10-9-2023

Applying Random Forests in Federated Learning: A Synthesis of Aggregation Techniques

Mattis Bodynek
Karlsruher Institut für Technologie, Germany, mattis.bodynek@student.kit.edu

Florian Leiser
Karlsruher Institut für Technologie, Germany, florian.leiser@kit.edu

Scott Thiebes
Karlsruher Institut für Technologie, Germany, scott.thiebes@kit.edu

Ali Sunyaev
Karlsruher Institut für Technologie, Germany, sunyaev@kit.edu

Applying Random Forests in Federated Learning: A Synthesis of Aggregation Techniques

Research Paper

Mattis Bodynek, Florian Leiser, Scott Thiebes, and Ali Sunyaev

Karlsruhe Institute of Technology, Karlsruhe, Germany
mattis.bodynek@student.kit.edu
{florian.leiser,scott.thiebes, sunyaev}@kit.edu

Abstract. Random forests (RFs) are a versatile choice for many machine learning applications. Despite their promising efficiency and simplicity, RFs are seldom used in collaborative scenarios like federated learning (FL). In FL, training data is scattered among a federation of clients. To train federated models, a central server aggregates inputs from all clients. For RFs as non-parametric models, coordinating the training phase and aggregating the global model is non-trivial. Design choices regarding the evaluation of candidate splits and the aggregation of decision trees prove to be context-specific. In this work, we identify aggregation techniques proposed in extant literature. The identified techniques are categorized across dimensions like training coordination, inference process, and privacy. We find an important distinction between synchronous and asynchronous techniques and evaluate the practical suitability of aggregation techniques by comparing advantages and drawbacks for prediction robustness and technical feasibility. Our results facilitate design choices of future federated RFs.

Keywords: Random Forest, Federated Learning, Literature Review

1 Introduction

Across many data-heavy applications like healthcare or finance, machine learning (ML) applications are becoming increasingly relevant. Supervised ML applications require large amounts of training data, which is troublesome when that data is residing in siloed environments Rieke et al. (2020). Collaborations could enrich the data sets and improve the predictions for all parties, but the exchange of sensitive data is often not possible due to strict privacy regulations or the protection of proprietary knowledge (Kairouz et al. 2021). Federated learning (FL) seeks to overcome these impediments by adapting ML techniques to multi-client environments. Federated ML models are trained collaboratively without violating data privacy requirements. In traditional FL, clients start by training parametric models like neural networks on locally available data sets. In each iteration, the local model updates are aggregated by a central server. The aggregated model is then made available to all participating clients (McMahan et al. 2017).

Studies suggest that the success of ML models in practice is determined by a range of factors surpassing quality measures established in research (Ishikawa & Yoshioka 2019). Factors like explainability and user-friendliness are critical for the long-term adoption of

18th International Conference on Wirtschaftsinformatik
September 2023, Paderborn, Germany

a model, especially in areas with high accountability standards (Varghese 2020). Neural networks are the most researched models in FL but dissatisfy some of these requirements (Paleyes et al. 2022). More explainable, less complex predictors like decision trees or regression models may suffer from inferior predictive performance (Bell et al. 2022). Random forests (RFs) can bridge this gap between explainability and predictive quality Varghese (2020). RFs are supervised ML models built by compiling multiple decision trees in an ensemble and aggregating the outputs for predictions (Breiman 1996). They can process categorical and high-dimensional data with good predictive performance (Cutler et al. 2012), while facilitating the employment of explainability techniques (De Fazio et al. 2023). For a deep technical introduction into RFs, we suggest the review of Biau & Scornet (2016) as a starting point.

The unique balance between prediction quality and explainability (Varghese 2020) makes RFs appealing models for FL as well. However, the non-parametric nature of the underlying decision trees necessitates dedicated aggregation techniques for federated RFs (FRFs) (Li et al. 2021). A range of these techniques has been proposed so far. However, existing knowledge of different techniques may be inaccessible to potential users because authors use different terminology and focus on specific applications. By synthesizing the body of work on FRFs and characterizing existing techniques, we aim to guide researchers and practitioners in identifying relevant solutions. Thus, we ask the research question (RQ):

> *RQ:* Which aggregation techniques for FL with RFs exist in research, and how do they compare with each other?

To answer our RQ, we review aggregation techniques for FRFs found in scientific literature. The identified techniques are structured by applying categorical mappings to make them more accessible to the reader. Subsequently, we evaluate the techniques based on developed criteria regarding their suitability in practice. Together, the literature review and subsequent evaluation contribute to an improved understanding of different aggregation techniques for RFs and non-parametric models in general. Our results serve as a basis for informed and thus more confident decisions on the design of the aggregation process, ultimately facilitating design choices for the application of non-parametric models in FL.

2 Background on Federated Learning

With the increasing interest in collaborative ML models, FL gained a lot of interest over the past years (Yang et al. 2019). In traditional FL, raw training data remains distributed at each client and is not exchanged across client boundaries, achieving strong data protection. A central server receives "focused updates intended for immediate aggregation [...] used to achieve the learning objective" (Kairouz et al. 2021). A fundamental question for the application of FRFs is the interconnection of data provisioning by local clients and the aggregation by a central server, which can be designed in numerous ways. To describe this central concept, we use the term *aggregation technique* as the combined strategy for data provisioning by technically or legally separated clients and the aggregation of this data into a shared model.

