SCALABLE PRIVACY-PRESERVING DISTRIBUTED EXTREMELY RANDOMIZED TREES FOR STRUCTURED DATA WITH MULTIPLE COLLUDING PARTIES

Amin Aminifar[1] *Fazle Rabbi*[1,2] *Yngve Lamo*[1]

[1]Western Norway University of Applied Sciences, [2]University of Bergen

ABSTRACT

Today, in many real-world applications of machine learning algorithms, the data is stored on multiple sources instead of at one central repository. In many such scenarios, due to privacy concerns and legal obligations, e.g., for medical data, and communication/computation overhead, for instance for large scale data, the raw data cannot be transferred to a center for analysis. Therefore, new machine learning approaches are proposed for learning from the distributed data in such settings. In this paper, we extend the distributed Extremely Randomized Trees (ERT) approach w.r.t. privacy and scalability. First, we extend distributed ERT to be resilient w.r.t. the number of colluding parties in a scalable fashion. Then, we extend the distributed ERT to improve its scalability without any major loss in classification performance. We refer to our proposed approach as k-PPD-ERT or Privacy-Preserving Distributed Extremely Randomized Trees with k colluding parties.

Index Terms— Distributed Learning, Privacy-Preserving Data Mining, Extremely Randomized Trees, Secure Multiparty Computation, Structured Data

1. INTRODUCTION

A basic assumption in traditional data mining algorithms is that all training data are stored in one data center where mining algorithms run. However, this assumption is not practical in many of today's real-world applications. Today, data is generated and stored on various machines, often located in distributed places. For example, health data is generated and stored at various hospitals, health service providers, and patients' personal devices. Such raw data cannot be shared with a data mining center due to privacy and legal concerns [1, 2]. At the same time, if each party performs mining on its limited data, the performance of the resulting model will largely be subordinate to the performance of a model that can be learned from all the data. Therefore, new mining approaches are required to learn from data distributed across multiple sources while maintaining privacy.

The learning from distributed data in privacy-preserving fashion have been extensively studied over the past decades. The first category of solutions is based on sharing raw data with a trusted third party, which might not be practical in certain scenarios since individuals' privacy cannot be protected from that party [3]. On the other hand, several studies have focused on perturbation-based solutions, e.g., [4–8], to address this issue by adding noise to the data before sharing it. While perturbing the data improves privacy, it also reduces the data utility. Moreover, noise removal techniques cast doubt on the privacy of such approaches [9, 10]. In addition, several anonymization methods, e.g., [11, 12], have been proposed to alter data values, by adopting techniques such as generalization (in k-anonymization [13]) or encryption of data values (in [14]), to avoid reidentification of data subjects [15], e.g., through linking attack [13]. However, in such perturbation-based and anonymization techniques, there is a trade-off between data utility and privacy, which make them impractical in certain scenarios.

Existing literature on data mining over distributed platforms incorporate approaches based on cryptographic and secure multiparty computing techniques [16–20]. However, such methods significantly increase communication and computing overhead, making them inefficient and impractical for many real-world scenarios, where we have large-scale data or limited communication and computing features, e.g., in mobile phones or resource-limited wearable devices [21–24]. Several state-of-the-art solutions, such as [3, 25, 26], aim to address learning in distributed settings in terms of reducing communication and computational overheads. This is because the complexity and scalability of the approach, along with the quality of data mining results and privacy, are among the three primary metrics for evaluating privacy-preserving data mining algorithms [27].

In this paper, we focus on the Extremely Randomized Trees (ERT) algorithm [28], which has a competitive performance for structured data, where we have independently meaningful attributes, compared to the existing state-of-the-art techniques, e.g., standard deep neural networks [29]. We consider the ERT algorithm in the distributed setting to reduce the amount of raw data leaving a party and privacy concerns [30]. We extend this distributed ERT framework in order to improve its scalability and privacy. We adopt an efficient Secure Multiparty Computation (SMC) technique for secure aggregation of partial results in our approach, which is resilient to multiple colluding parties, similar to Shamir's secret sharing technique [31]. We further propose a practical implementation of our proposed framework to reduce its overhead and improve its scalability. Moreover, we extend our proposed framework for efficient handling of large scale data and where only a subset of the parties participate in the process of learning. Our proposed framework offers the opportunity to make a trade-off among performance, privacy, and overhead.

