IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 28 July 2022, accepted 21 August 2022, date of publication 26 August 2022, date of current version 31 August 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3202008

RESEARCH ARTICLE

BOFRF: A Novel Boosting-Based Federated Random Forest Algorithm on Horizontally Partitioned Data

MERT GENCTURK[1,2], A. ANIL SINACI[2], AND NIHAN KESIM CICEKLI[1]
[1]Computer Engineering Department, Middle East Technical University, 06800 Ankara, Turkey
[2]SRDC Software Research & Development and Consultancy Corporation, ODTU Teknokent, 06800 Ankara, Turkey

Corresponding author: Mert Gencturk (mert@srdc.com.tr)

This work was supported by the European Union's Horizon 2020 Research and Innovation Programme under Agreement 824666.

ABSTRACT The application of federated learning on ensemble methods is a common practice with the goal of increasing the predictive power of local models. However, although existing federated solutions utilizing ensemble methods can achieve this when the datasets of sites are balanced and of good quality, i.e., the local models are already above a certain accuracy threshold, they usually fail to provide the same level of improvement to the models of sites that have an unsuccessful classifier because of their poor quality or imbalanced data. To address this challenge, we propose a novel federated ensemble classification algorithm for horizontally partitioned data, namely Boosting-based Federated Random Forest (BOFRF), which not only increases the predictive power of all participating sites, but also provides significantly high improvement on the predictive power of sites having unsuccessful local models. We implement a federated version of random forest, which is a well-known bagging algorithm, by adapting the idea of boosting to it. We introduce a novel aggregation and weight calculation methodology that assigns weights to local classifiers based on their classification performance at each site without increasing the communication or computation cost. We evaluate the performance of our proposed algorithm in different federated environments that we set up by using four healthcare datasets. The empirical results show that BOFRF improves the predictive power of local random forest models in all cases. The advantage of BOFRF is that the level of improvement it provides for sites having unsuccessful local models is significantly high unlike existing solutions.

INDEX TERMS Ensemble learning, federated learning, machine learning, privacy-preservation, random forest classification.

I. INTRODUCTION

In today's technological age, information systems are incessantly collecting massive amounts of data in their repositories. The analysis of such data and extracting knowledge from them has become an important concept for many domains such as security, finance, healthcare, and transportation where data are clustered in a number of different systems and organizations. Machine learning algorithms can be used to interpret information by building mathematical models on existing data to make predictions or decisions without human intervention, if these data are combined on a large scale [1].

The associate editor coordinating the review of this manuscript and approving it for publication was Xiong Luo.

However, sharing sensitive data is strictly controlled by laws and regulations such as the General Data Protection Regulation (GDPR) [2] enforced by the European Union in 2018 to preserve the privacy of personal information; hence, data providers are reluctant to allow data to leave their premises. Consequently, personal data cannot be shared between sites or collected by a third party and remain in isolated data silos, hampering the extraction of their inherent actionable insights and intelligence [3].

Privacy has become an important issue in many machine-learning applications as in all other fields that deal with sensitive data. To prevent the reveal of sensitive data to the outside world, several privacy preservation methods have been developed, such as k-anonymity and l-diversity [4].

Such techniques mainly achieve privacy preservation by modifying or removing some parts of the original data. Although the transformation of data may provide privacy, it may reduce the quality of the data, which is formally known as utility, causing defective results in machine learning. In addition, these methods are open to adversarial attacks that are used to reveal hidden sensitive information, especially when the attacker has some background knowledge of the data or when combined with some publicly available data [5].

A way of dealing with these challenges is to apply the concept of federated learning. The aim of federated learning is to build a joint machine learning model based on the data residing at multiple sites by exchanging only information about locally trained models, not actual data [6]. Federated learning has been proven to be a powerful mechanism as it enables multiple parties to build a better global model than the individual local models in a privacy-preserving setting. It can be applied to both horizontally partitioned data, where the datasets at different sites share the same set of features for different ID spaces, and vertically partitioned data, where the datasets contain different sets of features for the same ID space [7]. Although federated learning is a new concept proposed by Google in 2016 [8], in the literature, there have been many implementations of it using different techniques and focusing on different aspects, such as providing more security [9], [10], decreasing communication cost [11], [12], reducing computation cost [13], and increasing model performance.

The application of federated learning on ensemble methods is a common practice with the goal of increasing the power of the predictive model. Most state-of-the-art horizontal federated learning approaches utilizing ensemble methods are built on boosting techniques, especially AdaBoost, which is a well-known and extensively studied algorithm in the literature [14], [15]. In these approaches, the sites first execute the AdaBoost algorithm locally on their training data, and then the local models are transferred to a third-party coordinator or shared between sites to build an integrated model. Various algorithms handle this integration process differently. For instance, AdaBoost.PL aims to combine like-minded classifiers, hence sorts the weak classifiers in the local models with respect to their weights and merges the classifiers with similar correctness at the same sorted level [14]. BOPPID focuses on the data distribution differences between the sites during the integration and assigns weights to the local models according to the sampling size of the sites and by giving more importance to the site's own local model [15]. However, although existing federated solutions utilizing ensemble methods can improve the prediction power of local models when the datasets of sites are balanced and of good quality (i.e., the local models are already above a certain accuracy threshold), they usually fail to provide the same level of improvement to the models of sites that have an unsuccessful classifier because of their bad quality or imbalanced data. For example, BOPPID, which is one of the most successful federated implementations of AdaBoost [15], always assigns more weight to the site's own local model compared to any other local model with the same sample size, because it assumes that other sites might have a different data distribution; hence, they should not be given more importance. This prevents a site having an unsuccessful classifier from taking advantage of successful classifiers of other sites to improve its local model, which results in the site not achieving an accuracy as good as the others.

In this paper, we propose a novel and practical federated ensemble classification algorithm for horizontally partitioned data, namely Boosting-based Federated Random Forest (BOFRF), which not only increases the predictive power of all participating sites, but also provides significantly high improvements on the predictive power of sites having unsuccessful local models. In this regard, we implement a federated version of random forest, which is a well-known bagging algorithm, by adapting the idea of boosting to it. In the integration step, we introduce a novel aggregation and weight calculation methodology that assigns weights to local classifiers based on their classification performance at each site instead of proportioning them with the sample size or site index without increasing the communication or computation cost. Our algorithm operates in six steps: First, we build random forest models consisting of a number of decision trees at each site. Second, we share these local models with every other site and calculate the performance statistics (true/false positive/negative values) of all decision trees at each site. Then, we compute global statistics by aggregating local performance statistics, calculate the weight for each decision tree by utilizing the Matthews correlation coefficient (MCC), and finally generate the final federated ensemble classifier.

The primary contributions of our work can be listed as follows:

1) We propose a novel Boosting-based Federated Random Forest algorithm that combines both bagging and boosting ensemble techniques and adapts them to horizontal federated learning to enable different sites to perform joint machine learning operations without sharing any real data between them or with a third party, hence preserving privacy. Specifically, we consider the decision trees in each local random forest model as weak classifiers and calculate their weights in a federated manner.
2) We introduce a novel aggregation and weight calculation methodology that enables participating sites to generate a global federated model that can improve prediction capability of all sites, regardless of whether they had a successful or unsuccessful local model.
3) We evaluate the performance of our algorithm in several federated environments that we set up by using four healthcare datasets.
4) We show with our experiments that our proposed algorithm improves the prediction power of the baseline local random forest model in all cases and produces better results than similar approaches. In particular, the

percentage of improvement is significantly high for sites having unsuccessful local models because of their poor quality or imbalanced data.

The rest of this paper is organized as follows. In Section 2, we present the existing work on ensemble learning and its federated implementation. In Section 3, we propose our Boosting-based Federated Random Forest algorithm. Our experimental study and results are presented in Sections 4 and 5, respectively. Finally, we conclude our paper with a discussion in Section 6, and discuss future work in Section 7.

II. RELATED WORK

In this section, we present the concepts and work performed in the field of ensemble learning and federated learning utilizing ensemble techniques.

A. ENSEMBLE LEARNING

The idea of ensemble learning is to combine the predictions of multiple classifiers to produce a single classifier that is more accurate than any of the individual classifiers constituting the ensemble [16]. It has been shown that the probability of an ensemble classifier making an incorrect prediction is usually lower than that of a single classifier. Therefore, ensemble techniques have attracted the attention of many researchers. As a result, several ensemble methodologies have been developed, including bagging, boosting, arching, and stacking [17]. These methods have been applied in a variety of research fields, such as time-series forecasting, image segmentation, and classification [18], [19], [20], [21].

Bagging is an ensemble algorithm that takes a number of subsamples (with or without replacement) from the initial dataset, trains individual predictive models on those subsamples and obtains the final classifier by averaging the bootstrapped models, calculating weighted sums or majority voting [22]. Random forest (RF) is the best-known bagging algorithm that generates a number of decision trees by not only taking the random subset of data, but also using a random subset of features rather than all, to prevent strong predictors in the dataset from generating highly correlated models. It uses the majority voting approach while aggregating the results of the individual decision trees. In the literature, there are many extensions of random forests with different bootstrapping approaches focusing on aspects such as clustered or unbalanced data [23], [24], [25], [26].

Boosting is another popular ensemble method that is widely used in machine learning applications. In boosting, the aim is to create a strong classifier from a number of weak classifiers that are trained sequentially by using the information coming from the preceding ones [27]. AdaBoost is one of the most popular boosting algorithms owing to its high-performance and effective prediction capability [28], [29]. In AdaBoost, the weak classifiers are decision trees with only one node and two leaves, which is called a stump. In each iteration, a stump is generated, and weights are assigned to both the data points (instances) and weak classifiers based on the correctness of the predictions. Having trained a weak classifier, AdaBoost increases the weight of the misclassified instances and decreases the weight of the correctly classified instances so that subsequent classifiers can focus more on the samples on which the previous classifier made an error. The weights are calculated based on the error rate of the weak classifier. Gradient-boosted trees (GBT) [30] and XGBoost [31] are examples of other widely used boosting techniques.