An important distinction of FL regards the participating clients. Cross-device FL is concerned with training a global model by leveraging inputs of many unreliable clients whose individual contribution is asymptotically insignificant (Kairouz et al. 2021). Cross-silo FL on the other hand describes federated systems which involve fewer but more powerful and reliable clients, such as data centers (Kairouz et al. 2021). FL can further be differentiated regarding the partitioning of training data. The most common scenario is horizontal FL, where clients share the same feature space but hold data on different samples. Conversely, in vertical FL, clients share the same samples but hold data on different feature sets. In practice, vertically partitioned data is most relevant in the context of cross-silo FL. Federated transfer learning constitutes situations where neither feature nor sample spaces align (Kairouz et al. 2021).

An emerging paradigm in FL is multi-task learning (Kairouz et al. 2021), where every local learning task is considered separately. This is especially relevant when data is not identically and independently distributed (IID). While IID assumptions are often violated globally, they may hold for subsets of client data. Under these conditions, multi-task FL can outperform global models (Kairouz et al. 2021).

3 Review Methodology

To synthesize and assess the current state of research on FRFs, we conducted a scoping review. This method is well-suited for the exploration of research areas that have not been reviewed thoroughly yet (Mays et al. 2001). In particular, we followed the framework of Arksey & O'Malley (2005) to conduct the review.

3.1 Study Selection

Table 1 summarizes our search strategy, including search terms and inclusion criteria. Figure 1 visualizes the review process. The search yielded 293 unique records after removing duplicates (n = 159). In a first iteration, title and abstract of the records were screened and 234 manuscripts with no apparent relevance were excluded. The full texts of the remaining 59 records were retrieved and examined in more detail, where we excluded 38 publications due to insufficient eligibility. In the end, we charted 21 publications.

Table 1. Settings for the conducted scoping study

Criterion	**Detail**
Language	English
Timeframe	01/2016 - 10/2022
Databases	IEEE Xplore, ACM Digital Library, Scopus, ProQuest
Source	Journals, Conferences, arXiv preprints
Status	Peer-reviewed or currently under review
Search area	Abstract only
Search terms	(["distributed" OR "federated" OR "incremental"] AND ["learning" OR "machine learning" or "ML"] OR FedML) AND (forest OR tree OR trees)

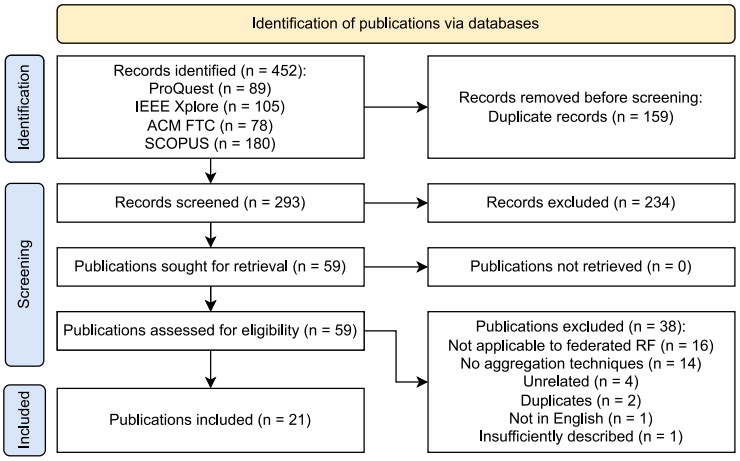

Figure 1. Outline of the review process

3.2 Data Charting

In addition to general metadata, we extracted information for six facets representing central characteristics of the identified aggregation techniques. These facets were determined through a targeted, exploratory analysis of the included publications. In this process, we prioritized categories with high discriminatory power. We developed an initial set of eight candidates, which we found to be relevant beyond the scope of FRFs. Two of these (namely, *explainability* and *problem domain*) were discarded, as they were not present in enough sources and thus did not provide sufficient insights into the differences between techniques within our sample. The remaining six facets are described below.

4 Review Results

An overview of the identified aggregation techniques and the evaluated mapping dimensions is displayed in Table 2. The overwhelming majority of the proposed systems apply cross-device FL. Clients are typically not only data sources but also influence the training phase and prediction process. Most studies investigate configurations between 2 and 100 clients (Hauschild et al. 2022, Liu et al. 2020b), where each client resembles an institutional data domain. The work of Kalloori & Klingler (2022) even aims at "involving corporate companies instead of mobile devices in the federated learning process".

Partitioning. The first facet captures the partitioning of training data across clients. We differentiate between *horizontal*, *vertical* FL, and *federated transfer learning*.

Researchers have explored data partitioning for FRFs with a clear tendency toward horizontal FL. Aggregation techniques within vertical FL appear to be particularly homogeneous with respect to our mapping dimensions like privacy (83% apply HE) and inference (83% joint inference). Interestingly, we did not find any examples for federated

Table 2. Overview of the identified aggregation techniques.

