Semi-supervised Deep Learning-based Methods for Indoor Outdoor Detection

Illyyne Saffar, Marie-Line Alberi-Morel, Kamal Deep Singh, César Viho

Semi-supervised Deep Learning-based Methods for Indoor Outdoor Detection

Illyyne Saffar
Nokia Bell Labs
Nozay, France
illyyne.saffar@nokia.com

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

Kamal Deep Singh
Laboratoire Hubert Curien,
Université de Saint-Etienne, Jean Monnet,
Saint-Etienne, France
kamal.singh@univ-st-etienne.fr

Cesar Viho
IRISA-UR1,
Rennes, France
Cesar.Viho@irisa.fr

***Abstract*—Detecting whether a mobile user is indoor or outdoor is an important issue which significantly impacts user behavior contextualization and mobile network resource management. Indoor Outdoor Detection (IOD) can be performed within mobile networks or in user terminals. Implementing IOD in mobile networks is interesting for operators as it is less costly, easier to deploy, and more energy efficient with centralized computing. This paper investigates hybrid/semi-supervised Deep Learning-based methods for detecting the environment of an active mobile phone user. They are based on both labeled and unlabeled large real radio data obtained from inside the network and from 3GPP signal measurements. We empirically evaluate the effectiveness of the semi-supervised learning methods using new real-time radio data, with partial ground truth information, gathered massively from multiple typical and diversified locations (indoor and outdoor) of mobile users. We also present an analysis of such schemes as compared to the existing supervised classification methods including SVM and Deep Learning.**

***Index Terms*—Environment Classification, Deep Learning, Indoor Outdoor Detection, Semi-Supervised Learning, 3GPP radio measurement, crowdsourcing, real user activity.**

I. Introduction

5G is the next evolution of mobile networks for accommodating the ever-growing user-demands, services and applications, guaranteeing a better Quality of Experience [1]. The improvement is possible thanks to additional cognition from information on user behavior, obtained through user behavior contextualization. The idea is to inject cognition learned from the consuming habits of individuals and communities as well as their behaviors into mobile 5G networks. The additional knowledge will help them grow smarter and be more efficient when faced with the increasing complexity of network management combined with numerous new applications and their heterogeneous needs. As a first step, in this paper we focus on detecting the environment of a mobile user connected to a cellular network. More precisely, to infer whether the mobile user is indoor or outdoor. The Indoor Outdoor Detection (IOD) is a cornerstone of the user behavior contextualization, which in turn can be used for learning user behavior, adapting the mobile network resources, etc. [2] [3].

Machine Learning (ML) is one of the key technologies to be used for user behavior contextualization, which in our case is detecting the user environment (either indoor or outdoor). Actually, ML extracts information from data to look for patterns, and then uses them as predictor functions when analyzing future data. Building such a representative knowledge requires a highly representative and diversified dataset. However, most of IOD works, showing excellent performances, are based on datasets collected in a "drive-test" mode, which is unfortunately limited to specific environments. This often makes it difficult to be generalized to a real user behavior. For this reason, to design an efficient IOD tool which is able to capture the real behavior of mobile users, the user data shall be collected within mobile networks using a crowdsourcing approach [4], [5]. This approach consists in gathering real and large network measurement data, which is derived within network or is sent to network by multiple mobile phones (or other connected devices) using standardized procedures. Indeed, due to their small size and popularity, mobile devices allow users to access wireless networks anywhere and anytime, while doing various activities in all kinds of environments. Additionally, it allows for a continual and fine-grained spatio-temporal monitoring and analysis. Thus, ML algorithms trained on datasets collected in crowdsourcing mode allow to learn very diverse real-world environments.

In literature most of IOD works mainly use the received signal power as input for the IOD model [12], [6]. Actually, this signal is highly correlated to user environments. However, only using it for IOD is not sufficient to guarantee IOD's good performance. Alone, it is not sufficient especially while facing ambiguous measurement points in mobile environments or in ambiguous user locations relative to eNB (evolved Node B). Hence, there is a need to vertically expand the dataset used to solve the IOD issue by adding other signals. Therefore, we propose to use new input signals which are related to the quality, the UE location and the user mobility. As a consequence, in this paper IOD uses Reference Signal Received Power ($RSRP$), Channel Quality Indicator (CQI), Timing Advance (TA) and Cell Id. They represent 3GPP signals or indicators. $RSRP$ and CQI are measured by UE and sent to eNB via standardized protocols. TA and Cell Id are derived inside the network when the user is connected. Note that IOD done in the network should consider the constraint of minimal human intervention.

