2019 International Conference on Indoor Positioning and Indoor Navigation (IPIN), 30 Sept. - 3 Oct. 2019, Pisa, Italy

Machine Learning applied to Wi-Fi fingerprinting: The experiences of the Ubiqum Challenge

Jordi Rojo*, Gabriel Ristow Cidral*, Germán Martín Mendoza-Silva,† Jorma Laiapea*,
Gerardo Parrello*, Arnau Simó*, Laura Stupin*, Deniz Minican*, María Farrés*, Carmen Corvalán*,
Florian Unger*, Sara Marín López*, Ignacio Soteras*, Daniel Castejón Bravo* and Joaquín Torres-Sospedra†
*Ubiqum Code Academy, Barcelona, Spain
†Institute of New Imaging Technologies, Universitat Jaume I, Castellón, Spain

***Abstract*—Wi-Fi Fingerprinting is widely adopted for smartphone-based indoor positioning systems due to the availability of already deployed infrastructure for communications. The UJIIndoorLoc database contains Wi-Fi data for indoor positioning in a large environment covering three multi-tier buildings collected with multiple devices. Since the evaluation set is private, developers and researchers are still allowed to evaluate their indoor positioning systems under the same evaluation conditions than the participants of the 2015 EvAAL-ETRI competition. This paper shows the results and the experiences of such kind of external evaluation based on a competition provided by the the students of the "Data Analytics and Machine Learning" program of the Ubiqum data academy, who applied the learned machine learning models taught during the program. The results show that state-of-art Machine Learning methods are providing good positioning results, but expertise on the problem is still needed.**

I. Introduction

Localization is at the core of numerous current applications and services [1]. While the localization problem is solved for most outdoor environments thanks to the inclusion of GNSS sensors into the mobile devices, it is still an open problem for challenging environments like urban canyons and indoors. Since most human activities are performed indoors, significant research and development efforts have been performed to develop indoor positioning systems (IPS) [2], mainly through technologies that use sensory system found in modern smartphones [3], which includes magnetometer, barometer, accelerometer, gyroscope, WiFi and BLE sensors. In particular, IPS based on WiFi received signal strength (RSS) readings are popular because the usual presence of already deployed WiFi infrastructure makes it an appealing low-cost solution applicable to many applications.

The preferred approach in WiFi based IPS is fingerprinting, in which RSS and the locations where they were collected are stored in a reference dataset, also known as the radio map. The radio map is later used to predict the location where a new RSS was measured. The nature of the propagation of WiFi signals in indoor spaces makes the WiFi fingerprinting a challenging subject [3], resulting in numerous fingerprinting methods being proposed [4], including the application of up-to-date machine learning (ML) approaches [5], [6].

Corresponding authors: Gabriel Ristow Cidral (`gabriel@ubiqum.com`) and Joaquín Torres-Sospedra (`jtorres@uji.es`)

The indoor localization problem based on WiFi fingerprinting, through the use of publicly available databases, is well suited for ML methods evaluation, given the high dimensionality problem and because it requires classification (building and floor estimation) and regression (longitude and latitude estimation). As a result, WiFi RSS datasets have been used for new ML method evaluation [7] and the study of ML methods application [8]. The Ubiqum code academy selected the UJIIndoorLoc dataset [9] for the "Ubiqum Challenge", which evaluated the ability of students of the "Data Analytics and Machine Learning program" [10] to apply ML approaches to real world problems. The students, organized in nine teams, were unfamiliar to indoor positioning. Although the UJIIndoorLoc has more than 21.000 samples, the students had to provide estimations for more than 5.000 additional blind samples. This test set, provided by the UJIIndoorLoc curator, did not contain location tags. The student estimations were evaluated under the same rules applied in the 2015 EvAAL-ETRI indoor competition [11], [12], [13], [14], where the private test set was first used.

The Ubiqum Challenge provided a great opportunity to explore the relation between ML approaches and WiFi fingerprinting IPS solutions. In this paper we explore that relation, having the following contributions:

- Showing that the direct application of state-of-the-art ML methods, without much knowledge of the indoor positioning problem, may challenge the results of many published proposals.
- Showing that, despite the expertise on the problem is helpful to improve the obtained accuracy, the nature of the problem has intrinsic difficulties that are not solved so far.
- The advantages of having blind evaluation procedures are shown for enhancing reproducible research and fair comparisons.