B. FEDERATED LEARNING

Federated learning (FL) enables multiple sites to build a joint machine learning model by exchanging only information about locally trained models, rather than actual data [6]. In FL, each site first builds a local model using their respective training data. The local models are then either transferred to a third-party coordinator, who is responsible for aggregating local models to build a global model, or exchanged between the sites in a peer-to-peer manner without the involvement of a third-party coordinator. After McMahan *et al.* [8] developed the first federated learning algorithm, namely FedAvg, in 2016, researchers have made significant efforts to advance existing studies on topics such as preventing privacy leakage, reducing communication cost, decreasing computation cost, and increasing model performance [9], [10], [11], [12], [13], in various fields of machine learning such as classification, regression, association rule learning, and deep learning [32], [33], [34]. In this study, we focus on the application of ensemble strategies to improve the classification performance in horizontal federated learning. Recently, researchers have performed several studies on implementing the random forest in a federated manner. Liu *et al.* [35] implemented a federated forest algorithm in which participants calculate the impurity improvement values for each feature locally, and the central coordinator selects the feature that gives the best split so that a joint random forest can be built. Ge *et al.* [36] improved the solution proposed by Liu *et al.* by optimizing the feature selection and pruning steps to create models with more accurate results. However, both approaches focus on vertically partitioned data, in which participants have different sets of features for the same sample space.

Studies on the application of federated learning for horizontally partitioned data mainly utilize the boosting technique, particularly AdaBoost. Gambs *et al.* [37] proposed MultBoost algorithm as a distributed implementation of AdaBoost, where the dataset is split between two or more participants, and the model is built in a privacy-preserving manner. The idea of MultBoost is to merge weak classifiers trained by participants in each iteration and compute the weights based on data instances misclassified by the merged weak classifier. However, one of the shortcomings of this algorithm is that it requires intense communication between the participants and a central coordinator, which creates a crucial performance issue in a federated environment. To reduce the communication cost and make computations in participants independent of each other, Palit *et al.* [14] introduced

the AdaBoost.PL algrotihm. In AdaBoost.PL, participants (or workers as the authors named) compute the local ensemble classifier by completing all the iterations of AdaBoost. Then, the workers sort the weak classifiers in increasing order with respect to their weights and merges the ''like-minded classifiers'' which are the ones located at the same index of sorted classifier list at each participant.

Li *et al.* [15] implemented the BOPPID (Boosting-based privacy-preserving integration of distributed data) algorithm by following a different methodology while combining the ensemble classifiers generated by the workers. In BOPPID, local classifiers are shared between participants, instead of being transferred to a central coordinator. Unlike AdaBoost.PL, which merges the weak classifiers of local ensemble classifiers, BOPPID updates their weights based on the following three criteria: (1) The local model of the participant performing the combination process is always assigned more weight compared to other local models with the same sample size. (2) The weights of participants' local model are proportional to their sample size such that the more data the participant has, the more weight its local model is assigned. (3) The weight of the local model of a participant performing the combining process has an upper bound; otherwise, the other models become insignificant.

In our work, we followed a different ensemble strategy than BOPPID where the weak classifiers were the decision trees in the local random forest models generated at each site. Moreover, we used a different weight calculation methodology when merging the weak classifiers. In the experiments, we compared the results of our proposed algorithm with only BOPPID, as Li *et al.* [15] showed that BOPPID performs better than its competitors, namely MultBoost and AdaBoost.PL.

III. METHODOLOGY

In this section, we first describe our Boosting-based Federated Random Forest (BOFRF) algorithm and give its mathematical formulation. Second, we analyze the computational complexity of it. Third, we study the privacy issues and present two different implementation of the proposed algorithm to increase the level of privacy. Description of the notations used in the paper while explaining the details of algorithm is shown in Table 1.

A. ALGORITHM

The federated environment consists of several sites, each of which contains datasets with both common and local features. Given N sites, the dataset of the nth site containing m_n instances can be represented as

$$D_n = \{(X_1, y_1), (X_2, y_2), \ldots, (X_{m_n}, y_{m_n})\} \tag{1}$$

where X_i is the vector of the feature values and $y_i \in \{+1, -1\}$ is the label used for binary classification. The datasets at each site are split into two sets: training set S_n and test set T_n, where $S_n \cup T_n = D_n$.

TABLE 1. Notations used in the paper while explaining the details of algorithm.

Notation	Description
N	Number of sites
D_n	Dataset of the nth site
m_n	Number of instances at the nth site
X	Vector of feature values
y	Label values $\in \{+1, -1\}$
S_n	Training set of the nth site
T_n	Test set of the nth site
R_n	The random forest classifier trained at the nth site
k_n	Number of decision trees at the nth site
d_i^n	The decision tree at the ith index of the nth site
α_i^n	The weight of decision tree at the ith index of the nth site
$c_i^n(q)$	The confusion matrix calculated by the qth site for decision tree at the ith index of the nth site
C_i^n	The final confusion matrix for decision tree at the ith index of the nth site
tp	True positive, i.e., the number of positive examples predicted positive
tn	True negative, i.e., the number of negative examples predicted negative
fp	False positive, i.e., the number of negative examples predicted positive
fn	False negative, i.e., the number of positive examples predicted negative
$M(C_i^n)$	The Matthews correlation coefficient value of confusion matrix C_i^n
τ	The threshold value
F_n	The local ensemble classifier at the nth site
F	The final global ensemble classifier

In our algorithm, we first train a random forest model on the training set, S, of each site. A random forest model comprises several decision trees, each of which is built by using a random subset of features at each split within the tree algorithm. The random forest classifier having k_n decision trees trained at the nth site is represented as

$$R_n = \{d_1^n, d_2^n, \ldots, d_{k_n}^n\} \tag{2}$$

where d_i^n is the ith decision tree generated in the random forest. In the standard random forest algorithm, after the model is built, each decision tree makes a prediction on the test set T, and the label predicted by the majority of decision trees constitutes the final prediction. In our algorithm, however, we apply a boosting methodology in a federated manner to calculate the weight α_i^n for each decision tree[1] so that their combination will constitute the final ensemble classifier. For this purpose, we first run each decision tree d_i^n of classifier R_n on the training set S_n. Then, for each d_i^n we calculate the confusion matrix c_i^n which is a two-dimensional array of the number of true positive, true negative, false positive and false negative predictions, such that

$$c_i^n(n) = (tp, tn, fp, fn). \tag{3}$$

In the second step, all sites share their own decision trees, in other words weak classifiers, with every other site, hence the qth site retrieves $((\sum_{i=1}^{N} k_i) - k_q)$ number of decision

[1] In our proposed solution, decision trees are the weak classifiers of the final ensemble classifier. We use both terms interchangeably; however, they always refer to the same thing.

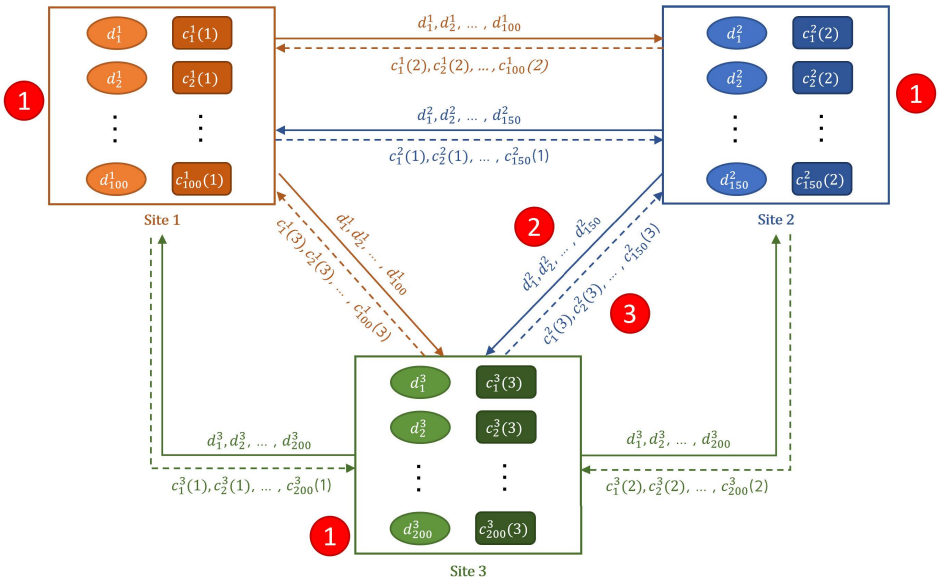

FIGURE 1. The first three steps of BOFRF in an example federated setup in which 3 sites participated. Ellipses and rounded rectangles represent decision trees and confusion matrices, respectively. Straight lines show items sent and dashed lines show items retrieved as a response. Circles indicate step numbers. Note that although all straight lines are part of step 2 and all dashed lines are part of step 3, these are shown in only one place for visual convenience.

trees. Each decision tree is then run on the site's training set S_q, and the corresponding confusion matrix is calculated. The confusion matrix calculated by the qth site for the decision tree d_i^n retrieved from the nth site can be represented as

$$c_i^n(q) = (tp, tn, fp, fn). \tag{4}$$

In the third step, each site sends the confusion matrices generated in the second step to the sites from which the corresponding decision tree was received. The final confusion matrix for a decision tree d_i^n is generated by adding up true positives with true positives, true negatives with true negatives, and so on, as in

$$C_i^n = \sum_{j=1}^{N} c_i^n(j). \tag{5}$$

Fig. 1 illustrates the first three steps of the algorithm in an example federated setup in which three sites participated. In the figure, each ellipse within a site corresponds to a decision tree, whereas the rounded rectangle next to it represents the corresponding confusion matrix. The arrows indicate the communication between sites where decision trees are sent, and confusion matrices are retrieved. The order of the steps is shown by circles with numbers.