For this, we propose to study semi-supervised Deep Learning-based methods for training automatic IOD classifiers. These methods are a mix of supervised and unsupervised approaches which can learn from partially labeled dataset. Such dataset reduces human intervention to the minimum possible. Indeed, the labeled data, used for ML training, is

either tagged manually or automatically. Manual data tagging can be expensive and complex and even unfeasible for certain mobile operators if they have to tag all the collected data. We investigate therefore three semi-supervised approaches that 1) learn from both labeled and unlabeled data and 2) make use of information on received power, quality, distance and mobility. The promise of semi-supervised learning is that we can get our ML algorithm to learn from "unlabeled" data, which in turn is easier to obtain. A single unlabeled example may be less informative than a single labeled example. Nevertheless, we can get tons of such less informative examples, by collecting huge crowdsourced unlabeled signals. Now, if our algorithms can exploit this unlabeled data effectively, then it will enable us to learn more possible environment types related to the user behavior. That way the data will closely reflect the users' habits. Coming back to ML, today, deep learning surpasses all classical ML algorithms [7], [8]. As said in [9]:"The more data we have, the wider the Neuronal Network is, the better the performances are."

The rest of paper is organized as follows. Section II presents the related works on IOD and mobility estimation. Section III analyses the input features (signals) selected for the dataset. Section IV, studies the impact of the mobility and distance information on IOD performance. Sections V and VI describes and compares three semi-supervised approaches: Cluster-then-label, Co-Training and Self-Training methods. Section VII evaluates the performance of Self-Training, shown to be the best, versus the volume of both labeled and unlabeled data. Finally, we conclude this paper with a brief discussion of open challenges, with a view to future research directions.

II. Related work

Works in literature have addressed IOD much more from mobile point of view than from the network or infrastructure point of view. In [10] authors propose to use a threshold of a signals set collected from some phone sensors related to: radio signals, cell signal strength, light intensity as well as the magnetic sensor to infer whether the mobile user is indoor or outdoor. Like [10], [11] also addressed IOD using the same set of sensors plus the sound intensity, battery temperature and the proximity sensor. The investigated IOD solution is based on ML algorithms and more precisely a semi-supervised ML approach. Their solution, implemented on different android devices shows a 92.33% of accuracy and provides the highest detection performance in comparison with existing methods including supervised classifier. This solution shows the interest of using semi-supervised ML approaches for IOD. Thus, this motivates us to try similar solutions on the network side.

In [12], authors have considered IOD at network side as a classification issue. Once the indoor or outdoor location is detected, it helps with other signals to localize the mobile user by estimating its longitude and latitude in a most possible accurate ways. For the IOD classification task, they used $RSRP$ and $RSRQ$ signals and tested many algorithms: Support Vector Machine (SVM), logistic regression and random forest. SVM was the solution retained since it performed best.

In [13] authors optimize the use of radio measurements in wireless networks. Literally, they use radio signal measurements collected in different situations of mobility with varying speed (low, medium, high). They dynamically estimate the signal attenuation. This in turn helps them to efficiently classify the mobile user environment (pedestrian, incar, non-moving) and it finally improves the handover process. This confirms that the user mobility is strongly correlated to his environment. Nevertheless, this proposition is still at an early stage and it has not been thoroughly developed yet.

Many works in literature have addressed the user mobility estimation. In [14], authors use RSRP measurement according to the speed dependent time variations of shadowing to compute the UE speed. They propose two methods: either based on a spectral analysis or based on a time-based spectrum spreading. For both methods the variation is compared to a reference curve or a look-up table (database) and according the difference analysis the UE speed is computed. In [15], authors propose a method for estimating the UE mobility, that relies on UE history information about the UE cell sojourn time. Then neighbouring eNBs exchange among them the learned network topology as well as the UE sojourn time history. Using such information, the eNB classifies the speed to one of the three mobility classes defined by 3GPP. Both methods for mobility estimation in [14] and [15] have shown good results, however they estimate the speed of UEs in some specific use cases. In addition, they are complex to setup.

For us, the issue is not to study the mobility itself, but rather to exploit a simplified mobility indicator in order to improve the IOD performance. For this reason, we employ the standardized 3GPP procedure of mobility estimation. Actually, this low complexity approach is advantageously simple. But, according to literature, the 3GPP procedure is not precise enough if one requires an accurate mobile speed estimation [15]. However, in our study, we aim to evaluate whether the mobility indicator, as an additional input, can bring enough rich information to improve the performance of IOD system.

In [16], [17], the user mobility is estimated and classified to one of the three categories (Normal, Medium, High). This estimation using standard 3GPP procedures is done as follows:

- 1- Compute the number of handovers or cell re-selections (denoted by NCR) during a sliding time window (denoted by TCR_{max}).
- 2- If a UE's NCR count is smaller than a threshold NCR_{medium}, then the UE's mobility state is determined as "Normal". If the UE's NCR is greater than NCR_{medium} but less than NCR_{high}, the state is determined as "Medium". Finally, if the UE's NCR is greater than NCR_{high}, then the state is determined as "High".