This research is supported by INTROducing Mental health through Adaptive Technology (INTROMAT) project.

2. BACKGROUND

Extremely Randomized Trees (ERT) is a tree-based ensemble supervised learning method [28]. This approach is robust to overfitting since it follows the logic of bagging, i.e., it generates an ensemble of different weak classifiers and finally classifies based on a majority vote among these classifiers. The randomness parameters for generating distinctive weak classifiers are data attributes and splitting points for generating candidate decision nodes.

This paper considers the distributed ERT framework, which is adapted for learning classifier models from structured data, with categorical/numerical attributes and categorical labels, distributed over an arbitrary number of sources. In such a setting, the training data is horizontally partitioned and distributed over multiple sources, i.e., different records are stored on different data holder parties. The raw data cannot be shared with a central server for mining due to privacy and legal concerns. Therefore, the distributed ERT learns from the data without direct access to it and merely by partial and limited information from parties that hold the training data.

Distributed ERT iteratively learns an ensemble of decision trees. Learning a decision tree requires selecting a decision node at each step. The selection of decision nodes is performed based on the information gain. Information gain is a measure/score that indicates how well a decision node, compared to others, classifies the data samples to have more pure sets of samples at every branch of the decision node considering samples' labels. To calculate the information gain, the classification results of candidate decision nodes are required (from all data holder parties). Therefore, in distributed ERT, every data holder party classifies its records with the randomly generated decision nodes and obtains partial results (two vectors representing the combination/mixture of record labels fall into True and False branches). The aggregation of such partial results from all data holder parties enables the calculation of scores/information gains.

The direct sharing of such partial results to other parties puts the privacy of data subjects at risk. For instance, assuming the party holds only one record, if the candidate decision node classifies the data based on a sensitive attribute, e.g., suffering from a mental disorder, then the partial result indicates if the data subject falls under a certain category. For calculating the score, only the aggregation of partial results is required. In distributed ERT, each party aggregates its partial results to the previous party's received result and sends it to the next party. Although this technique is more efficient compared to the employed techniques in similar studies [3], e.g., Shamir's secret sharing technique [31], the number of colluding parties to reveal a secret value, in the worst case, is one.

In this study, we extend the distributed ERT framework and the secure aggregation protocol to be resilient to k colluding parties, where k is determined by the user. We further propose an efficient implementation for our framework, which is scalable and robust for large scale data w.r.t. the participation of a subset of data holder parties.

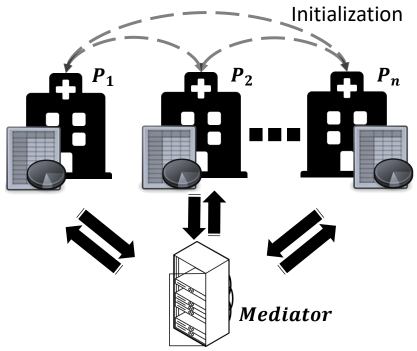


Fig. 1: Overall scenario for our privacy-preserving learning

3. APPROACH

In this section, we explain the proposed k-PPD-ERT algorithm. Section 3.1 describes the adopted secure aggregation technique for k-PPD-ERT. In Section 3.2, we explain how we can improve the scalability of the approach to learn from large scale data.

3.1. Privacy in the Presence of k Colluding Parties

Figure 1 illustrates the overall scenario for the proposed privacy-preserving learning framework. In the initialization phase of the k-PPD-ERT algorithm, each data holder party shares two seeds for the random function to other data holder parties (and receives two in return from each data holder party). The first seed (Seed for Selection of Parties, SSP) is unique for each sender but common for receivers, but the second seed (Seed for Secure Aggregation, SSA) is unique for each sender and receiver couple.