Once the final confusion matrices are calculated, the next step is to assign weights to each decision tree. Most ensemble methods utilize the accuracy or error rate, ratio of misclassified instances, and total number of instances to calculate weights for weak classifiers. However, this is not always the best solution, especially when one of the sites has imbalanced data. Assume that a site contains 98% of the data labeled as positive and 2% of the data labeled as negative. If we generate a learning model that always classifies all instances as positive, we obtain an accuracy of 98%. This model appears to be one of the best models that can be generated; however, it is useless. This problem is known as the "accuracy paradox" in the literature [38]. Instead, we can utilize other performance metrics such as area under curve (AUC), precision, recall, and F-score, depending on the type of problem we are trying to solve. Among them, the AUC is the most widely used performance metric, which can produce good results for both balanced and imbalanced datasets. However, the success of the AUC decreases with an increase in the imbalance or skewness in the dataset. In such cases, precision and recall would evaluate the model's performance better [39]. Precision is a useful metric for cases in which the number of false positives should be minimized, whereas recall, which is the accuracy on positive label, aims to minimize the number of false negatives. In order to benefit from both precision and recall at the same time, the F1-score was introduced as the harmonic mean of the two metrics. Although it is an effective metric for imbalanced datasets, it cannot be used as a generic weight calculation metric in a federated environment either, because it does not consider true negative predictions; hence, it is not able to detect true negative rates. In our work, we first used accuracy and F1-score to calculate the weights of each decision tree, but we observed that neither the federated model generated with accuracy nor the one generated with the F1-score always outperformed the local models. When we looked deeper into the results, we noticed that when a decision tree that predicts negative values well is combined with a decision tree that predicts positive values

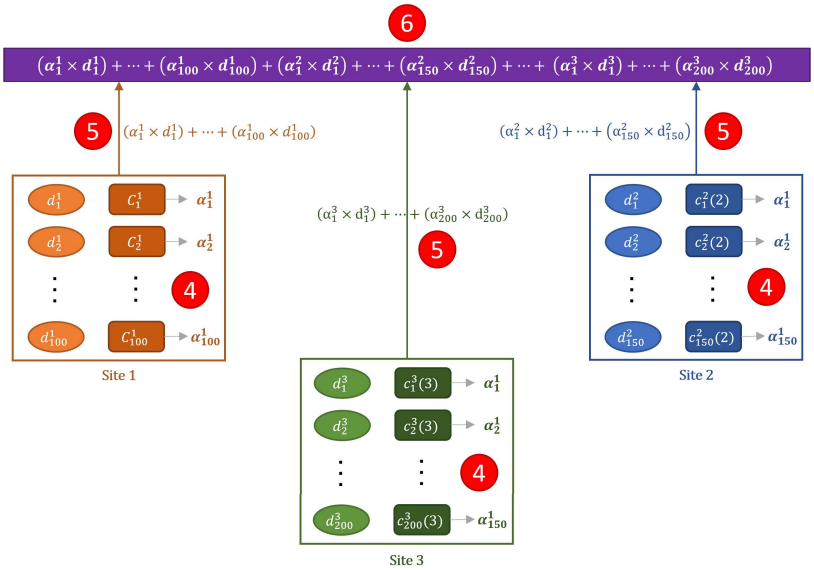

FIGURE 2. The weight calculation and aggregation steps of BOFRF in an example federated setup in which 3 sites participated. The rectangle on top represents the federated model generated as a combination of weighted decision trees coming from local RF models.

well, one degrades the performance of the other because of the aforementioned reasons.

The Matthews correlation coefficient (MCC), or phi coefficient, is a statistical measure used to discover the association between two binary variables [40]. It was adapted to the machine learning domain by Baldi *et al.* [41] in 2000 as a performance metric to show the correlation between predictions and real values. MCC considers all values in the confusion matrix; hence, it is not affected by imbalanced or skewed datasets [42]. It has been shown that it is more robust and reliable than the aforecited performance metrics when evaluating the classifier performance in both balanced and imbalanced cases [43]. Therefore, in our proposed methodology, we utilized the MCC to calculate the classifier weights. The MCC is calculated as

$$MCC = \frac{(tp \times tn) - (fp \times fn)}{\sqrt{(tp + fp) \times (tp + fn) \times (tn + fp) \times (tn + fn)}}. \quad (6)$$

The values of MCC range between $[-1, +1]$, where $+1$ indicates perfect classifier, 0 indicates random guessing classifier, while -1 indicates completely opposite classifier which predicts everything wrongly. Therefore, we are only interested in classifiers having positive MCC value. If all instances in the dataset belongs to only one label, either positive or negative, or if the classifier predicts everything as positive or negative, both the nominator and denominator in MCC become zero, which is undefined. In such cases, we can directly set the MCC to zero and ignore the classifier. Because the randomness of the classifier increases as the MCC converges to zero, we observed that removing weak classifiers having an MCC value below a certain threshold τ increases the performance of the final ensemble classifier. In our experiments, we obtained the optimal results with threshold $\tau = 0.2$. Based on these, the weight α_i^n of decision tree d_i^n having a final confusion matrix C_i^n is calculated as

$$\alpha_i^n = \begin{cases} M(C_i^n) & if M(C_i^n) > \tau \\ 0 & otherwise \end{cases} \quad (7)$$

where $M(C_i^n)$ is the MCC value calculated from confusion matrix C_i^n. After the weights of decision trees are calculated, the local ensemble classifier F_n at the nth site is generated as

$$F_n = (\alpha_1^n \times d_1^n) + (\alpha_2^n \times d_2^n) + \ldots + (\alpha_{k_n}^n \times d_{k_n}^n). \quad (8)$$

Finally, the linear combination of local ensemble classifiers, that is, all decision trees with their corresponding weights, constitutes the final global ensemble classifier $F = F_1 + F_2 + \ldots + F_n$, which can also be represented as

$$F = (\alpha_1^1 \times d_1^1) + \ldots + (\alpha_{k_1}^1 \times d_{k_1}^1) + (\alpha_1^2 \times d_1^2) + \ldots + (\alpha_{k_2}^2 \times d_{k_2}^2) + (\alpha_1^n \times d_1^n) + \ldots + (\alpha_{k_n}^n \times d_{k_n}^n). \quad (9)$$

The final steps of the algorithm are illustrated in Fig. 2. In general, F can be formulated as

$$F = \sum_{n=1}^{N} \sum_{i=1}^{k_n} \alpha_i^n \times d_i^n. \quad (10)$$

The pseudocode for our proposed Boosting-based Federated Random Forest (BOFRF) algorithm is shown in Algorithm 1. First, each site trains a standard random forest model on its training set and calculates the values of the confusion matrix (lines 1-8). The output random forest models, which contain a number of decision trees, are then sent to every other site (line 9). In lines 11-20, each site calculates the statistics for all decision trees of other sites on its own training data and sends back the results. The confusion matrices are

Algorithm 1 Boosting-Based Federated Random Forest (BOFRF)

Input: Dataset of N sites $\{D_1, D_2, \ldots, D_N\}$ where $D_i = \{(X_1, y_1), (X_2, y_2), \ldots, (X_{m_i}, y_{m_i})\}$.
Output: The final global ensemble classifier F
Procedure:

1: **for** $n \leftarrow 1$ to N **do in parallel** ▷*Step 1: Train a Random Forest model at each site*
2: $S_n \leftarrow$ Training set after D_n is split
3: $T_n \leftarrow$ Test set after D_n is split
4: $R_n \leftarrow RANDOM\_FOREST(S_n)$
5: **for** $i \leftarrow 1$ to k_n **do**
6: $y_{pred} \leftarrow PREDICT(d_i^n, S_n)$
7: $c_i^n(n) \leftarrow CONFUSION\_MATRIX(y_{pred}, S_n)$ ▷*Eq.* (3)
8: **end for**
9: Send R_n to every other site ▷*Step 2: Share decision trees with every other site*
10: **end for**
11: **for** $q \leftarrow 1$ to N **do in parallel** ▷*Step 3: Run each decision tree retrieved from other sites*
12: Create an array containing index of every other site $N' \leftarrow$ [1 to N except for q]
13: **for** n in N' **do**
14: **for** $i \leftarrow 1$ to k_n **do**
15: $y_{pred} \leftarrow PREDICT(d_i^n, S_q)$
16: $c_i^n(q) \leftarrow CONFUSION\_MATRIX(y_{pred}, S_q)$ ▷*Eq.* (4)
17: **end for**
18: **end for**
19: Send all $c_i^n(q)$ for d_i^n to the nth site which is the owner ▷*Step 3: Send confusion matrices*
20: **end for**
21: **for** $n \leftarrow 1$ to N **do in parallel**
22: **for** $i \leftarrow 1$ to k_n **do** ▷*Step 4: Merge confusion matrices and calculate weight*
23: $C_i^n \leftarrow \sum_{j=1}^{N} c_i^n(j)$ ▷*Eq.* (5)
24: $w \leftarrow MCC(C_i^n)$ ▷*Eq.* (6)
25: **if** $(w > \tau)$ **then** ▷*Eq.* (7)
26: $\alpha_i^n \leftarrow w$
27: **else**
28: $\alpha_i^n \leftarrow 0$
29: **end if**
30: **end for**
31: $F_n \leftarrow \sum_{i=1}^{k_n} \alpha_i^n \times d_i^n$ ▷*Step 5: Build the local ensemble model* (8)
32: **end for**
33: **return** $F = \sum_{n=1}^{N} \sum_{i=1}^{k_n} \alpha_i^n \times d_i^n$ ▷*Step 6: Build the global federated ensemble model* (9), (10)

then merged in line 23, and the corresponding weight is calculated in lines 24-29. In line 33, the weighted decision trees are combined, and the final ensemble classifier is built.

As a result, using a federated ensemble classifier F, the ensemble prediction becomes the sign of F for a given vector of the feature values X. An important point that should be highlighted here is that unlike traditional weak classifiers, which return either $+1$ or -1, the weak classifiers in BOFRF might also return 0 as a way of "abstaining" from answering [44]. In BOFRF, a weak classifier (decision tree) predicts 0 if any of its nodes contains a feature which does not appear in the feature space of the site where the prediction was made (line 6 and line 15). In such cases, related sites do not contribute to updating the confusion matrix of the corresponding weak classifier.

Last, but not least, after the federated ensemble classifier F is generated, each site makes predictions on its test set T_n by using both F and F_n, and compares the results to check which one is better. If F is better than F_n, the nth site uses F for future predictions; otherwise, it uses F_n. Thus, it is always ensured that the final model performs at least as well as the local model for each site.

B. COMPUTATIONAL COMPLEXITY

In this section, we analyze the computational complexity of BOFRF by going through each step of the algorithm. The first step of BOFRF is to build a standard random forest model containing k_n decision trees. The cost of building a decision tree at the nth site is $O(f_n m_n \log(m_n))$, where f_n is the number of features in the dataset (i.e., the length of X_n),

m_n is the number of instances, and $\log(m_n)$ is the depth of the tree in the worst-case scenario. Thus, the complexity of Random Forest becomes $O(k_n f_n m_n \log(m_n))$. It requires one pass through the training set for each decision tree to make a prediction and generate the confusion matrix; hence, the cost is $O(k_n m_n)$. As a result, the computational complexity of the first step (lines 1-10) is $O(k_n f_n m_n \log(m_n) + k_n m_n)$, which can be asymptotically rewritten as $O(k_n f_n m_n \log(m_n))$.

In the second step (lines 11-20), decision trees are shared between sites and confusion matrices are generated. From a complexity point of view, the only difference between steps 1 and 2 is the increase in the number of decision trees; hence, the cost is $O(Km_n)$, where K is the number of trees retrieved from other sites.

In the last step (lines 21-32), for each decision tree, we iterate through the N number of confusion matrices to calculate weight. Hence, the computational complexity is $O(k_n N)$.

