Federated Representation Learning for Indoor-Outdoor Detection in beyond 5G networks

Sid Ali Hamideche, Marie Line Alberi Morel, Kamal Singh, César Viho

Federated Representation Learning for Indoor-Outdoor Detection in beyond 5G networks

Sid Ali Hamideche
Nokia Paris Saclay
Nozay, France
sid.hamideche@nokia.com

Marie Line Alberi Morel
Nokia Paris Saclay
Nozay, France
marie_line.alberi-morel@nokia.com

Kamal Singh
Laboratoire Hubert Curien
Saint-Etienne, France
kamal.singh@univ-st-etienne.fr

César Viho
IRISA-UR1
Rennes, France
cesar.viho@irisa.fr

***Abstract*—Scarcity of labelled datasets makes it challenging to train robust mobile user environment detection models. Labelling and centralizing large data amounts for training is expensive. To address these issues, semi-supervised learning techniques aim to reduce data labelling, while Federated Learning (FL) avoids centralizing the data. In this work, we propose a novel approach for Indoor/Outdoor detection by combining the strengths of federated and semi-supervised learning. It consists of 2 steps: (1) Unsupervised Federated representation Learning to learn representations using large unlabelled data. We leverage *unlabelled data* from diverse sources situated across various geographical locations. Through FL, we develop high-quality representations by jointly learning from this distributed unlabelled data. (2) We then capitalize on the acquired representations and further employ transfer learning to achieve accurate detection using a *reduced amount of labelled data*. We also add an optimization module referred as *User Behavioural Optimizer* that corrects environment detection errors by tracking behavioural anomalies. We obtain an F1-score of 95.06% using only 30% of the entire amount of labelled data available.**

***Index Terms*—Federated representation Learning, Semi-supervised Learning, Indoor-Outdoor Detection**

I. Introduction

In 5G-advanced network and beyond, intelligent context-aware networks will be capable of exploiting sensing information, with "Network as a sensor" paradigm. Knowledge of environmental context of mobile user/device allows network operators to optimize deployments, operations [1] and enable accurate location-based services [2] with no or limited human intervention. In literature, some studies have utilized user's indoor-outdoor context for various applications. Detecting the environmental context of the mobile user or device corresponds to answering: where is the user or device (*Indoors or Outdoors*) while connecting to the network? This can be answered using machine learning (ML) techniques, which will be technological enablers for 6G communication networks. Such next-generation wireless networks are expected to be highly distributed. They will evolve to paradigms such as edge computing, bringing ML training or inference near to the data source, enabling a more flexible and scalable network architecture. However, intelligent context-aware networks gather user context to utilize this information externally for third parties or internally for value-added network services within the infrastructure. A simple solution entails data collection by the network, sent to a central entity for ML training and distributed entities are only used for the inference (sensing) process. To do this, the network collects data from mobile users/devices in different situations, which may not all have been seen by each distributed entity. However, this leads to large data transfer overhead, potential congestion, and context variance as users move to different geographical areas, each detected by different entities. It would be more efficient for them to share their common knowledge rather than working independently. Moreover, the training cannot be done regularly due to huge size of data needed to be continually processed by a central entity. This will lead to ML models which are not always up-to-date. Furthermore, sharing data with a centralized entity may not to be desired by users due to privacy and security reasons. Indeed, all data can be stolen or contaminated if the security of the central entity is breached. The above problems can be addressed by employing Federated Learning (FL) and semi-supervised learning (SSL).

Yet, training efficient machine learning models without full data transmission to a central entity, while ensuring low network bandwidth usage and safeguarding data privacy, remains a significant challenge. FL is a distributed ML technique which consists in training multiple models on different edge nodes and sharing with a central unit only the parameters learned, rather than whole data. For that, as an efficient way, the central entity is responsible for aggregating and sharing the parameters emitted by distributed entities, let's call them, FL-clients [3]. Distributed models can also train continuously, adapting to the network changes. Data labelling is another challenging aspect when using supervised machine learning in future mobile networks. Indeed, it can be quite expensive and sometimes even impossible in some cases. Therefore, semi-supervised approaches are preferable as they take advantage of unlabelled data availability and reduce the amount of data to be labelled, while preserving performances.

In this work, we propose a new approach consisting of 2 phases to detect the user environment from time series data:

1) Unsupervised representation learning using unlabelled data trained with Federated Learning in local entities.
2) Leveraging the above learned representations via Fine-tuning and Transfer learning on the server to improve *Indoor-Outdoor Detection* (IOD) accuracy with reduced amount of labeled data.