We suppose that the number of data holder parties is n. Therefore, after this initialization procedure, party m, P_m (where $1 \leq m \leq n$), receives two sets of $n-1$ seeds from other data holder parties ($\{SSP^{P_1}_{all}, \ldots, SSP^{P_n}_{all}\}$ and $\{SSA^{P_1}_{P_m}, \ldots, SSA^{P_n}_{P_m}\}$) and holds the seeds which were sent to other parties ($SSP^{P_m}_{all}$ and $\{SSA^{P_m}_{P_1}, \ldots, SSA^{P_m}_{P_n}\}$). Moreover, the secret value of party m is denoted by $secret\_val^{P_m}$.

The responsibilities of party m in one round of secure aggregation is explained in the following steps:

(a) **Identifying the k parties that participate in the secure aggregation for P_m:**
Party P_m uses $SSP^{P_m}_{all}$, in its random function, to identify which parties participate in secure aggregation for P_m, i.e., by randomly generating the party indices. Then, P_m generates random masks based on the *sent SSA* seeds ($\{SSA^{P_m}_{P_1}, \ldots, SSA^{P_m}_{P_n}\}$) of selected parties and aggregates them. It stores the result of aggregation in $rnd\_sum^{P_m}_{self}$.

(b) **Identifying the parties for which P_m participate in the secure aggregation:**
Party P_m uses its *received SSP* seeds ($\{SSP^{P_1}_{all}, \ldots, SSP^{P_n}_{all}\}$) to identify the parties with whose *received SSA* seeds, P_m must generate random masks. Then,

P_m generates random masks based on the *received* SSA seeds ($\{SSA^{P_1}_{P_m}, \ldots, SSA^{P_n}_{P_m}\}$) of selected parties and aggregates them. It stores the result of aggregation in $rnd\_sum^{P_m}_{others}$.

(c) **Aggregation and transfer of partial results ($P.R.$) to the mediator:**
Party P_m calculates $P.R.^{P_m}$ as follows: $P.R.^{P_m} = secret\_val^{P_m} - rnd\_sum^{P_m}_{self} + rnd\_sum^{P_m}_{others}$. Then, P_m sends $P.R.^{P_m}$ to the mediator.

The mediators calculates the desired result (aggregation of secret values) by aggregating all received partial results.

Privacy: We now show that the secret values of parties are kept private in our proposed protocol. The partial result $P.R.^{P_m}$, which is shared with the mediator consists of three components: $secret\_val^{P_m}$, $rnd\_sum^{P_m}_{self}$, and $rnd\_sum^{P_m}_{others}$. The two components, $rnd\_sum^{P_m}_{self}$ and $rnd\_sum^{P_m}_{others}$, mask the secret value. The value of $rnd\_sum^{P_m}_{self}$ can only be identified by collusion of k parties holding the random seeds for generating the random masks, which are the components of $rnd\_sum^{P_m}_{self}$. At the same time, $rnd\_sum^{P_m}_{others}$ can only be identified by collusion of k (potentially) other parties which generate the components of $rnd\_sum^{P_m}_{others}$. In the worst case, the k parties involved in $rnd\_sum^{P_m}_{self}$ and $rnd\_sum^{P_m}_{others}$ may be the same; hence, the minimum number of colluding data holder parties equals to k. Moreover, since the mediator receives the victim's partial result, the collusion of other parties without the mediator's participation is ineffective. Therefore, for identifying a secret value, the collusion of k data holder parties and the mediator is necessary.