Since all the three steps are performed in parallel at different sites, the complexity depends on the maximum number of decision trees, features, and instances among all the participating sites. Consequently, the computational complexity of BOFRF becomes $O(k'f'm' \log(m') + K'm' + k'N)$, where k',f',m',and K' represent the maximum values. Since $K' \leq k' * N$, because $k' * N$ also represents the total number of decision trees at all sites, the computational complexity of BOFRF becomes $O(k'f'm' \log(m') + k'm'N)$, which can be simplified to $O(k'm'(f' \log(m') + N))$.

C. PRIVACY ISSUES

The proposed BOFRF algorithm provides an adequate level of privacy in its current form as the actual data is not shared among the participants. However, sharing evaluation metrics in the presence of a sneaky site in the federated environment may still result in a privacy breach. For instance, if the sneaky site knows the characteristics of a particular patient, it can generate two models, one of which labels the individual as Class 0, and the other as Class 1. Then, by comparing the confusion matrices from other sites to find exactly 1 difference, the sneaky site can find out which site the patient belongs to. To prevent this from happening and increase the level of privacy, we propose two different implementations for our proposed BOFRF algorithm in this section: centralized implementation with a trusted third party and decentralized implementation using secure sum protocol.

In centralized implementation with a trusted third party, participating sites send their decision trees and confusion matrices to an orchestrator who is responsible for sending the decision trees to the other sites, retrieving output confusion matrices, and calculating the final confusion matrix for each decision tree without knowing anything about the information provided by the sites. The orchestrator then sends the final confusion matrix of each decision tree to the site that owns the respective tree. In this way, the participating sites never see the confusion matrices of other sites; hence, privacy is protected against certain types of attacks like patient-site assignment attack.

In decentralized implementation, instead of communicating with a third party, participating sites communicate with each other in a circular way. This approach uses the secure sum protocol, which allows participating sites to calculate the sum of their individual data without exposing their data to other sites [45]. In this protocol, one of the sites first adds a random noise to its own data and sends it to the next site. The site adds this number to its own data and sends it to the next site. This process is repeated until the site that started the protocol retrieves the sum. The site then subtracts the noise from the sum and finds out the actual sum. In decentralized implementation of BOFRF, each site executes the secure sum protocol for each confusion matrix. In the first round of this process, all the sites add random noise to their local confusion matrices and send them to the next site along with their local decision trees. The recipient site runs the retrieved decision trees on its own data, calculates the confusion matrices, adds these values to the received confusion matrices, and sends the decision trees and confusion matrices to the next site. The process continues until the circular cycle is completed. All sites then subtract the noise they added to confusion matrices at the first step and obtain the final confusion matrices. In this implementation, the participating sites see some values in confusion matrices, but they never know the actual values.

Although both implementations increase the level of privacy, it should be noted that there is a trade-off between the level of privacy and complexity. Centralized implementation increases the communication cost as all information exchange is done through a trusted third party, whereas decentralized implementation increases the time cost as execution is done sequentially rather than parallel.

IV. EXPERIMENTS

In this section, we describe in detail the datasets we considered and the environments we set up in our experiments to evaluate the effectiveness of the proposed BOFRF algorithm.

A. DATASETS

In our experiments, we used four healthcare datasets. The Pima Indians Diabetes dataset from Kaggle [46], [47] contains the data of 768 patients who were studied by the National Institute of Diabetes and Digestive and Kidney Diseases in Phoenix, Arizona. It has 8 independent features: age, body-mass index, number of pregnancies, plasma glucose, diastolic blood pressure, triceps skin fold thickness, insulin, diabetes pedigree function; and 1 dependent feature in which 1 is interpreted as ''diabetes positive''. We used this dataset to build machine-learning models to predict whether a patient in the dataset had diabetes.

The Diabetic Retinopathy dataset from the UCI Machine Learning Repository [48], [49] contains data extracted from 1151 eye fundus images to predict whether an image contains the sign of diabetic retinopathy, which is a visual impairment caused by diabetes mellitus. It consists of 19 independent features related to a lesion in the eye, anatomical

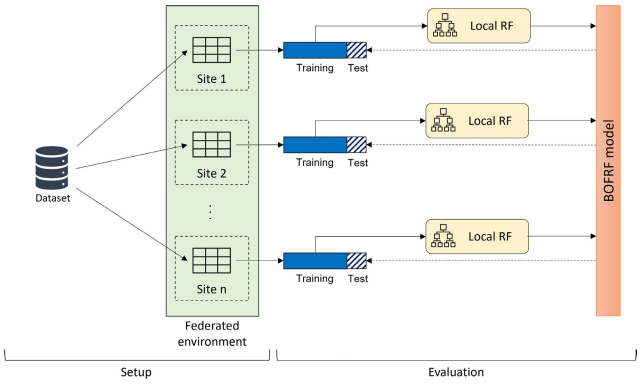

FIGURE 3. The setup and evaluation procedure of the experiments. In the setup, the dataset is split into *n* sites constituting the federated environment. In the evaluation, data at each site are split as training and test, and local Random Forest models are generated. After federated BOFRF model is built on top of the local models, it is evaluated on test data of each site.

information, or an image-level descriptor; and one dependent feature, where 1 indicates a sign of diabetic retinopathy.

The South African Heart Disease dataset contains data of 462 men living in the heart-disease high-risk region of Western Cape, South Africa. Researchers took a subset of it consisting of 9 features, which are systolic blood pressure, LDL cholesterol, family history of heart disease, adiposity, obesity, type-A behavior, tobacco and alcohol usage, and age, for predicting whether a patient will suffer from coronary heart disease or not. In this study, we used the dataset published by Bartley in Harvard Dataverse [50], [51].

The SHELTER study conducted research between 2009 and 2011 to assess the care needs and provision of care to nursing home (NH) residents in Europe [52] and collected the data of 4156 NH residents in eight countries (Czechia, England, Finland, France, Germany, Italy, The Netherlands, and Israel). Onder *et al.* [53] analyzed the determinants of excessive polypharmacy (usage of more than 10 drugs) in the SHELTER dataset and identified the factors that are associated with excessive polypharmacy. We utilized this information and extracted 14 features from the SHELTER dataset; and used this dataset to predict excessive polypharmacy risk of NH residents.

B. ENVIRONMENT SETUP

In our study, we conducted two types of experiments. In the first type, which we call observational experiments, we used the Pima Indians Diabetes, Diabetic Retinopathy, and South African Heart Disease datasets to set up different federated environments to prove the effectiveness of BOFRF in challenging scenarios of federated learning. In this regard, we set up federated environments consisting of different numbers of sites (i.e., 2, 3, or 4) by splitting the datasets into several datasets with different characteristics, i.e., datasets producing well-performing local models or datasets producing unsuccessful local models. The setup and evaluation procedures of these experiments are illustrated in Fig. 3.

TABLE 2. Number of sites, total number of records, and the number of positive and negative labelled instances in the federated environments built in the experiments.

Env. #	Dataset	Site	Records	Positive labels	Negative labels
Env. 1	Diabetes	Site 1	400	152	248
		Site 2	368	116	252
Env. 2	Diabetes	Site 1	300	114	186
		Site 2	200	68	132
		Site 3	268	86	182
Env. 3	Diabetes	Site 1	300	107	193
		Site 2	200	73	127
		Site 3	268	88	180
Env. 4	Diabetes	Site 1	100	33	67
		Site 2	250	94	156
		Site 3	280	57	223
		Site 4	138	84	54
Env. 5	Diabetes	Site 1	200	64	136
		Site 2	200	74	126
		Site 3	200	68	132
		Site 4	168	62	106
Env. 6	Diabetes	Site 1	100	37	63
		Site 2	250	97	153
		Site 3	300	89	211
		Site 4	118	45	73
Env. 7	Diabetic Retinopathy	Site 1	651	340	311
		Site 2	500	271	229
Env. 8	Diabetic Retinopathy	Site 1	400	218	182
		Site 2	300	158	142
		Site 3	451	235	216
Env. 9	Diabetic Retinopathy	Site 1	651	340	311
		Site 2	300	157	143
		Site 3	200	114	86
Env. 10	Heart Disease	Site 1	90	36	54
		Site 2	372	124	248
Env. 11	Heart Disease	Site 1	200	80	120
		Site 2	150	55	95
		Site 3	112	25	87
Env. 12	Heart Disease	Site 1	150	57	93
		Site 2	150	53	97
		Site 3	162	50	112
Environments on SHELTER		Czechia	500	170	330
		Germany	490	173	317
		England	507	179	328
		Finland	448	299	149
		France	490	209	281
		Israel	579	131	448
		Italy	540	89	451
		Netherlands	548	157	391

In the second type of experiments, which were conducted on the SHELTER dataset, we did not follow the same setup procedure because the dataset was already split according to origin countries. Instead, we built a total of 247 federated environments consisting of two, three, four, five, six, seven, and eight sites, with all possible combinations of the eight countries involved in the dataset. In these experiments, we performed statistical testing of BOFRF and observed how well it performs in real federated environments that we did not emulate, as well as how effective it is on larger datasets and increasing number of participants. Table 2 shows the details of the environments, including the total number of records at each site and the number of positive and negative labelled instances.

C. EVALUATION PROCEDURE

In each experiment, we first compared the results of our proposed BOFRF algorithm with the baseline, which is the local random forest algorithm of each participating site, to show how BOFRF can improve the predictive power of the baseline local model. While building the local random forest models, we applied 10-fold cross-validation and used the grid search method to avoid overfitting and determine the best model with the best hyperparameter combination. To ensure privacy of sensitive data and enable the calculation of the AUC, we put some restrictions on the hyperparameter values that are explored by the search grid, such as the minimum leaf size. For instance, a suspicious site may attempt to identify a subject through a decision tree that has a leaf node with only one subject. Therefore, a value of 1 for the minimum leaf size was not allowed so that none of the sites could generate such decision trees. This also makes the calculation of prediction probabilities possible for each decision tree since no leaf node contains a single class estimate. The prediction probabilities of decision trees are required to calculate the AUC values of the random forest models and the BOFRF model. Then, to evaluate the success of our proposed algorithm against existing solutions, we compared the results of BOFRF with those of the federated model generated by one of the most successful existing solutions in the field, namely BOPPID. For a fair comparison, we implemented the algorithm proposed by Li *et al.* in their paper [15] in Python and ran both BOPPID and local AdaBoost, which is the baseline of BOPPID in each experiment.

V. RESULTS

A. RESULTS OF OBSERVATIONAL EXPERIMENTS

1) COMPARISON WITH LOCAL RANDOM FOREST

We utilized the Pima Indian Diabetes dataset to set up six different environments with two different characteristics: (i) all sites had quite good models with relatively close AUC values (Environments 1 and 3), and (ii) one site had a very good model with a high AUC value, and at least one site had a poor model with a low AUC value (Environments 2, 4, 5 and 6). The results are presented in Table 3.