Year	Authors	Partitioning	Coordination	Task	Privacy	Architecture	Inference
2021	Aliyu et al.	Horizontal	Asynchronous	Class	-	Blockchain	Local
2022	Gencturk et al.	Horizontal	Asynchronous	Class	MPC	Server, P2P	Local
2019	Giacomelli et al.	Horizontal	Asynchronous	Class	HE	Server	Server
2022	Hauschild et al.	Horizontal	Asynchronous	Class	-	Server	Server
2022	Kwatra & Torra	Horizontal	Asynchronous	Class	A	Server	Local
2022	Ma et al.	Horizontal	Asynchronous	Class	-	Server	N/A
2021	Ma et al.	Horizontal	Asynchronous	Class	HE	Server	Server
2022	Markovic et al.	Horizontal	Asynchronous	Class	-	Server	N/A
2020	de Souza et al.	Horizontal	Asynchronous	Class	A	Blockchain	Local
2021	Zhang et al.	Horizontal	Asynchronous	Class	-	Server	Local
2021	Kwatra & Torra	Horizontal	Asynchronous	Class, Reg	-	Server	N/A
2022	Kalloori & Klingler	Horizontal	Synchronous	Class	DP	Server	Local
2022	Liu et al.	Horizontal	Synchronous	Class	DP	MC	Local
2020b	Liu et al.	Horizontal	Synchronous	Class, Reg	DP	Server	Local
2016	Guo et al.	Horizontal	Synchronous	Reg	-	Server	Local
2022	Ge et al.	Vertical	Synchronous	Class	-	Server	Joint
2021	Hou et al.	Vertical	Synchronous	Class	HE	MC-Server	Joint
2021b	Liu et al.	Vertical	Synchronous	Class	HE	MC-Server	N/A
2021	Liu et al.	Vertical	Synchronous	Class, Reg	HE	Server	Joint
2020	Wu et al.	Vertical	Synchronous	Class, Reg	HE, MPC, DP	MC	Joint
2022	Yao et al.	Vertical	Synchronous	Class, Reg	HE	Server	Joint

transfer learning (FTL) in the reviewed work. The notion of transfer learning precedes FL and describes differences in domains, tasks, and distributions (Pan & Yang 2009), which is a broader mandate than the FL-specific definition by Kairouz et al. (2021). Following the earlier notion, some have labeled personalized or multi-task FL as FTL (Saha & Ahmad 2021). We keep to the stricter definition, as it is more precise and does not overlap with the fields of horizontal and vertical FL.

Coordination. The second facet describes the coordination of the model training phase. We identified two architectural patterns, which we termed *asynchronous* and *synchronous* training, in consistency with other FL research (Zhang et al. 2020). The coordination of the model training phase determines the contents of the interaction between client and server. In asynchronous training of FRFs, clients build decision trees independently on their local data without model-specific communication. After finishing training, clients send their local models to the server, which aggregates the received data into a global RF. On the other hand, synchronous aggregation techniques collaboratively build decision trees across all clients under the coordination of an aggregation server. For each new node, clients provide information about the composition of their local data set. The server then finds and broadcasts the optimal split according to the provided data. With the globally selected split, nodes are expanded by recursively partitioning the local data at each client.

Across the 21 included publications, 11 employ asynchronous coordination, all within horizontal FL. The aggregating entity may apply a processing strategy to modify the composition of the federated model. The observed strategies fall into weighting, filtering, merging, or full compilation of all decision trees. Filtering reduces the size of the global RF by only considering a subset of local trees, either randomly (Hauschild et al. 2022) or performance-based (Markovic et al. 2022, Ma et al. 2021). Weighting techniques regulate the influence of each tree on the final prediction. Similar to filtering, we found different performance-based weighting methods (Gencturk et al. 2022, Ma et al.

2022). Lastly, merging techniques create decision trees by merging sets of local input trees. The identified approaches range from simple, node-wise selection rules (Kwatra & Torra 2022) to complex, transformation-based mechanisms (Kwatra & Torra 2021).

Ten of the reviewed sources instead employ synchronous training. In horizontal FL, each client holds a unique local data set with the respective labels. During training, clients provide summarized representations of the label distributions, so the central server can determine the global split (Guo et al. 2016, Kalloori & Klingler 2022). In systems relying on aggregating distribution data, clients communicate using histograms, which can be implemented for both classification (Kalloori & Klingler 2022) and regression tasks (Guo et al. 2016). A key property of histograms is the ability to merge adjacent bins. This enables a dynamic aggregation process, where the set of evaluated candidate splits is not explicitly defined at the time of aggregation. Alternatively, the server distributes a fixed set of split values to all clients (Liu et al. 2022, 2020b). In this case, the results are aggregated directly, since all clients evaluate the same candidate splits.

In vertical FL, clients either have access to all or none of the labels. If label data is available to all clients, they can autonomously compute the locally optimal split (Ge et al. 2022). The server can then determine the globally optimal split by comparing all local results. However, most vertical synchronous techniques assume labels are proprietary data of a single entity. In this scenario, clients can only share preliminary data for candidate splits, for instance in the form of quantile statistics (Yao et al. 2022, Liu et al. 2021b). This data always has to be forwarded to the entity holding the labels.

Task. The reviewed work covers both classification (*class*) and regression (*reg*) tasks. The type of prediction task was determined either by explicit comments of the authors or by analyzing the proposed algorithms.

Only one publication focuses exclusively on regression tasks, whereas five approaches allow for both classification and regression. The most commonly mentioned prediction task is binary or multi-class classification with 15 identified sources.