Correctness: We also show that the final value of aggregation of partial results is equal to the aggregation of secret values. Without loss of generality we consider $k = n - 1$. The aggregation of all partial results sent to the mediator is as follows:

$$\begin{aligned}\sum_{j=1}^{n} P.R.^{P_j} &= secret\_val^{P_1} - rnd\_sum^{P_1}_{self} + rnd\_sum^{P_1}_{others} \\ &\vdots \\ &+ secret\_val^{P_n} - rnd\_sum^{P_n}_{self} + rnd\_sum^{P_n}_{others} \\ &= \sum_{j=1}^{n} secret\_val^{P_j} - \sum_{j=1}^{n} rnd\_sum^{P_j}_{self} + \sum_{j=1}^{n} rnd\_sum^{P_j}_{others}.\end{aligned} \tag{1}$$

Based on (a), $rnd\_sum^{P_m}_{self} = \sum_{i=1}^{n} rnd^{P_m}_{P_i} - rnd^{P_m}_{P_m}$, where $rnd^{P_m}_{P_i}$ is the shared random mask between P_m and P_i. On the other hand, based on (b), $rnd\_sum^{P_m}_{others} = \sum_{i=1}^{n} rnd^{P_i}_{P_m} - rnd^{P_m}_{P_m}$. Substituting these two equations in equation 1, we obtain:

$$\begin{aligned}\sum_{j=1}^{n} P.R.^{P_j} &= \sum_{j=1}^{n} secret\_val^{P_j} - \sum_{j=1}^{n} rnd\_sum^{P_j}_{self} + \sum_{j=1}^{n} rnd\_sum^{P_j}_{others} \\ &= \sum_{j=1}^{n} secret\_val^{P_j} - \sum_{j=1}^{n} (\sum_{i=1}^{n} rnd^{P_j}_{P_i} - rnd^{P_j}_{P_j}) \\ &+ \sum_{j=1}^{n} (\sum_{i=1}^{n} rnd^{P_i}_{P_j} - rnd^{P_j}_{P_j}) = \sum_{j=1}^{n} secret\_val^{P_j}.\end{aligned} \tag{2}$$

The above equations show that the aggregation of partial results from data holder parties is equal to the aggregation of data holder parties' secret values.

3.2. Efficient Handling of Large Scale Data

In distributed ERT, all the data holder parties participate in (collaborate on) the process of selecting the best decision node/leaf at every round of the algorithm. However, in order to efficiently handle large scale data sets and reduce the communication/computation overheads, in k-PPD-ERT, only a subset of data holder parties participate in the process of learning at every round. The probability of participation of each party in the learning process at each round is a parameter that is set by the user.

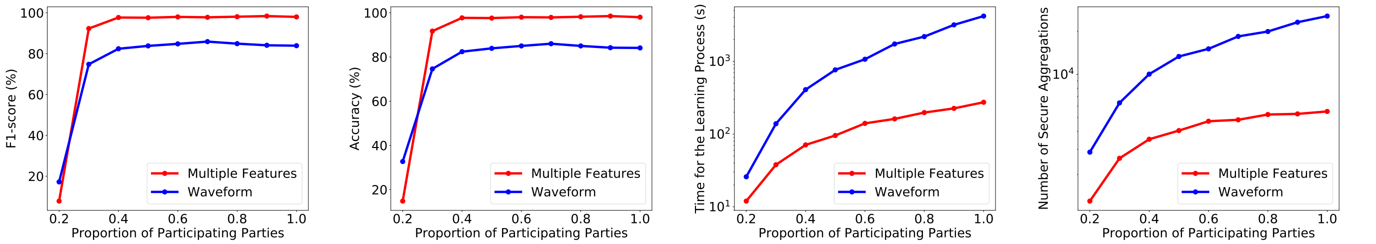
The algorithm uses the aggregation of data holder parties' partial results to calculate the candidate decision nodes' score/information gain. In certain rounds, the result of this aggregation is used to select a leaf for the tree. In k-PPD-ERT, when not all the parties participate in the aggregation process, the result of aggregation changes. However, in Section 4, we experimentally show that this technique does not lead to a major loss in the classification performance of our learned models.

