



Evaluation of Hyperparameter-Optimization Approaches in an Industrial Federated Learning System

Stephanie Holly^{*†}, Thomas Hiessl^{*}, Safoura Rezapour Lakani^{*},
Daniel Schall^{*}, Clemens Heitzinger[†] and Jana Kemnitz^{*}

^{*}Siemens Technology, 1210 Vienna, Austria

[†]TU Wien, 1040 Vienna, Austria

{e11703485@student, clemens.heitzinger}@tuwien.ac.at,

{hiessl.thomas, safoura.rezapour_lakani, daniel.schall, jana.kemnitz}@siemens.com

Abstract—Federated Learning (FL) decouples model training from the need for direct access to the data and allows organizations to collaborate with industry partners to reach a satisfying level of performance without sharing vulnerable business information. The performance of a machine learning algorithm is highly sensitive to the choice of its hyperparameters. In an FL setting, hyperparameter optimization poses new challenges. In this work, we investigated the impact of different hyperparameter optimization approaches in an FL system. In an effort to reduce communication costs, a critical bottleneck in FL, we investigated a local hyperparameter optimization approach that – in contrast to a global hyperparameter optimization approach – allows every client to have its own hyperparameter configuration. We implemented these approaches based on grid search and Bayesian optimization and evaluated the algorithms on the MNIST data set using an i.i.d. partition and on an Internet of Things (IoT) sensor based industrial data set using a non-i.i.d. partition.

Index Terms—Industrial federated learning, Optimization approaches, Hyperparameter optimization

I. INTRODUCTION

The performance of a machine learning algorithm is highly sensitive to the choice of its hyperparameters. Therefore, hyperparameter selection is a crucial task in the optimization of knowledge-aggregation algorithms. Federated Learning (FL) is a recent machine learning approach which aggregates machine learning model parameters between devices (henceforth clients) without sharing their data. The aggregation is coordinated by a server. Industrial Federated Learning (IFL) is a modified approach of FL in an industrial context [1]. In an FL setting, hyperparameter optimization poses new challenges and is a major open research area [2]. In this work, we investigate the impact of different hyperparameter optimization approaches in an IFL system. We believe that the data distribution influences the choice of the best hyperparameter configuration and suggest that the best hyperparameter configuration for a client might differ from another client based on individual data properties. Therefore, we investigate a local hyperparameter optimization approach that – in contrast to a global hyperparameter optimization approach – allows every client to have its own hyperparameter configuration. The local

approach allows us to optimize hyperparameters prior to the federation process reducing communication costs.

Communication is considered a critical bottleneck in FL [3]. Clients are usually limited in terms of communication bandwidth enhancing the importance of reducing the number of communication rounds or using compressed communication schemes for the model updates to the central server [3]. Dai et al. [4] introduced *Federated Bayesian Optimization* (FBO) extending Bayesian optimization to the FL setting. However, until now, there is no research on the impact of global and local hyperparameter optimization in FL. Therefore, we compare a local hyperparameter optimization approach to a global hyperparameter optimization approach, optimizing hyperparameters in the federation process.

The aim of this work is to i) analyze challenges and formal requirements in FL, and in particular in IFL, ii) to evaluate the performance of an Internet of Things (IoT) sensor based classification task in an IFL system, iii) to investigate a communication efficient hyperparameter optimization approach, and iv) to compare different hyperparameter optimization algorithms. Therefore, we want to answer the following questions.

- Q1: Does FL work for an IoT sensor based anomaly classification task on industrial assets with non-identically distributed data in an IFL system with a cohort strategy?
- Q2: Can we assume that the global and local hyperparameter optimization approach deliver the same hyperparameter configuration in an i.i.d. FL setting?
- Q3: Can we reduce communication costs in the hyperparameter optimization of a non-i.i.d. classification task in context of FL by optimizing a hyperparameter locally prior to the federation process?
- Q4: Does Bayesian optimization outperform grid search, both in a global and local approach of a non-i.i.d. IoT sensor based classification task?