Privacy. We identified four different privacy preservation techniques throughout training and prediction. Secure multi-party computation (*MPC*) allows for secure evaluation of a function based on private inputs of each client (Damgård et al. 2012). Homomorphic encryption (*HE*) transforms clients' inputs with homomorphic functions, scrambling the inputs but maintaining algebraic relations between them. Differential privacy (*DP*) introduces noise in models to mask a client's contribution (Kairouz et al. 2021). Lastly, anonymization (*A*) protects samples by removing or blurring any identifiable information.

Out of all included publications, seven integrate HE schemes to secure their communication. Wu et al. (2020) employ HE to reduce communication overhead compared to solely relying on MPC. Their HE scheme supports addition, multiplication, and the dot product. Additionally, they employ an MPC protocol based on additive secret sharing for the secure comparison of the computed split impurities (Wu et al. 2020). Gencturk et al. (2022) propose a P2P network where MPC allows for anonymous computing of the performance of local models on other clients' data. Four approaches apply DP, including Kalloori & Klingler (2022), who use histograms with added noise to communicate distribution data to the server. Liu et al. (2020b) propose a complex, multi-layer system for perturbing label information. Anonymization is employed in the form of k-anonymity

(Kwatra & Torra 2022) and by excluding identifiable information from the aggregation process (de Souza et al. 2020).

Architecture. Different communication architectures of aggregation techniques are compared in our fifth facet. *Server* describes the traditional setting, where all clients have the same role within the system and communication is routed through a central server. In architectures with a main client (*MC*), all coordination is handled by one of the clients. It embodies the role of the server, but still participates in the training. *MC-Server* scenarios are relevant for vertical FL, where one client holds all label data. Communication is handled by an aggregation server, which additionally forwards all local partition data to this main client. Peer-to-peer (*P2P*) architectures enable communication protocols without a centralized server. Finally, blockchain-based systems (*blockchain*) rely on decentralized ledgers, where clients store retrieval information for locally trained models.

The traditional client-server architecture is the prevalent communication setup among the reviewed approaches. Only three techniques with asynchronous training implement blockchain-based (Aliyu et al. 2021, de Souza et al. 2020) or P2P (Gencturk et al. 2022) architectures. We found variations regarding the entity controlling the flow of information among synchronously trained techniques. Some approaches introduce a main client with additional responsibilities for the protocol (Hou et al. 2021, Wu et al. 2020).

Inference. Our sixth facet describes the provision of the global model along with the involved parties. Three types of prediction processes can be distinguished. First, each client may apply the global model independently in a local inference system (*local*). Each client, therefore, has access to the federated model and has full control over the prediction process. Server-side inference (*server*) is initiated by a prediction request, which is processed by the aggregation server. Access to the federated model is provided via an API or a web application. Lastly, joint inference (*joint*) requires both clients and server to become active to generate predictions. Hence, the global model does not reside at one entity but is distributed across multiple clients. This design requires consensus across all stakeholders on the usage of the federated model.

Most systems apply local inference, especially in horizontal FL. A simple approach is letting the server distribute the global RF to all authorized recipients after completing training (Kwatra & Torra 2022). Alternatively, each individual client decides which decision trees should be incorporated into their local ensemble. Hereof, Aliyu et al. (2021) let clients publish local models to a blockchain, from where authorized users can download the models for local usage. In synchronous training, participants can keep track of the collaboratively built trees during training (Liu et al. 2020b), skipping the distribution step. Three approaches enable server-side inference, all applying asynchronous training. The global model may not be distributed due to privacy concerns, easier practical use, and monetized prediction processes. Joint inference between clients and server is only implemented in approaches with vertically partitioned data. The aggregation server only holds encrypted or referential information about the model (Liu et al. 2021). In addition to privacy protection, this can be employed to prohibit usage without the consent of all involved clients.

5 Evaluation

Within our study, we found that FRFs are typically applied in cross-silo environments. The nature of the clients eliminates many challenges associated with the cross-device setting prevalent in FL research, such as limited computational resources and unreliability of clients (Kairouz et al. 2021). Hence, cross-silo techniques should not be discussed in terms of cross-device requirements. We, therefore, developed evaluation criteria applicable to the cross-silo setting. As we set out to evaluate the suitability for a broad range of applications in practice, we limited our scope to domain-agnostic criteria. Hence, we settled on prediction robustness and technical feasibility. We excluded other interesting facets (e.g., explainability) since their implications remain largely independent of the applied techniques. The two selected dimensions allow us to draw generalizable conclusions regarding the practical suitability of aggregation techniques.

We focus on the differentiation between synchronous and asynchronous training phases, as our findings hint at the importance of this property. The simple, binary categorization directly applies to all reviewed techniques. Furthermore, it allows us to make assumptions about the data flow during training without losing generalizability of our results. Going forward, we first motivate and describe the developed criteria before presenting the results for synchronous and asynchronous training. An overview of the main findings can be found in Table 3. Importantly, our review did not identify asynchronous techniques within vertical FL. For the remainder of the evaluation, we will thus limit the scope to the horizontal setting.

