# BeyondFederated: truly decentralised learning at the edge

Quinten van Eijs

## 1 Problem Statement

In the context of a peer-to-peer system where users have their own music datasets, the task of searching for music in a decentralized manner poses several challenges. Privacy concerns arise in conventional distributed machine learning systems due to the communication between individual nodes and the centralized server during model updates, which compromises security and privacy. To address this issue, federated learning systems have emerged as a promising alternative. These systems update nodes within random cohorts based on simple update rules, eliminating the need for a central aggregator. However, existing systems in this domain have remained largely theoretical and have not fully capitalized on the structures of local updates with a learned peer-to-peer update rule. To bridge this gap, the objective of this work is to develop a robust and efficient peer-to-peer federated learning framework specifically tailored for a decentralized music search engine.

Developing a music search engine within a peer-to-peer (P2P) network presents significant challenges due to the limited availability of peers, lack of trust, and dynamic identities of peers. These factors add complexity to the task of building an efficient and reliable music search engine within a decentralized environment. The limited availability of peers results in incomplete search coverage and unreliable access to music files, compromising the effectiveness of the search engine. Furthermore, the absence of a central authority in P2P networks raises concerns about the authenticity and accuracy of shared information, undermining trust and compromising the reliability of search results. Moreover, the dynamic nature of peer identities makes it difficult to establish persistent connections and maintain accurate information about peer availability. Current federated learning approaches, which rely on a central server, are not well-suited to address these challenges in P2P networks as they contradict the principles of decentralization and introduce a single point of failure. To overcome these obstacles, alternative approaches that operate in a fully decentralized manner need to be explored in order to develop an effective and reliable music search engine within a P2P network.

# 2 Related Work

## 2.1 Papaya: Federated Learning, but Fully Decentralized

The authors propose a system architecture for their peer-to-peer learning system that is similar to BitTorrent. The system architecture consists of two main components: a tracker server and clients.

Any entity can start a peer-to-peer learning session by initializing a tracker server and storing its address in a torrent file. This torrent file is then distributed among many clients using a web server or some other method. Clients who take part in the peer-to-peer learning system would download the torrent file and update the tracker.

The tracker server maintains a list of all the clients that are currently participating in the peer-to-peer learning session. Clients can communicate with each other directly to share data and update their models. The authors note that this system architecture is scalable and can handle a large number of clients.

**notes** periodic sharing of parameters between random subset of nodes according to a learned trust matrix (social learning using DeGroot's model). **critique** Still highly experimental no actual implementation. Privacy concerns: maintains a list of all the clients. Fully Decentralized(?): But requires a tracker server.

## 2.2 DeceFL: a principled fully decentralized federated learning framework

The proposed DeceFL algorithm is a decentralized federated learning algorithm that allows clients to collaboratively train a machine learning model without sharing their private data. It aims to minimize the performance gap between a centralized model and a decentralized model.

An high-level explanation of the DeceFL algorithm:

- Communication Network Modeling: The algorithm models the communication network between clients as an undirected connected graph G = (N, E, W), where N represents the set of clients, E represents the set of communication channels, and W represents the weights associated with the communication channels.
- Initialization: Each client initializes its local model M and its local dataset D.
- Model Training and Communication: The algorithm iteratively performs the following steps: Local Model Training: Each client trains its local model M on its own local dataset D using its preferred learning algorithm. Model Update Exchange: Clients exchange their model updates with neighboring clients in the communication network. The model updates are aggregated at each client to obtain an updated global model. Model Update Incorporation: Each client incorporates the updated global model into its local model.

- Convergence and Performance Evaluation: The algorithm continues the iterations until a convergence criterion is met, such as a maximum number of iterations or a small enough change in the global model. The performance of the decentralized model is evaluated by comparing it with a centralized model using the global objective function.

The paper provides a detailed analysis of the convergence properties of the DeceFL algorithm, showing that it guarantees convergence and has a similar convergence rate as the centralized federated learning algorithm. It also demonstrates the performance of DeceFL through experiments and benchmarks against other federated learning frameworks.