In all cases, BOFRF successfully improved the performance of all local RF models. In the former, the percentage of improvements was usually around the same level at all sites, that is between 1-6%. In the latter, we observed a significant improvement in sites having poor models with low AUC values. BOFRF increased the local AUC value from 0.752 to 0.856 in Environment 2 (13.86% increase), from 0.734 to 0.882 in Environment 4 (20.17% increase), from 0.735 to 0.833 and from 0.714 to 0.849 in Environment 5 (11.7% and 15.86% increase, respectively), and from 0.686 to 0.904 in Environment 6 (24.29% increase). In these settings, the improvement on the best-performing model was usually around 0-2%, but this is something expected because these models can only be improved up to a certain level as they already have high AUC values. Fig. 4 shows the comparison of the local RF with BOFRF on the Pima Indian Diabetes dataset.

On the Diabetic Retinopathy dataset, we set up three environments, i.e., Environments 7, 8, and 9 as shown in Table 4. In Environment 8, where the participating sites had AUC values close to each other, we observed the same level of improvement as in the Pima Indian Diabetes dataset. In Environment 7, there were two sites, Site 1 and Site 2, with AUC values of 0.666 and 0.742, respectively. BOFRF improved

TABLE 3. Comparison of algorithms based on the AUC values in federated environments built on the Pima Indian Diabetes dataset. For better visualization, environments are abbreviated as E1, E2, etc., whereas sites are abbreviated as S1, S2, etc. The (*) near the values in the "Change" columns highlights the improvements that BOFRF provided significantly high.

Env. #	Site #	Local Ada	BOPPID	Change (%)	Local RF	BOFRF	Change (%)
E1	S1	0.792	0.811	+2.45	0.763	0.841	+10.29 (*)
	S2	0.805	0.816	+1.41	0.828	0.843	+1.76
E2	S1	0.776	0.809	+4.37	0.738	0.802	+8.59
	S2	0.772	0.823	+6.52	0.752	0.856	+13.86 (*)
	S3	0.883	0.906	+2.65	0.903	0.923	+2.24
E3	S1	0.832	0.834	+0.90	0.805	0.840	+4.29
	S2	0.857	0.859	+0.23	0.841	0.865	+2.83
	S3	0.868	0.845	-2.70	0.777	0.803	+2.81
E4	S1	0.840	0.841	+0.10	0.784	0.861	+9.74
	S2	0.769	0.819	+6.46	0.763	0.832	+9.03
	S3	0.757	0.843	+11.37	0.734	0.882	+20.17 (*)
	S4	0.896	0.911	+1.64	0.926	0.937	+1.18
E5	S1	0.839	0.885	+5.48	0.791	0.903	+12.31
	S2	0.819	0.833	+1.57	0.735	0.833	+11.7 (*)
	S3	0.895	0.885	-1.12	0.881	0.888	+0.76
	S4	0.656	0.745	+13.55	0.714	0.849	+15.86 (*)
E6	S1	0.718	0.818	+13.39	0.686	0.904	+24.29 (*)
	S2	0.812	0.816	+0.53	0.757	0.801	+5.47
	S3	0.927	0.936	+0.85	0.945	0.950	+0.46
	S4	0.755	0.750	-0.66	0.750	0.780	+3.85

TABLE 4. Comparison of algorithms based on the AUC values in federated environments built on the Diabetic Retinopathy and South African Heart Disease datasets. For better visualization, environments are abbreviated as E1, E2, etc., whereas sites are abbreviated as S1, S2, etc. The (*) near the values in the "Change" columns highlights the improvements that BOFRF provided significantly high.

Env. #	Site #	Local Ada	BOPPID	Change (%)	Local RF	BOFRF	Change (%)
E7	S1	0.692	0.718	+3.76	0.742	0.757	+2.03
	S2	0.693	0.695	+0.32	0.666	0.732	+9.80 (*)
E8	S1	0.710	0.720	+1.49	0.708	0.727	+2.69
	S2	0.703	0.768	+9.18	0.774	0.788	+1.85
	S3	0.684	0.739	+2.98	0.704	0.740	+5.12
E9	S1	0.692	0.705	+1.80	0.742	0.745	+0.31
	S2	0.729	0.747	+2.41	0.658	0.793	+20.44 (*)
	S3	0.707	0.823	+16.51	0.638	0.833	+30.72 (*)
E10	S1	0.509	0.607	+19.30	0.580	0.759	+30.77 (*)
	S2	0.708	0.708	0.00	0.758	0.793	+4.71
E11	S1	0.613	0.653	+6.52	0.618	0.709	+14.70 (*)
	S2	0.726	0.751	+3.39	0.771	0.773	+0.19
	S3	0.582	0.669	+14.92	0.573	0.696	+21.21 (*)
E12	S1	0.728	0.683	-6.17	0.692	0.718	+3.07
	S2	0.597	0.679	+13.81	0.599	0.685	+14.22 (*)
	S3	0.629	0.687	+9.13	0.724	0.741	+2.38

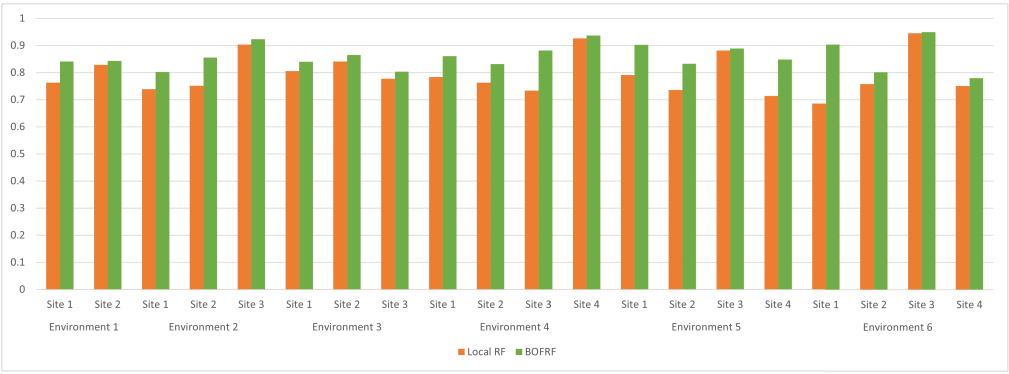

FIGURE 4. Comparison of the local RF with BOFRF in the Pima Indian Diabetes dataset. Orange bars represent the performance of local RF, whereas green bars represent the performance of BOFRF.

FIGURE 5. Comparison of the local RF and BOFRF in the Diabetic Retinopathy and South African Heart Disease datasets. Orange bars represent the performance of local RF, whereas green bars represent the performance of BOFRF.

the AUC values by 9.8% and 2.03%, respectively. In Environment 9, we kept Site 1 as it is and introduced two new sites having AUC values as low as Site 2 in Environment 7. In this case, BOFRF significantly improved the AUC values by 20.44% and 30.72%.

On the South African Heart Disease dataset, we introduced sites having the lowest AUC values in all experiments and evaluated the performance improvement. The results for Environments 10, 11, and 12 are shown in Table 4. In Environment 10, Site 1 had a local RF model with an AUC value of 0.580, and Site 2 had a local RF model with an AUC value of 0.758. After applying BOFRF, the AUC values increased by 30.77% and 4.71%, and became 0.759 and 0.793, respectively. In Environment 11, Site 3 had a very low AUC value of 0.573 due to its imbalanced data with only 22% positive. BOFRF provided 21.21% of improvement on this site and increased its AUC value to 0.696. Similarly, it increased the AUC of Site 1 in Environment 11 from 0.618 to 0.709 by 14.7%, and the AUC of Site 2 in Environment 12 from 0.599 to 0.685 by 14.22%. The overall comparison of local RF and BOFRF in the Diabetic Retinopathy and South African Heart Disease datasets is displayed in Fig. 5.

As a result of these experiments, we observed that for all sites, our proposed BOFRF algorithm successfully improved the prediction performance of the local RF. In particular, BOFRF significantly improved the prediction performance of sites whose local model was poor (i.e., having an AUC value less than 0.75). We highlighted such cases in Table 3 and Table 4 with "(*)" near the percentage of change values. For instance, in Environments 6, 9, and 10, we observed that BOFRF increased the local AUC value from 0.686 to 0.904 (24.29% increase), 0.638 to 0.833 (30.72% increase), and 0.580 to 0.759 (30.77% increase), respectively, with the help of other well-performing models. Furthermore, we also observed that even a site with a poor local model can provide a remarkable contribution to the other sites in BOFRF as presented in Environments 7 and 9.

2) COMPARISON WITH BOPPID

In our observational experiments, we observed that BOFRF provided better AUC results than BOPPID in 91.6% of the cases. In addition, the percentage of improvement provided by BOFRF was higher than that provided by BOPPID in 77.7% of the cases as shown in Fig. 6. The average change percentages of BOFRF and BOPPID were 9.04% and 4.67% respectively, which shows that BOFRF can improve the performance of the local models better than BOPPID.

FIGURE 6. Comparison of percentage of improvement provided by BOPPID and BOFRF on their baseline local models in observational experiments. Blue bars represent the percentage of improvement provided by BOPPID, whereas green bars represent the percentage of improvement provided by BOFRF.

TABLE 5. Achieved AUC values of countries along with their standard deviations in real federated environments built on the SHELTER dataset.

Site	Non-federated	Federated		
	Local RF (Baseline)	BOPPID	**BOFRF**	p-value
CZ	0.621 ± 0.039	0.630 ± 0.016	$\mathbf{0.659 \pm 0.026}$	0.000
DE	0.677 ± 0.039	0.645 ± 0.021	$\mathbf{0.688 \pm 0.005}$	0.000
EN	0.645 ± 0.023	0.681 ± 0.011	$\mathbf{0.692 \pm 0.012}$	0.001
FI	0.699 ± 0.032	0.687 ± 0.011	$\mathbf{0.715 \pm 0.019}$	0.000
FR	0.702 ± 0.007	0.685 ± 0.004	$\mathbf{0.714 \pm 0.005}$	0.000
IL	0.743 ± 0.026	0.751 ± 0.005	$\mathbf{0.777 \pm 0.003}$	0.000
IT	0.703 ± 0.046	0.687 ± 0.013	$\mathbf{0.705 \pm 0.015}$	0.000
NL	0.767 ± 0.015	$\mathbf{0.794 \pm 0.008}$	0.781 ± 0.003	1.000

TABLE 6. Achieved accuracy values of countries along with their standard deviations in real federated environments built on the SHELTER dataset.

