FEDERATED LEARNING FOR PERFORMANCE PREDICTION IN MULTI-OPERATOR ENVIRONMENTS

Xiaoyu Lan[1], Jalil Taghia[1], Farnaz Moradi[1], Mohammad Ali Khoshkholghi[2], Edvin Listo Zec[3], Olof Mogren[3], Toktam Mahmoodi[4], Andreas Johnsson[1,5]

[1]Ericsson Research, Ericsson, Sweden, [2]Middlesex University, London, UK, [3]RISE Research Institutes of Sweden, [4]Centre for Telecommunications Research, King's College London, UK, [5]Department of Information Technology, Uppsala University, Sweden

NOTE: Corresponding author: Xiaoyu Lan, xiaoyu.lan@ericsson.com

Abstract *– Telecom vendors and operators deliver services with strict requirements on performance, over complex and sometimes partly shared network infrastructures. A key enabler for network and service management in such environments is knowledge sharing, and the use of data-driven models for performance prediction, forecasting, and troubleshooting. In this paper, we outline a multi-operator service metrics prediction framework using federated learning that allows privacy-preserved knowledge-sharing across operators for improved model performance, and also reduced requirements on data transfer within an operator network. Federated learning is compared against local and central learning strategies for multi-operator performance prediction, and it is shown to balance the requirements on data privacy, model performance, and the network overhead. Further, the paper provides insights on how data heterogeneity affects model performance, where the conclusion is that standard federated learning has certain robustness to data heterogeneity. Finally, we discuss the challenges related to training a federated learning model with a limited budget on the communication rounds. The evaluation is performed using a set of realistic publicly available data traces, that are adapted specifically for the purpose of studying multi-operator service performance prediction.*

Keywords – Federated learning, machine learning, network automation and management, performance modeling

1. INTRODUCTION

Achieving zero-touch management of telecommunication networks is a challenging and demanding task. Telecom operators and providers, who deliver their services under strict Service Level Agreements (SLAs), continuously observe the network infrastructure and the service Key Performance Indicators (KPIs). Further, they apply data-driven techniques to learn performance models to be used for automation of management tasks such as service on-boarding, network slice dimensioning and admission, proactive service management, and root-cause analysis.

Applying data-driven and Machine Learning (ML)-based techniques to predict the performance of services, requires extensive measurements and data collection from the telecom infrastructure. One approach for predicting the performance of a service is to apply ML techniques on data observed from the network and infrastructure [1],[2]. It is typically assumed that the collected data can be transferred to a centralized location (cloud/data center) to be processed and used for training or fine- tuning ML models.

Measurements of the infrastructure and monitoring the service performance metrics generate large volumes of data [3]. Transferring such immense datasets over the network introduces a large overhead which can adversely impact the performance of the network, the specific service, and potentially other colocated services. Additionally, transferring data can be prohibited by privacy regulations and guidelines. The data related to service KPIs are clearly sensitive and must remain private. The infrastructure can be hosting services from different network slices sharing the common physical resources (radio, network, computer) which have to remain isolated from each other. Moreover, the different services can belong to different domains as they are either managed by different network operators or are executed over geographically distributed domains managed by the same operator.

To address the above-mentioned challenges with respect to data transfer, while aiming to achieve a high ML model performance using a large volume of data for training, distributed approaches such as Federated Learning (FL) can be used. FL can be seen as a privacy-preserving approach to distributed learning where agents participate in a federation to collaboratively learn a global model. The global model is an aggregated model computed from the local models of the agents. Crucially, in learning of the global model, agents do not share their local data, and it is only the model parameters that are shared [4].

General challenges of FL have been extensively studied in the literature [5], such as data that is not independent and identically distributed (non-i.i.d.), restrictions on the communication cost, privacy and security, compute

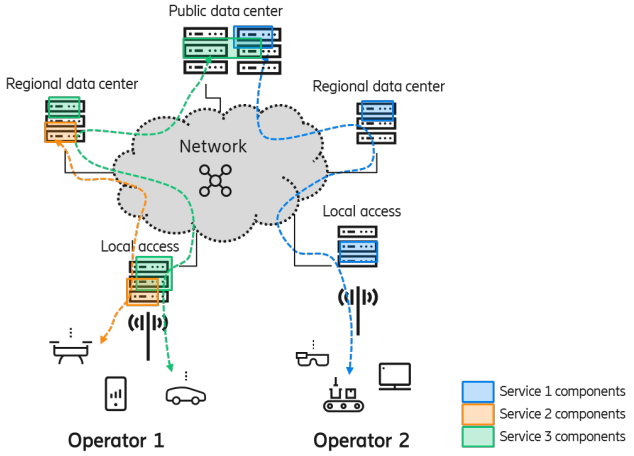


Fig. 1 – An illustration of a multi-operator environment where three services are managed by two operators which are distributed across multiple locations. The operators can use FL to learn a global model for service performance prediction

resource allocation, and system heterogeneity. However, the benefits and challenges of applying FL techniques for predicting service performance using infrastructure data in multi-operator environments have not received specific attention.

Fig. 1 illustrates a multi-operator scenario where three services managed by two operators are supported by service components across geographically distributed locations. Service performance prediction within one operator is dependent on features from all service components used by the service across the infrastructure (e.g., local access, regional data center, and public data center for service 3 in the figure). In the scenario with multiple operators sharing the same prediction task, FL can be used to train a global ML model on data collected from three services by sharing only the local model weights rather than transferring private data from multiple geographical sites to a central entity.

In this paper, we evaluate FL for service-level metrics prediction and compare it with traditional learning strategies, namely, local and central learning, using traces obtained from a realistic testbed at KTH university [6]. The data is split into 24 parts across different trace groups to emulate the multi-operator environment described above.

The main contributions of this paper are as follows: (i) we introduce the scenario of service performance prediction in multi-operator network environments and provide an evaluation based on traces from a realistic testbed[1]; (ii) we study the heterogeneity of the data and discuss how it can affect model performance; (iii) we compare different learning strategies and provide guidelines to balance model performance and privacy; (iv) we provide initial results on challenges in optimizing FL models for a fixed and limited budget on the communication rounds.

[1]The FL-prepared traces are available through [7].

The rest of this paper is organized as follows. Section 2 describes the FL background and the problem formulation. The datasets and the experimental setup are presented in Section 3. The results are described in Section 4. The discussion is summarized in Section 5. Related work is reviewed in Section 6. The paper ends with conclusions in Section 7.

2. BACKGROUND AND PROBLEM FORMULATION

2.1 Background: FL via federated averaging

FL was proposed as an approach for leveraging distributed datasets [4]. While providing a level of privacy, it can also give efficiency gains by leveraging edge compute and reducing communication needs [8]. Let K denote the number of agents indexed by k. Further, let x_k be the input data for the agent k with data distribution denoted by p_k. The optimization problem in FL can then be described as

$$\min_{w\in\mathbb{R}^d} \frac{1}{K}\sum_{k=1}^{K} \mathbb{E}_{x\sim p_k}\left[\ell_k(w;\ x_k)\right], \qquad (1)$$

where ℓ_k is the loss function for agent k, and w denotes the global model parameters that minimize the loss.