III. Dataset collection in crowdsourcing mode

In machine learning domain, data collection is the first main step for building the desired knowledge about the user environment. For this goal, we opt for a dataset composed of radio signals ($RSRP$, CQI), temporal features (TA, Time), a Mobility Indicator (MI), and finally the environment label

when it is known. Thus, our dataset is composed of a vector of following 6 features:

- Time: recording time of signal or burst data arrival (ms).
- $RSRP$: the average received power of the Reference Signal (RS). The $RSRP$ value lies between -140 dBm to -44 dBm [18].
- CQI: Channel Quality Indicator that is used to indicate the most appropriate transmission modulation and coding scheme to be used [19].
- TA: Timing Advance is used to control UL signal transmission timing [20].
- MI: the number of the Cell ID changes ($NCID$) in a sliding window of a given duration (TCR_{max}) [16], [17].
- Label: Indoor or Outdoor label in case it exists.

To estimate the value of mobility indicator, we derive the value of sliding window duration TCR_{max}. In urban environments, considering macro-cells, we assume a typical separation distance of around $900m$ between two base stations. Assuming this distance, we compute that a mobile user with a typical average speed of $30km/h$ moves to an other cell at least once at $100s$, excepting a few rare cases. Consequently, $TCR_{max} = 100s$ of history on visited cells (UE History Information) is sufficient for starting to observe cell ID changes.

However, this is valid only for urban environments. Figure 1 plots the time to cross a cell vs. typical user speed for three environment types - urban and suburban macrocell and small cell - assuming a trajectory model where a user follows a straight line. We assume a typical separation distance between two base stations of 1.5 km for suburban environments and 350 m for small cell deployments. As expected, we observe that the crossing time is a function of the environment. For a speed of $30km/h$, TCR_{max} value is around 100s. TCR_{max} has a lower value in the case of small cell deployments because such cell-types have smaller radius.

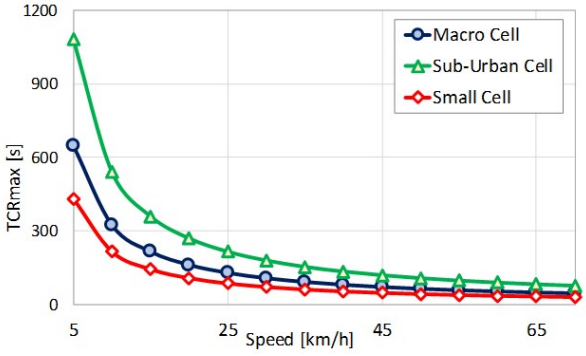

Fig. 1. Time to cross a cell (s) vs. user speed (km/h)

The crowdsourcing mode for data collection is a recent concept that is used by a large researcher community as well as big firms like Google, Netflix and Amazon. Actually, such data collection mode allows to get data more quickly, cheaply and in large quantities more than ever before. But most importantly, it allows to reflect better the users' behaviors by providing a huge diversity of their experiences. This, in turn, improves the ML performance because with this data collection mode we get more highly representative data.

In our case, our dataset, collected using the crowdsourcing mode, has been gathered since October 2017 on wards, 24h/7, with an average of 1 measurement per 15 seconds while the mobile phone session is active and 1 measurement per 2 minutes otherwise. Thus, we have collected around 2M lines of data per user. This number is still growing. In this paper, we used 250K lines of data which is specific to LTE networks. This dataset is made of 30% of labeled data and 70% of unlabeled data. The collection has been performed in many different indoor and outdoor environments. Indoor corresponds to the following locations: at home, in restaurant, in cafe, at work or in other types of building, etc. Whereas, outdoor is associated to forest, streets, parks, mountain and beach, to a pedestrian, a running user, or a user in car moving with high speed, etc. The gathering was done in many cities and places like countryside, small cities, metropolis, and different countries, but for this paper we are only studying data collected in France. This long collection period allows us to have data reflecting all weather types.

IV. Mobility and distance impact on IOD

In this section, we study the impact of adding two additional input features referred to as MI and TA to existing $RSRP$ and CQI on IOD performance. MI and TA represent respectively the mobility type and the distance between user and eNB. To allow fair comparison, the evaluation is done with the classical ML algorithms, used in [12], i.e., Support Vector Machine (SVM), with a supervised Deep Learning and with a clustering algorithm.

$RSRP$ and CQI are radio metrics directly linked to the mobile environment as they represent the extent of environment attenuation. But, using them is not enough to correctly classify some ambiguous points. For example, consider a mobile user travelling in train. The user is considered as outdoor meanwhile the received signal strength is bad because of not only the high speed of the train (Doppler effect), but also because of the surrounding structures. Indeed, the metal windows and carriages of train cause a significant attenuation of radio signal power. For example, in 2 GHz frequency band, the penetration losses from train carriages are usually in the range of 20 to 35 dB [21]. Figure 2 shows the indoor and outdoor Cumulative Distribution Functions (CDFs) derived from the crowdsourced data. The blue curve with full-line represents the data taken in normal speed: only from static locations and low speed points. The blue dotted curve depicts the data collected from all mobile locations (normal, medium, high). We note that the dotted line is closer to the indoor CDF. This leads to superimposed points located at the beginning of the tails of both the CDFs. These overlapped points are coming mainly from either deep indoor positions or high or medium speed mobile positions, thus, creating ambiguity.