Site	Non-federated	Federated		
	Local RF (Baseline)	BOPPID	**BOFRF**	p-value
CZ	0.680 ± 0.035	0.692 ± 0.010	$\mathbf{0.717 \pm 0.017}$	0.000
DE	0.658 ± 0.020	0.641 ± 0.014	$\mathbf{0.663 \pm 0.010}$	0.000
EN	0.661 ± 0.042	0.686 ± 0.013	$\mathbf{0.697 \pm 0.010}$	0.001
FI	0.623 ± 0.011	0.631 ± 0.048	$\mathbf{0.702 \pm 0.005}$	0.000
FR	0.609 ± 0.015	0.603 ± 0.013	$\mathbf{0.646 \pm 0.014}$	0.000
IL	0.831 ± 0.043	0.834 ± 0.005	$\mathbf{0.848 \pm 0.003}$	0.000
IT	0.849 ± 0.018	0.861 ± 0.005	$\mathbf{0.862 \pm 0.008}$	0.228
NL	0.744 ± 0.005	0.763 ± 0.016	$\mathbf{0.773 \pm 0.008}$	0.001

B. RESULTS OF STATISTICAL EXPERIMENTS

In the statistical experiments that we performed on the SHELTER dataset, we built 247 environments in total, consisting of every possible combination of countries in federated settings with two, three, four, five, six, seven, and eight sites. For example, Czechia (CZ) is involved in 7 settings with two sites (Czechia and another country), 21 settings with three sites (Czechia and two more countries), 35 settings with four sites, and so on. In each environment, we first run the local random forest model, which is the baseline of BOFRF, at each site (country) without using the federated learning approach. We then run the BOPPID and BOFRF algorithms in each of the 247 environments, and calculated the mean AUC and accuracy values for each country. In Tables 5 and 6, we compare these values along with their standard deviations and *p-value*s.

The statistical experiments confirmed our findings in observational experiments; that is, BOFRF improved the prediction power of the baseline local random forest model in all cases. In particular, in Table 5, the best improvement was observed in Czechia (CZ) and England (EN), which had the lowest AUC value in their baseline random forest models, that is, the two most unsuccessful local classifiers. On the other hand, BOPPID provided the best improvement on the site having the best performing local model, the Netherlands (NL), which is also the only case in which BOPPID produced better result than BOFRF. This is understandable because BOPPID is designed to give more importance to the site's

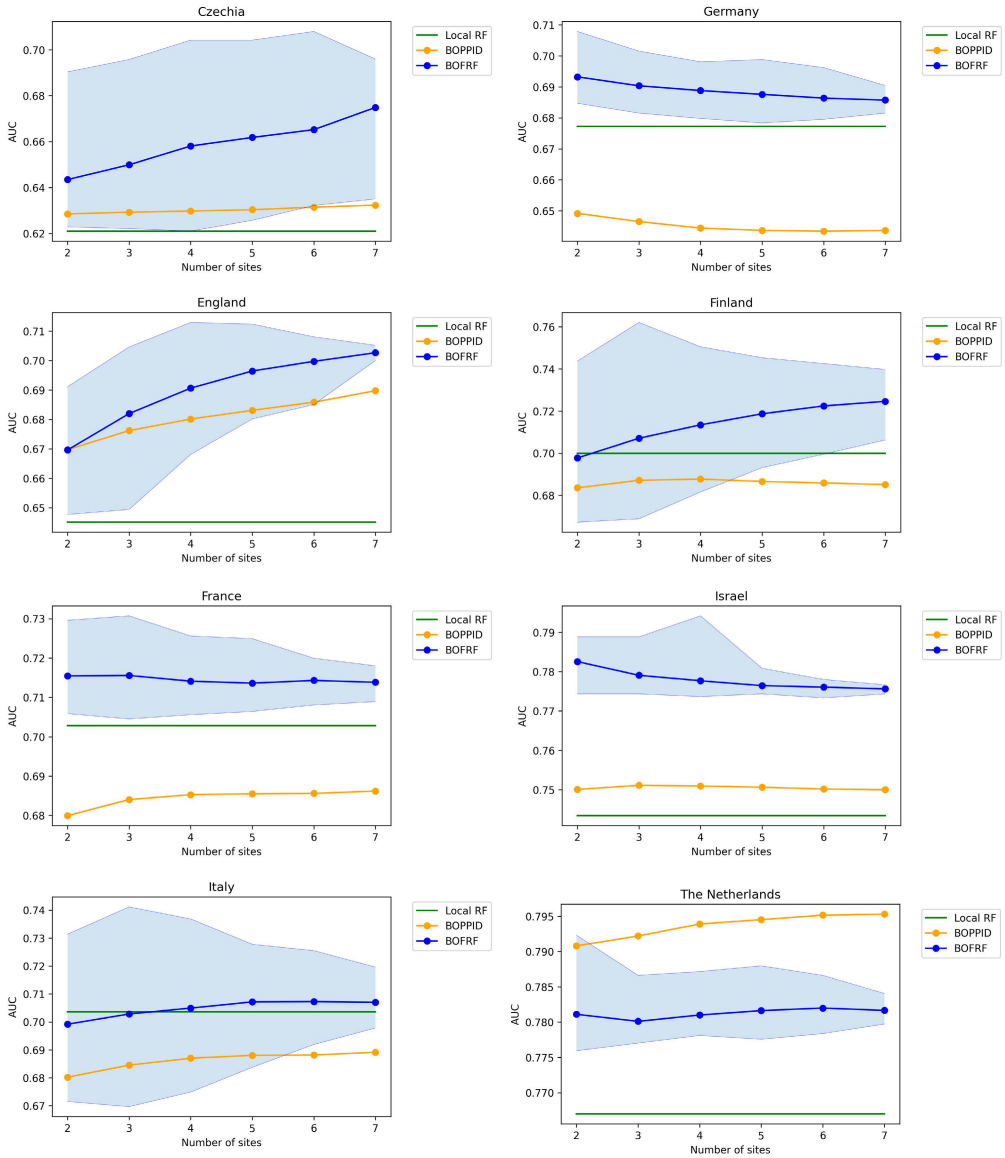

FIGURE 7. Overall AUC comparison of BOFRF with the baseline local RF model and BOPPID. The blue, orange, and green lines in each chart represent the BOFRF, BOPPID, and local RF, respectively. The blue-painted areas show the range that the AUC values of BOFRF change.

own local model than the others in the federated model. However, in all the other cases, BOFRF produced better results than BOPPID. To statistically verify that BOFRF outperforms BOPPID, right-tailed Z-test was applied. The alternative hypothesis H_1 was that the mean AUC value of BOFRF (μ) is greater than the mean AUC value of BOPPID (μ_0) at a given level of significance. The null hypothesis was just the opposite as formulated in (11).

$$H_0 : \mu \leq \mu_0$$
$$H_1 : \mu > \mu_0 \tag{11}$$

After the Z-tests were run, the corresponding *p-value* of each test were calculated based on the output *Z-score*s. The level of significance was determined as $\alpha = 0.05$. If the *p-value* ≤ 0.05, the null hypothesis should be rejected, otherwise it cannot be rejected. The null hypothesis can also be rejected if the *Z-score* ≥ 1.645, which is the critical value for the significance level of 0.05. As shown in Table 5, the *p-value* of all sites except the Netherlands were either 0.000 or 0.001, which means that the null hypothesis is rejected, hence the alternative hypothesis is true. The reason that *p-value*s of the sites had such low values was because their corresponding *Z-score*s had extremely high values, ranging from 7 to 52. The results obtained with the accuracy values were in line with those obtained with the AUC values. As shown in Table 6, BOFRF gave better accuracy results than BOPPID in all cases. The only *p-value* that was not below 0.05, hence not rejecting the null

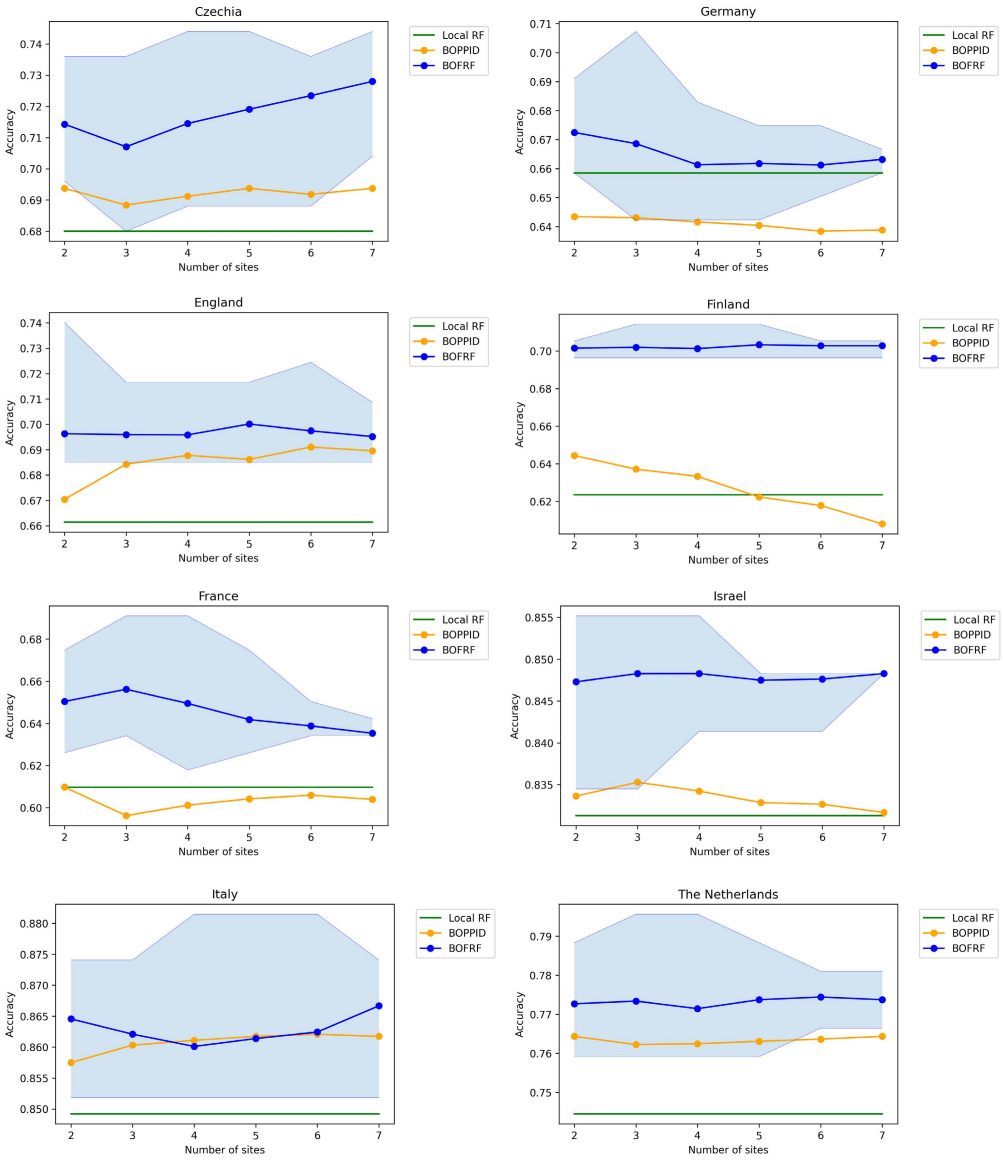

FIGURE 8. Overall accuracy comparison of BOFRF with the baseline local RF model and BOPPID. The blue, orange, and green lines in each chart represent the BOFRF, BOPPID, and local RF, respectively. The blue-painted areas show the range that the accuracy values of BOFRF change.

hypothesis, was that of Italy. Consequently, in seven out of eight sites in both AUC and accuracy, we had statistically significant evidence to confirm that BOFRF outperforms BOPPID.