The most prevalent algorithm to solve the FL optimization problem is federated averaging (FEDAVG) [4]. When using FEDAVG, a copy of the global model is trained locally for T epochs on the data of each agent. The resulting local models are then communicated to a central server which aggregates the model parameters using the arithmetic mean. The new global model is then communicated back to the agents, which resume training. This process is repeated for a given number of global communication rounds. McMahan et al. [4] showed that this works very well when data is independently and identically distributed (i.i.d.). Meanwhile, the performance of FEDAVG deteriorates if the data distributions of the agents differ too much from each other. How to best learn a global model in this case is an open research question [9].

2.2 Problem formulation

The multi-operator environment under consideration is illustrated in Fig. 1, where a set of clients are interacting, over a network, with services managed by multiple operators and are executing in multiple data centers. We consider each component of a service running in a data center to be an agent. In order to predict the service-level metrics for each service, infrastructure metrics collected from each agent's execution environment are used as input features.

We consider three learning strategies, namely, local learning, central learning and federated learning, for the problem of service-level metrics prediction. Fig. 2 shows the learning strategies. Due to the nature of

the environment, a given target component may not have the representative infrastructure metrics for effective prediction of the service-level metrics. To be able to do that, it needs to have access to the infrastructure metrics from other components distributed across services. However, there are challenges with sharing data between the components due to the data privacy concerns and cost associated with transferring data. We can see that both local learning and central learning strategies are not well suited for the problem described here, in the case of local learning, the solution is inherently suboptimal and in the case of the central learning, the solution is impractical because of data privacy and communication cost related to data transfer.

Having said that, in this multi-operator setup, since the agents are located across geographically distributed data centers and are managed by different operators, the problem of service-level metrics prediction using FL is well motivated as it allows for structured exchange of knowledge between the components without the need for sharing data. However, FL itself comes with communication overhead cost and there are challenges in efficient training of the FL models. Hence, it is important to study how much improvement can be expected from FL compared to the local learning, and how the performance of FL compares to the central learning.

It is known that communication cost can be a bottleneck in training FL models particularly when the number of participating agents is high and the size of the model is large [5]. With a fixed number of agents and size of the model, it is a linear relation between the number of rounds and communication cost. Therefore, one approach to reduce the communication cost in FL is to reduce the number of rounds. In this paper we have focused on scenarios where there is a cost associated to the number of communication rounds. Knowing *when and how often* the local models of the agents need aggregation is crucial for model convergence, especially true in non-i.i.d. settings. If the communication rounds are too infrequent and the agents train for too many local epochs, the resulting models may start to diverge in parameter space from the global model [10].

If there were no budget constraints on the number of communication rounds, a reliable training strategy would be to train for a few epochs per round (as few as a single epoch) and instead continue federation for many rounds. However, for real-world problems, there is an overhead cost associated with the communication rounds. In practice, there could be a limited budget on the communication cost and consequently on the communication rounds.

In this work, we consider a fixed and limited budget on the communication rounds. Within a fixed budget, the question is which training strategy would be the best? That is, for how many epochs the agents shall train their

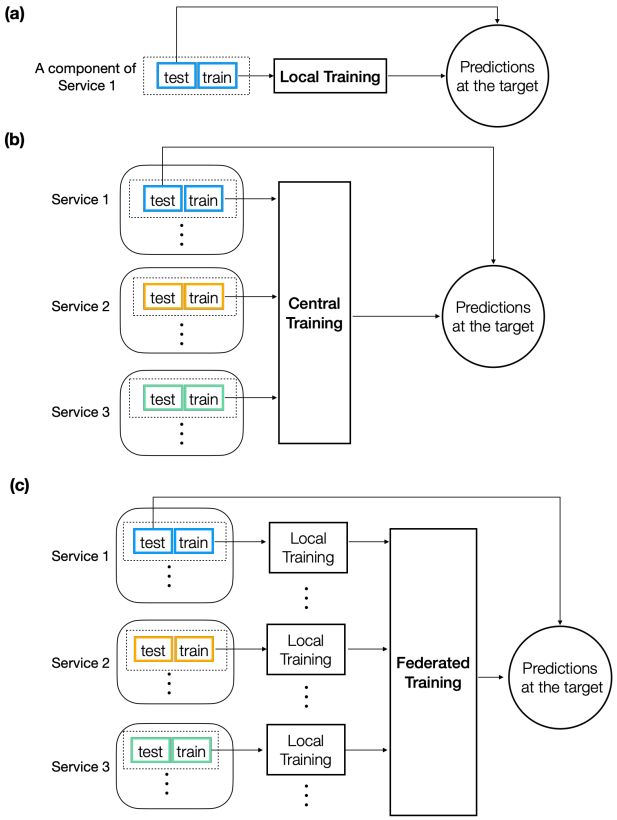


Fig. 2 – Learning strategies for the multi-operator environment illustrated in Fig. 1: (a) local learning, (b) central learning, (c) federated learning

models using their local data before the global aggregation? In this work, we show that this is indeed a nontrivial problem, and depending on the data and the use case, our choice on the number of epochs per round has consequential influence on the outcome.

3. DATASET AND EXPERIMENTS

The evaluation in this paper is based on realistic traces [6] obtained from a testbed at the KTH university. A brief overview of the scenarios and experimental infrastructure are provided below, and additional details are available in [11].

3.1 Dataset

The traces are generated by executing experiments with different configurations of services and load patterns in a testbed environment consisting of a server cluster and a set of clients. The features are collected from the server cluster and the service-level metrics are collected on the client machines.

There are two services running on these machines, namely Video-on-Demand (VoD) and Key-Value (KV) store. The VoD service provides single-representation streaming with a varying frame rate. The server cluster consists of six machines for VoD service: two networked-storage machines, three web server and transcoding

Table 1 – Trace configurations and specifications

Trace Name	Measured Service	Load Pattern	Number of Services	Number of Tasks
KV-BothApps-FlashCrowd	KV	FlashCrowd	2	2
KV-BothApps-Periodic	KV	Periodic	2	2
KV-SingleApp-FlashCrowd	KV	FlashCrowd	1	2
KV-SingleApp-Periodic	KV	Periodic	1	2
VoD-BothApps-FlashCrowd	VoD	FlashCrowd	2	3
VoD-BothApps-Periodic	VoD	Periodic	2	3
VoD-SingleApp-FlashCrowd	VoD	FlashCrowd	1	3
VoD-SingleApp-Periodic	VoD	Periodic	1	3

Table 2 – Summary of the tasks for KV and VoD services

Acronym	Service	Description
ReadsAve	KV	Average read latency
WritesAve	KV	Average write latency
NetReadAvgDelay	VoD	Net value of read average delay
AvgInterDispDelay	VoD	Average display delay
AvgInterAudioPlayedDelay	VoD	Average audio play delay

machines, and one load-balancer machine. The KV service is running on the same machines as the VoD service, and can execute in parallel. The six physical machines jointly provide the KV service. Two load generators are running in parallel in the testbed, one for the VoD application and another for the KV application. Both generators produce loads according to two distinct load patterns, namely periodic load and flash crowd load.