II. ALGORITHMIC CHALLENGES AND FORMAL REQUIREMENTS FOR INDUSTRIAL ASSETS

In FL, new algorithmic challenges arise that differentiate the corresponding optimization problem from a distributed

optimization problem. In distributed learning settings, major assumptions regarding the training data are made which usually fail to hold in an FL setting [5]. Moreover, non-i.i.d. data, limited communication, and limited and unreliable client availability pose further challenges for optimization problems in FL [2]. Kairouz et al. [2] considered the need for addressing these challenges as a major difference to distributed optimization problems. The optimization problem in FL is therefore referred to as federated optimization emphasizing the difference to distributed optimization [5]. In an IFL setting, additional challenges regarding industrial aspects arise [1]. In this section, we want to formulate the federated optimization problem and discuss the algorithmic challenges of FL in general, and in particular of IFL.

A. Problem Formulation

We consider a supervised learning task with features x in a sample space \mathcal{X} and labels y in a label space \mathcal{Y} . We assume that we have K available clients, $K \in \mathbb{N}_{\geq 2}$, with

$$D_k := D_{\mathcal{X},k} \times D_{\mathcal{Y},k} \subseteq \mathcal{X} \times \mathcal{Y}$$

denoting the data set of client k and $n_k := |D_k|$ denoting the cardinality of the client's data set. Let \mathcal{Q} denote the distribution over all clients, and let \mathcal{P}_k denote the data distribution of client k . We can then access a specific data point by first sampling a client $k \sim \mathcal{Q}$ and then sampling a data point $(x, y) \sim \mathcal{P}_k$ [2]. Then, the local objective function is

$$F_k(w) := \mathbb{E}_{(x,y) \sim \mathcal{P}_k} [f(x, y, w)], \quad (1)$$

where $w \in \mathbb{R}^d$ represents the parameters of the machine learning model and $f(x, y, w)$ represents the loss of the prediction on sample (x, y) for the given parameters w . Typically, we wish to minimize

$$F(w) := \frac{1}{K} \sum_{k=1}^K F_k(w). \quad (2)$$

B. Federated Learning

One of the major challenges concerns data heterogeneity. In general, we cannot assume that the data is identically distributed over the clients, that is $\mathcal{P}_k = \mathcal{P}_l$ for all k and l . Therefore, F_k might be an arbitrarily bad approximation of F [5].

In the following, we want to analyze different non-identically distributed settings as demonstrated by Hsieh et al. [6] assuming that we have an IoT sensor based anomaly classification task in an industrial context. Given the distribution \mathcal{P}_k , let $P_{\mathcal{X},\mathcal{Y}}^k$ denote the bivariate probability function, let $P_{\mathcal{X}}^k$ and $P_{\mathcal{Y}}^k$ denote the marginal probability function respectively. Using the conditional probability function $P_{\mathcal{Y}|\mathcal{X}}^k$ and $P_{\mathcal{X}|\mathcal{Y}}^k$, we can now rewrite the bivariate probability function as

$$P_{\mathcal{X},\mathcal{Y}}^k(x, y) = P_{\mathcal{Y}|\mathcal{X}}^k(y|x)P_{\mathcal{X}}^k(x) = P_{\mathcal{X}|\mathcal{Y}}^k(x|y)P_{\mathcal{Y}}^k(y) \quad (3)$$

for $(x, y) \in \mathcal{X} \times \mathcal{Y}$. This allows us to characterize different settings of non-identically distributed data:

Feature distribution skew: We assume that $P_{\mathcal{Y}|\mathcal{X}}^k = P_{\mathcal{Y}|\mathcal{X}}^l$ for all k, l , but $P_{\mathcal{X}}^k \neq P_{\mathcal{X}}^l$ for some k, l . Clients that have the same anomaly classes might still have differences in the measurements due to variations in sensor and machine type.

Label distribution skew: We assume that $P_{\mathcal{X}|\mathcal{Y}}^k = P_{\mathcal{X}|\mathcal{Y}}^l$ for all k, l , but $P_{\mathcal{Y}}^k \neq P_{\mathcal{Y}}^l$ for some k, l . The distribution of labels might vary across clients as clients might experience different anomaly classes.