Figure 3 shows the CDF of mobility indicator values for different environments (car, pedestrian, buildings, train, mall, bus) for $TCR_{max} = 100s$. As expected, the curves imply that the number of cell ID changes ($NCID$) is correlated with the environment type. Indeed, the indoor user (e.g. in

buildings) either doesn't change cells or changes very few when he is located at borders of multiple cells. However, when he moves outdoor (e.g in transportation), the number of handovers increases as he covers large distances. Note that the figure implies that $NCID$ is smaller in pedestrian case than in mall case. One reason is that in some cases relatively longer walks occur in malls.

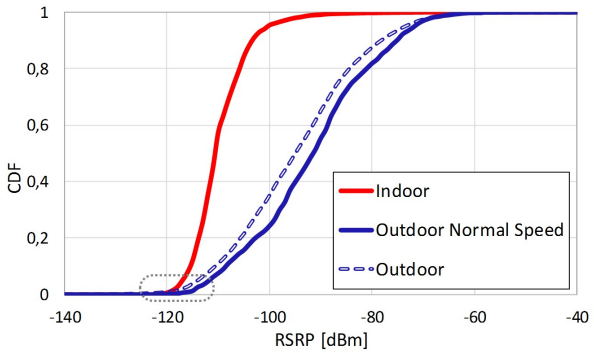

Fig. 2. Empirical CDFs for measured $RSRP$. (full) outdoor static and low speed, (dotted) outdoor variable speed.

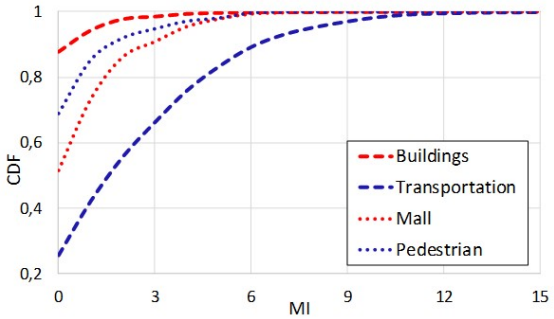

Fig. 3. Cumulated Distribution Function of the mobility indicator vs. Environment - one user

Consequently, the insertion of TA and MI should eliminate the ambiguities as they provide additional information that is relevant to the IOD system. Indeed, with them the IOD system exploits information on the distance of mobile users from the base station and is aware of their mobility type, respectively. On one hand, TA would help to classify the ambiguous points which correspond to, for e.g., measurement points with low $RSRP$, but near to eNB. On the other hand, the user mobility highly correlated to the user environment will ease the classification of outdoor measurement points with low $RSRP$, for example, while inside a high speed train. The indoor user moves slowly as compared to outdoor where he can move more quickly. Therefore, using both the additional signals can help to classify the ambiguous measurement points and improve the overall performance of the supervised classifier. However, a question may be asked: how much do these parameters contribute to IOD performance? Figure 4 depicts the relative optimal ordering of the four input features related to their relevance for IOD. The order of these items is obtained using "Extra-Trees-Classifier" algorithm. It reveals that $RSRP$ will contribute most to the IOD performance. The ranking scores of TA and MI are close. They thus impact the IOD performance almost identically. Furthermore, both signals together will contribute a little higher than $RSRP$ and CQI combined. Thus, an improvement in IOD performance is expected by introducing these two additional parameters.

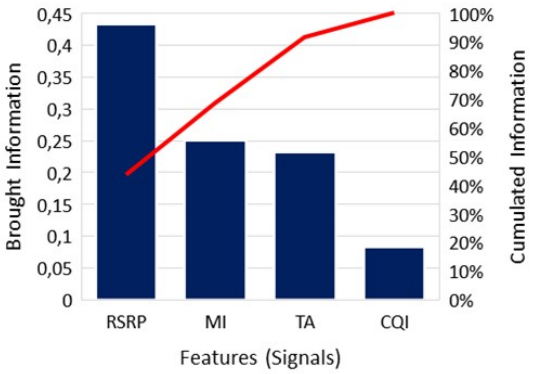

Fig. 4. Feature ranking based on cumulative information brought by them.

For the evaluation, the comparison is done in three cases where different structures of learning datasets are examined: the first one contains only $RSRP$ and CQI, the second one includes in addition the timing advance, but not the mobility indicator and the third one includes all data. In each case, the model is trained and evaluated on labeled data which is split into two subsets composed of 70% of data for training and 30% of data for tests and validation. As shown in the table I the user mobility is correlated to the IOD issue. Added to $RSRP$ and CQI, TA and MI enhance the classical machine learning performance with up to 8% of gain, approximately. We note also that Deep Learning (DL) outperforms the other classical machine learning algorithms.