II. Materials and Methods

This section introduces the material and methods used in this work. First the data set used in the challenge is described. Later, the Ubiqum course and challenge are introduced. The approaches followed by the students are also described.

A. UJIIndoorLoc Dataset and EvAAL-ETRI Competition

The UJIIndoorLoc [9] is a WiFi fingerprinting public data set available at the UCI repository [15], [16] of machine learning and the IndoorLoc Platform [17], which was donated in order to ease IPS method development and contribute to reproducible research [18]. The public release contains a training set (19937 samples) and a validation set (1111 samples), whereas a third unlabelled test set (5179 samples) is available for external blind evaluation upon request.

The UJIIndoorLoc is a challenging database for ML problems, mainly to test WiFi fingerprinting and address known challenges, including: sparse collection, large dimensionality, device diversity, and large covered area. Along the years, several publications have used the UJIIndoorLoc public datasets to test their proposals, and it has been used even for ML student projects [8]. Some works using the UJIIndoorLoc achieved positioning mean errors around 6m to 7m [19], [20], [21], [22], [17], while others have reported mean errors larger than 8m [23], [7], [24], [25], [26]. The number of works using the UJIIndoorLoc is large, although it is not always possible to compare the reported results on this database given that some studies either because they report results individually for each building [27] or individually for latitude and longitude [28], use only a subset of the data [29], or they report numbers different from the mean error [30].

The IPIN Competition (formerly known as the EvAAL and EvAAL-ETRI Indoor Location Competition) aims at establishing benchmarks and evaluation metrics for comparing AAL solutions [12], [31]. In the 2015 edition [13], in conjunction with the 2015 IPIN Conference, the competition included an off-site Track devoted to "WiFi fingerprinting in large environments". Competitors had access to the public version for the *UJIIndoorLoc* database [9] available at the UCI repository [15], [16], to which they had to apply their algorithms off-line. For the competition, a private testing data without labels (ground-truth locations) was provided to the competitors and they had 6 weeks to provide their final predictions. The evaluation procedure was completely blind since the the competitors did not have access to the position labels and they did not have any feedback about their estimates. All competitors were able to submit up to five results files and only the best performing one (according to the competition metric) was considered for the competition and final rank.

The competition organizers evaluated all the submited results based on the positioning error, which was computed in two dimensions (latitude and longitude) as the euclidean distance between the real and estimated locations, penalties are added for floor error (4 m) and building error (50 m):

$$\begin{aligned} error =& distance_{lat,lon}(estimated, real) \\ &+ 4 * abs(estimated_{floor} - real_{floor}) \\ &+ 50 * (estimated_{bld} != real_{bld}) \end{aligned} \tag{1}$$

Finally, the metric used in the competition was the third quartile (75th percentile) of the error in all test samples.

B. Ubiqum course and Challenge

On Ubiqum's Data Analytics & Machine Learning program, students acquire the complex skills of a 21st century business strategist by working through a series of realistic projects, for example, time series forecasting and indoor localisation through data visualisation, data wrangling, and advanced ML techniques.

Ubiqum adopts a project-based, learn by doing methodology that allows students to reach a defined goal through the help of expert mentors. Simulating the activities of a professional data analyst, students come out armed with the right skills and knowledge to start their careers in data analytics.

The challenge proposed in the program was to apply the available techniques to unlabelled testing data in a evaluation procedure performed by an external actor and provide good models within three weeks. It is worth to highlight that the mentors and the students did not have any inside data about the ground truth of the final evaluation dataset.

C. Machine Learning Methods

Machine learning has been intensively applied to indoor localization. As a consequence, the works that propose novel indoor localization solutions usually include comparisons to well known ML methods already applied to the indoor localization problem [32], [20]. Given the large number of possible applicable method and their variants, comparative studies and works that review ML methods, even specifically to WiFi fingerprinting base positioning, are also available [5], [33], [6], [34].

