqahtani, M. Abbas, M. M. Alshahrani, and B. O. Soufiene, "Eeg-based epileptic seizure detection using binary dragonfly algorithm and deep neural network," *Scientific Reports*, 2023.