



**HAL**  
open science

## Mobile User Environment Detection using Deep Learning based Multi-Output Classification

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

► **To cite this version:**

Illyyne Saffar, Marie-Line Alberi-Morel, Mohanned Amara, Kamal Deep Singh, César Viho. Mobile User Environment Detection using Deep Learning based Multi-Output Classification. WMNC 2019 - 12th IFIP Wireless and Mobile Networking Conference, Sep 2019, Paris, France. pp.16-23, 10.23919/WMNC.2019.8881474 . hal-02437854

**HAL Id: hal-02437854**

**<https://hal.science/hal-02437854v1>**

Submitted on 19 Jun 2021

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Mobile User Environment Detection using Deep Learning based Multi-Output Classification

Illyne Saffar<sup>\*‡</sup>, Marie Line Alberi Morel<sup>\*</sup>, Mohammed Amara<sup>\*</sup>, Kamal Deep Singh<sup>†</sup>, Cesar Viho<sup>‡</sup>

<sup>\*</sup>Nokia Bell Labs, Nozay, France      <sup>‡</sup>Univ Rennes, Inria, CNRS, IRISA, France

<sup>†</sup>Univ Lyon, UJM Saint-Etienne, CNRS, Laboratoire Hubert Curien UMR 5516, Saint-Etienne, France

Email: illyne.saffar@nokia.com, marie\_line.alberi-morel@nokia-bell-labs.com, mohammed.amara@nokia.com, kamal.singh@univ-st-etienne.fr, cesar.viho@irisa.fr

**Abstract**—Future mobile networks can hugely benefit from cognition of mobile user behavior. Indeed, knowing what/when/where/how the user consumes their mobile services can notably improve the self-adaptation and self-optimization capabilities of these networks and, in turn, ensure user satisfaction. The cognition of mobile user behavior will thus help 5G networks to face the variable consuming habits of users which in turn impact the network conditions, by predicting them in advance. In this paper, we focus on the “where” part, i.e., the detection of the environment where a given user consumes different mobile applications. A statistical study on the real activity of users reveals that there are multiple various environment types corresponding to the mobile phone usage. A Deep Learning based model is introduced to intelligently detect the user environment, using supervised and semi-supervised multi-output classification. Relevant multi-class schemes are proposed to efficiently regroup the multiple environment categories in more than two classes. We empirically evaluate the effectiveness of the proposed model using new real-time radio data, gathered massively from multiple typical and diversified environments of mobile users.

**Index Terms**—User Environment Detection, Multi-output classification, Deep Learning, Semi-supervised Learning, 3GPP signals, real user activity.

## I. INTRODUCTION

Autonomic 5G networks are the next evolution of mobile networks for accommodating the ever-growing user-demands for better services and applications with high Quality of Experience with lowest cost of management and operation [1]. Such objectives are achievable thanks to the analysis of mobile user data in order to extract actionable knowledge related to user profile or the use context of mobile services.

According to Kipling’s method [2], a mobile user profile is defined in terms of how/ when/ what/ where/ who/ why (5W1H) the user consumes mobile applications. The cognition and knowledge gained about the consuming habits of individuals and communities are then exploited by mobile autonomic 5G networks to grow smarter. Furthermore, this can make 5G networks more efficient when faced with the increasing complexity of network management combined with numerous new applications and their heterogeneous needs. In this paper, we focus on detecting the environment where a mobile user connected to a cellular network experiences his or her mobile applications.

In literature, the environment detection issue has been studied mainly considering an Indoor Outdoor Detection (IOD) binary classification to detect the environment [3], [4]. However, identifying the user’s environment is a more complicated task than just IOD. Actually, the environment deeply affects the way a user interacts with his or her mobile phone. He or she usually tends to have different attitudes at home than at work, at a café than in transport, during the working week than during the week-end [5], etc. The way, individuals use their phones, varies notably as a function of these factors. Classifying the environment as only two states (Indoor or Outdoor) can be seen as a first level of analysis, but it does not reveal the complexity of the situation. So, a more granular classification is desired in order to detect/predict where a mobile user uses/prefers certain mobile applications. However, a more detailed classification impacts the distribution of measurement instances into the resulting classes. This is because users prefer to use their mobile phone in specific situations [6]. As a consequence, some classes (the most popular) have much more measurement points than others. This unbalancing between the categories of various environments, observed in case of two classes in [3], is also augmented when classifying with more than two classes. Therefore, achieving a multi-output classification will first require to tackle this phenomenon.

In this paper, we investigate the user environment detection using a supervised as well as a semi-supervised multi-output classification technique. Multi-output refers to the detection of multiple types of environment. We study relevant multi-class schemes that smartly regroup the multiple environment categories in more than two classes. However to what level and detail the model requires to classify a user’s environment? Which classification scheme leads to better balanced data? Can we classify the user environment with detailed classes and with good performance? In the following, a comparative analysis of classification between schemes of three, four and five classes will show that using four classes with relevant labels enables a good trade-off between balancing and more granularity.