The traces used in the experiments are summarized in Table 1. The trace name is encoded according to *service under investigation* (KV or VoD services), *number of concurrent services* (SingleApp or BothApps), and *load pattern* (Periodic or Flashcrowd load). As an example, VoD-BothApps-Flashcrowd corresponds to a trace where both services, KV and VoD, are running under the flash crowd load, while the service-level metrics of VoD are being measured on the client side.

Features are collected from the Linux kernels that run on the server cluster machines. There are about 1700 features and some examples are CPU utilization per core, memory utilization, and disk I/O. Service-level metrics serve as the target values in ML tasks. Table 2 summarizes the service-level metrics that are considered in the experiments for KV and VoD services.

3.2 Emulation of multiple datasets for FL agents

In this subsection, we present an approach for emulation of the multi-operator scenario, given the above described traces. Further, we investigate the heterogeneity of the data at each agent.

The data traces are used to emulate the multi-operator scenario shown in Fig. 1, where the traces correspond to the services and the servers correspond to the service components. The original data set is divided to fit multiple FL agents in the experiments across different trace groups and server machines. Both KV and VoD services have four trace groups with different execution types (SingleApp or BothApps) and load patterns (periodic or flash crowd load), which are illustrated in Table 1. Each trace group contains features which are collected from six server machines, as described in Section 3.1. Hence the dataset could be divided into 24 FL agents as illustrated in Table 3. The agents are named according to trace groups (0-3) and machines (0-5). For example, agent "1_3" contains features from trace group 1 with BothApps and PeriodicLoad and are collected from machine 3. Only the same type of features across machines are kept. There are 197 features for agents of KV service, and 182 features for agents of VoD service. The data splits used in this work are available through [7].

In order to evaluate the degree of data heterogeneity of agents, we modeled the underlying data distributions across 24 agents for both input features X and service-level metrics Y, which are described as follows.

3.2.1 Input feature analysis

Symmetrized Kullback-Leibler (KL) divergence D_{sym} [12] is used to measure the similarity between underlying distributions of the input features across agents. It is calculated as:

$$D_{sym}\left(P_i, P_j\right) = \frac{1}{2}\left(D_{KL}\left(P_i||P_j\right) + D_{KL}\left(P_j||P_i\right)\right), \quad (2)$$

where P_i and P_j correspond to the probability density functions for features of agent i and agent j, respectively, and

$$D_{KL}\left(P_i||P_j\right) = \int p_i(x)\log\frac{p_i(x)}{p_j(x)}\,\mathrm{d}x. \quad (3)$$

All agents' data is standardized by removing the mean and scaling to the unit variance. Dimension reduction was done on normalized data by applying Principal Component Analysis (PCA)[12] on the input data features. The number of principal components was chosen such that their sum of explained variances is greater than the percentage 85%. The resulting number of principal components for KV and VoD were 14 and 12, respectively. The selected principal components were modeled by a Gaussian Mixture Model (GMM). The initial number of Gaussian components were set to 20. The optimal number of Gaussian components were chosen automatically through Bayesian model selection in variational inference [12]. Here, we used Scikit-Learn[13] implementation of the Bayesian GMMs. As the result, the input distribution of each agent is represented by a mixture of Gaussian distributions.

Next, we computed the KL divergence between agents' input distributions. Since there is no closed-form solution to the KL-divergence between two GMMs, it was approximated through Monte Carlo sampling (10^6 Monte Carlo samples). Fig. 3 shows the heatmap of KL divergence scores among all 24 agents for KV and VoD services. The large KL score between two agents suggests that the

Table 3 – A diagram of the data division by trace groups and machines

Trace	Execution type - Load pattern					
0	BothApps - FlashcrowdLoad					
1	BothApps - PeriodicLoad					
2	SingleApp - FlashcrowdLoad					
3	SingleApp - PeriodicLoad					
Machine	0	1	2	3	4	5

agents do not share similar underlying distributions. The main observation is that agents show various degrees of heterogeneity, and taking as a whole, the agents form a dataset that is largely heterogeneous.

The following other observations are notable. For both KV and VoD, the KL scores on the diagonal lines are relatively small. This implies that the data distributions of the same machines across execution types and load generation patterns are similar. In the testbed, machines 0,1 and machines 2,3,4 are executing different functionalities, i.e., network storage and web server, respectively for VoD services. In Fig. 3(b), the KL scores between the agents of machines 0,1 are bounded with blue boxes, and the KL scores among agents of machines 2,3,4 are bounded with green boxes. Relatively speaking, the KL scores within blue and green boxes are smaller. This implies that the features that are collected from machines with the same service function have similar distributions.

3.2.2 Service-level metrics

Note that across six machines per trace group, not only the same class of service-level metrics are measured but also the measured values of the service-level metrics are the same. However, across trace groups for the same machine, the same class of service-level metrics are measured but the measured values may be different. Fig. 4 shows the histograms of the service-level metrics used in experiments for both KV and VoD services. It can be seen that the service-level metrics AvgInterDispDelay and AvgInterAudioPlayedDelay of VoD services have bi-modal distributions while the other service-level metrics have a uni-modal distribution. In the experiments, we show that the presence of multi-modality can adversely affect the performance of FL.

3.3 Experiments setup

3.3.1 Evaluation framework

To facilitate evaluation of the learning strategies described in Section 2, we design an evaluation framework where the learning strategies can be compared against each other. The data of the agents are divided into a training set and a test set where the sizes of the training and the test sets are the same across all agents, respectively. Note that for a target agent, we assume the same training and test sets across all learning strategies to ensure a fair comparison. A conceptual illustration of these learning strategies are shown in Fig. 2.

Table 4 – Predictive net used in all experiments for KV and VoD traces

Net	Number of Units	Activation Function	Batch Normalization	Drop Out
Input Layer	D_{input}	Tanh	True	0.2
Hidden Layer 1	50	ReLU	True	0.2
Hidden Layer 2	50	ReLU	True	0.2
Output Layer	D_{output}	Linear	False	False
Loss Function	SmoothL1Loss, $\beta = 1.0$			
Optimizer	Adam, learning rate = 0.001			

Table 5 – Scenarios considered in the experiments

Name	KV Traces		VoD Traces	
	N_{training}	N_{test}	N_{training}	N_{test}
Small Set	1211	23014	1457	27694
Tiny Set	242	23983	291	28860

Local learning For a target agent, data in the training set are normalized by the standard normalization, that is removing the mean and scaling to the unit variance. Data in the test set is normalized using the same normalizer learned on the training set. Next, the predictive local ML model is learned using the data in the training set. Finally, the performance of the learned local model is evaluated on the test set.