Our approach can also be seen as semi-supervised learning, as it consists of both unsupervised and supervised training. The advantage of our approach is that collecting labels is not necessary in the server along with unlabelled data from users in distributed entities. The unlabelled data collected can be continuously mined, while collecting only a small amount of labelled data. The representation learning used with transfer techniques, which have been an area of focus in recent years, shows state-of-the-art performances in fields like image recognition and natural language processing [4]–[6]. They can mine large sets of unlabelled data to learn about the patterns embedded and the most relevant information that can be extracted from data. This useful knowledge can then be exploited by other models applied on other downstream tasks, particularly for supervised learning tasks where the amount of data is insufficient to extract such knowledge. In our IOD use case based on time series classification, the labelled dataset may not contain all the patterns of more recent daily situations that can be observed in real world. To ensure accurate IOD models with short time series, we propose to use the module called *User Behavioural Optimizer (UBO)* [7]. UBO aims to correct environment detection errors by tracking behavioural anomalies, that diverge from typical behaviour of a human.

The main paper contributions are:

1) A study of our approach applied for IOD *with 3GPP standard radio data* and Transfer Learning for supervised IOD *on radio access network side*
2) Our experiments show that our IOD approach *with three times less labelled data* obtains similar F1-scores as compared to conventional supervised learning.

This paper is organized as follows. Section II presents some related work. Section III proposes our approach. Section IV discusses the results. The final section presents our conclusion.

II. Related Work

Using Federated Learning (FL) in new-generation mobile networks has lately become common. Most of such work, however, mainly focuses on IoT and how to leverage the sensors available at users' devices [8] [9]. Also, they focus on topics other than user context detection, which has mainly been studied with centralized learning techniques.

FL works have been mainly studied with supervised approaches where the labels are available within the FL-clients [3]. Recently, FL with semi-supervised approaches have emerged, for example Aouedi *et al.* [10] used it for attack detection in IoT. To overcome the lack of labelled data, they trained auto-encoders on distributed entities, then sent the learned parameters to the server where the aggregation is done. The aggregated parameters are then used to initialize the supervised classification model, using a minimal amount of labelled data. However, they do not consider time series data. Zhao *et al.* [11] proposed a similar approach using LSTMs for time-series multi-modal human activity recognition. Jeong *et al.* [12] proposed a novel federated semi-supervised approach for 2 cases: (i) where clients do not have any labels (ii) where data is partially labelled. In addition to these works, FL has also been applied to future generation of radio access networks such as O-RAN, but most of them focus on topics such as network slicing using reinforcement learning techniques [13]. Moreover, while widely used in natural language processing such as GPT [4], BERT [5] and in image classification such as SimCLR [6], self-supervised or unsupervised representation learning techniques are not well explored in the context of mobile networks and radio signal data.

Detecting whether user is *Indoors* or *Outdoors* has also been studied in context of mobile networks, most of the works however focus on using sensors available on user equipment [14], [15], while others [16], [17] use fingerprinting of radio signals such as Wi-Fi. Our work only uses radio measurements available on the network side, as our approach is intended to be deployed in network instead of user equipment.

To the best of our knowledge, the paper presents the first work that uses representation learning on radio measurements. The learned representations are then fine-tuned and transferred for supervised IOD.

III. Federated Representation Learning for Real Life Environment Sensing

This section presents how we exploit representation learning techniques combined with FL on large amount of unlabelled data which is easily acquirable in distributed mobile networks. We also show how to transfer the learned representation to a downstream task which is *Indoor-Outdoor* Detection (IOD).

A. Data description

The data used was collected by volunteers. Following [18] and [7], the data used consists of LTE signals which are 3GPP radio measurements gathered at base station (BS). They are Reference Signal Receiver Power (RSRP) and Timing Advance (TA). This collection was conducted during 33 Days with a frequency of 1 measurement per second, for a given user. The data also contains some breaks.Nevertheless, the collection was carried out in different situations and environments, such as home, outdoors, transport, forest..., and at different times of the day (day and night) and week (weekday and weekend). Furthermore, the collection was done passively, which means that while the collection application was running, users were acting normally and doing their usual activities,