Same label, different features: We assume that $P_{\mathcal{Y}}^k = P_{\mathcal{Y}}^l$ for all k, l , but $P_{\mathcal{X}|\mathcal{Y}}^k \neq P_{\mathcal{X}|\mathcal{Y}}^l$ for some k, l . The same anomaly class can have significantly different features for different clients due to variations in machine type, operational- and environmental conditions.

Same features, different label: We assume that $P_{\mathcal{X}}^k = P_{\mathcal{X}}^l$ for all k and l , but $P_{\mathcal{Y}|\mathcal{X}}^k \neq P_{\mathcal{Y}|\mathcal{X}}^l$ for some k, l . The same features can have different labels due to operational- and environmental conditions, variation in manufacturing, maintenance et cetera.

Quantity skew: We cannot assume that different clients hold the same amount of data, that is $n_k = n_l$ for all k, l . Some clients will generate more data than others.

In real-world problems, we expect to find a mixture of these non-identically distributed settings. In FL, heterogeneity does not exclusively refer to a non-identical data distribution, but also addresses violations of independence assumptions on the distribution \mathcal{Q} [2]. Due to limited, slow and unreliable communication on a client, the availability of a client is not guaranteed for all communication rounds. Communication is considered a critical bottleneck in FL [3]. In each communication round, the participating clients send a full model update w back to the central server for aggregation. In a typical FL setting, however, the clients are usually limited in terms of communication bandwidth [3]. Consequently, it is crucial to minimize the communication costs.

C. Industrial Federated Learning

In an industrial setting, FL experiences challenges that specifically occur in an industrial context. Industrial assets have access to a wealth of data suitable for machine learning models, however, the data on an individual asset is typically limited and private in nature. In addition to sharing the model within the company, it can also be shared with an external industry partner [1]. FL leaves possibly critical business information distributed on the individual client (or within the company). However, Zhao et al. [7] proved that heterogeneity, in particular, a highly skewed label distribution, significantly reduces the accuracy of the aggregated model in FL. In an industrial context, we expect to find heterogeneous clients due to varying environmental and operational conditions on different assets. Therefore, Hiessl et al. [1] introduced a modified approach of FL in an industrial context and termed it *Industrial Federated Learning* (IFL). IFL does not allow arbitrary knowledge exchange between clients. Instead, the knowledge exchange only takes place between clients that have sufficiently similar data. Hiessl et al. [1] refer to this set of clients as a *cohort*. We expect the federated learning

approach in a cohort to approximate the corresponding central learning approach.

III. HYPERPARAMETER OPTIMIZATION APPROACHES IN AN IFL SYSTEM

In an FL setting, hyperparameter optimization poses new challenges and is a major open research area [2]. The performance of a machine learning model is linked to the amount of communication [8]. In an effort to reduce communication costs, a critical bottleneck in FL [3], we investigated a communication efficient hyperparameter optimization approach, a local hyperparameter optimization approach that allows us to optimize hyperparameters prior to the federation process. Kairouz et al. [2] introduced the idea of a separate optimization of hyperparameters and suggest a different hyperparameter choice for dealing with non-i.i.d. data.

Dai et al. [4] investigated a communication efficient local hyperparameter optimization approach and introduced Federated Bayesian Optimization (FBO) extending Bayesian optimization to the FL setting. In FBO, every client locally uses Bayesian optimization to find the optimal hyperparameter configuration. Additionally, each client is allowed to request for information from other clients. Dai et al. [4] proved a convergence guarantee for this algorithm and its robustness against heterogeneity. However, until now, there is no research on the impact of global and local hyperparameter optimization.

In the LocalHPO algorithm 1, we perform local hyperparameter optimization. We optimize the hyperparameter configuration λ^k for each client k . In the GlobalHPO algorithm 2, we perform global hyperparameter optimization. We optimize the hyperparameter configuration λ in the federation process. The LocalOptimization method in the LocalHPO algorithm 1 and the GlobalOptimization method in the GlobalHPO algorithm 2 can be based on any hyperparameter optimization algorithm.