Central learning Here, it is assumed that the target agent has access to the data of all other agents. The target agent constructs a training set from the training data of all agents, named central learning set. Note that the test set here is the same test set used for local learning. Data is normalized using the same approach as in local learning. The predictive central model is learned using the data in the central learning set, and its performance is evaluated on the test set.

Federated learning For a target agent, there is no possibility of accessing the data from other agents. Agents participate in a federation to collaboratively learn a joint model, named the global model. All agents are initialized using the same predictive model which they have all received from the server entity. Here, we consider federated averaging as the choice of FL scheme, as discussed in Section 2.

For a target agent, data is normalized as in the case of local learning. The target agent participating in the federation learns collaboratively the global model. The performance of the global model is then evaluated on the test set. Note that the test set here is the same test set used for the local and the central learning.

3.3.2 Models

We compare the following models, (i) the local model,(ii) the central model, and (iii) the federated model, obtained from the evaluation framework. For the federated learning, we limit the number of rounds R to 20,

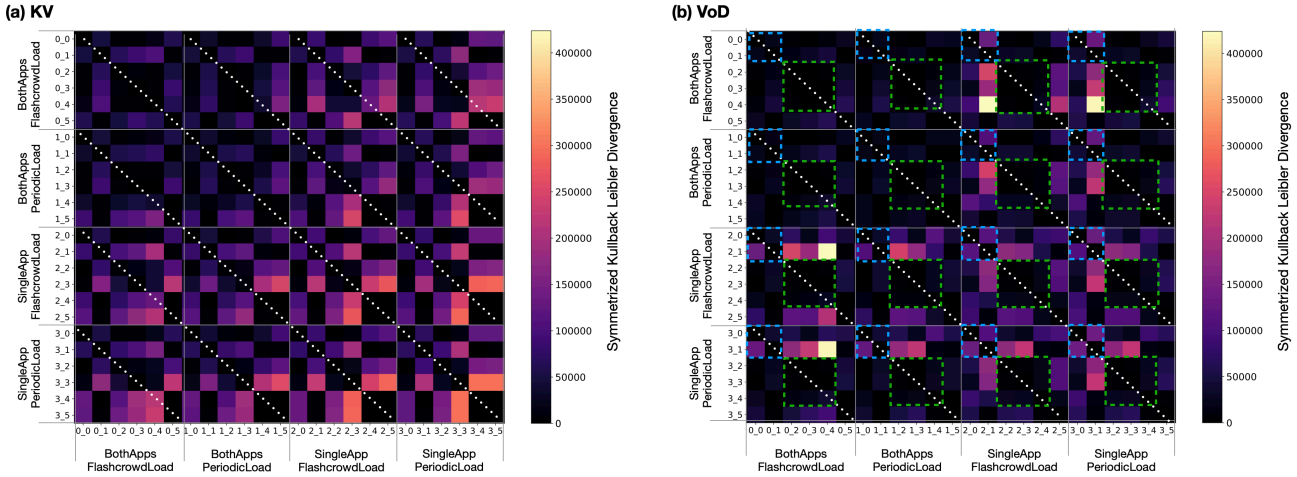


Fig. 3 – A heatmap of KL divergence scores of features among 24 agents for KV and VoD services

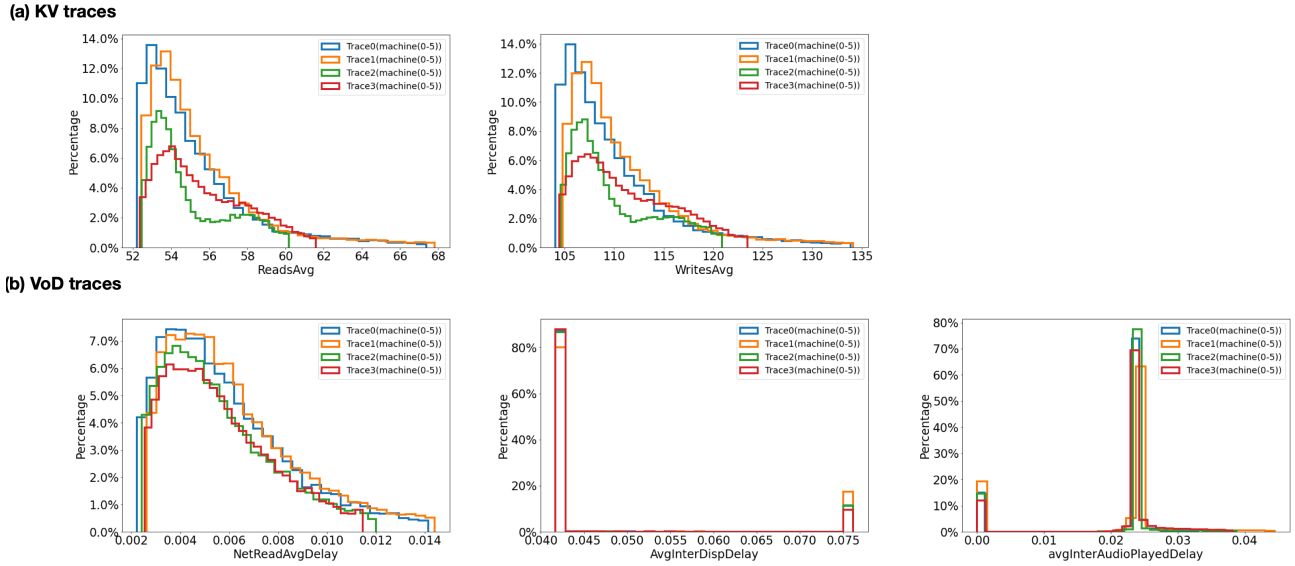


Fig. 4 – A histogram of the service-level metrics of trace groups for KV and VoD services. The figure legends are introduced in Table 3

however, we vary the number of epochs T per round according to $T \in \{10, 50, 200\}$. This is to study various FL learning strategies when there is a fixed and limited budget on the communication rounds, as motivated in Section 2.2. For the local and central models, the number of epochs is fixed and it is set to 200. Table 4 summarizes the design choices used in construction of the predictive models for KV and VoD traces. All models are implemented in PyTorch [14].

3.3.3 *Scenarios*

We consider different scenarios by varying the size of the training set N_{training} and the test set N_{test}. This is to study the performance of the models with respect to the availability of data in the execution environments. Table 5 summarizes these scenarios.