Fig. 7 and Fig. 8 elaborate the comparison of AUC and accuracy values shown in Table 5 and Table 6 respectively, and provide line charts to depict the influence of the increasing number of sites on the performance of the algorithms. In the figures, the blue lines represent the mean values of BOFRF, whereas the orange lines represent the mean values of BOPPID. The AUC/accuracy value of the baseline local RF model is represented by a straight green line. To illustrate, in Fig. 7, the mean AUC values of BOFRF for Czechia in federated settings with two, three, four, five, six, and seven sites were calculated as 0.643, 0.649, 0.658, 0.661, 0.665, and 0.678, respectively. The blue-painted areas in the figure show the range in which the values of BOFRF change, that is, the area between the minimum and the maximum values, and the blue lines indicate the average values for each setting. In Czechia, England, Finland, Italy, and the Netherlands, the blue line followed an ascending path, implying that for an increasing number of sites, the performance of the BOFRF increased as well. In the case of France, increasing number of sites did not significantly affect the result; hence, the line followed a straight path, indicating that the same level of improvement was observed in all scenarios. A decrease

in performance for an increasing number of sites was only observed in Germany and Israel, but in both cases, the rate of decrease was limited and BOFRF performed better than local RF and BOPPID in all scenarios.

VI. DISCUSSION

The experiments showed that our proposed algorithm can successfully generate a powerful global model for all sites participating in the federated setting, regardless of whether they have well-performing or under-performing local models. In this section, we evaluate the results further and discuss the advantages and limitations of the proposed solution.

First, the empirical results show that BOFRF consistently improves the prediction performance of local RF model, hence achieves the main objective of federated learning because BOFRF considers the performance of each decision tree across all sites with equal emphasis. When compared to one of the best-performing existing solutions, namely BOPPID, the most important advantage of BOFRF is that it provides significantly high prediction improvement for the sites whose local model is not performing well, suggesting that the algorithm also attained the expected impact. The particularly important feature of BOFRF that enables it to achieve this is its novel MCC-based weight calculation methodology, which takes both target classes (0 and 1) into account when calculating the weights of each decision tree. In our experiments, we observed that the closer the AUC value of the local site is to 0.5, which is regarded as the failure limit, the more improvement BOFRF can provide given successful local models from other sites. This is important because in federated environments, enhancing the prediction capability of unsuccessful sites is much more important than the others in many cases. Thanks to BOFRF, these sites could benefit from the advantages of federated machine learning, that is, the ability to make accurate predictions despite the insufficient data they have in their repositories. Therefore, unless the usage of AdaBoost models is more convenient than RF models for some datasets (e.g., datasets producing high bias and low variance models), we believe that BOFRF would be a better choice than BOPPID because of the better prediction improvement it can provide for the sites, especially those with relatively unsuccessful local models.

In the literature, there are numerous solutions utilizing deep learning methods for federated learning [54], [55]. Although federated deep-learning approaches can produce better results than traditional federated learning methods in certain settings, they are computationally expensive, bring a communication overhead, and require a large amount of training data. Therefore, the implementation of federated solutions on traditional machine learning methods, including random forest, is still an important topic that has been widely studied by researchers. In this regard, we believe that the proposed solution is an essential contribution to state-of-the-art federated solutions.

Using decision trees as weak classifiers is a useful approach to follow in federated settings because decision trees have many advantages over other machine learning methods, such as the ability to handle categorical data and the ability to deal with outliers and noisy or missing data. Furthermore, although the participating sites mostly share the same set of features, they may still have some local features that are not present in others. This is not an issue in random forest because the standard random forest algorithm already uses the idea of taking a random subset of features to build each decision tree to prevent overfitting; hence, the local features are not present in every decision tree generated at sites. However, this is valid up to a point because the more local features differ from each other, the more vertical they become, which may prevent the BOFRF algorithm from producing successful results. Therefore, we note that our work is currently limited to horizontally partitioned data; hence, it may not be suitable for applications in vertical settings. Furthermore, in this study, we did not focus on improving the robustness of the proposed algorithm. Future enhancements could include the addition of a validation step, where the sites approve the retrieval of decision trees sent by other sites.

VII. CONCLUSION

In this paper, we propose a novel federated ensemble classification algorithm for horizontally partitioned data, namely Boosting-based Federated Random Forest (BOFRF), in which we adapt the idea of boosting to random forest in a federated manner and introduce a new aggregation and weight calculation methodology in the integration phase. We present a number of observational and statistical experiments conducted on four healthcare datasets to measure the performance of BOFRF. The empirical results show that BOFRF improves the prediction performance of the local random forest models in all cases. Moreover, and more importantly, it provides significantly high improvement on the predictive power of sites having unsuccessful local classifiers because of their poor quality or imbalanced data unlike existing solutions.

In future work, to further improve the prediction capability of participating sites, we aim to extend our solution by adapting the concept of personalized federated learning [56], [57], [58], where each site obtains a site-specific federated model instead of a single global model. Furthermore, we aim to analyze the effect of applying feature selection techniques [59], [60] prior to running the actual algorithm to handle heterogeneous data.

ACKNOWLEDGMENT

This work was supported by the FAIR4Health project [61], which has received funding from the European Union's Horizon 2020 Research and Innovation Programme under Agreement 824666.

REFERENCES

[1] X.-D. Zhang, "Machine Learning," in *A Matrix Algebra Approach to Artificial Intelligence*. Singapore: Springer, 2020, pp. 223–440.

[2] (May 2016). *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, and Repealing Directive 95/46/EC (General Data Protection Regulation) (Text With EEA Relevance)*. [Online]. Available: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32016R0679

[3] V. Koutkias, "From data silos to standardized, linked, and FAIR data for pharmacovigilance: Current advances and challenges with observational healthcare data," *Drug Saf.*, vol. 42, no. 5, pp. 583–586, May 2019.

[4] C. C. Aggarwal and P. S. Yu, "A general survey of privacy-preserving data mining models and algorithms," in *Privacy-Preserving Data Mining: Models Algorithms*. Boston, MA, USA: Springer, 2008, pp. 11–52.

[5] N. Li, T. Li, and S. Venkatasubramanian, "T-closeness: Privacy beyond k-anonymity and l-diversity," in *Proc. IEEE 23rd Int. Conf. Data Eng.*, Apr. 2007, pp. 106–115.

[6] Q. Yang, Y. Liu, Y. Cheng, Y. Kang, T. Chen, and H. Yu, "Federated learning," *Synth. Lectures Artif. Intell. Mach. Learn.*, vol. 13, no. 3, pp. 1–207, 2019.

[7] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning," *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 2, pp. 1–19, 2019.

[8] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Y. Arcas, "Communication-efficient learning of deep networks from decentralized data," 2016, *arXiv:1602.05629*.

[9] S. Truex, N. Baracaldo, A. Anwar, T. Steinke, H. Ludwig, R. Zhang, and Y. Zhou, "A hybrid approach to privacy-preserving federated learning," in *Proc. 12th ACM Workshop Artif. Intell. Secur. (AISec)*, 2019, pp. 1–11.

[10] K. Wei, J. Li, M. Ding, C. Ma, H. H. Yang, F. Farokhi, S. Jin, Q. S. T. Quek, and H. V. Poor, "Federated learning with differential privacy: Algorithms and performance analysis," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 3454–3469, 2020.

[11] W. Zhang, "Dynamic-fusion-based federated learning for COVID-19 detection," *IEEE Internet Things J.*, vol. 8, no. 21, pp. 15884–15891, Nov. 2021.

[12] H. Wang, M. Yurochkin, Y. Sun, D. Papailiopoulos, and Y. Khazaeni, "Federated learning with matched averaging," 2020, *arXiv:2002.06440*.

[13] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Process. Mag.*, vol. 37, no. 3, pp. 50–60, May 2020.

[14] I. Palit and C. K. Reddy, "Scalable and parallel boosting with MapReduce," *IEEE Trans. Knowl. Data Eng.*, vol. 24, no. 10, pp. 1904–1916, Oct. 2012.

[15] Y. Li, C. Bai, and C. K. Reddy, "A distributed ensemble approach for mining healthcare data under privacy constraints," *Inf. Sci.*, vol. 330, pp. 245–259, Feb. 2016.

[16] Z.-H. Zhou, "Ensemble Learning," in *Machine Learning*. Singapore: Springer, 2021, pp. 181–210.

[17] O. Sagi and L. Rokach, "Ensemble learning: A survey," *WIREs Data Mining Knowl. Discovery*, vol. 8, no. 4, Jul. 2018, Art. no. e1249.

[18] S. Zhang, Y. Chen, W. Zhang, and R. Feng, "A novel ensemble deep learning model with dynamic error correction and multi-objective ensemble pruning for time series forecasting," *Inf. Sci.*, vol. 544, pp. 427–445, Jan. 2021.

[19] A. Kumar, J. Kim, D. Lyndon, M. Fulham, and D. Feng, "An ensemble of fine-tuned convolutional neural networks for medical image classification," *IEEE J. Biomed. Health Informat.*, vol. 21, no. 1, pp. 31–40, Jan. 2017.