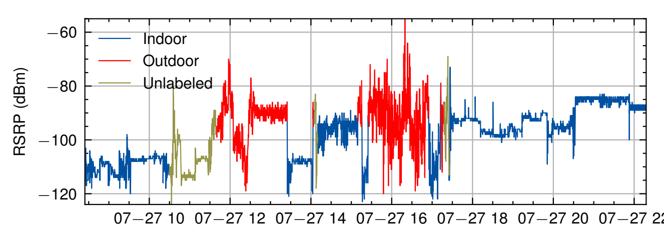

Fig. 1. RSRP variations over time

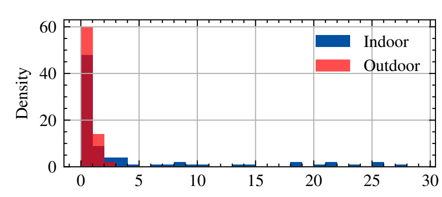

Fig. 2. Sojourn time distribution in I/O environment. Number of instances are as follows. Indoor: 1554010 - Outdoor: 134300 - Unlabelled: 232453

which they would do even if they were not collecting data. We therefore expect the collected data to be representative of real world data and the environment to be real-life. Fig. 1 shows the variation of RSRP data for both environments.

As seen in Fig. 2, users spend more time indoors than outdoors. This is because people mostly stay indoors at night, which is a big portion of time. Also, users in this dataset work indoors, unlike some users who can work outdoors. Also, most of the collected data is labelled. We also see that on average, users stay in the same environment for long period of time. The dataset is made of two parts, an unlabelled part denoted as $\mathcal{D}^u = \{(t_1, \boldsymbol{x}_1)...(t_m, \boldsymbol{x}_m)\}$ where $\boldsymbol{x}_i \in R^d$ is the i^{th} data point of dimension $d = 2$ features and t_i is the timestamp, and a labelled part denoted as $\mathcal{D}^l = \{(t_1, \boldsymbol{x}_1, y_1)...(t_n, \boldsymbol{x}_m, y_n)\}$ where $y_i \in \{1, \cdots, C\}$ is the label with $C = 2$ in this case, the label being either *Indoors* or *Outdoors*. As labels are difficult to obtain in real world, we investigate the impact of limitation of labelled data amount on the performances of our proposal.

B. Semi-supervised Federated Learning for IOD

With FL, ML models are trained locally in distributed entities, as opposed to centralized learning with all data. The parameters of FL models are then sent regularly to a central server where they are aggregated then returned. As mentioned in Section I, this is well suited for future mobile networks, which will be highly distributed with a desegregated architecture. One example with O-RAN, depending on the implementation, could be that a centralized entity may refer to Central Units (CU) and distributed entities (FL-clients) may refer to Distributed Unit (DU). The basic implementation of FL is supervised, where each FL-client has the labels of its data. However, getting labels is costly, and sometimes unfeasible, especially in mobile networks where crowdsourced data are collected online. To overcome this, we combine unsupervised representation learning method and supervised central method (with reduced amount of labelled data), that is time series classification, within FL architecture. In this system, only a small amount of labelled data is available and unlabelled data is not sent to the central server. Our approach for mobile user IOD, shown in Fig. 3, consists of 2 main steps:

1) **Unsupervised federated representation learning:** this step's goal is to learn meaningful representations using the large amount of unlabelled data which is available. The idea is to train a Denoising Autoencoder (DAE) [19] in a Federated Learning way to learn effective representations and capture relevant patterns. As in Alg. 1, this consists of 4 sub-steps repeated for a desired number of iterations T, where K is the number of FL-clients, W_j^t are the weights of the model j at iteration t, and W_G^t the resulted weights from the aggregation (denoted by G):

$$W_G^{t+1} = \frac{1}{K} \sum_{j=1}^{K} W_j^t \qquad (1)$$

   (1) Initialize DAE models of FL-clients with W_G^t then train
   (2) Send the learned DAE parameters W_j^t of each FL-client j to FL-server
   (3) Aggregate the parameters on FL-server using equation 1 into W_G^{t+1}
   (4) Send W_G^{t+1} to FL-clients and repeat from sub-step (1)

2) **Supervised IOD:** this step exploits the learned representations from step 1, using transfer learning techniques then training a supervised IOD with some labelled data. As shown in Alg. 1, this consists of sub-steps (5) and (6):
   (5) Initialize the first layers of supervised model with the encoder parameters learned in step 1
   (6) Train supervised model using labelled data D^l