Algorithm 1: LocalHPO

Server executes:

```

initialize  $w_0$ 
for each client  $k = 1, \dots, K$  do
  |  $\lambda^k := \text{LocalOptimization}(k, w_0)$ 
end
return  $(\lambda^k)_{k=1}^K$ 

```

Algorithm 2: GlobalHPO

Server executes:

```

 $\lambda := \text{GlobalOptimization}()$ 
return  $\lambda$ 

```

We want to differentiate between a global hyperparameter λ_i whose value is constant for all clients and a local hyperparameter λ_i^k whose value depends on a client k . Here, λ_i^k denotes the hyperparameter λ_i on client k . We notice that this differentiation is only relevant for settings with non-i.i.d.

data. In an i.i.d. setting, we assume that a hyperparameter configuration that works for one client also works for another client. In our experiments, we verified this assumption for a proxy data set.

IV. DATA, ALGORITHMS AND EXPERIMENTS

In the next section, we want to make our benchmark design explicit and present our experimental setup. We will present the machine learning tasks including the data partition of the training data, the machine learning models, the optimization algorithms and our experiments. We considered an image classification task on a data set, the MNIST data set of handwritten digits, and an IoT sensor based anomaly classification task on industrial assets.

A. Data

In order to test the IFL system on the MNIST data set, we still need to specify on how to distribute the data over artificially designed clients. To systematically evaluate the effectiveness of the IFL system, we simulated an i.i.d. data distribution. This refers to shuffling the data and partitioning the data into 10 clients, each receiving 6000 examples. Following the approach of McMahan et al. [5], we applied a convolutional neural network with the following settings: 2 convolutional layers with 32 and 64 filters of size 5×5 and a ReLu activation function, each followed by a max pooling layer of size 2×2 , a dense layer with 512 neurons and a ReLu activation function, a dense layer with 10 neurons and a softmax activation function.

The industrial task concerns IoT sensor based anomaly classification on industrial assets. The data was acquired with the SITRANS multi sensor, specifically developed for industrial applications and its requirements [9]. We considered multiple centrifugal pumps with sensors placed at different positions, in different directions to record three axis vibrational data in a frequency of 6644 Hz. Per minute, 512 samples were collected. We operated the pumps under 6 varying conditions, including 3 healthy states and 3 anomalous states. A client is either assigned data of an asset in a measurement, or data of several assets in a measurement ensuring that each client sees all operating conditions. However, since in the process of measurement, the assets were completely dismantled and rebuilt, we consider the data to be non-i.i.d. regarding its feature distribution. We applied an artificial neural network with the following settings: a dense layer with 64 neurons and a ReLu activation function, a dropout layer with a dropout rate of 0.4, a dense layer with 6 neurons and a ReLu activation function, a dropout layer with a dropout rate of 0.4, and a softmax activation function. We remapped the features using the Kabsch algorithm [10], applied a sliding window, extracted the Melfrequency cepstral coefficients, applied the synthetic minority oversampling technique [10], and normalized the resulting features.

B. Algorithms

Our evaluations include the Federated Averaging (FedAvg) algorithm according to McMahan et al. [5], and the

hyperparameter optimization approaches LocalHPO 1 and GlobalHPO 2. We implemented these approaches based on grid search and Bayesian optimization. In this section, we give their pseudocode. We searched for the learning rate η with fixed fraction of participating clients C , number of communication rounds R , number of local epochs E , and mini-batch size B .

In algorithm 3, we give the pseudocode of the LocalOptimization method in LocalHPO 1 based on the grid search algorithm with a fixed grid G . We iterate through the grid G , train the model on the training data of client k based on the ClientUpdate method used in the FedAvg algorithm [5] with the learning rate η as an additional argument, and validate the performance of the model w_η on the validation data $\mathcal{D}_{\text{valid}}^k$ of client k . Finally, the learning rate that yields the highest accuracy A_η on the validation data is selected. Here, w_η denotes the resulting model trained on the training data with learning rate η and $A(\mathcal{D}_{\text{valid}}^k, w_\eta)$ denotes the accuracy of the model tested on the validation data $\mathcal{D}_{\text{valid}}^k$ of client k .