Algo	RSRP-CQI		RSRP-CQI-TA		RSRP-CQI-TA-MI	
	Acc.	F1-S.	Acc.	F1-Sc.	Acc.	F1-S.
kMeans	78.73%	75.81%	66.61%	45.93%	75.83%	67.38%
Logis. Regress.	84.63%	82.26%	87.59%	85.93%	89.67%	88.44%
SVM	85.54%	**83.71%**	90.17%	89.11%	92.32%	91.44%
DL	**85.60%**	83.66%	**93.45%**	**92.77%**	**95.72%**	**95.30%**

TABLE I
CLUSTERING AND SUPERVISED CLASSIFICATION PERFORMANCE: ACCURACY & F1-SCORE VS. TIME ADVANCE & MOBILITY INDICATOR

V. DEEP LEARNING-BASED SEMI-SUPERVISED APPROACHES

Assuming that we have a sufficiently powerful learning algorithm, one of the most reliable ways to get better performance is to feed the algorithm with more data. Indeed, the quality of the model is generally constrained by the quality and the volume of the training data. DL and other modern nonlinear machine learning techniques get better with more data. Thus, there is a need to look for a way to enlarge volume of the training data. The idea is then to use unlabeled data, which is easy to obtain, and mix it with available labeled data, which is costly to obtain, for classifier training. Hybrid and semi-supervised approaches are the best candidates for this.

In this section, we compare and discuss the IOD performance using semi-supervised IOD approaches. As investigated

in [22], [23], [24], [25], [11], we consider the 3 classic following approaches of hybrid learning that make use of both the labeled and the unlabeled data:

- Cluster-then-label: a clustering method is used to label the unlabeled data.
- Co-training: multiple supervised classifiers learn from each other's outputs.
- Self-training: a supervised classifier trained on a small labeled dataset learns iteratively from its own classification of additional unlabeled data.

We evaluate the above 3 ways of learning using our dataset. Let S_{Total} be the total dataset made of:

$$S_{Total} = S_{Labeled} \cup S_{Unlabeled}$$

where $S_{Labeled} \in \mathbb{R}^6$ is the subset of the labeled data and $S_{Unlabeled} \in \mathbb{R}^5$ is the subset of the unlabeled data. Note that Card $(S_{Unlabeled}) \approx 3 \times$ Card $(S_{Labeled})$ in case of our collected data, where Card() gives the number of data points. For the performance evaluation on new environments unknown to the classifier we use $S_{Test} \in \mathbb{R}^6$, where $S_{Test} \not\subset S_{Total}$ and Card $(S_{Test}) \approx \frac{1}{3} \times$ Card $(S_{Labeled})$.

A. Cluster-then-Label

Our proposed system of Cluster Then Label (CTL) approach is composed of two main modules described in Figure 5. The first module handles the unlabeled data by applying a clustering algorithm on S_{Total} to make emerge 2 clusters: indoor and outdoor. We use labels of $S_{Labeled}$, as well as the priori information that users are much more indoor than outdoor, to associate labels to $S_{Unlabeled}$. Then an optimizer is used to correct the wrong labels, as much as it can, during the clustering phase. For correction, we use the idea that a user can not change his environment twice in 30 seconds. This is because a user cannot change its environment two times so quickly. The second module uses $S_{Labeled}$ and also newly labeled data of $S_{Unlabeled}$ to train a supervised classifier.

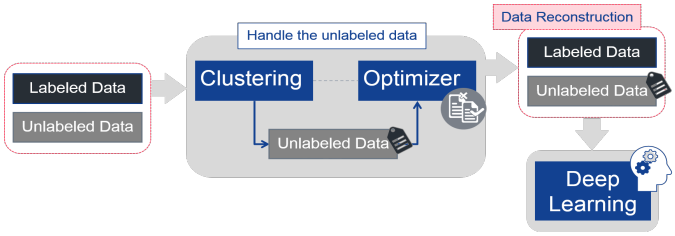

Fig. 5. The Cluster-then-label semi supervised approach model

We evaluate the CTL method with different clustering algorithms (K-Means, Expectation Maximization, Hierarchical Clustering, Bayesian Gaussian Mixture (BGM)). BGM showed better performance and was retained. Deep feed forward neuronal Network (DL) was used as the supervised classifier.

B. Co-Training

In general, the Co-training (CT) approach explores the results of two or more classifiers at the same time. There are many implementations of the CT according to the needs and the use cases. However the most common one splits the dataset vertically according to features (signals in our case) and thus forming feature-based sub-datasets. As shown in Figure 6, two DL classifiers are trained on $S_{Labeled}$ data. Then each data instance in $S_{Unlabeled}$ is classified by the two classifiers and the intersection result with high classification probability is used to retrain and improve a final DL. The assumption is that the classifiers working with different sets of features are able to complement each other. The main issue of CT is how to split data vertically to form the subsets that have the same amount of information. The idea is that if there are 2 primary features and 2 other secondary ones, then we will build subsets in a way that each subset has a primary feature and a secondary one. This guarantees that each subset and the associated classifier has its fair share of effective features to attain good performance.