Algorithm 1: Semi-supervised Federated Learning

1) Unsupervised federated representation learning
$W_G^1 \leftarrow IntilizeModel()$
foreach $t = 1, 2, \ldots, T$ **do**
  // Client Side
  foreach $j = 1, 2, \ldots, K$ **do**
    (1) $W_j^t \leftarrow DAEtraining(D_j^u, W_G^t)$
    (2) $SendParams(W_j^t)$ // To server
  // Server Side
  (3) $W_G^{t+1} \leftarrow AggregateModels(W_1^t, W_2^t, \ldots, W_K^t)$ using Eqn. 1
  (4) $SendParms(W_G^{t+1})$ // To clients
$W_G \leftarrow W_G^T$
2) Supervised IOD
$W_E \leftarrow GetEncoderParams(W_G)$
(5) $InitilizeSupervisedModel(W_E)$
(6) $IODclassifierTraining(D^l)$

C. Denoising Autoencoder DAE for representation learning

As mentioned before, for the unsupervised representation learning part we use Denoising Autoencoders (DAE) with LSTM and CNN, which are well suited for time series data. Autoencoders consist of 2 components: an encoder (Orange part in Fig. 3) which reduces the dimensionality of the input into a lower space representation with a bottleneck layer, and a decoder which uses that representation to reconstruct the

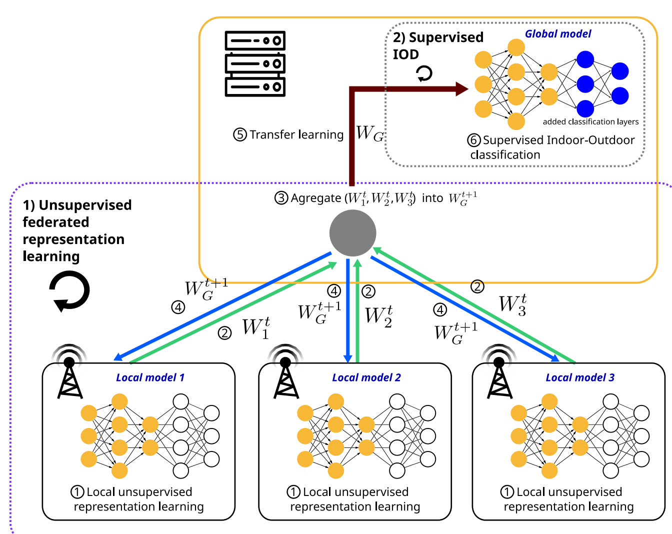

Fig. 3. Overview of our semi-supervised Federated Learning approach

input. Several loss functions have been used with autoencoder in the literature, such as the mean squared error (MSE) or mean absolute error (MAE). DAEs differ from the basic Autoencoders by adding a noise to data, with the goal of reconstructing the original input not containing the noise.

This adds some complexity which pushes the model to learn more meaningful representations because the denoising task needs a better data understanding, as compared to the task of compression [19]. Several types of noises can be added to the radio signal time series, i.e., we can mask some random samples. This obliges the model to learn and understand how do radio signals change relative to previous and next measurement, rather than memorizing the whole sequence. The same intuition is used in [5] for natural language processing. They explain that a model needs to learn the contextual information to be able to predict masked words in a text. In general, the task becomes harder with more noise because this requires the model to understand more complex features of data. However, it may have an opposite effect, where the task becomes too hard for the model to understand. Therefore, as a first step, we opted for only 3 types of noises, represented by g.

Let $X_i = \{\boldsymbol{x}_{i-L+1}, \ldots, \boldsymbol{x}_i\}$ be the i^{th} input time series of length L, the type of noises are:

1) Adding a random Gaussian noise to help model become invariant to small variations: $g(X_i) = X_i + \epsilon$, $\epsilon_j \sim \mathcal{N}(0, \sigma^2), \quad j = 1, \ldots, L$ where $\sigma = 0.1$.
2) Masking random samples of the time series: $g(X_i) = X_i \cdot m, \quad m_j \sim \text{Bernoulli}(p), \quad j = 1, \ldots, L$ where $p > 0.8$. As explained before, this pushes the model to learn and understand: how do radio signals measurements change relatively to previous and next measurement.
3) Mask random sub-sequences of length l where $l < 0.1L$. As compared to second noise, a whole sub-sequence is masked instead. While, second noise helps to understand the relative values of the measurements, this helps the model to also have a global vision on the sequence which is necessary to interpolate a whole sub-sequence.