Algorithm 3: Local Grid Search

```

LocalOptimization( $k, w_0$ ):
for each learning rate  $\eta \in G$  do
   $w_\eta := \text{ClientUpdate}(k, w_0, \eta)$ 
   $A_\eta := A(\mathcal{D}_{\text{valid}}^k, w_\eta)$ 
end
 $\eta_k^* := \arg \max_{\eta \in G} A_\eta$ 
return  $\eta_k^*$ 

```

Algorithm 4: Global Grid Search

```

GlobalOptimization():
for each learning rate  $\eta \in G$  do
   $w_\eta := \text{FederatedAveraging}(\eta)$ 
  for each client  $k = 1, \dots, K$  do
     $A_\eta^k := A(\mathcal{D}_{\text{valid}}^k, w_\eta)$ 
  end
   $A_\eta := \frac{1}{K} \sum_{k=1}^K A_\eta^k$ 
end
 $\eta^* := \arg \max_{\eta \in G} A_\eta$ 
return  $\eta^*$ 

```

In algorithm 4, we give the pseudocode of the GlobalOptimization method in GlobalHPO 2 based on the grid search algorithm with a fixed grid G . We iterate through the grid, perform the FedAvg algorithm [5] with the learning rate η as an additional argument, validate the performance of the model w_η on the validation data $\mathcal{D}_{\text{valid}}^k$ for all clients k and compute the average accuracy of all clients. Finally, the learning rate that yields the highest average accuracy A_η is selected.

In algorithm 5, we give the pseudocode of the LocalOptimization method in LocalHPO 1 based on Bayesian optimization. The objective function f takes the learning rate η as an argument, trains the model on the training data of client k based on the ClientUpdate method used in the FedAvg algorithm [5] with the learning rate η as an additional

argument, validates the performance of the model w on the validation data $\mathcal{D}_{\text{valid}}^k$ of client k , and returns the resulting accuracy. We initialize a Gaussian process GP for the objective function f with n_{init} sample points. Then, we find the next sample point $\eta_{n_{\text{init}}+i}$ by maximizing the acquisition function, evaluate $f(\eta_{n_{\text{init}}+i})$, and update the Gaussian process GP . Finally, we select the learning rate η^* that yields the highest accuracy. We repeat this for n_{iter} iterations.

In algorithm 6, we give the pseudocode of the GlobalOptimization method in GlobalHPO 2 based on Bayesian optimization. The objective function f takes the learning rate η as an argument, performs the FedAvg algorithm [5] with the learning rate η as an additional argument, validates the performance of the model w on the validation data $\mathcal{D}_{\text{valid}}^k$ for all clients k , computes the average accuracy of all clients and returns the resulting accuracy. We initialize a Gaussian process GP for the objective function f with n_{init} sample points. Then, we find the next sample point $\eta_{n_{\text{init}}+i}$ by maximizing the acquisition function, evaluate $f(\eta_{n_{\text{init}}+i})$, and update the Gaussian process GP . Finally, we select the learning rate η^* that yields the highest average accuracy. We repeat this for n_{iter} iterations.

Algorithm 5: Local Bayesian Optimization

```

LocalOptimization( $k, w_0$ ):
initialize a Gaussian process  $GP$  for  $f$ 
evaluate  $f$  at  $n_{\text{init}}$  initial points
for  $i = 1, \dots, n_{\text{iter}}$  do
  find sample point  $\eta_{n_{\text{init}}+i}$  that maximizes acquisition
  function
  evaluate objective function  $f$  at  $\eta_{n_{\text{init}}+i}$ 
  update the Gaussian process  $GP$ 
end
 $\eta^* := \arg \max_{i=1, \dots, n_{\text{init}}+n_{\text{iter}}} f(\eta_i)$ 
return  $\eta^*$ 
objective function:
 $f(\eta)$ :
 $w := \text{ClientUpdate}(k, w_0, \eta)$ 
 $A := A(\mathcal{D}_{\text{valid}}^k, w)$ 
return  $A$ 