3.3.4 *Performance evaluation*

Model performance is evaluated using the normalized Mean-Absolute-Error (nMeanAE) score between the true and predicted performance metrics defined as:

$$\text{nMeanAE} := \frac{1}{\bar{y}} \left(\frac{1}{N_{\text{test}}} \sum_{n=1}^{N_{\text{test}}} |y_n - \hat{y}_n| \right), \qquad (4)$$

where $\hat{y}_n$ is the model prediction for the n-th measured performance metric y_n, and $\bar{y}$ is the average quantity across all samples in the test set.

4. RESULTS

In this section, we present the results of the experiments described in Section 3.3.

Fig. 5 summarizes the performance of the models for the prediction of KV and VoD service-level metrics. The figure shows the nMeanAE error per agent averaged across

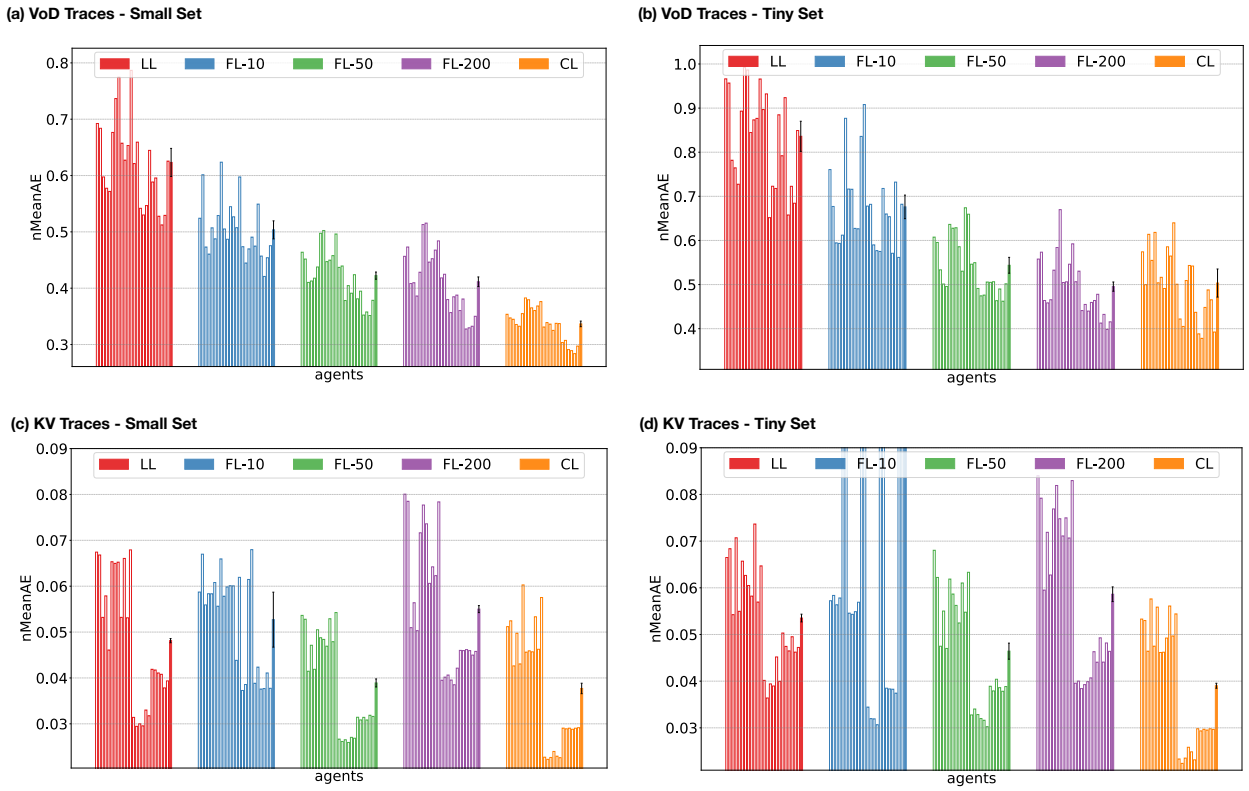


Fig. 5 – Error in prediction of the VoD and KV service-level metrics per agent averaged across all tasks. The error is measured in terms of nMeanAE between measured and predicted service-level metrics. The right-most solid boxes in all the categories show the averaged nMeanAE scores across all agents. The empty boxes show the nMeanAE scores per agent. The figure shows the average result across 10 experiment trials. LL stands for local learning, CL stands for central learning, and FL stands for federated learning. As an example FL-10 indicates the FL where agents perform 10 epochs per round at the local learning phase

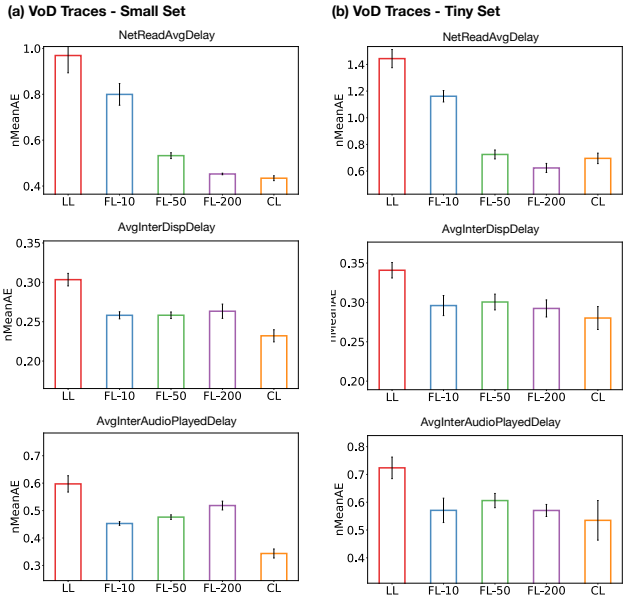


Fig. 6 – Error in prediction of the VoD service-level metrics per task averaged across all agents. The error is measured in terms of nMeanAE between measured and predicted service-level metrics. The figure shows the average result across 10 experiment trials.