Only one of these noises is randomly applied to the input each time, rather than all at once. The DAE is then trained by minimizing the reconstruction loss function MAE (Eq. 2), where f_{enc} and f_{dec} represent encoder and decoder. $\hat{X}_i = f_{dec}(f_{enc}(g(X_i)))$ represents the output of the autoencoder.

$$\mathcal{L} = \frac{1}{L \cdot d} \sum_{j=i-L+1}^{i} \sum_{k=1}^{d} |x_{j,k} - \hat{x}_{i,k}| \tag{2}$$

Fig. 4 shows the implementation of LSTM DAE architecture.

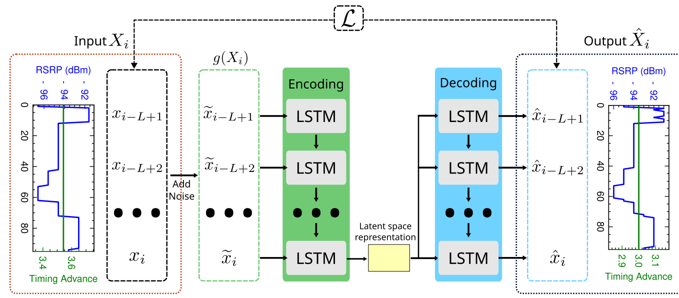

Fig. 4. LSTM Denoising Autoencoder architecture

Supervised Learning (SL): in this scenario, we only use labelled data $\mathcal{D}^l$ using supervised techniques, namely LSTM and CNN for time series classification. This serves as baseline for classification to study the benefit brought by representation learning, as in this scenario we don't use unlabelled data.

Centralized Semi-supervised Learning (C-SSL): in this scenario, we use unlabelled data $\mathcal{D}^u$ as a single chunk to pre-train a single Denoising Autoencoder (DAE) which parameters are transferred to the IOD classifier that is next fine-tuned by training it similarly to SL scenario using $\mathcal{D}^l$. The performances achieved with C-SSL are used as reference for the proposed approach with decentralized data.

Federated semi-supervised Learning (FL-SSL): This corresponds to the proposed approach. For this scenario, we split the unlabelled data into K smaller chunks $\{\mathcal{D}^u_1, ..., \mathcal{D}^u_K\} = \mathcal{D}^u$ where $\mathcal{D}^u_i$ represents data from same distributed entity (FL-client) containing multiple Cell IDs of same group, this mimics a real network data. We then train the model using the approach described above to learn representations which are transferred to IOD classifier similarly to C-SSL scenario. As default, our tests use $K = 3$ distributed entities.

IV. Experiments

Here, we first explain the setup and then discuss the results.

A. Scenarios

We want to exploit FL with representation learning to leverage unlabelled data that are easy to acquire. To understand the benefits of our method, 3 scenarios are defined:

B. Configuration setup

To evaluate the model performances for different scenarios, we use forward-chaining cross-validation [7]. It is more suitable for sequential data as it avoids any kind of data leakage which occurs if data were randomly shuffled and split like in basic K-Fold cross-validation. This validation method consists in splitting data into N blocks without shuffling and launching N-1 experiments. We opted for $N = 6$. For each experiment, we train the same model 5 times with different random initialization. The performances of IOD models is done using Balanced Accuracy and F1-score. We then compute the mean and the standard deviation of these metrics over the five experiments. Moreover, to investigate the added value of using unlabelled data, we conduct a study on the impact of different amounts (%) of labelled training data used. The percentages are $\{30\%, 50\%, 100\%\}$, where 100% means that all labelled data are used. Our objective is to study the impact of limited data labels on the performance.