We reviewed a set of 25 papers from the 4 last years that applied several ML methods to the indoor localization problem, and detected that the most used for WiFi fingerprinting, either as main method or for comparisons are the k Nearest Neighbours (k-NN) [35], Random Forest (RF) [36] or Decision Trees (DT) [37], Support Vector Machines (SVM) [38], Neural Networks (NN) [39] including Deep Neural Networks (DNN) [40] and Extreme Learning Machine (ELM) [41], and Naive Bayes (NB) [42]. Among the reviewed papers, the k-NN method was in 20 out of the 25 studies, while the RF/DT and SVM were and 13 and 12 out of 25, respectively. It is also common to find the previous methods that include or are combined with other techniques like boosting, bagging or ensembling that improve their performance.

The competing teams mainly applied the ML models that they were taught in the ML course, although they tested several methods available in ML toolkits of their choosing. The models included k-NN and two tree-based methods. The later methods has been reported to to provide remarkable results, especially if they involve ensembling methods, like in the case of RF and Gradient Boosted Trees (GBM and XGBoost). The rest of this section describes the taught methods.

Tree-based models use a variation of decision trees to analyse a training set and further prediction. A decision tree is an algorithm that partitions the data into subsets. This process starts with a binary split and continues until no further splits can be made. Various branches of variable length are formed.

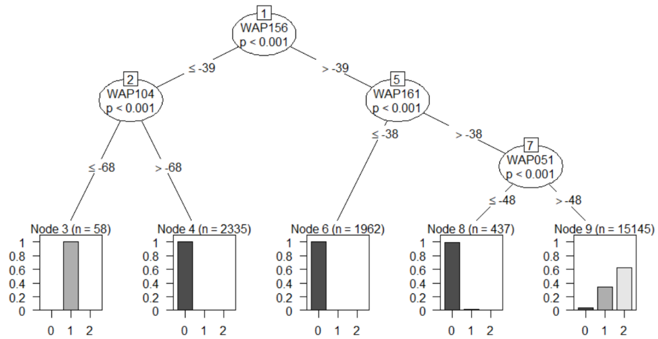

Fig. 1. Example of a tree-based model for Building prediction using the RSSI values on four APs

The goal of a decision tree is to encapsulate the training data in the smallest possible tree (see Figure 1)

1) Random Forest: A random forest is an ensemble learning method that operates by constructing a multitude of decision trees, normally trained with the bagging ensemble technique. This technique works by building many independent predictors/models/learners and combine them using some model averaging techniques. More specifically, the random forest produces several trees, and each tree is constructed by applying an algorithm on random subsets of a training set. This randomness helps to make the model more robust than a single decision tree, and less likely to overfit the training data.

The prediction of the random forest is obtained by a majority vote (classification problems) or an average (regression problems) over the predictions of the individual trees.".Unlike competing non-parametric techniques such as kernel methods or neural networks, random forests require very little tuning; experience has shown that one can often obtain good predictive models out-of-the-box with standard software like $randomForest$ for R [43].

The prediction of the random forest is obtained by a majority vote (classification problems) or an average (regression problems) over the predictions of the individual trees.".

In terms of parameters, the most common and impactful ones are:

- Number of trees: number of independent trees to be generated. In general, the more trees you use the better get the results. However, the improvement decreases as the number of trees increases, i.e. at a certain point the benefit in prediction performance from learning more trees will be lower than the cost in computation time for learning these additional trees.
- Mtry: It represents the number of variables randomly sampled as candidates at each split. Due to the conflicting evidence reported in the literature, different packages in R and Python have different optimal calculation estimates. In case of R, the $randomForest$ function uses as the default the square root of the number of predictor variables (rounded down) for classification models,. For regression models, it is the number of predictor variables divided by 3 (rounded down).

2) Gradient Boosted Trees: Gradient Boosted Trees differ from Random mostly on the way the trees are constructed. Instead of bagging, GBT leverage their trees by a boosting method. The idea behind is whether a weak learner can be modified to become better. Different than the Random Forest model, in which the trees are built in parallel, GBM trees are created sequentially. This method was already used in WiFi Fingerprinting works [8], [44].

For model tunning, there are typically three parameters:

- Number of trees: number of consecutive trees to be generated.
- Depth of trees: Represents the depth of each tree in the forest. The deeper the tree, the more splits it has and it captures more information about the data (again, by a cost of computational time).
- Learning rate: learning rate shrinks the contribution of each tree by a rate. This determines the impact of each tree on the final outcome. GBM works by starting with an initial estimate which is updated using the output of each tree. The learning parameter controls the magnitude of this change in the estimates. Lower values are generally preferred as they make the model robust to the specific characteristics of tree and thus allowing it to generalize well. Lower values would require higher number of trees to model all the relations and will be computationally expensive.