```

C. Experiments

In order to systematically investigate the impact of global and local hyperparameter optimization, we compared the global and local hyperparameter optimization approach in an i.i.d. setting, the MNIST machine learning task, as well as in a non-i.i.d. setting, the industrial task. Therefore, we implemented the global and local optimization approach based on grid search with a grid $G := [0.0001, 0.001, 0.01, 0.1]$, and based on Bayesian optimization with the widely used squared exponential kernel and the upper confidence bound acquisition function. We searched for the learning rate η with fixed R , C , E and B .

In order to evaluate the global and local optimization approaches in a direct comparison, we chose the number of epochs E in the local optimization approach as $E = E_{\text{global}}R$, where E_{global} is the number of epochs in the global

Algorithm 6: Global Bayesian Optimization

```

GlobalOptimization():
initialize a Gaussian process  $GP$  for  $f$ 
evaluate  $f$  at  $n_{init}$  initial points
for  $i = 1, \dots, n_{iter}$  do
    find sample point  $\eta_{n_{init}+i}$  that maximizes acquisition
    function
    evaluate objective function  $f$  at  $\eta_{n_{init}+i}$ 
    update the Gaussian process  $GP$ 
end
 $\eta^* := \arg \max_{i=1, \dots, n_{init}+n_{iter}} f(\eta_i)$ 
return  $\eta^*$ 
objective function:
 $f(\eta)$ :
 $w := \text{FederatedAveraging}(\eta)$ 
for each client  $k = 1, \dots, K$  do
     $A^k := A(\mathcal{D}_{valid}^k, w)$ 
end
 $A := \frac{1}{K} \sum_{k=1}^K A^k$ 
return  $A$ 
    
```

optimization approach and R is the number of communication rounds. In the global optimization task, we set $R := 10$, $C := 1$, $E := 1$ and $B := 128$ for the MNIST data, and $R := 10$, $C := 1$, $E := 5$ and $B := 128$ for the industrial data. In the local optimization task, we set $E := 10$ and $B := 128$ for the MNIST data, and $E := 50$ and $B := 128$ for the industrial data. For the evaluation of the global hyperparameter optimization approach, we optimized the learning rate using the global approach, trained the federated model with a global learning rate, and tested the resulting federated model on the cohort test data. Then, we optimized the learning rate using the local approach, trained the federated model with local individual learning rates for each client in the cohort, and tested the resulting federated model on the cohort test data.

V. EXPERIMENTAL RESULTS

Following the approach of Hiessl et al. [1], we demonstrated the effectiveness of the IFL System for the industrial task and showed that the IFL approach performs better than the individual learning approach and approximates the central learning approach. Fig. 1 shows the test accuracy on the central cohort test data for each client, for i) a model trained on the individual training data of the client (individual learning), ii) a central model trained on the collected training data of all clients in the cohort (central learning), and iii) the federated model trained in the cohort.

Fig. 2 a) shows the results for the MNIST data. The optimization approaches are based on the grid search algorithm. For the training posterior to the optimization, we set $R := 10$, $C := 1$, $E := 1$, and $B := 128$ in the IFL system. The color indicates the optimized learning rate on the corresponding client. Since the MNIST data is i.i.d., there is only one cohort and all clients have the same federated model and thus the same test accuracy. Our results show that the grid search algorithm selected 10^{-3} in the local optimization of

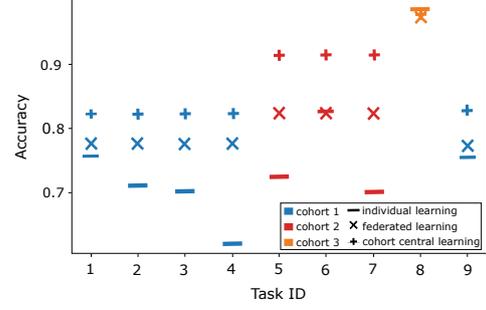


Fig. 1. Comparison of individual learning, central learning, and federated learning on the industrial data set for all clients (Task ID).