C. Performance comparison

Table I shows that using unlabelled data to pre-train the model improves IOD performances in all scenarios. In particular, for the case of only 30% of labelled data, we can observe

Labeled data	Method	Case	Accuracy	F1-score	Training Time (s)
30%	LSTM	SL	91.94 $\pm$1.28	91.75 $\pm$1.33	299
		C-SSL	$\mathbf{94.57} \pm 0.70$	$\mathbf{94.51} \pm 0.72$	3638
		FL-SSL	$\mathbf{94.43} \pm 0.42$	$\mathbf{94.35} \pm 0.45$	3619
	CNN	SL	88.16 $\pm$3.72	87.63 $\pm$4.08	272
		C-SSL	$\mathbf{91.72} \pm 0.45$	$\mathbf{91.58} \pm 0.45$	3530
		FL-SSL	$\mathbf{91.58} \pm 0.87$	$\mathbf{91.44} \pm 0.90$	3466
50%	LSTM	SL	96.11 $\pm$0.42	96.10 $\pm$0.44	311
		C-SSL	$\mathbf{97.26} \pm 0.34$	$\mathbf{97.25} \pm 0.34$	3668
		FL-SSL	$\mathbf{97.48} \pm 0.09$	$\mathbf{97.48} \pm 0.09$	3697
	CNN	SL	93.48 $\pm$1.28	93.39 $\pm$1.33	290
		C-SSL	$\mathbf{94.96} \pm 0.58$	$\mathbf{94.95} \pm 0.59$	3591
		FL-SSL	$\mathbf{94.93} \pm 0.48$	$\mathbf{94.92} \pm 0.49$	3519
100%	LSTM	SL	97.56 $\pm$0.11	97.56 $\pm$0.11	337
		C-SSL	$\mathbf{97.89} \pm 0.20$	$\mathbf{97.89} \pm 0.20$	3716
		FL-SSL	$\mathbf{97.88} \pm 0.15$	$\mathbf{97.88} \pm 0.15$	3777
	CNN	SL	95.24 $\pm$2.48	95.20 $\pm$2.54	322
		C-SSL	$\mathbf{96.76} \pm 0.47$	$\mathbf{96.76} \pm 0.47$	3530
		FL-SSL	$\mathbf{96.99} \pm 0.06$	$\mathbf{96.99} \pm 0.06$	3552

TABLE I

PERFORMANCE COMPARISON OF THE THREE IOD MODELS FOR DIFFERENT AMOUNTS OF LABELLED DATA

(for LSTM and CNN respectively) an increase in F1-score of $+2.76\%$ and $+3.95\%$ in C-SSL and $+2.60\%$ and $+3.81\%$ in FL-SSL as compared to SL. Fig. 5 shows that improvement brought by unlabelled data is less significant with increasing labelled data, which is expected. Certainly, with ample labeled data, supervised learning can theoretically reach peak performance with minimal room for improvement. As data labelling is expensive, improvements with limited amounts of labelled data remain profitable for network operators. Fig. 5 also shows a lack of performance difference between FL-SSL vs. C-SSL. However, FL-SSL leverages available unlabeled data within the network without the mentioned drawbacks of central data transfer. As observed in Table I, both the FL modes have training times around ten times longer than the SL mode one. This is due to the additional processing of the unlabelled data. However, this is not detrimental if the training is done in background. Moreover, we observed that LSTM outperformed CNN. In our case, the most recent measurements are more crucial for predicting the environment than older ones. LSTM, with its forget gate, can prioritize important data across the sequence, while CNN tends to evenly emphasize neighbouring data in a sequence.

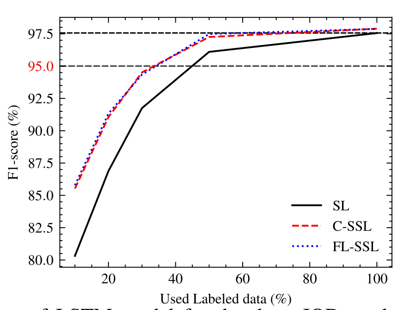

Fig. 5. F1-score of LSTM model for the three IOD models with different amount (%) of labelled data

Let us analyse more deeply the performance of the three algorithms by evaluating their convergence time in terms of the number of epochs. For both C-SSL and FL-SSL, the DAE (Denoising Autoencoder) is trained for a total of 200 epochs. In FL-SSL, the parameters are sent to the server and

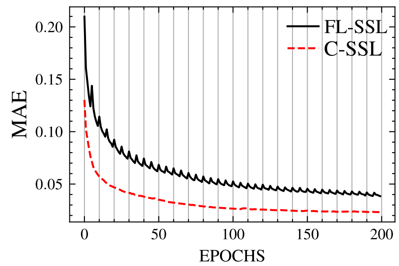

Fig. 6. DAE reconstruction loss (MAE) for C-SSL and FL-SSL

aggregated before returned to FL-clients every 5 epochs. Fig. 6 shows that the reconstruction loss (MAE) of DAE converges pretty well for both cases. In C-SSL the MAE converges faster, which is expected, as a single model is observing all data, where in FL-SSL it needs to aggregate the model periodically, resulting in weight readjustments. This also explains the loss spikes with FL-SSL after every aggregation round, as seen in Fig. 6. Despite, this small difference, the IOD performances are similar because, in transfer learning, the performances of the source task (DAE) don't need to be as perfect. Indeed, in some cases, over-fitting the source task can have the opposite effect and can hinder the transfer and reduce the downstream task performance, but this did not occur in our case.