Depending on the employed loss function, base model and optimization scheme, the boosting model might differ in terms of accuracy. The Generalized Boosted Regression Models (GBM) package [45] takes the approach described in [46], [47], whereas eXtreme Gradient Boosting (XGB) package [48] is providing a scalable end-to-end system for tree boosting based on a novel sparsity aware algorithm [49].

3) k-NN: The *k-Nearest Neighbor* rule (k-NN) is a distance-based classifier which compares a current sample to all the labelled samples from a database [35]. This classifier requires generating a database for the comparisons (commonly known as a training set) where all the samples are properly labelled. In the case of indoor positioning, the samples are WiFi fingerprints, i.e., vectors with the acess points (AP) intensities, and the labels are the numerical values related to the real-world coordinates (longitude, latitude, altitude/floor and building).

There are two steps to estimate the position of a current fingerprint with k-NN:

1) The distance (or similarity) with respect to the current fingerprint is calculated for all the training fingerprints.
2) The k nearest (or most similar fingerprints) in the feature space are used to obtain the estimated position. In the simplest scenario, $k = 1$, the k-NN algorithm calculates the distance of a current fingerprint with respect to all the training fingerprints. The current position corresponds to the position (as longitude, latitude and altitude) of the training fingerprint which reported the lowest distance (or highest similarity).

The k-NN distance-based classifier has only one parameter (apart from the distance/similarity measure used to rank the training instances) which has to be properly set to obtain optimal results: the value of k. This values represents the number of samples from the database (fingerprints whose position is well-known) which are used to estimate the position of a device from a WiFi fingerprint. Setting a low value of k, such as 1, may be not adequate since only a single sample is considered to estimate the final position, whereas a high k value may consider points which are very far from the current position and, therefore, degrade the IPS accuracy.

D. Implemented methods

Each student team had up-to-five opportunities to submit results as the EvAAL-ETRI competitors had. This section describes the approach and model which provided the best results for each team. The teams have been anonymized and sorted according to the main evaluation metric, being "Team 01" the one reporting the best results in the Ubiqum challenge.

1) Team 01: They applied a cascading approach, in which XGBoost is used for building classification, then XGBoost is used for floor classification, and finally kNN is applied for regression on the latitude and longitude. The team include some data pre-processing that included considering for training only APs seen in the validation set, replacing the non-detection indicator value for -105dBm, and using a positive representation of the RSS values.

2) Team 02: The team used GBM for estimating the building, RF for floor prediction and two separate k-NN models for coordinates regression (latitude and longitude), cascading the predictions. Before training the models, they first removed APs observations higher than -30 dBm and lower than -90 dBm and then removed fingerprints and APs that were correlated to others in the database.

3) Team 03: They used RF to building, floor, latitude and longitude in that order using a cascading approach. The pre-processing steps that the team did were removing rows and columns with no variance and removing non-intersecting columns between train and validation sets.

4) Team 04: The team used RF for estimating in cascade the building, floor, and coordinates. Their pre-processing steps included replacing the non-detection indicator value for -105dBm, removing duplicate fingerprints and considering only APs common to both training and validation datasets.

5) Team 05: The team used RF building classification, floor classification and coordinates regression in that order. Before applying the model, they transform the data so that only APs common to the public and private set were kept, replacing the non-detection indicator value for -110dBm and removing fingerprints that did not contain valid RSS measurements.

6) Team 06: This team used k-NN for building classification, then RF for floor classification and later XGBoost for coordinates regression, using a cascading approach. They only used a subset if the training dataset and replaced the non-detection indicator value for -110dBm before the application of the models.

7) Team 07: They applied RF in a cascading manner to predict building, floor, longitude and latitude, in that order. Their pre-processing steps included removal of duplicates and zero variance attributes, replacement of the non-detection indicator value for -110dBm, and application of PCA to reduce dimensionality.