Random participation of data holders in the described process changes the result of aggregation and, consequently, the learning. However, the learning results are not noticeably affected (shown experimentally in Section 4). The randomness in the participation of data holder parties, similar to the randomness in the generation of candidate decision nodes in the distributed ERT, is another source of randomness in our approach. Introducing another source of randomness in ensemble learning methods while keeping the algorithm's ability to generate weak classifiers is in accordance with the logic of bagging.

To determine which parties participate at each round, the mediator shares a common random seed (Seed for Participating Parties, *SPP*) with all data holder parties. Therefore, by using this seed and the constant probability of participation, every party determines the participating parties in that round of secure aggregation (for selecting the best candidate decision node/leaf). Then, each participating party picks its k peer parties for secure aggregation based on the available parties in that round, determined by SPP.

4. EVALUATION AND DISCUSSION

In this section, first, we evaluate the adopted secure aggregation technique. We compare our technique with distributed ERT and Shamir's techniques. These secure aggregation techniques are evaluated based on the communication cost in one round of secure aggregation and the minimum number of parties that need to participate in collusion in order to identify a secret value. Then, we examine the limited participation of data holder parties in the process of selecting the best candidate decision node/leaf. We evaluate the classification performance and the scalability of k-PPD-ERT offered by adopting this approach.



(a) F1-score w.r.t. proportion of participating parties

(b) Accuracy w.r.t. proportion of participating parties

(c) Elapsed learning time w.r.t. proportion of participating parties

(d) Secure aggregations w.r.t. proportion of participating parties

Fig. 2: Analysis of the classification performance, the elapsed learning time, and number of secure aggregations for learning based on different proportions of participating parties in the learning process

Table 1 exhibits and compares communication costs (in one round of secure aggregation) and the minimum number of parties necessary to collude for identifying a secret value. According to the table, the communication complexity of k-PPD-ERT has the lowest order, while offering the highest minimum number of colluding parties for identifying a secret value. The communication complexity of the k-PPD-ERT technique is $O(n)$, similar to the distributed ERT, while this equals to $O(n^2)$ for Shamir's technique. On the other hand, the minimum number of colluding parties for k-PPD-ERT is k data holder parties plus the mediator, which is the highest. Therefore, the k-PPD-ERT's secure aggregation technique offers privacy with multiple colluding parties, while preserving the algorithm's scalability.

Table 1: Scalability and privacy comparison against existing techniques

Approach	Party	Communication (N is the number of parties)			Min Number of
		Send	Receive	Total (All N parties)	Colluding Parties
Distributed ERT	All	1	1	$2N$	1
k-PPD-ERT	Data Holders	1	0	$2(N-1)$	$k+1$ $(k < N)$
	Mediator	0	$N-1$		
Shamir [31]	k-1 Parties	N	$N-1$	$2(N^2-N+k-1)$	k $(k < N)$
	One Party	$N-1$	$N+k-2$		
	The Rest	$N-1$	$N-1$		

In k-PPD-ERT's secure aggregation technique, the total number of send and receive messages in k-PPD-ERT is independent of k, so we can always set k to $n-1$. This does not introduce any cost, concerning the communication, in our algorithm.

We now evaluate data holder parties' limited participation at every round of a selecting decision node/leaf. To investigate this feature, we use Multiple Features [32] and Waveform Database Generator (Version 1) [33] datasets, and allocate $2/3$ of each dataset for learning and the rest for testing. We distribute the training data evenly among ten parties. The mediator learns an ensemble of 25 decision trees by k-PPD-ERT in every experiment. We repeat the learning process for situations in which the proportion of participating parties at every round of selecting the best decision node/leaf is $0.2, 0.3, 0.4, ..., 1$. Figure 2 visualizes the results of these experiments. In every experiment, we record: the classification performance, shown in Figure 2a and 2b, the elapsed time for learning process, in Figure 2c, and the number of required secure aggregations for the learning process, in Figure 2d. The Y-axis in Figure 2c and 2d has a logarithmic scale (because of the differences in the magnitude of results for Multiple Features and Waveform datasets).