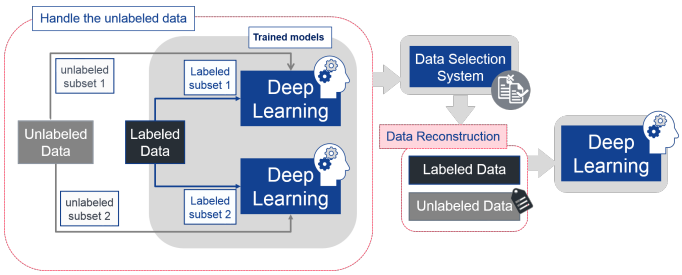

Fig. 6. The Co-Training (CT) semi supervised approach model

We analyzed the features in consideration with different machine learning techniques. According to Figure 4 we divided the whole dataset vertically to two subsets composed of (1) $[RSRP, CQI, Class]$ and (2) $[TA, MI, Class]$. This vertical division ensures the same information weight ($\simeq 50\%$), so we offer a fair opportunity of equal learning to both the DL classifiers. After the training phase of these two classifiers, we apply the same vertical division on the $S_{Unlabeled}$ set in order to predict their labels. Each data instance is classified by the two different classifiers. For the final step, only the intersection between resulting labels of two classifiers is kept. The kept part of $S_{Unlabeled}$ added to $S_{labeled}$ are used to train and improve the final DL.

C. Self-training

The Self-Training (ST) approach is one of the semi-supervised learning methods that alternatively repeats classifier training and labeling unlabeled data in training set. The main issue with ST approach is the amplification of error while labeling the unlabeled data. That means if the current trained classifier makes errors while classifying the unlabeled data then the wrong label of the unlabeled data will provide an inaccurate information for the classifier of the next step [26]. By iterating these two steps, the overall error of the final classifier will become larger. To remedy this error amplification phenomenon and to have a generic classifier, we propose a data selection system between the two phases (Figure 7). To eliminate the wrongly labeled data, we again apply the assumption that a user can not change his environment twice in 30 seconds. A label is therefore considered wrong if in 30s the

user goes from environment 1 to environment 2 and then from environment 2 to environment 1 again. Thus, we eliminate this labeling error. We also delete data that was classified with a low classification probability. That means, we eliminate data that was classified with a classification probability lower than 65%. This threshold of 65% was fixed after a statistical study to avoid both risks of over-fitting or error amplification.

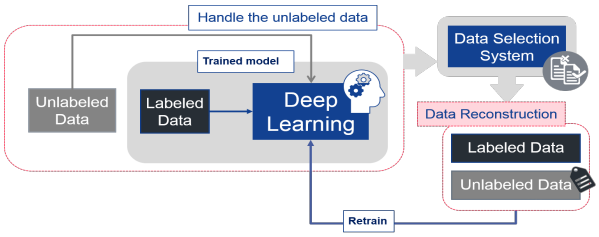

Fig. 7. The Self-taught/Self training semi supervised approach model

VI. Experimental Setup and Results

In ML language we call hyper-parameters the set of parameters that are fixed before the training phase. By contrast, the values of other parameters are derived via training. In Deep Learning, the hyper-parameter optimization is the biggest challenge because of their high number. On top of that, individual models can be very slow to train mainly when training from scratch. Actually, there are two ways for training neural networks: (1) either to train from scratch or (2) to use a pre-trained model/ architecture/ weights/ to initialize training. The multi-layer neural network requires a large amount of computational resources for training. Thus, the pre-trained option is needed for faster training of the neural network. Using weights from pre-training models suffers from a problem that it may aggravate error amplification because misconception generated during previous self-training phase is propagated to the next self-training phase. To avoid the over-fitting we set a dropout layer to regularize the learning in the next-training phase.

For the IOD study, we set Deep Learning hyper-parameters obtained from a GridSearch algorithm (which simply is an exhaustive search through a manually specified subset of the hyper-parameter for Deep Learning). We have used both scikit-learn and keras in python for the implementation. The DL module is a feed forward neuronal network (fully connected) with 8 hidden Layers using $tanh$ as the activation function. Actually, $tanh$ is one of the widely used activation function while designing neural networks today. It is used mainly in classification tasks which will lead to faster training process and convergence. As for the last layer (the output layer) we used a sigmoid activation function to smooth the results since we look for a binary classification either 0 or 1 (for in-door/outdoor environments). This experimentation performed on the $S_{Labeled}$ data, provides a DL model with 95.30% of $F1-score$. The model is saved and will serve as an initialization for the next training steps.

The 3 learning approaches - CTL, CT and ST - are evaluated on S_{Total} by computing each-time the $F1-score$ and the accuracy of each approach. The performance evaluation is carried out on both S_{Test} and $S_{Labeled}$. As shown in table II, CTL has the lowest $F1-score$ compared to CT and ST. This is explained by the fact that more the unlabeled data volume increases, the more the performance of the supervised DL gets limited to the clustering performance and errors. In our case, the BGM cluster used has a $F1-score$ of 79.71%, which gives a low $F1-score$ of CTL. CT and ST show close performances with slightly better performance of the latter since both the DL classifiers trained with their own tagged data subsets provide the same $F1-score$ of average of 85%. However, CT is very complex and greedy in resource use. CT takes lot of training time as it deals with 3 neuronal networks. Therefore ST is the best choice for IOD system since on the first phase (trained only on labeled data) as well as the last phase (trained on both labeled and unlabeled data) has showed the best performances reaching an $F1-score$ of 96.18%.