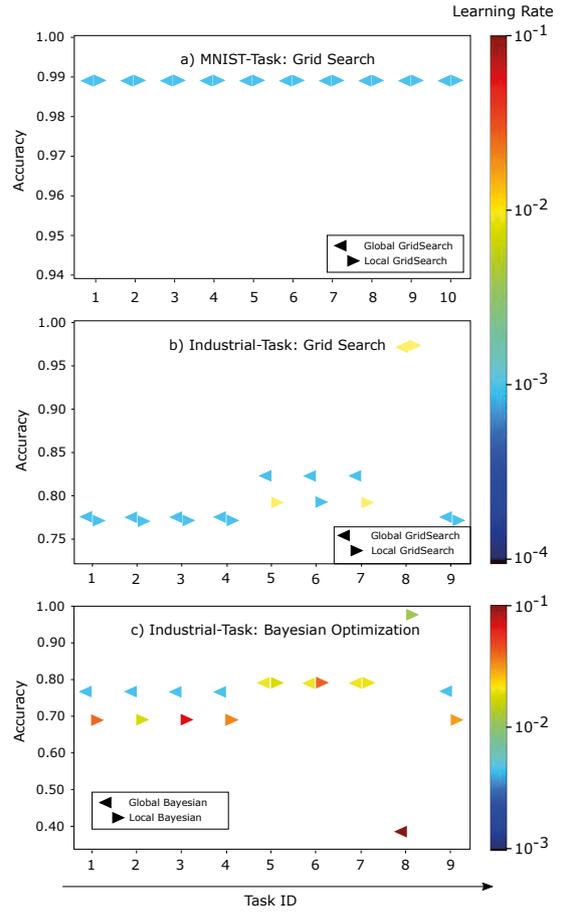


Fig. 2. Comparison of the optimization approaches based on a) grid search for the MNIST task, b) grid search for the industrial task, and c) Bayesian optimization for the industrial task for all clients (Task ID).

the learning rate on each client. According to our expectation, the global optimization approach yielded the same learning rate.

For the industrial task, we evaluated the global and local optimization approach based on grid search and Bayesian optimization. For the training posterior to the optimization, we set $R := 20$, $C := 1$, $E := 5$, and $B := 128$ in the IFL system. Fig. 2 b) shows the results for the industrial data

with the optimization approaches based on the grid search algorithm. The results show that, in all cohorts, the global approach yielded an equal or larger accuracy than the local approach.

Fig. 2 c) shows the results for the industrial data with the optimization approaches based on the Bayesian algorithm. Note that the search space of the learning rate was $[10^{-4}, 10^{-1}]$ in the optimization while the scale in the plot starts from 10^{-3} . The results show that the global approach yielded a larger accuracy than the local approach in cohort 0 and cohort 1.

The local Bayesian approach yielded different learning rates, see Fig. 2 c), on clients with no difference in data, that is, the same number of samples, the same class distribution, and the same measurement protocol. However, the local grid search approach yielded the same learning rate as the global grid search approach, see Fig. 2 b). Therefore, we suggest that the reason lies in the implementation of the Bayesian optimization approach and a not sufficiently large number of iterations to guarantee convergence.

In order to compare the optimization approaches for the industrial task, we performed a paired t-test regarding the test accuracy to determine the statistical significance, see table I. We observe that the global optimization approach is significantly better than the local approach, both for the grid search approach ($p = 0.028$) and for the Bayesian approach ($p = 0.012$). Furthermore, the results show that the grid search approach is significantly better than the Bayesian approach, both for the global approach ($p = 0.004$) and for the local approach ($p = 0.008$). Note that we considered cohort 2 an outlier and excluded this cohort from our calculations. Cohort 2 only consists of client 8, a client whose data was not generated according to the standard measurement protocol. Without outlier removal, the global grid search approach is still significantly better than the local grid search approach ($p = 0.032$), and the local grid search approach is significantly better than the local Bayesian approach ($p = 0.010$). However, there is no significant difference in the global Bayesian approach vs. the local Bayesian approach ($p = 0.755$) and in the global grid search approach vs. the global Bayesian approach ($p = 0.230$).