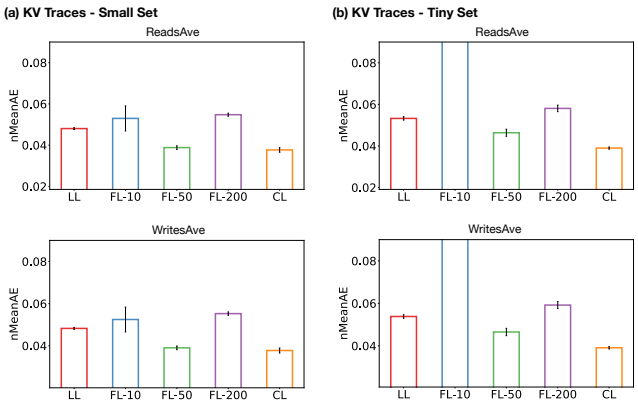


Fig. 7 – Error in prediction of the KV service-level metrics per task averaged across all agents. The error is measured in terms of nMeanAE between measured and predicted service-level metrics. The figure shows the average result across 10 experiment trials

all tasks for Small Set and Tiny Set. The following observations are notable. For the VoD traces, all FL models outperform local learning. Among FL models, FL-200 is the best performer while both FL-200 and FL-50 outperform FL-10. Similar observations are made for both Small Set and Tiny Set. For the KV traces, there can be seen a rather considerable difference between the performance of the FL models. For Small Set shown in Fig. 5(c) and Tiny Set shown in Fig. 5(d), FL-50 is the best performer and it outperforms the local learning. However, FL-10 and FL-200 performance is inferior to the local learning.

Our observations for KV traces differ from the ones for VoD traces. This suggests that for a fixed and limited budget on the communication rounds, choosing a common strategy that works well for all scenarios is not straightforward. As an example, while FL-200 is the best performer for the VoD traces, it is among the worst performers for the KV traces. In other words, it is difficult to make a general statement about the best approach for choosing the number of epochs per round in training FL models given a fixed budget on the communication rounds. Indeed this observation is one of the main results of the experiments. The considerable difference between the behavior of the FL models for KV and VoD traces can be explained partly by noting the difference between the degree of heterogeneity across input distributions of the agents. In Fig. 3, we measured the degree of heterogeneity across agents. While both KV and VoD traces were shown to be heterogeneous, a clear difference between their profiles can be seen.

Fig. 6 and Fig. 7 show the performance of the models per task averaged across all agents for VoD and KV traces, respectively. For the case of VoD traces, between the FL models, FL-200 is the best performer for NetReadAvgDelay while FL-10 is the best performer for AvgInterDispDelay and AvgInterAudioPlayedDelay. This observation further highlights the difficulty in choosing a reasonably optimal training strategy for FL when there is a fixed and limited budget on the communication rounds.

It can be seen that for the KV traces, across tasks, FL models perform similarly, e.g., FL-50 is the best performer for both ReadsAve and WritesAve. However, for the VoD traces, FL models perform differently. As an example, FL-10 is the best performer for AvgInterAudioPlayedDelay while it is the worst performer for NetReadAvgDelay. The difference in behavior of the FL models can be explained by looking into the distribution of the tasks. As shown in Fig. 4(a), ReadsAve and WritesAve have similar distributions while, as shown in Fig. 4(b), NetReadAvgDelay, AvgInterDispDelay and AvgInterAudioPlayedDelay have clearly different distributions.

For the task NetReadAvgDelay in VoD traces, the best performing FL model (FL-200) achieves similar performance as the central learning model. The same observation can be made for the best performing FL model (FL-50) for ReadsAve and WritesAve tasks in KV traces. However, for AvgInterDispDelay and AvgInterAudioPlayedDelay in VoD traces, the difference between the performances of the best performing FL model and the central learning model is relatively large. This can be explained by taking into consideration the distribution of the tasks. The task distribution for NetReadAvgDelay, ReadsAve and WritesAve are uni-modal while for AvgInterDispDelay and AvgInterAudioPlayedDelay, the task distributions are bi-modal. This implies that FL can be expected to achieve similar performance figures as the central learning for uni-modal task distributions. However, for bi-modal task distributions, performance of the FL remains inferior to the central learning.

5. DISCUSSION

In this work, we studied the problem of service-level metrics prediction in multi-operator environments. Due to the nature of the environment, we argued that local learning and central learning are not well-suited. In the case of the local learning, taking into consideration merely the infrastructure metrics collected from the individual components results in models that are inherently suboptimal as the local data will not be fully representative of the prediction task. For the case of the central learning, having access to all data allows for learning models that outperform models learned using only the local data. However, learning a central model may simply not be possible due to the data ownership rights, privacy concerns, and the cost of data transfer to a central location. Thus, studying FL is well positioned.

While FL can potentially outperform local learning and approach central learning, its success depends on many factors, among others, the budget on the communication cost. The communication cost budget is an important factor that needs to be taken into consideration for real-world applications of FL. For a fixed and limited budget on the communication rounds, we showed that training FL models that perform reasonably well is far from trivial. Importantly, performance of the FL model depends largely on the degree of model fitness at the local training phase, determined by the number of training epochs. Prior to the global aggregation, if agents contribute with local models that are under-fitted by training for too few epochs, the global model may not converge given the limited budget. On the flip side, if they contribute over-fitted models by training for too many epochs, the global model may diverge in the parameter space. Knowing the right degree of model fitness at the local training phase of the FL has consequential influence on the quality of the global model and ultimately success of the federation.

Motivated by the discussion above, our experiments were designed to show the potential and limitations of the FL. We emulated a multi-operator scenario and constructed a federation of 24 agents and described characteristics of the data. In particular, we measured the degree of

heterogeneity across agents showing considerable variations across agents, and we visualized the complexity of the prediction tasks showing the presence of bi-modality in the task space. Leveraged on the unique characteristics of data, we designed an evaluation framework for comparing the performance of different learning strategies, namely, local, central and federated learning. In this evaluation framework, we limited the budget on the communication cost (to 20 rounds), and evaluated the quality of the federated models under three different training strategies in terms of the number of epochs per round, namely, FL-10, FL-50 and FL-200. We observed that the quality of the trained FL models varies largely depending on the degree of fitness of the models before global aggregation. In other words, it was shown that it is not straight-forward to decide on the optimal number of epochs per round when there is a fixed budget on the communication rounds. With that in mind, here, we call for future research in this direction. More specifically, a data-driven training strategy for FL where there is a fixed and limited budget on the communication rounds.

A unique aspect of the multi-operator use case introduced here is that the components belonging to the same service have the same measured service-level metrics. In our emulated data, this effectively means that the agents belonging to the same server machines have the same measured service-level metrics for different infrastructure metrics. Although we considered FEDAVGas the choice of FL method, belonging to the class of horizontal federated learning, a hybrid technique of vertical and horizontal federated learning [15, 16] fits particularly well to this use case, where the agents belonging to the same trace group can form a vertical federation, as they share the same measured service-level metrics, and agents across the trace groups can form horizontal federation. Future studies are needed to evaluate the effectiveness of such FL techniques.

6. RELATED WORK

FL is a privacy-preserving distributed learning approach which enables training machine learning models collaboratively over remote data centers or devices without sharing local data. FL is a potential candidate to train predictive models for service-level metrics in telecommunication networks where data privacy and communication overhead are two major challenges of performance prediction techniques [17].