5.1 Prediction Robustness

In FL, clients independently collect their own training data, which is never shared with other parties. Because data is collected differently across clients, the common assumption of IID data seldom holds in practice (Kairouz et al. 2021). In reality, the sizes of local data sets can differ substantially across clients. Additionally, not all local data sets may follow the same distribution. Robustness against statistical heterogeneity has been identified as a core challenge in FL (Li et al. 2020).

Across our sample, we saw two node expansion strategies for synchronous training. First, the server can directly compute their aggregated quality if all clients locally

Table 3. Main findings of the conducted comparison

Dimension	Synchronous Training	Asynchronous Training
Statistical heterogeneity	No performance loss compared to non-federated baseline	Prediction quality may degrade with imbalanced distributions and dataset sizes
Multi-task capabilities	Requires prior knowledge about the similarity of local datasets	Optimal composition of aggregated model(s) can be determined a posteriori
Communication overhead	May increase considerably with the size and complexity of the system	Highly parallelizable and scalable due to only a single communication phase
Adaptability	Rigid model structure necessitates complete retraining for updates	Client contributions are fully decoupled and hence dynamically modifiable

evaluate the same set of candidate splits (Liu et al. 2022, 2020b). Second, clients may send distribution data summarizing their local sample to the server (Kalloori & Klingler 2022, Guo et al. 2016). The server then computes candidate splits after aggregating the local distributions. In terms of prediction robustness, the second strategy is preferable, since merging local distribution data reveals the structure of the global data set to the server, resulting in an optimal split which is agnostic to size disparities. Shifts in local distributions do not affect the global model either, as it is built on representations of the global distribution. Depending on the granularity of local distribution data, like the histogram bin size (Guo et al. 2016), the resulting model can be equivalent to one trained on the union of local data sets. This is also referred to as "lossless" (Chen et al. 2021) performance.

In asynchronous training, clients are fundamentally decoupled and candidate splits can only be evaluated on locally available data. Each local decision tree thus only incorporates information extracted from one data set. If size imbalances are unaccounted for in the aggregation process, the prediction output may be biased toward clients with larger data sets. Weighting the trees with respect to the size of the training data set can counter this effect, albeit not consistently across different configurations Hauschild et al. (2022). On top of size imbalances, distribution heterogeneity can negatively impact the prediction quality of asynchronously trained models due to the lack of data exchange across client boundaries during training. This cannot be countered easily, as the introduction of inter-client knowledge would go against the asynchronous approach.

The creation of individualized models for all clients can elegantly circumvent the issue of statistical heterogeneity (Li et al. 2020). In this context, we found asynchronous training to be more suitable because each decision tree is based on the local distribution of a single client. The utility of the model can thus be maximized by letting clients decide the composition of the model for their local prediction task (de Souza et al. 2020). In synchronously trained FRFs, each tree is based on contributions from all clients. Multi-task strategies are only practical if clients are known to have similar local distributions prior to training. Quantifying the similarity of local data sets a priori is possible (Liu et al. 2022) but adds further complication and is a potential source for data leakage. Synchronous aggregation techniques are hence less suitable for multi-task FL.

5.2 Technical Feasibility

The practical suitability of aggregation techniques also depends on their technical feasibility. For a start, communication efficiency can be a central bottleneck in FRFs, since the communication delay between clients and server makes up a large share of the total training time in cross-silo FL (Zhang et al. 2020). Furthermore, technical feasibility also entails the ability to adapt to dynamic environments (Ishikawa & Yoshioka 2019). This becomes especially relevant when requirements are insufficiently defined from the start or when recurring changes are anticipated (Ishikawa & Yoshioka 2019). An aggregation technique capable of dynamically reacting and adapting to systemic changes is more versatile and thus more likely to be applied in practice.

Synchronous coordination has a few drawbacks regarding technical feasibility. First, building a decision tree synchronously is communication-intensive. Clients first need to

share their input data with the server, which then distributes the optimal split (Kalloori & Klingler 2022), resulting in at least two communication phases per node. Node expansion within the same tree layer can be parallelized. However, the server cannot expand a node until all clients have provided their data. The slowest client determines the processing speed for each node (Zhang et al. 2020). In asynchronous training, only one large communication phase is necessary to make local models available to the server. Since clients train independently, their contributions are fully decoupled. The communication overhead associated with asynchronous training hence does not grow problematically with the number of clients or the size of local models.

Each synchronously built decision tree is based on multiple local data sets. When new clients are introduced, or incremental updates are necessary, the global RF thus has to be completely retrained to incorporate new information. The cost-benefit relation of maintaining a synchronously trained model becomes less economical the more frequently a system is modified. Asynchronous FRFs on the other hand allow for continuous integration of new clients and facilitate the revocation of inputs. Moreover, they enable individualized updates to decision trees within the global RF, for instance when data is accumulated over time and models need to be continuously retrained. In case future system requirements are not fully understood from the start, the flexibility associated with asynchronous training ensures that the federated system remains adaptable.

Additionally, the impact of privacy preservation techniques on communication efficiency must also be considered. Encryption techniques in particular require a significant amount of computation and can dominate the processing time (Zhang et al. 2020). Since encryption has to be applied before sending local data, the cumulative overhead of encryption grows with the number of transmission phases and the amount of transmitted data. Summed up, the communication overhead for synchronous aggregation techniques is comparably high and can depend on a range of factors. The training complexity and the rigidity of the resulting FL model limit synchronous training to static environments.