D. Signalling overhead comparison

Now, let us compare the network signalling overhead of C-SSL vs. FL-SSL approach. Lets estimates, for a training phase (e.g. each day) how much data can be collected by one FL-client for different numbers of users in one day and compare it with the number of parameters computed and exchanged. The number of the trained LSTM autoencoder parameters is 36738. This amount of parameters is exchanged between each FL-client and the FL server during each of the $200/5 = 40$ rounds. In comparison in case of C-SSL, the amount of data transmitted by the FL-client to the central server is $\sim 232k \times M \times 2$ features if the FL-client serves M mobile users. This results in $\sim 464k \times M$ instances to be transmitted to central server. To estimate the signaling overhead, we derive the ratio of total number of parameters exchanged (in FL-SSL) divided by the amount of data exchanged (in C-SSL). Fig. 7 shows ratio vs. number of users served in case of one single FL-client. We assume that parameters and instances are quantified with same number of bits. Our curve shows that for more than 9 users per FL-client, it becomes beneficial to use our approach FL-SSL instead of C-SSL. We can observe that with 100 users attached to a cell in mobile network, our approach saves 90% of data from being transported as compared to the C-SSL.

E. User behavioural knowledge based optimizer

The results showed that we can get acceptable performance with only 30% of the total number of labelled data. To improve further, we use UBO as proposed in [7]. It consists in using prior knowledge, about typical human mobility behaviour, to correct behavioural anomalies in ML model predictions. It exploits the knowledge that users don't switch environment back and forth in a short amount of time. UBO corrects

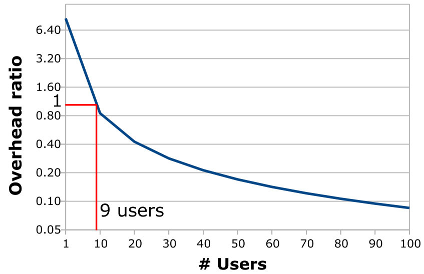

Fig. 7. Overhead ratio for 40 rounds: amount of parameters needed to be sent (in FL-SSL) divided by data amount (in C-SSL) for different no. of users

predicted environment, if there is an environment switch in less than $30s$ and if the prediction confidence represented by the softmax output of the classifier is lower than a given threshold. Fig. 8 shows the obtained F1-score for LSTM with 30% of labelled data. The figure shows that UBO improves the model performances, especially for the case of C-SSL and FL-SSL where we exceed 95% which is a very good target performance for real world IOD. Therefore, even with reducing the labelled data to 30% we were able to get the target performances by combining federated representation learning with UBO.

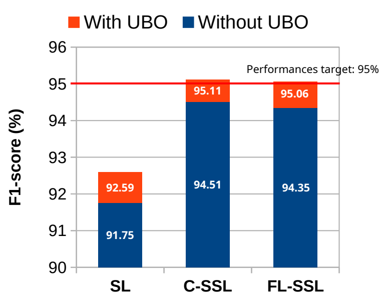

Fig. 8. F1-score for LSTM with 30% of labelled data for the 3 IOD models

V. Conclusion

We have exploited Federated Learning and distributed architecture of future mobile network that are able to collect a large amount of unlabelled radio measurement data for representation learning. We have shown that these representations can be effectively transferred to a target task, which is Indoor-Outdoor Detection, where we improved the performances, especially for reduced number of labelled data. We have also shown that we can obtain similar results with the models that learn the representation in a centralized way without the drawbacks of centralized learning. By combining this approach with User Behaviour Optimizer, we were able to obtain a F1-score of 95.06% with only 30% of the total data amount. These results show the potential of both Federated Learning and unsupervised representation learning, which are promising techniques that leverage large unlabelled datasets. Moreover, the learned representations can be transferred to tasks other than IOD. In future work, we plan also to explore other more advanced representation learning techniques or source tasks other than Denoising Autoencoder, such as forecasting or predicting the time elapsed between two time series. More generally, we would like to explore tasks that will help the model to learn the most meaningful representations.