In [18], the authors have proposed both centralized and federated approaches for Virtual Network Function (VNF) autoscaling in multi-domain 5G networks. The proposed FL-based solution is able to predict the future number of VNF instances according to the expected traffic demand and service requirements in order to maximize the QoS while keeping data localized. The authors, then, have compared the performance of the centralized and federated approaches based on the model accuracy and the resource usage. In [19] and [20], the authors point out the challenges of data-sharing between network operators for service QoE-model development, and propose FL approaches to overcome these challenges.

In [21], a distributed network slicing framework based on a federated-orchestrator (F-orchestrator) is proposed to coordinate the computational resources without sharing base stations' local data. In this framework, each base station trains a model on its local data to allocate the resources to its users for a set of IoT services. The F-orchestrator then creates a global model based on the local models to optimally allocate the resources across all the base stations. Minimizing the raw monitoring data exchange among different domains, is a key enabler for both sustainability and scalability in 5G networks. Hence, the authors in [22] introduce statistical FL provisioning models that can learn over live network non-i.i.d. datasets in an offline fashion while respecting SLA long-term statistical constraints. The results show that the FL technique dramatically reduces communication overhead compared to a centralized deep learning.

Another work [23], addresses the challenges of predicting the performance of network slices in 5G networks. The authors proposed an FL approach to predict a KPI (average duration of user attachment) of running network slices with respect to data isolation in each network slice. In [24], an FL approach is proposed for forecasting telecom KPI values in radio network cells. In this work, the goal of applying FL is to shorten the training time of new models, as well as minimizing the transferred data to a central server.

The authors in [25], developed a hierarchical FL approach using multiaccess edge computing in order to tackle the data privacy and communication bottlenecks of IoT heterogeneous networks with imbalanced and non-i.i.d. data. User association to the edge nodes and resource utilization are two KPIs that the trained FL models aim to optimize. In [26], an FL architecture named Blaster is proposed for routing network traffic through distributed software defined network-enable edge networks with varying network conditions and high volume of generated traffic, in order to improve the service performance of the applications while reducing the communication overhead caused by model training.

Similar to previous work, in this study, we investigate and evaluate the efficiency of FL in telecom networks. The focus of this paper, in contrast to other related work is the comparison of different learning strategies (including FL) for service performance prediction using data obtained from a multi-domain infrastructure where we investigate the impact of data heterogeneity, limited budget on the communication rounds, and training data size on FL model performance.

7. CONCLUSION

In this paper, we introduced and discussed the challenges of service-level performance prediction in multi-operator network environments using federated learning, with the aim of reducing network overhead and preserving data privacy across operators. In an experimental study, we used traces collected from a testbed to emulate a multi-operator environment and designed an evaluation framework for comparing the performance of models trained using an FL approach against local learning and central learning. The evaluation framework was designed to illustrate potential benefits and limitations of FL. In particular, we studied FL under limited budget on the communication rounds. The results showed that FL can potentially outperform local learning and approach the performance of central learning. However, the performance of FL can vary largely depending on the employed training strategy at the local training phase of FL. With this study, we call for further research on tackling challenges in the application of FL in multi-operator network environments.

ACKNOWLEDGEMENT

This research has been supported by Sweden's Innovation Agency (VINNOVA) through the Celtic-Next project ANIARA (C2019/3-2).

REFERENCES

[1] Raouf Boutaba, Mohammad A. Salahuddin, Noura Limam, Sara Ayoubi, Nashid Shahriar, Felipe Estrada Solano, and Oscar Mauricio Caicedo Rendon. "A comprehensive survey on machine learning for networking: evolution, applications and research opportunities". In: *J. Internet Serv. Appl.* 9.1 (2018), 16:1–16:99.

[2] Michel Gokan Khan, Javid Taheri, Mohammad Ali Khoshkholghi, Andreas Kassler, Carolyn Cartwright, Marian Darula, and Shuiguang Deng. "A Performance Modelling Approach for SLA-Aware Resource Recommendation in Cloud Native Network Functions". In: *2020 6th IEEE Conference on Network Softwarization*. IEEE. 2020, pp. 292–300.

[3] Binfeng Wang and Jinshu Su. "FlexMonitor: A flexible monitoring framework in SDN". In: *Symmetry* 10.12 (2018), p. 713.

[4] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. "Communication-Efficient Learning of Deep Networks from Decentralized Data". In: *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*. Ed. by Aarti Singh and Jerry Zhu. Vol. 54. Proceedings of Machine Learning Research. PMLR, 2017, pp. 1273–1282.

[5] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. "Federated Learning: Challenges, Methods, and Future Directions". In: *IEEE Signal Processing Magazine* 37.3 (2020), pp. 50–60. DOI: 10.1109/MSP.2020.2975749.

[6] F. S. Samani. *Data traces for efficient learning on high-dimensional operational data*. https://github.com/foroughsh/KTH-traces. 2021.

[7] Jalil Taghia, Xiaoyu Lan, Farnaz Moradi, Hannes Larsson, Andreas Johnsson, Forough Shahab Samani, and Rolf Stadler. *Telecom Data Traces for Distributed Learning*. https://www.kaggle.com/datasets/jaliltaghia/data-traces-from-a-data-center-testbed. Dec. 2022.

[8] Wei Yang Bryan Lim, Nguyen Cong Luong, Dinh Thai Hoang, Yutao Jiao, Ying-Chang Liang, Qiang Yang, Dusit Niyato, and Chunyan Miao. "Federated Learning in Mobile Edge Networks: A Comprehensive Survey". In: *IEEE Communications Surveys Tutorials* 22.3 (2020), pp. 2031–2063. DOI: 10.1109/COMST.2020.2986024.

[9] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. "Advances and open problems in federated learning". In: *arXiv preprint arXiv:1912.04977* (2019).

[10] Q. Li, Yiqun Diao, Quan Chen, and Bingsheng He. "Federated Learning on Non-IID Data Silos: An Experimental Study". In: *ArXiv* abs/2102.02079 (2021).

[11] Rerngvit Yanggratoke, Jawwad Ahmed, John Ardelius, Christofer Flinta, Andreas Johnsson, Daniel Gillblad, and Rolf Stadler. "A service-agnostic method for predicting service metrics in real time". In: *International Journal of Network Management* 28.2 (2018), e1991.

[12] Christopher M. Bishop. "Pattern Recognition and Machine Learning". In: Springer, 2006. Chap. 10.

[13] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[14] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. "PyTorch: An Imperative Style, High-Performance Deep Learning

Library". In: *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019, pp. 8024–8035.

[15] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. "Federated Machine Learning: Concept and Applications". In: *ACM Trans. Intell. Syst. Technol.* 10.2 (Jan. 2019). ISSN: 2157–6904. DOI: 10.1145/3298981.

[16] Qinbin Li, Zeyi Wen, Zhaomin Wu, Sixu Hu, Naibo Wang, Yuan Li, Xu Liu, and Bingsheng He. "A Survey on Federated Learning Systems: Vision, Hype and Reality for Data Privacy and Protection". In: *IEEE Transactions on Knowledge and Data Engineering* (2021), pp. 1–1. DOI: 10 . 1109 / TKDE . 2021 . 3124599.