For the environment detection task, the deep-learning based approach has shown the best performance over classical machine learning [12]. Actually, this approach is preferred to achieve the multi-output classification since it is appropriated for resolving problems where modeling relationships between large number of parameters are not tractable. This is the case

when classifying the environment into more than two classes. Indeed the model has to extract the complexity of the variety of situations met by the mobile users. This detection of environment will help the operator to perform online adaptation and optimisation of network and radio parameters according to users' environment types, i.e, depending on what percentage of users are in indoor environment, how many in transport, etc. This will also help the operator to provide context-aware services. The evaluation of the proposed solution will be done using real and large 4G LTE radio data collected through a crowdsourcing approach, i.e. collected by the phone device and sent to the mobile network via 3GPP procedures [3]. Thus, Machine Learning algorithms trained on datasets collected in crowdsourcing mode allow to learn very diverse real-world environments. Deep learning lately is gaining much popularity due to its supremacy in terms of accuracy when trained with huge amounts of data. Deep Learning's performance continues to improve when more and more data is used for training. This results in it outperforming the traditional models/algorithm of Machine Learning [7], [8]. However, using real data requires to tackle the problem of data cleaning and also needs to overcome the noise inherent to real data, which in turn is not synthetic. In addition, we investigate optimization solutions during the training phase. We focus mainly on the initialization of the model parameter values before training and on the generalization of the model for various user profiles using an additional Gaussian Layer.

The remaining paper is organized as follows. In Section 2, related existing works are discussed. Section 3, provides an analysis of the user activity according to the environment type. Section 4, describes the data collection and investigates multiple scheme of multi-output classification. The solution adopted to optimize the classification is detailed in Section 5. In Section 6, results of the environment detection system are analysed and discussed. Finally conclusion and perspectives are presented in Section 7.

## II. RELATED WORK

In the literature, to the best of our knowledge, the user environment detection (UED) issue has not been largely studied. So far, what has been carefully studied in literature, is UED as a binary classification or detection problem, with only indoor and outdoor as the classes.

In [9], authors detected the user environment according to three environment types: indoor (inside a building), semi-outdoor (close to or semi-open building) and outdoor (outside a building). For such detection, they used a combination of signals collected from some phone sensors related to radio signals, cell signal strength, light intensity as well as the magnetic field to infer whether the environment state is indoor, outdoor or semi-outdoor. Detecting user environment with only such granularity does not provide sufficient knowledge for the mobile network to relate the detected use context with the user's application preferences. The user consumption will be almost the same in an indoor environment as in semi-outdoor. In [4], authors studied UED with only indoor and

outdoor as environment classes. For IOD, they propose a semi-supervised approach using a co-training based solution. They used 2 classifiers in parallel with a weighted score of classification probability to improve the final performance of IOD. To train these classifiers they not only considered the same measurements set coming from the phone sensors as [9], but also added sound intensity, battery temperature and the proximity sensor. The set of these papers deals with the IOD from phone devices side with UE-specific data which in turn is in most cases not known by the mobile networks. Moreover [9] uses a computed threshold to make a classification decision between two states. Extending such methods to make a classification decision between a plurality of states (more than 2) is too complicated and is not performant.

The work in [10], used a Bayesian detector that combines the signals measured from the cellular (RSRP provided by the cellular modem) and GNSS receiver (the confidence radius of the location, provided by the active localization sensor). This information provided by both measurements is combined with a joint posteriori probability based on the distributions of RSRP and GPS measurement to perform indoor/outdoor detection. In [11], authors localize mobile users by accurately estimating their longitude and latitude. As a first step of localization process, they also detect the user environment type: indoor or outdoor. For this IOD step, they used *RSRP* and *RSRQ* signals and tested many learning algorithms: SVM, logistic regression and random forest. In both [3] and [12], authors studied UED as a binary classification (Indoor vs. Outdoor) and using a Deep Learning approach. In [3], authors used 4 signals for the UED task: a power signal *RSRP*, a quality signal *CQI* that replaced the *RSRQ* because the latter is not often reported to the network and finally the *Timing Advance*. In [12], authors used the same signals as [3], but with addition of a mobility indicator to solve some difficult cases of detection like when the user is in train (outdoor environment), and suffers from a drastic deterioration of the RSRP signal, then the Mobility Indicator helps to better detect such case. Both papers [3] and [12] show good performance of the UED

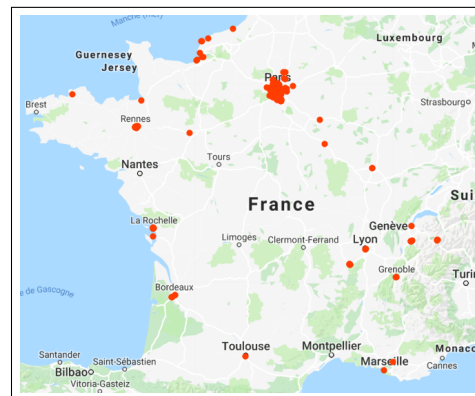


Fig. 1: Data collection points in France: multiple places

binary classification, i.e., around 95% of F1-score (a measure of a machine learning accuracy).

However, considering UED as a classification with multiple outputs (multiple environment classes) can be of interest for user profiling. In this paper, we rather focus on UED for user profiling. Thus, we use a deep learning approach trained on a large and real dataset. We consider several input signals which are collected within the network side. We look at the performance in terms of F1-scores of supervised or semi-supervised UED deep learning methods.

### III. DATA COLLECTION

Data constitutes the fuel and the core of any machine learning or deep learning process. For our study, the collected data comes from a large crowdsourcing campaign, collected by phone devices and sent to the mobile network. Using measurements done by a phone device is the most reasonable choice for user behaviour analyses and user profiling as it allows to capture the real behavior of mobile users. Indeed, due to their small size and popularity, the portable phone devices are always with users during their various activities in all kinds of environments.