On the one hand, the results in Figure 2a and 2b show that random participation of only 40% of data holder parties at each round leads to high classification performance. The difference in classification performance for 40% of participation and more (even when all parties participate, similar to distributed ERT) is negligible. Furthermore, in some experiments with data holders' limited participation, we obtain models with higher classification performance. The logic behind bagging and the introduced source of randomness in k-PPD-ERT may explain these improvements.

On the other hand, the results in Figure 2c and 2d show improvements concerning the scalability when fewer data holders participate in learning at each round. Figure 2c shows the decrease of elapsed time for learning a model by reducing the number of participating parties. In addition, Figure 2d exhibits the continuous growth of secure aggregation rounds by increasing the number of parties that participate in different rounds of selecting a decision node/leaf for our decision trees.

The results in Figure 2 show that our algorithm's scalability improves by limiting the number of data holder parties that participate in every round of selection of a decision node/leaf. However, the learning performance and its resulting models will not have any noticeable loss.

5. CONCLUSION

In this paper, we consider the distributed ERT framework and extend it by adopting a secure aggregation technique that is resilient to the collusion of up to k data holder parties and the mediator. We further proposed a scalable implementation for our framework, which is efficient w.r.t. the communication overhead. Moreover, we investigated the efficient handling of large scale data with the limited participation of data holder parties at every round of the learning process. Our evaluation demonstrates the privacy preservation and resilience of the proposed framework w.r.t. the number of colluding parties and its scalability and robustness for large scale data w.r.t. the participation of a subset of data holder parties.

6. REFERENCES

[1] Samuel D Lustgarten, Yunkyoung L Garrison, Morgan T Sinnard, and Anthony WP Flynn, "Digital privacy in mental healthcare: current issues and recommendations for technology use," *Current Opinion in Psychology*, 2020.

[2] Damian Pascual, Alireza Amirshahi, Amir Aminifar, David Atienza, Philippe Ryvlin, and Roger Wattenhofer, "Epilepsygan: Synthetic epileptic brain activities with privacy preservation," *IEEE Transactions on Biomedical Engineering*, 2020.

[3] Fatih Emekçi, Ozgur D Sahin, Divyakant Agrawal, and Amr El Abbadi, "Privacy preserving decision tree learning over multiple parties," *Data & Knowledge Engineering*, 2007.

[4] Rakesh Agrawal and Ramakrishnan Srikant, "Privacy-preserving data mining," in *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, 2000.

[5] Shipra Agrawal and Jayant R Haritsa, "A framework for high-accuracy privacy-preserving mining," in *21st International Conference on Data Engineering (ICDE'05)*. IEEE, 2005.

[6] Alexandre Evfimievski, Ramakrishnan Srikant, Rakesh Agrawal, and Johannes Gehrke, "Privacy preserving mining of association rules," *Information Systems*, 2004.

[7] Shariq J Rizvi and Jayant R Haritsa, "Maintaining data privacy in association rule mining," in *VLDB'02: Proceedings of the 28th International Conference on Very Large Databases*. Elsevier, 2002.

[8] Rakesh Agrawal, Ramakrishnan Srikant, Johannes Gehrke, and Alexandre Evfimievski, "Privacy preserving mining of association rules," *Information systems*, 2004.

[9] Hillol Kargupta, Souptik Datta, Qi Wang, and Krishnamoorthy Sivakumar, "On the privacy preserving properties of random data perturbation techniques," in *Third IEEE international conference on data mining*. IEEE, 2003.

[10] Zhengli Huang, Wenliang Du, and Biao Chen, "Deriving private information from randomized data," in *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, 2005.

[11] Ashwin Machanavajjhala, Daniel Kifer, Johannes Gehrke, and Muthuramakrishnan Venkitasubramaniam, "l-diversity: Privacy beyond k-anonymity," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 2007.

[12] Noman Mohammed, Benjamin CM Fung, Patrick CK Hung, and Cheuk-kwong Lee, "Anonymizing healthcare data: a case study on the blood transfusion service," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2009.