6 Discussion

6.1 Principal Findings

In this study, we analyzed extant literature on the aggregation of RFs in FL. We identified 21 studies investigating FRFs. Therein, we found the majority of approaches to apply horizontal FL. Our results suggest that a key feature determining the properties of an aggregation technique is the coordination of the training phase. The choice regarding synchronous or asynchronous training determines the information flow between the involved parties. The behavior of the resulting system depends on whether the decision trees are built on local data of a single client per tree, or in a round-based process involving multiple clients. Thus, we compared synchronous and asynchronous approaches in terms of their prediction robustness and their technical feasibility.

Our results indicate that synchronously trained models can produce superior federated models in the presence of non-IID data, where local data sets are of unequal size or follow different underlying distributions. This is because the resulting global RF is structurally similar to its non-federated counterpart. The impact of size imbalances in

asynchronous training can be mitigated by size-based weighting (Hauschild et al. 2022). Still, we found no other examples of size-based processing techniques. Additionally, the effects of distribution heterogeneity cannot be mitigated in asynchronous training because cross-client information is unavailable. The combined impact of size imbalances and distribution heterogeneity on asynchronous FRFs has not been investigated yet, but the prediction quality may degrade as well. However, building a global model in the presence of statistical heterogeneity may not align with the practical requirements. Individualized models tend to perform better here since they are tailored to the non-IID local distributions (Kairouz et al. 2021). For individualization approaches like multi-task FL, asynchronous training has an advantage due to the decoupled nature of the decision trees. Synchronous techniques are less applicable, as they produce decision trees with tightly interwoven client contributions. This can undermine the utility for individual stakeholders, a crucial factor for the success of a federated system (Huang et al. 2022).

Regarding the technical feasibility, asynchronous training incurs far less communication overhead because the training process for individual clients is largely decoupled. This makes it the preferable option for environments that require scalability and adaptability. The global model allows for targeted, dynamic updates, which hints toward potential applications in incremental FL. Since each tree only depends on the input of a single client, their influence on the final prediction can easily be adjusted. In synchronous training, there may be scenarios where frequent retraining becomes uneconomical due to the complex training process. Additionally, there is a trade-off in synchronous training between performance and a more accurate representation of local data (Guo et al. 2016). Analogously, the cost of privacy-preservation techniques such as encryption comes into play, since the associated overhead increases with the frequency and size of data transmissions.

6.2 Implications

Our work yields several important implications for research and practice. For one, we provide a structured categorization of extant work on the aggregation of RFs in FL. With the rising interest in both RF and FL, their intersection becomes ever more relevant. By synthesizing and highlighting recent efforts in this intersection, we demonstrate the benefits of non-parametric approaches, ultimately contributing to the transition from inferior local models to federated systems. With the categorization of existing approaches, our work sets the stage for gaining deeper insights into the structural relationships between aggregation techniques. Our results show that the choice of synchronous or asynchronous training is an important driver of the robustness and applicability of FRFs. With this distinction, we complement the standard comparison from a data distribution perspective (horizontal vs. vertical) with a training coordination perspective (synchronous vs. asynchronous). This distinction allows for novel, model-agnostic insights into FL systems. Finally, the investigation into the practical suitability of aggregation techniques enables more knowledgeable design choices. Concretely, asynchronous coordination of the training phase tends to be more feasible in practical applications. This is due to its simplicity, superior efficiency, and the decoupled nature of client contributions. Nevertheless, the ability of synchronous FRF implementations to produce a potentially

lossless global model may be a decisive trait under certain circumstances. Complex decisions often hinder the instantiation of ML systems. Our research facilitates the decision-making process and makes an informed application of FRFs more attainable. Our results are applicable across many disciplines.

6.3 Limitations and Future Research

FRFs are used across many domains, making it likely that we did not capture all relevant literature. However, our search covered a large share of extant work. We assumed cross-silo FL in the evaluation, since this is what most reviewed approaches focused on. Yet, some techniques can be applied in cross-device FL as well (Aliyu et al. 2021). Furthermore, our review found no aggregation techniques within vertical FL implementing asynchronous training. Since we focused on comparing synchronous and asynchronous coordination, we limited the evaluation to horizontal FL. Whether asynchronous training could be applied successfully in scenarios with vertically partitioned data is an open question. The evaluation speaks toward the practical applicability of techniques with asynchronous training. Our considerations for the prediction robustness in the presence of statistical heterogeneity are less conclusive and depend on assumptions about the underlying data. A quantitative investigation into the effects of statistical heterogeneity on synchronous and asynchronous training is the logical next step.

7 Conclusion

RFs are a popular choice for ML tasks because they are simple to use and reliably produce good results. Their application in FL environments, however, has thus far received limited attention. We conducted a comprehensive review on FRFs and synthesized aggregation techniques. Furthermore, we evaluated the identified aggregation techniques based on previously developed criteria to expose differences. Our results show that extant research generally assumes the cross-silo setting. As for the partitioning of local data, vertical FRFs have been explored less than horizontal FRFs. The most influential design choice for FRFs is the coordination of the model training phase since synchronous or asynchronous training have implications for the robustness of the resulting system against statistical heterogeneity and its applicability in dynamic, uncertain environments. We present implications for both future research on FL protocols and the practical application of FRFs. Future research can derive recommendations for suitable aggregation techniques from our findings. More research is necessary to better understand the characteristics of federated data sets and their implications on design choices for cross-silo FL. Going forward, we encourage further research into the decision for synchronous or asynchronous training, specifically in the form of quantitative studies.