As shown in Figure 1 our dataset has been collected in different locations in France: red dots on the map indicate the main ones, and the roads linking these locations where users are moving with different speeds.

The data has been collected during 16 months, 24h/7, with an average of 1 measurement per 15 seconds, when the mobile phone session is active, and 1 measurement per 2 minutes otherwise. Thus, around 2M lines of data per user have been collected. The collection has been performed in many different indoor/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 in France. This long collection period allows us to have data

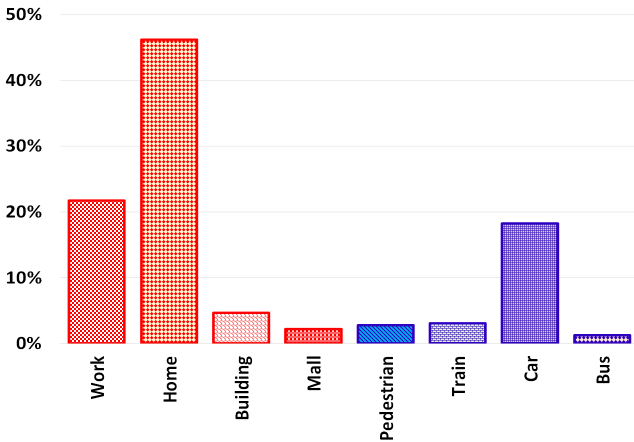


Fig. 2: User activity per environment

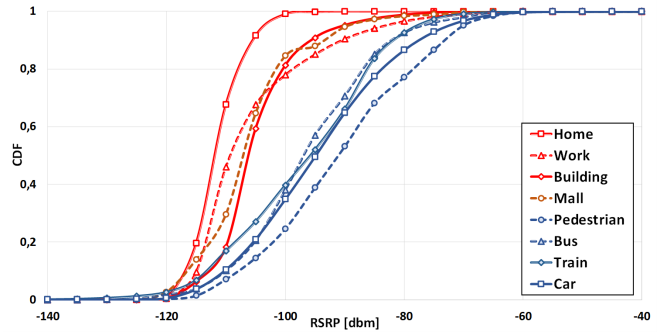


Fig. 3: CDF of RSRP

reflecting all weather types. With such collecting mode the dataset built closely captures the complexity and the variety of a mobile user moving in real world.

For the user environment detection, a specific dataset is built consisting of radio signals (RSRP, CQI), time related features (TA, Time) and mobility indicator (MI) and finally the environment label when it exists. The dataset is composed of 6-feature samples as following:

- Time: the recording time of signal or burst data arrival.
- RSRP: the average received power of the reference signal. RSRP varies between -140 dBm and -44 dBm.
- CQI: Channel Quality indicator which is the selection tool for the best fitting transmission modulation and coding scheme to be used in communication.
- TA: Timing Advance is used to control UL signal transmission timing.
- MI: the number of the Cell ID changes during a sliding window of length equal to 100s as fixed in [12].
- Environment: the label of environment crossed by user during the measurement campaign.

We consider the following categories of environment: [Work, Home, Building, Bus, Car, Mall, Pedestrian, Train]. They have been chosen for labelling the data. This is because they are highly representative of the locations where data have been collected. The environment labeling was processed using GPS

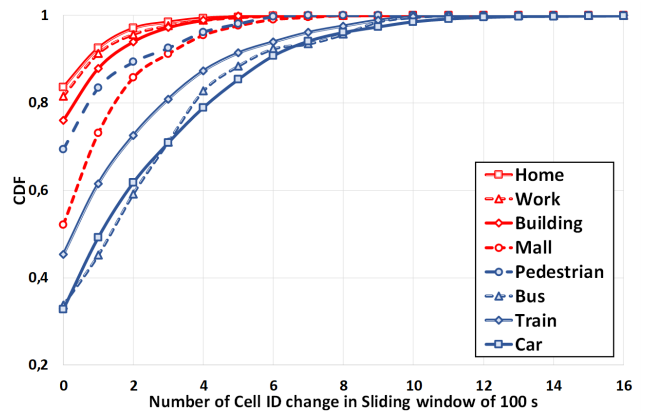


Fig. 4: CDF of number of Cell ID changes during 100s

coordinates for users who allowed us to collect them.

#### IV. MULTI-OUTPUT CLASSIFICATION SCHEMES

##### A. Relation between user activity and environment type

A better identification of the environment classification problem can be achieved when analysing the user’s activity according to his or her environment. This investigation is done with respect to the eight types of environment that characterise the locations crossed during the measurement campaign: [*Work, Home, Building, Bus, Car, Mall, Pedestrian, Train*]. These environment categories were selected in order to reflect the complexity of a user’s daily life and capture the variety of his movements in real world.

The ‘phone usage’ state is defined as the state when the user uses his or her phone or equivalently the screen is on and unlocked and there is data exchanged. In order to track this state, following additional data features are also collected during the campaign. These features are the instance index or timestamp, the screen state and the label of environment type. Analysing this data gives us a clear picture of mobile phone usage in different environments. Figure 2 depicts the user activity by plotting the phone usage ratio for different environment categories. User activity is measured as the ratio between the number of instances the user has been using his phone effectively and the total number of instances. This figure illustrates the percentage of total time the user is connected to 4G network and exchanges data with it. We observe that most of the activity is spent more indoor (70 %), mainly at home and at work, as compared to being outdoor. Figure 2 highlights the user activity trends that we observed after statistical analysis on mobile user behavior in literature.