8) Team 08: The team used RF for building classification, for floor classification, and for latitude and longitude regression, in that order, using a cascade approach that considering the predicted building but not the predicted floor. As pre-processing, they included the removal of APs and fingerprints composed only by non-detection indicator values, removal of unmatching APs between training and test sets, replacement of the non-detection indicator value by -105dBm, and the addition of a new column indicating which the strongest detected AP in a fingerprint.

9) Team 09: They used RF for predicting in a cascade fashion building, floor and coordinates. Before the model application, the team replaced the non-detection indicator value by -105dBm, and reduce the training data by removing the least relevant APs.

III. Results

This section introduces the results of the EvAAL-ETRI Competitors [23], [50], [51], [52] and the competitors of the Ubiqum Code Academy Challenge. Moreover, some discussion is introduced about the results, the EvAAL-ETRI competition and the Ubiqum Code Academy Competition. Since databases and the evaluation restrictions were the same in both cases, a fair comparison is possible.

The results of the competing teams (EvAAL-ETRI and Ubiqum) are shown in Table I. Again, the mean error and the percentile values are based on the positioning error with penalties previously described. For Teams "Team 07", "Team 08" and "Team 09" we also provide the competition scores after manually fixing their estimates ("Team 07*", "Team 08*" and "Team 09*" rows).

First, Teams "Team 01", "Team 02" and "Team 03" have provided results between the "RTLSUM" and "ICSL", who were the winner and running-up teams in the EvAAL-ETRI competition according to the main evaluation metric of the competition (the third quartile or 75th percentile). The results of these three teams have some merit since the teams had no prior experience on the Indoor Positioning topic, so domain knowledge was minimum. Despite that lack of experience in indoor positioning and fingerprinting, the models developed by the competitors provided good accuracy given the challenging context of the employed data set (multi-building, multi-user and multi-device). Moreover, the results on the final test set are remarkable due to the time provided to students to create the models (only three weeks). On the other hand, "Team 07" , "Team 08" and "Team 09" provided the worst positioning errors, mainly due to the wrong floor detection. In cascade prediction, a wrong initial estimation might have a very negative impact on the final model accuracy.

TABLE I
EVAAL-ETRI COMPETITION AND UBIQUM CODE ACADEMY CHALLENGE RESULTS

	MeanError	25thperc	50thperc	75thperc	95thperc	100thperc	Floor Hit Rate	Building Hit Rate
RTLS@UM	6.20	2.51	4.57	8.34	15.81	52.27	93.74	100
ICSL	7.67	3.10	5.88	10.87	19.68	39.14	86.93	100
HFTS	8.49	3.69	6.99	11.60	19.93	40.70	96.25	100
MOSAIC	11.64	3.26	6.72	12.12	21.54	313.33	93.86	98.65
Team 01	6.98	2.99	5.64	9.62	17.20	36.07	84.09	100.00
Team 02	7.24	3.35	5.96	9.77	17.95	214.99	82.14	99.98
Team 03	8.04	3.76	7.18	10.67	18.55	41.88	82.83	100.00
Team 04	8.92	4.08	7.47	11.80	21.21	127.94	82.02	99.81
Team 05	9.34	4.11	7.75	12.54	21.32	156.84	83.24	99.83
Team 06	10.13	3.54	8.48	14.32	24.96	132.46	82.29	99.98
Team 07	12.54	7.99	11.08	15.60	24.62	59.83	3.69	100.00
Team 08	12.80	8.05	11.48	15.92	25.22	46.60	2.76	100.00
Team 09	14.21	8.76	12.26	17.04	29.39	129.17	1.87	99.88
Team 07*	8.83	4.20	7.40	11.97	20.75	55.83	81.29	100.00
Team 08*	9.02	4.25	7.68	12.12	21.43	42.60	81.85	100.00
Team 09*	10.37	4.92	8.41	13.19	25.40	125.17	85.48	99.88

Further analysis on the results showed that the wrong floor identification of the bottom ranked teams was due to a programming error. The original floor identifier was moved from range $[0, 1, 2, 3, 4]$ to range $[1, 2, 3, 4, 5]$ because of internal indexing by these teams. After manually correcting their estimates for the later set of results the errors were coherent with the errors reported by the EvAAL-ETRI competitors.