References

Aliyu, I., Feliciano, M. C., van Engelenburg, S., Kim, D. O. & Lim, C. G. (2021), 'A blockchain-based federated forest for sdn-enabled in-vehicle network intrusion detection system', *IEEE Access* **9**, 102593–102608.

Arksey, H. & O'Malley, L. (2005), 'Scoping studies: towards a methodological framework', *International Journal of Social Research Methodology* **8**(1), 19–32.

Bell, A., Solano-Kamaiko, I., Nov, O. & Stoyanovich, J. (2022), It's just not that simple: An empirical study of the accuracy-explainability trade-off in machine learning for public policy, *in* '2022 ACM Conference on Fairness, Accountability, and Transparency', FAccT '22, Association for Computing Machinery, New York, NY, USA, p. 248–266.

Biau, G. & Scornet, E. (2016), 'A random forest guided tour', *Test* **25**, 197–227.

Breiman, L. (1996), 'Bagging predictors', *Machine Learning* **24**(2), 123–140.

Chen, X., Zhou, S., Guan, B., Yang, K., Fao, H., Wang, H. & Wang, Y. (2021), Fed-eini: An efficient and interpretable inference framework for decision tree ensembles in vertical federated learning, *in* '2021 IEEE International Conference on Big Data (Big Data)', IEEE, pp. 1242–1248.

Cutler, A., Cutler, D. R. & Stevens, J. R. (2012), 'Random forests', *Ensemble machine learning: Methods and applications* pp. 157–175.

Damgård, I., Pastro, V., Smart, N. & Zakarias, S. (2012), Multiparty computation from somewhat homomorphic encryption, *in* R. Safavi-Naini, ed., 'Advances in cryptology - CRYPTO 2012', Lecture Notes in Computer Science, Springer, Berlin and Heidelberg, pp. 643–662.

De Fazio, R., Di Giovannantonio, R., Bellini, E. & Marrone, S. (2023), 'Explainabilty comparison between random forests and neural networks—case study of amino acid volume prediction', *Information* **14**(1), 21.

de Souza, L. A. C., Antonio F. Rebello, G., Camilo, G. F., Guimaraes, L. C. B. & Duarte, O. C. M. B. (2020), Dfedforest: Decentralized federated forest, *in* '2020 IEEE International Conference on Blockchain (Blockchain)', IEEE Computer Society, pp. 90–97.

Ge, N., Li, G., Zhang, L. & Liu, Y. (2022), 'Failure prediction in production line based on federated learning: an empirical study', *Journal of Intelligent Manufacturing* **33**(8), 2277–2294.

Gencturk, M., Sinaci, A. A. & Cicekli, N. K. (2022), 'Bofrf: A novel boosting-based federated random forest algorithm on horizontally partitioned data', *IEEE Access* **10**, 89835–89851.

Giacomelli, I., Jha, S., Kleiman, R., Page, D. & Yoon, K. (2019), 'Privacy-preserving collaborative prediction using random forests', *AMIA summits on translational science proceedings* **2019**, 248.

Guo, T., Kutzkov, K., Ahmed, M., Calbimonte, J.-P. & Aberer, K. (2016), Efficient distributed decision trees for robust regression, *in* 'Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2016, Riva del Garda, Italy, September 19-23, 2016, Proceedings, Part II 16', Springer, pp. 79–95.

Hauschild, A.-C., Lemanczyk, M., Matschinske, J., Frisch, T., Zolotareva, O., Holzinger, A., Baumbach, J. & Heider, D. (2022), 'Federated random forests can improve local performance of predictive models for various health care applications', *Bioinformatics* **38**(8), 2278–2286.

Hou, J., Su, M., Fu, A. & Yu, Y. (2021), 'Verifiable privacy-preserving scheme based on vertical federated random forest', *IEEE Internet of Things Journal* **9**(22), 22158–22172.

Huang, C., Huang, J. & Liu, X. (2022), 'Cross-silo federated learning: Challenges and opportunities', *arXiv preprint* .

Ishikawa, F. & Yoshioka, N. (2019), How do engineers perceive difficulties in engineering of machine-learning systems? - questionnaire survey, *in* '2019 IEEE/ACM Joint 7th International Workshop on Conducting Empirical Studies in Industry (CESI) and 6th International Workshop on Software Engineering Research and Industrial Practice (SER&IP)', IEEE, pp. 2–9.

Kairouz, P., McMahan, H. B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A. N., Bonawitz, K., Charles, Z., Cormode, G., Cummings, R. et al. (2021), 'Advances and open problems in federated learning', *Foundations and Trends in Machine Learning* **14**(1–2), 1–210.

Kalloori, S. & Klingler, S. (2022), Cross-silo federated learning based decision trees, *in* J. Hong, M. Bures, J. W. Park & T. Cerny, eds, 'Proceedings of the 37th ACM/SIGAPP Symposium on Applied Computing', ACM, New York, NY, USA, pp. 1117–1124.