VI. CONCLUSION AND FUTURE WORK

The results show that the federated learning approach approximates the central learning approach, while outperforming individual learning of the clients. In this work, we investigated the impact of global and local optimization approaches in an IFL System based on a proxy data set and a real-world problem. In our experiments on the industrial data, local optimization yielded different learning rates on different clients in a cohort. However, the results show that a globally optimized learning rate, and thus, a global learning rate for all clients in a cohort improves the performance of the resulting federated model. Therefore, we conclude that the global optimization approach outperforms the local optimization approach resulting in a communication-performance trade-off

TABLE I
TEST ACCURACY OF FEDERATED MODEL ON CENTRAL COHORT TEST DATA OF INDUSTRIAL TASK POSTERIOR TO CORRESPONDING OPTIMIZATION APPROACH AND TRAINING

client	global grid	local grid	global Bayesian	local Bayesian
1	0.7756	0.7720	0.7659	0.6897
2	0.7756	0.7720	0.7659	0.6897
3	0.7756	0.7720	0.7659	0.6897
4	0.7756	0.7720	0.7659	0.6897
5	0.8230	0.7921	0.7882	0.7889
6	0.8230	0.7921	0.7882	0.7889
7	0.8230	0.7921	0.7882	0.7889
8	0.9740	0.9749	0.3867	0.9736
9	0.7756	0.7720	0.7659	0.6897

in the hyperparameter optimization in FL. In our experiments on the proxy data set, however, the local approach achieved the same performance as the global approach.

A limitation of our study is that we only considered one hyperparameter in our optimization task. Hence it would be interesting to explore whether we can confirm these observations for a hyperparameter configuration of more hyperparameters. The results show that the grid search approaches outperform the Bayesian approaches, both globally and locally. However, we suggest a convergence analysis for the Bayesian approach.

REFERENCES

- [1] T. Hiessl, S. Rezapour Lakani, J. Kemnitz, D. Schall, and S. Schulte, "Cohort - based federated learning services for industrial collaboration on the edge," *TechRxiv. Preprint*. <https://doi.org/10.36227/techrxiv.14852361.v1>, 2021.
- [2] P. Kairouz, H. B. McMahan, and et al., "Advances and open problems in federated learning," *Foundations and Trends® in Machine Learning*, vol. 14, no. 1, 2021.
- [3] J. Konečný, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," *NIPS Workshop on Private Multi-Party Machine Learning*, 2016.
- [4] Z. Dai, B. K. H. Low, and P. Jaillet, "Federated Bayesian optimization via Thompson sampling," *Advances in Neural Information Processing Systems* 33, 2020.
- [5] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*, vol. 54, pp. 1273–1282, 2017.
- [6] K. Hsieh, A. Phanishayee, O. Mutlu, and P. B. Gibbons, "The non-iid data quagmire of decentralized machine learning," *International Conference on Machine Learning (ICML)*, pp. 4387–4398, 2020.
- [7] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, "Federated learning with non-iid data," *arXiv: 1806.00582*, 2018.
- [8] A. Nilsson, S. Smith, G. Ulm, E. Gustavsson, and M. Jirstrand, "A performance evaluation of federated learning algorithms," *Proceedings of the Second Workshop on Distributed Infrastructures for Deep Learning (DIDL)*, pp. 1–8, 2018.
- [9] T. Bierweiler, H. Grieb, S. von Dosky, and M. Hartl, "Smart sensing environment - use cases and system for plant specific monitoring and optimization," *Automation 2019*, pp. 155–158, 2019.
- [10] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, 2002.
- [11] F. L. Markley, "Attitude determination using vector observation: A fast optimal matrix algorithm," *J. Astronaut. Sci.*, vol. 41, no. 2, pp. 261–280, 1993.