[17] Sawsan Abdulrahman, Hanine Tout, Hakima Ould-Slimane, Azzam Mourad, Chamseddine Talhi, and Mohsen Guizani. "A survey on federated learning: The journey from centralized to distributed on-site learning and beyond". In: *IEEE Internet of Things Journal* 8.7 (2020), pp. 5476–5497.

[18] Tejas Subramanya and Roberto Riggio. "Centralized and federated learning for predictive VNF autoscaling in multi-domain 5G networks and beyond". In: *IEEE Transactions on Network and Service Management* 18.1 (2021), pp. 63–78.

[19] Selim Ickin, Markus Fiedler, and Konstantinos Vandikas. "QoE Modeling on Split Features with Distributed Deep Learning". In: *Network* 1.2 (2021), pp. 165–190.

[20] Selim Ickin, Konstantinos Vandikas, Farnaz Moradi, Jalil Taghia, and Wenfeng Hu. "Ensemble-based synthetic data synthesis for federated QoE modeling". In: *2020 6th IEEE Conference on Network Softwarization*. IEEE. 2020, pp. 72–76.

[21] Yingyu Li, Anqi Huang, Yong Xiao, Xiaohu Ge, Sumei Sun, and Han-Chieh Chao. "Federated orchestration for network slicing of bandwidth and computational resource". In: *arXiv preprint arXiv:2002.02451* (2020).

[22] Hatim Chergui, Luis Blanco, and Christos Verikoukis. "Statistical Federated Learning for Beyond 5G SLA-Constrained RAN Slicing". In: *IEEE Transactions on Wireless Communications* (2021).

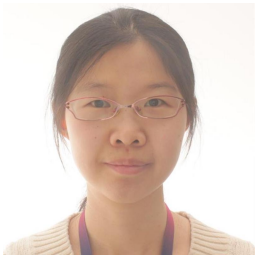
[23] Bouziane Brik and Adlen Ksentini. "On Predicting Service-oriented Network Slices Performances in 5G: A Federated Learning Approach". In: *2020 IEEE 45th Conference on Local Computer Networks*. IEEE. 2020, pp. 164–171.

[24] Jose Carlos Alcantara et al. "De-centralized Learning for Radio Network Key Performance Indicator Prediction". In: (2020).

[25] Alaa Awad Abdellatif, Naram Mhaisen, Amr Mohamed, Aiman Erbad, Mohsen Guizani, Zaher Dawy, and Wassim Nasreddine. "Communication-efficient hierarchical federated learning for IoT heterogeneous systems with imbalanced data". In: *Future Generation Computer Systems* (2021).

[26] Alessio Sacco, Flavio Esposito, and Guido Marchetto. "A federated learning approach to routing in challenged sdn-enabled edge networks". In: *2020 6th IEEE Conference on Network Softwarization*. IEEE. 2020, pp. 150–154.

AUTHORS



Xiaoyu Lan received her M.Sc. degree in scientific computing in 2011 from KTH Royal Institute of Technology, Sweden. She is currently a senior researcher with Ericsson Research, Stockholm. She has developed machine learning solutions in various domains, is the co-inventor of multiple patent applications and co-author of several research papers. Her current research focus is on distributed learning and federated learning for network management.

Jalil Taghia is currently with Ericsson Research, Stockholm. He received his PhD in Electrical Engineering from KTH Royal Institute of Technology in 2014. His research interests include probabilistic inference in machine learning and Bayesian statistics.

Farnaz Moradi received her PhD in computer science and engineering from Chalmers University of Technology in 2014. She is currently a research engineer with Ericsson Research, Stockholm. Her research interests include distributed learning, transfer learning, and machine learning for network and service management.

Mohammad Ali Khoshkholghi received his Ph.D. degree in computer science from University Putra Malaysia in 2017, and bachelor's and master's of computer science, in 2007 and 2011, respectively. He is currently an assistant professor at Middlesex University London. Before joining Middlesex University, he worked as a postdoctoral research fellow at King's College London from 2020 to 2022 as well as at Karlstad University in Sweden, from 2018 to 2020. He serves as a referee, TPC and editorial board member for many prestigious journals and conferences. His research interests lie in the area of edge and cloud computing, 5G/6G networks and machine learning.

Edvin Listo Zec is a machine learning researcher at RISE Research Institutes of Sweden and a PhD student in federated learning at KTH Royal Institute of Technology. His research focuses on representation learning using deep neural networks in decentralized and federated systems, with a particular interest in continual learning. In addition, he is passionate about applying ML to tackle issues related to climate change.

Olof Mogren is a research leader at RISE Research Institutes of Sweden. He has a PhD in computer science from Chalmers University of Technology, and is the organizer of RISE learning machines seminars. Olof works on problems within applied AI often related to environmental questions. His work includes federated learning, privacy-preserving representation learning, and generative adversarial networks for all data modalities, including natural language, images, and sound. Some of his ongoing projects include soundscape analysis using AI methods, the federated learning testbed, predictive maintenance of district heating networks, the Swedish medical language data lab, and smart fire detection using machine listening.

Toktam Mahmoodi received a B.Sc. degree in electrical engineering from the Sharif University of Technology, Iran, in 2002, and a Ph.D. degree in telecommunications from King's College London, U.K, in 2009. She was a visiting research scientist with F5 Networks, San Jose, CA, in 2013, a post-doctoral research associate with the ISN Research Group, Electrical and Electronic Engineering Department, Imperial College from 2010 to 2011, and a mobile VCE researcher from 2006 to 2009. She has also worked in the mobile and personal communications industry from 2002 to 2006, and in an R&D team on developing DECT standard for WLL applications. She has contributed to, and led number of FP7, H2020 and EPSRC funded projects, advancing mobile and wireless communication networks. Toktam is currently Head of the Centre for Telecommunications Research in the Department of Engineering, and is a professor of communication engineering in the same department at King's College London. Her research interests include network intelligence, and mission critical networking, with impact in healthcare, automotive, smart cities and emergency services.

Andreas Johnsson (Senior Member, IEEE) received an M.Sc. degree in computer science and mathematics from Uppsala University in 2002, and a Ph.D. degree in computer science from Mälardalen University in 2007. He is a research leader with Ericsson Research, Stockholm. His group has made contributions within the area of machine learning and automation for 5G and 6G networks. He has been appointed adjunct Associate Professor of Computer Communication at Uppsala University since 2019. He has received two best paper awards from IEEE ComSoc, and received multiple research grants from both Swedish and European funding agencies. He has co-authored numerous patents, papers, and standards. His research interests include machine learning for network and service management, network measurements, cloud, and IoT.