Kwatra, S. & Torra, V. (2021), A survey on tree aggregation, *in* '2021 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)', IEEE, pp. 1–6.

Kwatra, S. & Torra, V. (2022), A k-anonymised federated learning framework with decision trees, *in* 'Data Privacy Management, Cryptocurrencies and Blockchain Technology: ESORICS 2021 International Workshops, DPM 2021 and CBT 2021, Darmstadt, Germany, October 8, 2021, Revised Selected Papers', Springer, pp. 106–120.

Li, Q., Wen, Z., Wu, Z., Hu, S., Wang, N., Li, Y., Liu, X. & He, B. (2021), 'A survey on federated learning systems: Vision, hype and reality for data privacy and protection', *IEEE Transactions on Knowledge & Data Engineering* (01), 1–1.

Li, T., Sahu, A. K., Talwalkar, A. & Smith, V. (2020), 'Federated learning: Challenges, methods, and future directions', *IEEE Signal Processing Magazine* **37**(3), 50–60.

Liu, B., Tan, C., Wang, J., Zeng, T., Shan, H., Yao, H., Huang, H., Dai, P., Bo, L. & Chen, Y. (2021b), 'Fedlearn-algo: A flexible open-source privacy-preserving machine learning platform', *arXiv preprint* .

Liu, S., Wang, J. & Zhang, W. (2022), 'Federated personalized random forest for human activity recognition', *Mathematical biosciences and engineering : MBE* **19**(1), 953–971.

Liu, Y., Chen, M., Zhang, W., Zhang, J. & Zheng, Y. (2020b), 'Federated extra-trees with privacy preserving', *arXiv preprint* .

Liu, Y., Ma, Z., Yang, Y., Liu, X., Ma, J. & Ren, K. (2021), 'Revfrf: Enabling cross-domain random forest training with revocable federated learning', *IEEE Transactions on Dependable and Secure Computing* **19**(6), 3671–3685.

Ma, X., Liao, L. X., Li, Z. & Chao, H.-C. (2022), 'Asynchronous federated learning for elephant flow detection in software defined networking systems', *Journal of Physics: Conference Series* **2216**(1), 012085.

Ma, Z., Ma, J., Miao, Y., Liu, X., Choo, K.-K. R. & Deng, R. H. (2021), 'Pocket diagnosis: Secure federated learning against poisoning attack in the cloud', *IEEE Transactions on Services Computing* **15**(6), 3429–3442.

Markovic, T., Leon, M., Buffoni, D. & Punnekkat, S. (2022), Random forest based on federated learning for intrusion detection, *in* 'Artificial Intelligence Applications and Innovations: 18th IFIP WG 12.5 International Conference, AIAI 2022, Hersonissos, Crete, Greece, June 17–20, 2022, Proceedings, Part I', Springer, pp. 132–144.

Mays, N., Roberts, E. & Popay, J. (2001), 'Synthesising research evidence'.

McMahan, B., Moore, E., Ramage, D., Hampson, S. & Aguera y Arcas, B. (2017), 'Communication-efficient learning of deep networks from decentralized data', *Artificial Intelligence and Statistics* pp. 1273–1282.

Paleyes, A., Urma, R.-G. & Lawrence, N. D. (2022), 'Challenges in deploying machine learning: a survey of case studies', *ACM Computing Surveys* **55**(6), 1–29.

Pan, S. J. & Yang, Q. (2009), 'A survey on transfer learning', *IEEE Transactions on knowledge and data engineering* **22**(10), 1345–1359.

Rieke, N., Hancox, J., Li, W., Milletarì, F., Roth, H. R., Albarqouni, S., Bakas, S., Galtier, M. N., Landman, B. A., Maier-Hein, K., Ourselin, S., Sheller, M., Summers, R. M., Trask, A., Xu, D., Baust, M. & Cardoso, M. J. (2020), 'The future of digital health with federated learning', *NPJ digital medicine* **3**, 119.

Saha, S. & Ahmad, T. (2021), 'Federated transfer learning: Concept and applications', *Intelligenza Artificiale* **15**(1), 35–44.

Varghese, J. (2020), 'Artificial intelligence in medicine: chances and challenges for wide clinical adoption', *Visceral medicine* **36**(6), 443–449.

Wu, Y., Cai, S., Xiao, X., Chen, G. & Ooi, B. C. (2020), 'Privacy preserving vertical federated learning for tree-based models', *Proceedings of the VLDB Endowment* **13**(12), 2090–2103.

Yang, Q., Liu, Y., Chen, T. & Tong, Y. (2019), 'Federated machine learning', *ACM Transactions on Intelligent Systems and Technology* **10**(2), 1–19.

Yao, H., Wang, J., Dai, P., Bo, L. & Chen, Y. (2022), 'An efficient and robust system for vertically federated random forest', *arXiv preprint* .

Zhang, C., Li, S., Xia, J., Wang, W., Yan, F. & Liu, Y. (2020), Batchcrypt: Efficient homomorphic encryption for cross-silo federated learning, *in* '2020 USENIX annual technical conference (USENIX ATC 20)', pp. 493–506.

Zhang, H., Bosch, J., Olsson, H. H. & Koppisetty, A. C. (2021), Af-dndf: Asynchronous federated learning of deep neural decision forests, *in* '2021 47th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)', IEEE, pp. 308–315.