Compared to most of EvAAL-ETRI competition, only 4 teams of the Ubiqum Code Academy challenge were able to provide a perfect model to detect or identify the building. Failing at detecting the building in a cascade/waterfall prediction has the intrinsic positioning error of providing the final location (latitude, longitude and floor) far from the expected one plus the 50 m penalty.

In terms of floor identifier estimation, all teams provided a floor hit rate much lower compared to the EvAAL-ETRI competitors. The difference between the best competitor of the EvAAL-ETRI and Ubiqum Challenge is higher than 10% which affected the mean positioning error due to the introduced penalty of 4 m. Moreover, teams “Team 07”, “Team 08” and “Team 09” completely failed at determining the correct floor. Although it initially seems that the advanced Machine Learning methods did not succeed in detecting or estimating the floor, the error was because of a programming mistake.

Finally, Figure 2 shows the Cumulative Distribution Function (CDF) of the positioning errors for all the teams participating in the challenge. This figure can graphically show a better overview of the positioning errors plus penalties. Three main groups can be identified. First, “Team 01” and “Team 02” provide the best results of the Challenge and their accuracy is very similar through the plot. Second, “Team 07”, “Team 08” and “Team 09” provide the worst results of the challenge and the vast majority of positioning errors are above 4 m, mainly because of the wrong floor estimation. Third, the remaining teams (“Team 03”, “Team 04”, “Team 05” and “Team 06”) provide intermediate results.

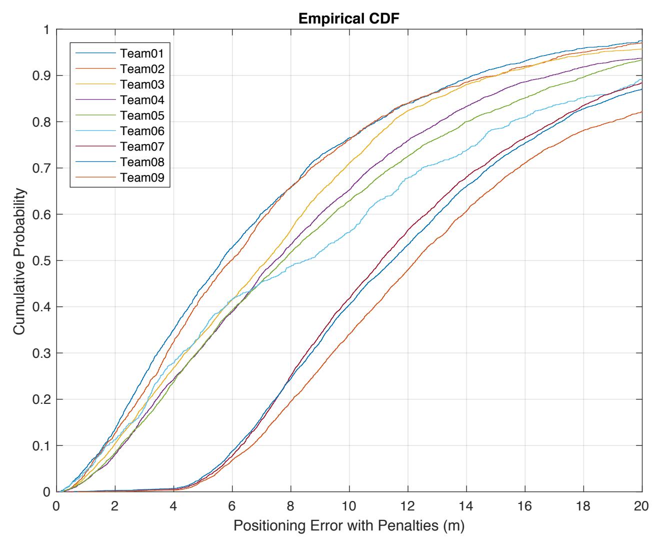

Fig. 2. CDF of the Ubiqum Code Academy Challenge Teams.

Regarding the last group (“Team 03”, “Team 04”, “Team 05” and “Team 06”), there are two teams showing particular results: “Team 03” provides similar results than “Team 01” and “Team 02” for the 75th superscript and above. “Team 06” provided good results for low percentile values, in-line with the best performing teams, but then later the positioning accuracy degrades providing large positioning errors in-line to the worst performing teams.

It should be highlighted that the Ubiqum code academy challenge students only had three weeks to provide the estimated positions, instead of the six weeks that the EvAAL-ETRI competitors had. Nevertheless, avoiding using the validation test for initial test leaded to a error with ranked these teams to the last positions with a poor evaluation grade when they could have been ranked in better positions.

IV. Discussion

The mean errors commonly reported from other works using only the public training and validation sets of UJI-IndoorLoc are similar to those reported from the EVAAL-ETRI competition, which also used the private dataset for evaluation. This similarity hints at the challenging nature of this database for the localization problem and the precision challenge intrinsic to WiFi fingerprinting for large spaces. Apart from the EVAAL-ETRI competition results, the only known work which provided results for the private dataset is Torres-Sospedera et al. [53], achieving 6.19m as mean error. This work is of a major importance given that the author was a EVAAL-ETRI competition organizer and the curator of the UJIIndoorLoc. The facts that some Ubiqum Challenge competitor results are closer to the best ones from the EVAAL-ETRI competitors and that the Ubiqum Challenge competitor results are inline with those from works using the public dataset suggest that the proper application of state-of-the-art methods from known ML toolkits like keras [54], sklearn [55], weka [56] or some R packages [48], [45] should suffice for achieving position accuracies not worse than 1m to 2m in the case of collection setting like than of the UJIIndoorLoc, which is not that significant given a best mean error of around 6m. Also, the comparison of the result confirms that knowing the problem and expertise in positioning field are relevant for fine tuning a machine learning system in order to achieve non-trivial improvements in the positioning precision.