[13] Latanya Sweeney, "k-anonymity: A model for protecting privacy," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 2002.

[14] Amin Aminifar, Yngve Lamo, Ka I Pun, and Fazle Rabbi, "A practical methodology for anonymization of structured health data," in *Proceedings of the 17th Scandinavian Conference on Health Informatics*, 2019.

[15] "Health informatics — Pseudonymization," Standard, International Organization for Standardization, 2017.

[16] Chris Clifton, Murat Kantarcioglu, Jaideep Vaidya, Xiaodong Lin, and Michael Y Zhu, "Tools for privacy preserving distributed data mining," *ACM Sigkdd Explorations Newsletter*, 2002.

[17] Yehuda Lindell and Benny Pinkas, "Privacy preserving data mining.," *Journal of cryptology*, 2002.

[18] Murat Kantarcioglu, "A survey of privacy-preserving methods across horizontally partitioned data," in *Privacy-preserving data mining*. Springer, 2008.

[19] Jaideep Vaidya, "A survey of privacy-preserving methods across vertically partitioned data," in *Privacy-preserving data mining*. Springer, 2008.

[20] Yoshinori Aono, Takuya Hayashi, Lihua Wang, Shiho Moriai, et al., "Privacy-preserving deep learning via additively homomorphic encryption," *IEEE Transactions on Information Forensics and Security*, 2017.

[21] Benny Pinkas, "Cryptographic techniques for privacy-preserving data mining," *ACM Sigkdd Explorations Newsletter*, 2002.

[22] Dionisije Sopic, Amin Aminifar, Amir Aminifar, and David Atienza, "Real-time classification technique for early detection and prevention of myocardial infarction on wearable devices," in *2017 IEEE Biomedical Circuits and Systems Conference (BioCAS)*. IEEE, 2017.

[23] Dionisije Sopic, Amin Aminifar, Amir Aminifar, and David Atienza, "Real-time event-driven classification technique for early detection and prevention of myocardial infarction on wearable systems," *IEEE transactions on biomedical circuits and systems*, 2018.

[24] Farnaz Forooghifar, Amir Aminifar, and David Atienza, "Resource-aware distributed epilepsy monitoring using self-awareness from edge to cloud," *IEEE transactions on biomedical circuits and systems*, 2019.

[25] Jakub Konečnỳ, H Brendan McMahan, Daniel Ramage, and Peter Richtárik, "Federated optimization: Distributed machine learning for on-device intelligence," *arXiv preprint arXiv:1610.02527*, 2016.

[26] H Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, et al., "Communication-efficient learning of deep networks from decentralized data," *arXiv preprint arXiv:1602.05629*, 2016.

[27] Elisa Bertino, Dan Lin, and Wei Jiang, "A survey of quantification of privacy preserving data mining algorithms," in *Privacy-preserving data mining*. Springer, 2008.

[28] Pierre Geurts, Damien Ernst, and Louis Wehenkel, "Extremely randomized trees," *Machine learning*, 2006.

[29] Scott M Lundberg, Gabriel Erion, Hugh Chen, Alex DeGrave, Jordan M Prutkin, Bala Nair, Ronit Katz, Jonathan Himmelfarb, Nisha Bansal, and Su-In Lee, "From local explanations to global understanding with explainable ai for trees," *Nature machine intelligence*, 2020.

[30] Amin Aminifar, Fazle Rabbi, Ka I Pun, and Yngve Lamo, "Privacy preserving distributed extremely randomized trees," in *Proceedings of the 36th Annual ACM Symposium on Applied Computing*, 2021.

[31] Adi Shamir, "How to share a secret," *Commun. ACM*, 1979.

[32] "UCI Machine Learning Repository: multiple features data set," https://archive.ics.uci.edu/ml/datasets/Multiple+Features, Accessed: 2021-02-09.

[33] Leo Breiman, Jerome Friedman, Charles J Stone, and Richard A Olshen, *Classification and regression trees*, CRC press, 1984.