CTL		CT		ST	
Accuracy	F1-Score	Accuracy	F1-Score	Accuracy	F1-score
83.05%	79.89%	95.77%	95.34%	**96.50%**	**96.18%**

TABLE II
Semi-Supervised approach (CTL, CT, ST) performances: Accuracy & F1-score

VII. The impact of data volume

In this section, we discuss the impact of data volume and optimization of the semi-supervised self-training. Here, we address an important question: how much percentage of labeled data is needed for target satisfactory performance of the IOD system?

To answer this question, a study of the impact of the labeled data and unlabeled data volume on the training model is conduced. For this, $F1-score$ is evaluated for various $S_{Unlabeled}$ and $S_{Labeled}$ which leads to two scenarios:

- Scenario 1: $S_{Labeled}$ is fixed and the volume of the $S_{Unlabeled}$ is variable. The ST training performance is evaluated progressively according to the percentage of unlabeled data which reaches at maximum 72.69%.
- Scenario 2: $S_{Unlabeled}$ is fixed and the volume of the $S_{Labeled}$ is variable. The ST training performance is evaluated progressively according to the percentage of labeled data which reaches at maximum 27.31%.

Results of both scenarios are shown in Figure 8. The figure plots the $F1-score$ values of ST versus the size ratio between $S_{Labeled}$ or $S_{Unlabeled}$ and S_{Total}. The double X-axis refers then to the percentage of labeled data (the bottom X-axis) or unlabeled data (the top X-axis) related to the total volume of data. As expected, the addition of unlabeled data improves the IOD system performances. In scenario 1, ST uses all the labeled data and a variable part of unlabeled data. $F1-score$ increases with the size of $S_{Unlabeled}$. However, there is only moderate improvement. By using all of the labeled data ST starts already at 95% to converge toward $96,18\%$. This state corresponds to the case where all the unlabeled data is used. The information brought by all $S_{Labeled}$ data is sufficiently rich. This is unlike the scenario II, where

the $F1-score$ augmentation is more pronounced. Availability of less $S_{Labeled}$ is realistic assumption as collecting labeled data is expensive. In any case the labeled data contains more relevant information. If the mobile operator targets an error percentage of 5% for IOD (namely $F1-score=95\%$), the red curve indicates that a distribution of 20% and 80% of labeled and unlabeled data respectively is sufficient for the training phase. Consequently, the mobile network operators wanting to implement IOD inside their network may use similar percentages of labeled and unlabeled data during the updating phase of IOD learning model. They may need to manually label only 20% of collected data. During online labeling it enables first to alleviate the network overhead by limiting the amount of UL signalling (all labels) sent to eNB and, secondly, reduces the complexity and the required time for tagging data.

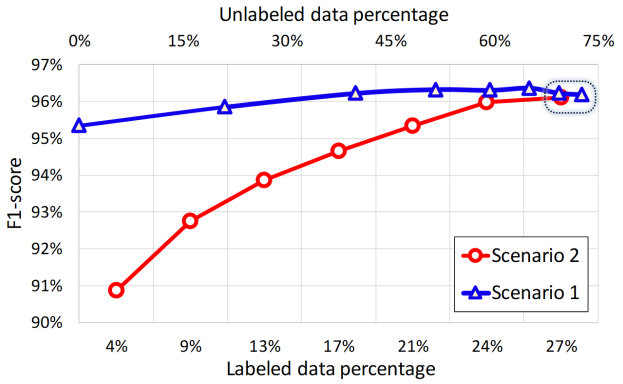

Fig. 8. Data volume Impact: (Blue Line) Scenario 1: Variation of the $S_{Unlabaled}$ volume. (Red Line) Scenario 2: Variation of $S_{Labeled}$ volume

VIII. Conclusion

In this paper we studied the IOD problem from the network side. We used a ML approach using 3GPP signals. The inputs are: $RSRP$, CQI, TA, MI. We first showed the importance of this judicious choice of inputs. The addition of TA and MI to the $(RSRP, CQI)$ couple has shown an improvement of 10% in the overall performance of IOD. A diversified partially labeled dataset was used for evaluation. It allows to be as close as possible to the real behaviors of mobile users in daily life. Secondly, in order to exploit also the unlabeled data, a comparative study of semi-supervised approaches was conducted. It showed that Self-Training approach is the best one for IOD. The ST training model obtained with a sharing of (20%,80%) between labeled and unlabeled data provides a $F1-score$ of 95%. Such an evaluation - namely the required sharing between labeled and unlabeled data for a target IOD performance - could be of interest to operators. Avoiding to tag all data strongly reduces the labeling efforts and constraints for the operators wanting to implement IOD algorithm. ST can thus perform well without requiring a complete labeling of data. In future, we plan to extend our work to user behavior contextualization by investigating other user behavior attributes.