Contrary to traditional TV users, who just watch scheduled programs, mobile service users are free to choose the content they want, at any point in time and space. During a day, a user can be in different situations, such as walking outdoor, in a car, at work, in a mall, in a café or at home. As a matter of fact, mobile users’ preferences for certain applications or contents is linked with the usage situations [5]. In the literature, some statistical studies show that mobile phones are mostly used in a building for internet service (80%) and for a call (70 %) [6]. This can be explained by the fact that the different use contexts pose their own limitations, which in turn impact the potential application usages. In [13], [14], [15], the most commonly mentioned physical environments of application usage are indoor (waiting halls or lounges, work, home and cafes), but also include vehicles, such as public transportation and private cars. The home environment is preferred by users, while the bus or car environment is not for using the phone. This is explained by the calm and the pleasant atmosphere which is more suitable for focusing on viewing. Whereas, in motion, the complicated tasks generally result in an unpleasant entertainment experience. Furthermore, people spend most of their time indoor than in mobility and outdoor. Consequently, outdoor labels are less represented than indoor labels in phone usage data.

Figure 2 illustrates the multiple diversity of indoor and outdoor situations met by the users of mobile phone. These situations are [*Work, Home, Building, Bus, Car, Mall, Pedestrian, Train*]. Consequently, it is preferable to consider the user environment detection task as a multi-output classification problem. We observed also in Figure 2 that the data instances are distributed unequally between the different categories. It shows that the data proportion in groups of label “Train”, “Bus”, “Mall”, “Building” or “Pedestrian” is very low compared to the other groups. This unbalanced nature of data remains a challenge for Machine Learning algorithms

##### B. Classification schemes

In this section, the objective is to define relevant classification schemes with multiple classes by regrouping smartly the various environment categories given in section III.

As a matter of fact, the issue of unbalanced data is not related to the data collection constraints, but it is rather, inherently, due to the nature of real human activities. As a consequence, in order to ensure an efficient scheme of environment classification for real user activity, we have to find a trade-off to limit this inherent data bias for different classes. We target to design a classification scheme that detects the detailed environment types of mobile users, with a fine granularity and a low decision error margin. We fixed this margin at 5%. An investigation has to be done to guess the relevant minimal value. But, it is out of scope of this paper.

We focus on studying the relevant trade-offs in our case of multi-output classification. We consider different possibilities ranging from a simple binary classification problem to a more complex classification task for detecting detailed information about a user’s environment. To ensure this, we decide to regroup the environment categories {*Work, Home, Building, Mall, Bus, Car, Pedestrian, Train*}. Furthermore, these categories have thick inter-relationships and we can merge them into bigger consistent and recognizable groups. The created merged groups shall also include data with similar statistical properties in order to optimize the classification results. This is obtained by observing the cumulative distribution curves of the collected data {*RSRP, CQI, TA, MI*} as well as the variance  $\sigma^2$  of the phone activity for each proposed multi-class schemes. The variance is written as:  $\sigma^2 = \frac{\sum(X_i - \bar{X})^2}{N}$  where  $X_i$  is the number of instance in the class  $i$  and  $N$  the number of classes. This represents the percentage of total instances linked to this environment. The variance is an appropriate metric to measure the degree of unbalance by quantifying the variance of the percentage of total instances per environment category from its average.

Figures 3 and 4 depict respectively the cumulative distribution curves of *RSRP* and *MI*. We observed in Figure 3 that three groups of similar curves can be extracted: a set with only *Home*, a set regrouping {*Work, Building, Mall*} and another set assembling {*Bus, Car, Pedestrian, Train*}. This results into a further split of indoor and outdoor classes. Furthermore, the environment “Home” is detached from the set of other indoor situations. Analysing the Figure 4, we also note a similar

separation between indoor and outdoor curves. Moreover, CDF curves of MI highlight a clear separation between “pedestrian” and the remaining set of outdoor labels and between “Mall” and the others indoor situations. Clearly these two environment types are associated to users moving very slowly (walking) as compared to others that are either static or high speed.

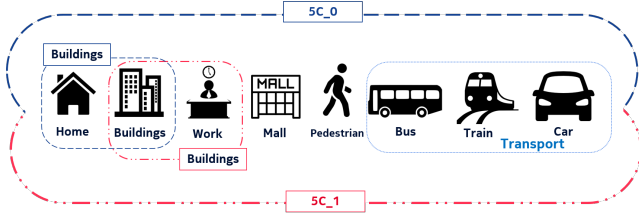


Fig. 5: Multiple class schemes example: “5C\_0” and “5C\_1”

Based on these observations, we propose eight grouping schemes derived from the classes “indoor” and “outdoor” investigated in [12]. The label “indoor” refers to “home”, “work”, “mall” and “buildings”. The label “outdoor” regroups the environments “pedestrian”, “car”, “bus” and “train”. Grouping schemes of 3, 4 and 5 classes are then investigated to better model the diversity of environments. The studied schemes are as follows (for illustration, some examples are provided in Figure 5):