References

[1] R. Sung, S.-h. Jung, and D. Han, “Sound based indoor and outdoor environment detection for seamless positioning handover,” *ICT Express*, vol. 1, no. 3, pp. 106–109, 2015.

[2] H. Zou, H. Jiang, Y. Luo, J. Zhu, X. Lu, and L. Xie, “Bluedetect: An ibeacon-enabled scheme for accurate and energy-efficient indoor-outdoor detection and seamless location-based service,” *Sensors*, vol. 16, no. 2, p. 268, 2016.

[3] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, “Communication-efficient learning of deep networks from decentralized data,” in *Artificial intelligence and statistics*. PMLR, 2017, pp. 1273–1282.

[4] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever *et al.*, “Improving language understanding by generative pre-training,” 2018.

[5] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” *arXiv preprint arXiv:1810.04805*, 2018.

[6] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, “A simple framework for contrastive learning of visual representations,” in *International conference on machine learning*. PMLR, 2020, pp. 1597–1607.

[7] S. A. Hamideche, M. L. A. Morel, K. Singh, and C. Viho, “Indoor-outdoor detection using time series classification and user behavioral cognition,” in *2022 14th IFIP Wireless and Mobile Networking Conference (WMNC)*. IEEE, 2022, pp. 7–14.

[8] J. Mills, J. Hu, and G. Min, “Communication-efficient federated learning for wireless edge intelligence in iot,” *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 5986–5994, 2019.

[9] D. C. Nguyen, M. Ding, P. N. Pathirana, A. Seneviratne, J. Li, and H. V. Poor, “Federated learning for internet of things: A comprehensive survey,” *IEEE Communications Surveys & Tutorials*, vol. 23, no. 3, pp. 1622–1658, 2021.

[10] O. Aouedi, K. Piamrat, G. Muller, and K. Singh, “Federated semisupervised learning for attack detection in industrial internet of things,” *IEEE Transactions on Industrial Informatics*, vol. 19, no. 1, pp. 286–295, 2022.

[11] Y. Zhao, P. Barnaghi, and H. Haddadi, “Multimodal federated learning on iot data,” in *2022 IEEE/ACM Seventh International Conference on Internet-of-Things Design and Implementation (IoTDI)*. IEEE, 2022, pp. 43–54.

[12] W. Jeong, J. Yoon, E. Yang, and S. J. Hwang, “Federated semi-supervised learning with inter-client consistency & disjoint learning,” *arXiv preprint arXiv:2006.12097*, 2020.

[13] H. Zhang, H. Zhou, and M. Erol-Kantarci, “Federated deep reinforcement learning for resource allocation in o-ran slicing,” in *GLOBECOM 2022-2022 IEEE Global Communications Conference*. IEEE, 2022, pp. 958–963.

[14] A. Esmaeili Kelishomi, A. Garmabaki, M. Bahaghighat, and J. Dong, “Mobile user indoor-outdoor detection through physical daily activities,” *Sensors*, 2019.

[15] Y. Zhu, H. Luo, F. Zhao, and R. Chen, “Indoor/outdoor switching detection using multisensor densenet and lstm,” *IEEE Internet of Things Journal*, vol. 8, no. 3, pp. 1544–1556, 2020.

[16] K. Chintalapudi, A. P. Iyer, and V. N. Padmanabhan, “Indoor localization without the pain,” *Proceedings of the sixteenth annual international conference on Mobile computing and networking*, 2010. [Online]. Available: https://api.semanticscholar.org/CorpusID:10281765

[17] G. Shtar, B. Shapira, and L. Rokach, “Clustering wi-fi fingerprints for indoor–outdoor detection,” *Wireless Networks*, vol. 25, pp. 1341–1359, 2019.

[18] I. Saffar, M. L. A. Morel, K. D. Singh, and C. Viho, “Semi-supervised deep learning-based methods for indoor outdoor detection,” in *ICC 2019-2019 IEEE International Conference on Communications (ICC)*. IEEE, 2019, pp. 1–7.

[19] D. Erhan, A. Courville, Y. Bengio, and P. Vincent, “Why does unsupervised pre-training help deep learning?” in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 2010, pp. 201–208.