References

[1] PIRINEN, Pekka. A brief overview of 5G research activities. In : 5G for Ubiquitous Connectivity (5GU), 2014 1st International Conference on. IEEE, 2014. p. 17-22.

[2] KAASINEN, Eija. User acceptance of mobile services: Value, ease of use, trust and ease of adoption. 2005.

[3] SIRIS, V. A., BALAMPEKOS, K.,MARINA, Mahesh K. Mobile quality of experience: Recent advances and challenges. In : Pervasive Computing and Communications Workshops (PERCOM Workshops), 2014 IEEE International Conference on. IEEE, 2014. p. 425-430.

[4] MARINA, Mahesh K., RADU V., et BALAMPEKOS, K. Impact of indoor-outdoor context on crowdsourcing based mobile coverage analysis. In : Proceedings of the 5th Workshop on All Things Cellular: Operations, Applications and Challenges. ACM, 2015. p. 45-50.

[5] CAINEY, Joe, GILL, Brendan, JOHNSTON, Samuel, et al. Modelling download throughput of LTE networks. In : Local Computer Networks Workshops (LCN Workshops), 2014 IEEE 39th Conference on. IEEE, 2014. p. 623-628.

[6] MEKKI, Sami, KARAGKIOULES, Theodoros, et VALENTIN, Stefan. HTTP adaptive streaming with indoors-outdoors detection in mobile networks. In : 2017 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS). IEEE, 2017. p. 671-676.

[7] ZHANG, Chaoyun, PATRAS, Paul, et HADDADI, Hamed. Deep Learning in Mobile and Wireless Networking: A Survey. arXiv preprint arXiv:1803.04311, 2018.

[8] LECUN, Yann, BENGIO, Yoshua, et HINTON, Geoffrey. Deep learning. nature, 2015, vol. 521, no 7553, p. 436.

[9] Andrew Ng Slides, https://media.nips.cc/Conferences/2016/Slides/6203-Slides.pdf

[10] ZHOU, P., ZHENG, Y., LI, Z., et al. Iodetector: A generic service for indoor outdoor detection. In : Proceedings of the 10th acm conference on embedded network sensor systems. ACM, 2012. p. 113-126.

[11] RADU, Valentin, KATSIKOULI, Panagiota, SARKAR, Rik, et al. A semi-supervised learning approach for robust indoor-outdoor detection with smartphones. In : Proceedings of the 12th ACM Conference on Embedded Network Sensor Systems. ACM, 2014. p. 280-294.

[12] RAY, Avik, DEB, Supratim, et MONOGIOUDIS, Pantelis. Localization of LTE measurement records with missing information. In : Computer Communications, IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on. IEEE, 2016. p. 1-9.

[13] ALAYA-FEKI, Afef Ben Hadj, LE CORNEC, Alain, et MOULINES, Eric. Optimization of Radio Measurements Exploitation in Wireless Mobile Networks. JCM, 2007, vol. 2, no 7, p. 59-67.

[14] HADDAD, Majed, HERCULEA, Dalia Georgiana, ALTMAN, Eitan, et al. Mobility state estimation in LTE. In : Wireless Communications and Networking Conference (WCNC), 2016 IEEE. IEEE, 2016. p. 1-6.

[15] HERCULEA, D., CHEN, C. S., HADDAD, M., et al. Straight: Stochastic geometry and user history based mobility estimation. In : Proceedings of the 8th ACM International Workshop on Hot Topics in Planet-scale mObile computing and online Social neTworking. ACM, 2016. p. 1-6.

[16] 3GPP TS36.304: User Equipment (UE) procedures in idle mode.

[17] 3GPP TS36.331: Radio Resource Control (RRC); Protocol specification.

[18] 3GPP TS36.133: Requirements for support of radio resource management.

[19] 3GPP TS36.213: Physical layer procedures.

[20] 3GPP TS36.321: Medium Access Control (MAC) protocol specification".

[21] LAIYEMO, Ayotunde Oluwaseun, High speed moving networks in future wireless systems, 2018. school of computer science.

[22] ZHU, Xiaojin et GOLDBERG, Andrew B. Introduction to semi-supervised learning. Synthesis lectures on artificial intelligence and machine learning, 2009, vol. 3, no 1, p. 1-130

[23] ZHU, Xiaojin. Semi-supervised learning literature survey. Computer Science, University of Wisconsin-Madison, 2006, vol. 2, no 3, p. 4.

[24] ZHOU, Zhi-Hua et LI, Ming. Tri-training: Exploiting unlabeled data using three classifiers. IEEE Transactions on knowledge and Data Engineering, 2005, vol. 17, no 11, p. 1529-1541.

[25] ZHU, X., LAFFERTY, J., et ROSENFELD, R. Semi-supervised learning with graphs. 2005. Thèse de doctorat. Carnegie Mellon University, language technologies institute, school of computer science.

[26] LEE, H.W., KIM, N., et LEE, J.H.. Deep neural network self-training based on unsupervised learning and dropout. International Journal of Fuzzy Logic and Intelligent Systems, 2017, vol. 17, no 1, p. 1-9.