- “3C\_0” and “3C\_1”: these two schemes include three classes in total. Both include a class “outdoor” that groups all the outdoor categories. The scheme “3C\_0” contains two additional indoor classes “work” and “building” (grouping “home”, “mall” and various “buildings”). Whereas, the scheme “3C\_1” contains two additional indoor classes which are “home” and “building” (grouping “home”, “mall” and various “building”).
- “4CO\_0” and “4CO\_1”: these two schemes include four classes in total. Both include a unique class “outdoor”. The scheme “4CO\_0” contains three more indoor classes “work”, “mall” and “building” (assembling “home” and various “building”). Whereas, the scheme “4CO\_1” contains three other indoor classes which are “home”, “mall” and “building” (assembling “work” and various “building”).
- “4CI\_0” and “4CI\_1”: these two schemes include four classes. Both include a class “outdoor” split further in two classes “pedestrian” and “in-transport” (which in turn regroups “bus”, “train” and “car”). The scheme “4CI\_0” contains two additional indoor classes “mall” and “building” (regrouping “home”, “work” and various “buildings”). Meanwhile, the scheme “4CI\_1” contains two other indoor classes “home” and “building” (merging “work”, “mall” and various “buildings”).
- “5C\_0” and “5C\_1”: these two schemes include five classes. Both include a class “mall” and two classes “pedestrian” and “in-transport”. The scheme “5C\_0” contains three additional classes “mall”, “work” and “building” (namely “home” and various “buildings”). Mean-

while, “5C\_1” contains three classes “mall”, “home” and “building” (namely “works” and various “buildings”).

Figure 6 shows the variance of data size when the data is divided according to the above schemes. This is to quantify the balance or unbalance of data between different classes. When the variance is high the dispersion, in terms of data size in different classes, is important and, thus, the scheme is very unbalanced. As shown in Figure 6, the schemes “3C\_0” and “3C\_1” deliver the smallest variance. They are followed by “4CO\_1”, “4CI\_1” and “5C\_1”. Indeed, the scheme 3C results in the most balanced data among different classes. This scheme regroups the instances coming from outdoor labels and splits the indoor label instances. All the schemes are illustrated In Figure 10. The figures represents the phone activity according the environment type for the 8 proposed schemes.

Later, we will evaluate the relevance of a more detailed environment detection as compared to outdoor or indoor.

## V. CLASSIFICATION OPTIMIZATION

### A. Hyper-parameter Tuning

The hardest step in every deep learning training procedure is to find the best hyper-parameters set for the model. Hyper-parameters are the set of variables to be optimized before applying any learning algorithm to a dataset. The set of hyper-parameters is composed of number of hidden layers, batch size, epoch size, the weight initialization, the activation function, the loss function, the learning rate, the momentum and the dropout fraction. The challenge with these hyper-parameters is that there is no magic combination that always works. The best hyper-parameters combination depends on each task and also on each dataset. In this paper, we focus on 3 main strategies: i) Manual Search, ii) Grid Search, iii) Bayesian model-based optimization. Manual Search is the traditional way to set hyper-parameters, i.e, setting them manually by trial and error many times and quantifying the errors for each trial. This method is the fastest way to set the hyper-parameters, but it never guarantees that we will have the right combination and good classification performance. In [3], [12] authors used grid search to optimize the hyper-parameters of their deep learning model. Actually grid search

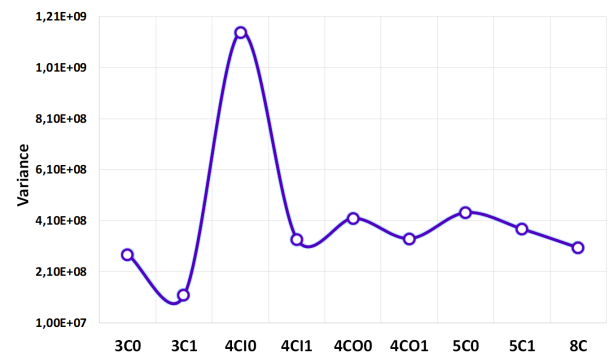


Fig. 6: Variance of phone activity for the multiple class schemes

consists of comparing all possible combinations of hyper-parameters and then returning the best set in terms of the returned objective function value. The challenge with hyper-parameters optimization is to minimize the number of times the objective function is evaluated. Actually, the evaluation of the objective function is generally very expensive. With a complex model, a large dataset, and a high dimensionality of hyper-parameters space, the problem becomes intractable and grid search becomes inefficient because of its greediness.

Recently a new approach using a Bayesian optimization for tuning hyper-parameters has been considered [16]. Based on a probabilistic model of the objective function called the surrogate function, Bayesian optimization reduces the frequency of calls to the the objective function, to the lowest possible. The surrogate model is represented by the probability of the score knowing a fixed hyper-parameters' set ( $P(score | hyperparameters)$ ). Actually, the use of a surrogate model enables the algorithm to select the most promising hyper-parameters for the objective function evaluation, so that, the search does not spend a significant amount of time on looping on bad combination of hyper-parameters. That is to say, such optimization leads to a faster convergence of the deep learning based model.