Overall, the DeceFL algorithm enables clients to collaboratively train a model while preserving privacy and achieving comparable performance to centralized federated learning algorithms.

**critique** Still Experimental with some code base, the paper suggests that privacy algorithms such as blockchain or homomorphic encryption can be applied to the DeceFL framework to enhance data privacy protection and secure communication. While the paper acknowledges that these techniques have not been considered in the study. Future work lies in the integration of such techniques to the proposed DeceFL algorithm to make the global objective function unknown to clients. Mainly focussing on decentralized model updates but not within a peer-2-peer enviroment.

## 2.3 Fully Decentralized Joint Learning of Personalized Models and Collaboration Graphs

The proposed approach differs from traditional methods in that it allows for fully decentralized learning without the need for a central server to aggregate updates. Instead, the approach leverages a similarity graph to collaboratively learn personalized models for each user. The graph describes the relationships between user personal tasks, and it is learned jointly with the models. The algorithm alternates between updating the models and updating the graph until convergence.

The personalized models are learned in a fully decentralized manner, with each user only communicating with a small number of peers. This approach allows for the models to be adapted to each user's distinct behaviors/preferences while still benefiting from sharing information with similar peers. The authors demonstrate the effectiveness of their approach on several real-world datasets, including image classification and sentiment analysis tasks.

Overall, the proposed approach offers a promising direction for scalable and robust machine learning in distributed environments. The approach is more scalable and does not have a single point of failure or communication bottlenecks. The authors suggest that future work could explore the use of more sophisticated similarity graphs and the integration of privacy-preserving techniques.

**critique** Not focussing on federated learning instead: Decentralized collaborative learning using a collaboration graph. Lack of privacy considerations: The paper does not extensively address privacy concerns associated with fully decentralized learning. As user data is shared and utilized in a collaborative manner, privacy-preserving techniques should be considered to protect sensitive information.

### 2.4 Swarm Learning for decentralized and confidential clinical machine learning

The Concept of Swarm Learning is a decentralized approach to machine learning that allows for individual nodes to train a common machine learning model collaboratively without sharing the training data. This is achieved by individual nodes sharing parameters (weights) derived from training the model on the local data. The Swarm Learning framework builds on two proven technologies, distributed machine learning and blockchain, and is designed to make it possible for a set of nodes, each possessing some training data locally, to train a common machine learning model collaboratively without sharing the training data. Swarm Learning provides security measures to support data sovereignty, security, and confidentiality. It is an alternative to cloud computing and federated computing approaches, which have disadvantages such as data duplication, increased data traffic, and challenges for data privacy and security.

**critique** Requires Private Permisonned Blockchain Network, Does not only require Swarm Learning Nodes which could be implemented as potential clients but still requires Swarm Network (SN) nodes. raises the questions if this would be applicable within a fully peer-2-peer application?

## 3 High-Level Overview of Components

The system harnesses the collective knowledge and data available on multiple devices to enhance the search capabilities and provide personalized search for songs stored on local devices.

- Local Song Indexing: Each device maintains a local song database that indexes the songs available on that device. This indexing includes song metadata, file locations, and potentially some audio features.

  Create Datset using the FMA: A Dataset For Music Analysis, and scraped pandacd items.

  https://github.com/mdeff/fma

  Create text Searcher model based on ScaNN(Scalable Nearest Neighbors)

  https://www.tensorflow.org/lite/models/modify/model_maker/text_searcher

https://github.com/google-research/google-research/tree/master/scann

- Model Training: Devices perform local training on their indexed songs using their local models. This involves employing content-based or collaborative filtering techniques to learn patterns and preferences from the local data.

- Model Exchange and Aggregation: Devices periodically exchange model updates with neighboring devices, incorporating their knowledge and aggregating it with their own models. This process enables collaborative learning and enhances the overall song search and recommendation capabilities.

- Song Search and Recommendations: When a user initiates a song search, their local device leverages the collective knowledge of the federated models across the network. The device utilizes its trained model to generate personalized recommendations or perform search queries on local song databases based on user preferences or search criteria.