[20] M. Abdar, M. Zomorodi-Moghadam, X. Zhou, R. Gururajan, X. Tao, P. D. Barua, and R. Gururajan, "A new nested ensemble technique for automated diagnosis of breast cancer," *Pattern Recognit. Lett.*, vol. 132, pp. 123–131, Apr. 2020.

[21] T. Zhou, H. Lu, Z. Yang, S. Qiu, B. Huo, and Y. Dong, "The ensemble deep learning model for novel COVID-19 on CT images," *Appl. Soft Comput.*, vol. 98, Jan. 2021, Art. no. 106885.

[22] L. Breiman, "Bagging predictors," *Mach. Learn.*, vol. 24, no. 2, pp. 123–140, 1996.

[23] R. Hornung and M. N. Wright, "Block forests: Random forests for blocks of clinical and omics covariate data," *BMC Bioinf.*, vol. 20, no. 1, pp. 1–17, Dec. 2019.

[24] A. Hajjem, F. Bellavance, and D. Larocque, "Mixed-effects random forest for clustered data," *J. Stat. Comput. Simul.*, vol. 84, no. 6, pp. 1313–1328, Jun. 2014.

[25] C. A. Field and A. H. Welsh, "Bootstrapping clustered data," *J. Roy. Stat. Soc., Ser. B, Stat. Methodol.*, vol. 69, no. 3, pp. 369–390, Jun. 2007.

[26] M. Samanta and A. H. Welsh, "Bootstrapping for highly unbalanced clustered data," *Comput. Statist. Data Anal.*, vol. 59, pp. 70–81, Mar. 2013.

[27] P. Bühlmann and B. Yu, "Boosting," *WIREs Comput. Statist.*, vol. 2, no. 1, pp. 69–74, Jan. 2010.

[28] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *J. Comput. Syst. Sci.*, vol. 55, no. 1, pp. 119–139, Aug. 1997.

[29] Y. Cao, Q.-G. Miao, J.-C. Liu, and L. Gao, "Advance and prospects of AdaBoost algorithm," *Acta Automatica Sinica*, vol. 39, no. 6, pp. 745–758, Jun. 2013.

[30] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *Ann. Statist.*, vol. 29, no. 5, pp. 1189–1232, Oct. 2001.

[31] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 785–794.

[32] F. Wang, H. Zhu, R. Lu, Y. Zheng, and H. Li, "A privacy-preserving and non-interactive federated learning scheme for regression training with gradient descent," *Inf. Sci.*, vol. 552, pp. 183–200, Apr. 2021.

[33] Y. Liu, Z. Ma, Z. Yan, Z. Wang, X. Liu, and J. Ma, "Privacy-preserving federated k-means for proactive caching in next generation cellular networks," *Inf. Sci.*, vol. 521, pp. 14–31, Jun. 2020.

[34] Y. Huang, L. Chu, Z. Zhou, L. Wang, J. Liu, J. Pei, and Y. Zhang, "Personalized cross-silo federated learning on non-IID data," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 7865–7873.

[35] Y. Liu, Y. Liu, Z. Liu, Y. Liang, C. Meng, J. Zhang, and Y. Zheng, "Federated forest," *IEEE Trans. Big Data*, vol. 8, no. 3, pp. 843–854, Jun. 2022.

[36] N. Ge, G. Li, L. Zhang, and Y. Liu, "Failure prediction in production line based on federated learning: An empirical study," *J. Intell. Manuf.*, May 2021. [Online]. Available: https://link.springer.com/article/10.1007/s10845-021-01775-2 and https://arxiv.org/abs/2101.11715

[37] S. Gambs, B. Kégl, and E. Aïmeur, "Privacy-preserving boosting," *Data Mining and Knowledge Discovery*, vol. 14, no. 1, pp. 131–170, 2007.

[38] T. Bruckhaus, "The business impact of predictive analytics," in *Knowledge Discovery and Data Mining*. Hershey, PA, USA: IGI Global, 2007, pp. 114–138.

[39] J. Davis and M. Goadrich, "The relationship between precision-recall and ROC curves," in *Proc. 23rd Int. Conf. Mach. Learn. (ICML)*, 2006, pp. 233–240.

[40] B. W. Matthews, "Comparison of the predicted and observed secondary structure of T4 phage lysozyme," *Biochim. Biophys. Acta-Protein Struct.*, vol. 405, no. 2, pp. 442–451, Oct. 1975.

[41] P. Baldi, S. Brunak, Y. Chauvin, C. A. F. Andersen, and H. Nielsen, "Assessing the accuracy of prediction algorithms for classification: An overview," *Bioinformatics*, vol. 16, no. 5, pp. 412–424, May 2000.

[42] S. Boughorbel, F. Jarray, and M. El-Anbari, "Optimal classifier for imbalanced data using Matthews correlation coefficient metric," *PLoS ONE*, vol. 12, no. 6, Jun. 2017, Art. no. e0177678.

[43] D. Chicco and G. Jurman, "The advantages of the Matthews correlation coefficient (MCC) over f1 score and accuracy in binary classification evaluation," *BMC Genomics*, vol. 21, no. 1, p. 6, Dec. 2020.

[44] R. E. Schapire and Y. Singer, "Improved boosting algorithms using confidence-rated predictions," *Mach. Learn.*, vol. 37, no. 3, pp. 297–336, 1999.

[45] C. Clifton, M. Kantarcioglu, J. Vaidya, X. Lin, and M. Y. Zhu, "Tools for privacy preserving distributed data mining," *ACM SIGKDD Explor. Newslett.*, vol. 4, no. 2, pp. 28–34, Dec. 2002.

[46] J. W. Smith, J. E. Everhart, W. C. Dickson, W. C. Knowler, and R. S. Johannes, "Using the ADAP learning algorithm to forecast the onset of diabetes mellitus," in *Proc. Symp. Comput. Appl. Med. Care*, Nov. 1988, pp. 261–265.

[47] *Pima Indians Diabetes Database*. Accessed: Mar. 25, 2022. [Online]. Available: https://www.kaggle.com/uciml/pima-indians-diabetes-database

[48] B. Antal and A. Hajdu, "An ensemble-based system for automatic screening of diabetic retinopathy," *Knowl.-Based Syst.*, vol. 60, pp. 20–27, Apr. 2014.

[49] D. Dua and C. Graff, *UCI Machine Learning Repository, School of Information and Computer Science*. Irvine, CA, USA: Univ. California, 2019. [Online]. Available: http://archive.ics.uci.edu/ml

[50] C. Bartley, *Replication Data for: South African Heart Disease*. Irvine, CA, USA: Harvard Dataverse, 2016.

[51] *Replication Data for: South African Heart Disease Dataset*, Mar. 2022, doi: 10.7910/DVN/76SIQD. [Online]. Available: https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/76SIQD
[52] G. Onder, I. Carpenter, H. Finne-Soveri, J. Gindin, D. Frijters, J. C. Henrard, T. Nikolaus, E. Topinkova, M. Tosato, R. Liperoti, F. Landi, and R. Bernabei, "Assessment of nursing home residents in Europe: The services and health for elderly in long TERm care (SHELTER) study," *BMC Health Services Res.*, vol. 12, no. 1, p. 5, Dec. 2012.
[53] G. Onder, R. Liperoti, D. Fialova, E. Topinkova, M. Tosato, P. Danese, P. F. Gallo, I. Carpenter, H. Finne-Soveri, J. Gindin, R. Bernabei, F. Landi, and f. the SHELTER Project, "Polypharmacy in nursing home in Europe: Results from the SHELTER study," *J. Gerontol. Ser. A, Biol. Sci. Med. Sci.*, vol. 67A, no. 6, pp. 698–704, Jun. 2012.
[54] G. Xu, H. Li, Y. Zhang, S. Xu, J. Ning, and R. Deng, "Privacy-preserving federated deep learning with irregular users," *IEEE Trans. Depend. Secure Comput.*, vol. 19, no. 2, pp. 1364–1381, Apr. 2022.
[55] M. Hao, H. Li, G. Xu, S. Liu, and H. Yang, "Towards efficient and privacy-preserving federated deep learning," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2019, pp. 1–6.
[56] S. Liu, J. Wang, and W. Zhang, "Federated personalized random forest for human activity recognition," *Math. Biosci. Eng.*, vol. 19, no. 1, pp. 953–971, 2021.
[57] Y. Deng, M. M. Kamani, and M. Mahdavi, "Adaptive personalized federated learning," 2020, *arXiv:2003.13461*.
[58] A. Fallah, A. Mokhtari, and A. Ozdaglar, "Personalized federated learning: A meta-learning approach," 2020, *arXiv:2002.07948*.
[59] R. Sheikhpour, M. A. Sarram, S. Gharaghani, and M. A. Z. Chahooki, "A robust graph-based semi-supervised sparse feature selection method," *Inf. Sci.*, vol. 531, pp. 13–30, Aug. 2020.
[60] L. Chamakura and G. Saha, "An instance voting approach to feature selection," *Inf. Sci.*, vol. 504, pp. 449–469, Dec. 2019.
[61] *FAIR4Health Project Website*. Accessed: Mar. 25, 2022. [Online]. Available: https://www.fair4health.eu/

MERT GENCTURK received the B.S. and M.S. degrees in computer engineering from Middle East Technical University, Turkey, in 2012 and 2015, respectively, where he is currently pursuing the Ph.D. degree in computer engineering.

He is also the Technical Manager and a Researcher with SRDC Corporation, Ankara, Turkey. His research interests include machine learning, federated learning, interoperability, and their application in the healthcare domain.

A. ANIL SINACI received the B.S., M.S., and Ph.D. degrees from the Department of Computer Engineering, Middle East Technical University, in 2007, 2009, and 2014, respectively.

He is a Senior Researcher and the Principal Solutions Architect with SRDC Corporation. He works as a Part-Time Instructor with the Department of Computer Engineering, Middle East Technical University. He has authored many papers published in peer-reviewed journals and conferences. His current research interests include distributed data management, data interoperability, machine learning, federated machine learning, big data analytics, and eHealth infrastructures.

NIHAN KESIM CICEKLI received the bachelor's degree in computer engineering from the Middle East Technical University, in 1986, the master's degree in computer engineering from Bilkent University, in 1988, and the Ph.D. degree in computer science from the Imperial College, London, U.K., in 1993.

She was a Visiting Faculty at the University of Central Florida, from 2001 to 2003, and a Teaching Professor at Syracuse University, from 2017 to 2019. She is a Professor of computer engineering with the Department of Computer Engineering, Middle East Technical University (METU), Ankara, Turkey. Her recent research interests include multimedia databases, information retrieval, recommendation systems, web information systems, and text mining.