### B. Adding Gaussian noise

In [17], the authors have shown that by allowing a little random margin of inaccuracy in Feed Forward Neural Networks, a model can perform better on both training and accuracy. Neural Networks are able to model functions that change their outputs spectacularly in response to just a small input variation. Some noise is introduced to the neural network model via a Gaussian Noise layer during the training phase. Noise permits to have some robustness in the output and smoothed decision borders. Thus, noise helps reduce the chance of over-fitting, even when training dataset is small, and aids in the generalization of the model. Thus, we add a Gaussian Noise layer inside our model. It consists in adding noise to output of the layer before the activation function. The added random value,  $\epsilon_i$ , follows a normal distribution  $\epsilon_i \sim \mathcal{N}(0, \Sigma)$ , i.e., with a mean equal to zero and the covariance matrix of features  $\Sigma$ .

Precisely, the figure indicates the percentage of time a user is active in his or her phone.

## VI. EXPERIMENTAL SETUP AND RESULTS

### A. Configuration and performance metric

For our supervised multi-output classification problem, we used a Feed Forward Neural Network architecture with a total of 7 hidden layers that is depicted in Figure 7. To evaluate the impact of injecting more data during the training phase [18], we used a semi-supervised training model detailed in [12] that is depicted in Figure 8. The model is trained with a part of labelled data and another part of unlabeled data. Authors in [12] demonstrate its good performance. We plan to evaluate this brought when using in a semi-supervised multi-output classification problem.

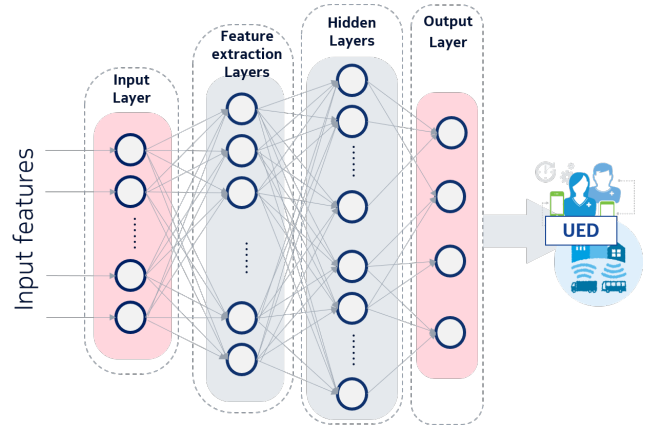


Fig. 7: Supervised Training Model (Feed Forward Neural Network)

The set of hyper-parameters (e.g. the number of hidden layers, batch size, epoch size, the weights) have been tuned using Bayesian optimization. The implementation is done under python and using keras with tensorflow as a back-end. For training and test, we have collected around 2M lines of data per user. In this paper, we used 270K lines of data collected only in France and corresponding to LTE networks. The dataset is made of 50% of labeled data and 50% of unlabelled data. The training is done using the labeled part of our dataset. For training, we used (70%) of the labeled data and we used the (30%) remaining for the model performance evaluation. As the multi-output classification problem leads to a issue of unbalanced classes, F1-score metric in addition to the accuracy is used for performance evaluation. F1-score is one of the most used metric in case of unbalanced data classes. The metric by definition is the weighted average of *Precision* and *Recall* according to the following relation:

$$F1 - score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$$

where *Precision* is the number of correct positive results divided by the number of all positive results returned by the classifier, and *Recall* is the number of correct positive results divided by the number of all relevant samples.

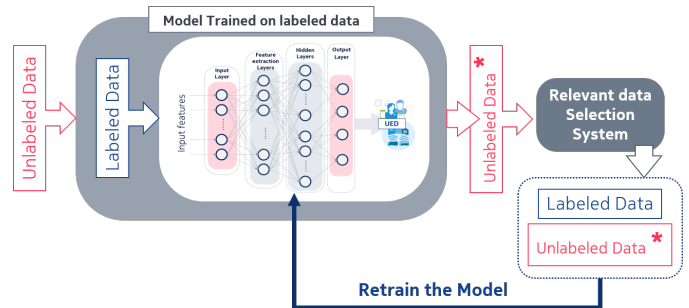


Fig. 8: semi-supervised Training Model (Self-Training [12])

	$2C$		$3C_0$		$4CO_0$		$4CI_0$		$5C_0$	
	Accuracy	F1-Score	Accuracy	F1-Score	Accuracy	F1-Score	Accuracy	F1-Score	Accuracy	F1-score
Supervised	<b>95.8%</b>	<b>95.76%</b>	91.89%	91.84%	91.58%	91.53%	<b>93.18%</b>	<b>93.11%</b>	90.70%	90.69%
Semi-Supervised	<b>96.51%</b>	<b>96.45%</b>	94.20%	94.19%	94.24%	94.24%	<b>94.92%</b>	<b>94.92%</b>	93.42%	93.38%
			$3C_1$		$4CO_1$		$4CI_1$		$5C_1$	
	Accuracy	F1-Score	Accuracy	F1-Score	Accuracy	F1-Score	Accuracy	F1-Score	Accuracy	F1-score
Supervised			92.06%	92.04%	91.42%	91.38%	<b>94.31%</b>	<b>94.28%</b>	90.79%	90.72%
Semi-Supervised			94.14%	94.13%	94.18%	94.17%	<b>94.53%</b>	<b>94.55%</b>	93.66%	93.63%

TABLE I: Deep Learning-based supervised and semi-supervised multi-ouput classification performance: F1-score vs. classification schemes - Training with Noise Layer with and without using self-learning