A basic training in ML approaches was enough for the Ubiqum Challenge students to known that an exploratory analysis and pre-processing was required before the application of ML models. The UJIIndoorLoc contains measurements of 520 APs, but an AP is usually detected only in a given environment, which can be a building, or some floors of a building. As a result, the students realized the convenience of using a cascade or waterfall approach in which, for a given test sample, the building estimation is used as a refinement for floor estimation, and the building (and sometimes floor) estimation is used as a refinement for coordinates estimation as well. Also, the high dimensionality suggested to filter out non-relevant APs, mainly those than were not common to all measurement sets and those whose measurements were correlated to other AP measurements. Additionally, it was clear to the students to deal building and floor determination as supervised classification problem, and the coordinates estimation as supervised regression problems.

The teams mainly replaced the non-detection indicator value by a value lower than the minimum detected RSS (e.g., $-110dBm$). A significant number of studies usually use this approach, although it has been shown that omitting the usage of non-detected AP may improve accuracy. Furthermore, the students applied filtering over extreme RSS values (e.g., APs reporting $0dBm$), which is also a common approach in WiFi fingerprinting works. The applied filters successfully led them to remove outliers from the set (or sets) used for training purposes.

Some teams noted that the validation set has a rich spatial diversity, which led them to merge the training and validation sets. As a consequence, the competitors had better radio maps, and thus were able to build better models, than if they had used only the training set. However, it should only be applied if the selected model has been previously tested using the training set for training and the validation set for testing purposes. Using the validation data for testing is highly encouraged to select the best model parameters with independent data sets and, moreover, test that the estimations are coherent with the expected outputs. In fact, the results of the worst ranked teams indicate that they had not validated the model before submitting their solution, which led to a very poor floor identification and a penalty of 4 m in almost all samples.

The applied ML models are in-line with those taught in the course but they also are in-line with the ones used in WiFi fingerprinting works. The students tried several method in the toolkits they decided to use, and RF, for classification and regression, was the most often selected. RF not only produced good estimations, but also was able to handle in a reasonable time the training of the whole dataset. The two best performing teams used k-NN, which is a widely used method in WiFi fingerprinting IPS works for regression on the coordinates. The two best performing teams also used gradient boosting, either as XGBoost or GBM. Gradient boosting requires much more experience to configure the hyperparameters manually than RF or k-NN to obtain good accuracy, which might have make it less appealing to the students.

It seems that even using the procedures provided in the toolkits to optimise the model hyperparameters selection , the advanced ML models are not able to provide the best full-position estimations and improve k-NN based solutions, like the one proposed by UMinho team in [52]. The best team in the Ubiqum Challenge was based on k-NN for longitude and latitude estimation and its performance was slightly worse than the method proposed in [52]. One of the possible causes is that the UJIIndoorLoc data set includes hundreds of indoor locations in three buildings and, therefore, all the APs detected across them are in the data set. For each fingerprint, the average detected APs is around 18, so most of values in each fingerprint vector corresponds to missing values. Although assigning a low RSS as default non-detected value is a valid solution for pattern-matching algorithms such as k-NN, it might not be the best strategy for other models such as ANN or RF. In those advanced models, clustering approaches in combination with feature selection should be considered to minimize the effect of the missing values.

WiFi IPS deployed in real scenarios should be able to handle the large datasets required to characterize large environment, likely including several buildings with several floors. Some teams reported that the times required to first train their models, and later to compute the estimations, were large. Although the training time should not be a severe concern, the estimation times are important to consider. Despite the good accuracy provided by the competitors solutions, their approaches might not be adequate for real-time deployments.