### B. Optimization performance results

To quantify the impact of the optimization method, we fix the classification scheme to be  $4CI_1$ . Now, if we fix the number of the hidden layers to 7, the training step of the neural network takes in average  $130.86second$ . The challenge is to find the best combination of the hyper-parameters that can learn the best UED classifier. For optimization, we consider 7 hyper-parameters: the epoch size, the batch size, the activation function, the initialization, the learning rate, the dropout rate, the optimizer, and the number of neurons per layer. Different values of these hyper-parameters and their combinations are very large. This means that the convergence time to train the best model is also very high. Let us assume that we loop among 5 possibilities or values of the epoch size, 5 possibilities of the batch size, 3 possibilities of the activation function, 3 possibilities of the initialization, 3 possibilities of the learning rate, 5 possibilities of the dropout rate, 5 possibilities of the optimizer, and 4 possibilities of the number of neurons per layer. This leads us to test 472500 neural networks and it would take around  $472500 * 130.86 seconds$  for them to train. To limit the run time, we investigated 3 methods: manual search, grid search and Bayesian search as shown in Fig. 9. We set a threshold of 50 iterations for both the Bayesian optimization and the grid search (an iteration = one combination of hyper-parameter for training a model). As expected the manual search is the fastest one since it is conducted by a human (10 runs at most). As shown in Fig. 9 and supported by analysis before, the Grid search is too slow and the computation time is high. In the worst case, the best model is delivered after looping on all combinations of hyper-parameters. Whereas, Bayesian method finds the right range and parameters space from the first iterations.

### C. Multi-ouput classification performance

Table I presents the performance results of a supervised and semi-supervised multi-output classification for the 8 multi-output classification schemes as well as the two-class scheme. They are evaluated in terms of F1-score and then compared with a classical IOD binary classification. We observe that all schemes deliver F1-scores higher than 90% that corresponds to acceptable performance in terms of classification. However, among all the schemes the two-class scheme and the two schemes " $4CI_1$ " and " $4CI_0$ " give the best performance.

As shown in Table I, the scheme " $2C$ " is equal to 95.76% when supervised training is used. The F1-scores of " $4CI_1$ "

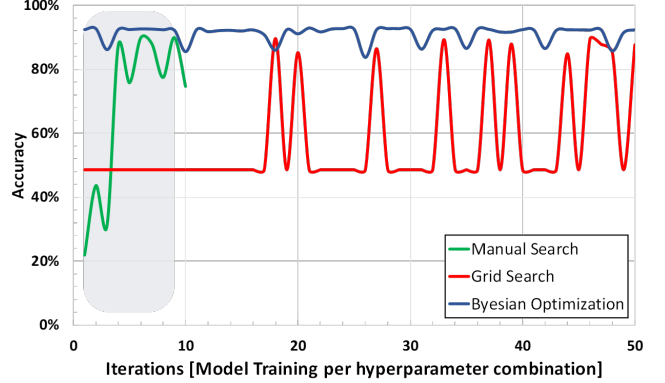


Fig. 9: Impact of optimization methods on the accuracy computing

and " $4CI_0$ " obtained in supervised case are equal to 94.28% and 93.11%, respectively. We notice a slight improvement of F1-score using the two-class scheme as compared to the F1-score obtained under same conditions in [12]. Indeed, this enhancement is due to the fact that now our algorithms process two times more real radio data collected in real conditions of crowd-sourcing. Consequently, it impacts the performance in a favorable way. Using the semi-supervised method for multi-output detection is furthermore positive. It still enhances the scores thanks to the addition of unlabelled data in the training phase. F1-scores for the three schemes are equal to 96.45%, 94.55% and 94.92%, respectively. In both methods, we also observe a maximum loss of around 6% when using a five-class scheme as compared to the binary classification. The loss is reduced to around 2% when using a 4-class scheme.

Thus, we show that a detailed learning of the environment can be achieved with a very minimal loss of performance. This is obtained using a smart cutting of the groups "indoor" and "outdoor". We observe that the coarse learning of the environment benefits more from the diversity brought by the introduction of labels "pedestrian" and "in-transport" than by the split of the group "indoor". Furthermore, we also note that the level of unbalance in data has an influence on the F1-score. The most unbalanced scheme delivers small F1-scores. The two best schemes (outside the two-class scheme) offer relatively balanced classes.



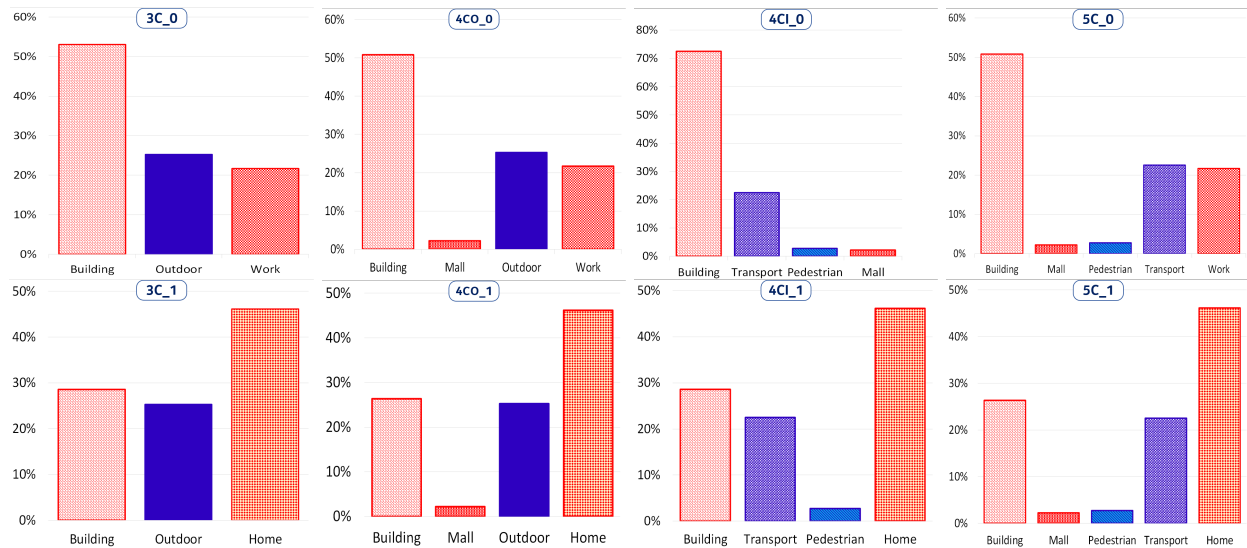


Fig. 10: Phone Activity for various classification schemes of environment

## VII. CONCLUSION

In order to deliver context-aware services, it is important to first detect the use context. In this paper, we focused on detecting user environment which is an important part of the use context. Instead of just detecting the environment as indoor or outdoor, we focused on detecting more detailed classes of the user environment.

We showed that the mobile network can detect the user environment using a multi-class classifier trained on real data. This classifier uses standardised mobile network signals as input. We studied how to divide the initial indoor/outdoor classes further into more detailed environment types. However, real data based training poses the problem that the data corresponding to different classes can be highly unbalanced. This unbalance is not due to the way with which we collected data collection, but is strongly related to the inherent user behavior and his preferences to services consumption. Knowing that this data unbalance can be a problem for machine learning, we tried to find a compromise between the data instances per class and the granularity degree by varying the number of environment classes. Thus, we studied different combinations of splitting the environment into different classes and compared their performance. The performance peaked for 2 class classifier as it is the easiest and for 4 class classifier which offers more detailed environment classification than just indoor and outdoor.

In future, we plan to build a classifier that can detect more components corresponding to the use context and not only the environment types. We aim also to quantify the impact of the environment and the other components on the context-aware services.

## REFERENCES

- [1] PIRINEN, Pekka. A brief overview of 5G research activities. In : 1st International Conference on 5G for Ubiquitous Connectivity. IEEE, 2014. p. 17-22.
- [2] JAVAID Nadee, SHER Arshad, NASIR Hina Nasir, GUIZANI Nadra. Intelligence in IoT-Based 5G Networks: Opportunities and Challenges. In : IEEE Communications Magazine, October 2018. p. 94-100.
- [3] SAFFAR, Illyne, ALBERI MOREL, Marie Line, SINGH, Kamal Deep, et al. Machine Learning with partially labeled Data for Indoor Outdoor Detection. In : 2019 16th IEEE Annual Consumer Communications and Networking Conference (CCNC). IEEE, 2019. p. 1-8.
- [4] 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.
- [5] S. Ickin et al., Factors influencing quality of experience of commonly used mobile applications, IEEE Communications Magazine, vol. 50, no. 4, pp. 48-56, 2012.
- [6] <http://www.hetnetforum.com/>, HetNet Forum, 2018
- [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] ZHOU, Pengfei, ZHENG, Yuanqing, LI, Zhenjiang, 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.
- [10] MEKKI, Sami, KARAGKIOULES, Theodoros, et VALENTIN, Stefan. HTTP adaptive streaming with indoors-outdoors detection in mobile networks. arXiv preprint arXiv:1705.08809, 2017.
- [11] RAY, Avik, DEB, Supratim, et MONOGILOUDIS, 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.
- [12] SAFFAR, Illyne, ALBERI-MOREL, Marie-Line, SINGH, Kamal Deep, et al. Semi-supervised Deep Learning-based Methods for Indoor Outdoor Detection. In : IEEE International Conference on Communications (ICC). 2019.
- [13] V. A. Siris, K. Balampekos and M. K. Marina, Mobile Quality of Experience: Recent Advances and Challenges, Workshop on Information Quality and Quality of Service for Pervasive Computing, 2014
- [14] T. Soikkeli, J. Karikoski, H. Hämäläinen, Diversity and End User Context in Smartphone Usage Sessions, Next Generation Mobile Applications, Services and Technologies (NGMAST), 5th International Conference, 2011, pp. 7-12.
- [15] E. Kaasinen, User Acceptance of Mobile Services - Value, Ease of Use, Trust and Ease of Adoption, 2005
- [16] BERGSTRA, James, YAMINS, Dan, et COX, David D. Hyperopt: A python library for optimizing the hyperparameters of machine learning

algorithms. In : Proceedings of the 12th Python in science conference. 2013. p. 13-20.

- [17] SIETSMA, Jocelyn et DOW, Robert JF. Creating artificial neural networks that generalize. Neural networks, 1991, vol. 4, no 1, p. 67-79.
- [18] Andrew Ng Slides, <https://media.nips.cc/Conferences/2016/Slides/6203-Slides.pdf>