V. Conclusions

This paper introduced the results of the Ubiqum Code Academy Competition (Ubiqum challenge) for evaluating the Student's skills on Machine Learning in a real-world use case. Nine teams participated in a fair competition where the participants provided position estimations for 5,179 blind test WiFi fingerprints, i.e., neither the students nor the mentors had access to the ground truth. The blind test fingerprints constitute the private set of the UJIIndoorLoc, a large dataset with publicly available training and validation data sets. The UJIIndoorLoc database curator evaluated the up-to-three solution sets, provided by each team, with estimations on all the blind fingerprints, which enabled to compare the teams results with the ones provided by the competitors of the EvAAL-ETRI Competition in 2015.

Through the comparison of the team estimations with those from EvAAL-ETRI Competition in 2015, in addition to reviewing the accuracies reported from works that have used the public datasets of the UJIIndoorLoc, this paper showed that students with only a basic training in ML methods in without experience in indoor positioning, are able to obtain accuracies results similar to many published IPS proposals. Two participants of the Ubiqum challenge ranked in the second and third positions in the overall ranking that considers both competition events. The student were not able to rank first in the overall ranking and, in general, they obtained poorer floor detection rates than the EvAAL competitors. Also, the students reported time issues for training their models and performing the estimations. The students results and the comparisons presented in this paper hint usefulness of ML for indoor positioning, the importance of experience in indoor positioning problems for IPS development, and that new fundamental approaches may be needed to overcome the known challenges of WiFi fingerprinting.

Contrary to the Machine Learning research field, where the proposed methods are usually tested on a large set of diverse databases, a small and laboratory made data set is usually used to test indoor positioning systems. Hopefully, this trend is changing and researchers are starting to use the UJIIndoorLoc database and similar ones for comprehensive evaluation, but there is still a long path to walk to reach the evaluation standards used in machine learning. As an output of this research, it is important to highlight that, apart from the training and evaluation/test data sets, the database curator should provide an independent validation set for parameter selection and internal evaluation of the model. This allows to set the model parameters with independent data, not correlated with training data, and ensures that the model is providing, more or less, estimates that are coherent with the expected outputs avoiding some format or programming issues. This validation set is of especial relevance in WiFi fingerprinting due to the high correlation between those fingerprints sequentially taken in a row in the same location during a short period of time.

However, the evaluation procedure settled by the UJIIndoorLoc curators might raise one important concern. If the dataset is kept private by one research team, who ensures that the database curators are giving the same chances and opportunities to all teams?. Thus the necessity of providing some rules to the indoor positioning community to provide more high-quality datasets and providing the ways to fairly evaluate indoor positioning systems without giving the researchers access to the ground truth of the testing samples.

Acknowledgment

Germán M. Mendoza-Silva gratefully acknowledges funding from grant PREDOC/2016/55 by Universitat Jaume I.

References

[1] H. Huang and G. Gartner, "Current trends and challenges in location-based services," *ISPRS International Journal of Geo-Information*, vol. 7, no. 6, 2018. [Online]. Available: http://www.mdpi.com/2220-9964/7/6/199

[2] R. F. Brena, J. P. García-Vázquez, C. E. Galván-Tejada, D. Muñoz-Rodriguez, C. Vargas-Rosales, and J. Fangmeyer, "Evolution of indoor positioning technologies: A survey," *Journal of Sensors*, vol. 2017, 2017.

[3] P. Davidson and R. Piché, "A survey of selected indoor positioning methods for smartphones," *IEEE Communications Surveys Tutorials*, vol. 19, no. 2, pp. 1347–1370, Secondquarter 2017.

[4] S. Xia, Y. Liu, G. Yuan, M. Zhu, and Z. Wang, "Indoor fingerprint positioning based on wi-fi: An overview," *ISPRS International Journal of Geo-Information*, vol. 6, no. 5, p. 135, 2017.

[5] S. Bozkurt, G. Elibol, S. Gunal, and U. Yayan, "A comparative study on machine learning algorithms for indoor positioning," in *2015 International Symposium on Innovations in Intelligent SysTems and Applications (INISTA)*, Sep. 2015, pp. 1–8.

[6] B. A. Akram, A. H. Akbar, B. Wajid, O. Shafiq, and A. Zafar, "Locswayamwar: finding a suitable ml algorithm for wi-fi fingerprinting based indoor positioning system," in *International Telecommunications Conference*. Springer, 2019, pp. 111–123.

[7] C. J. Tonde, "Supervised feature learning via dependency maximization," Ph.D. dissertation, Rutgers University-Graduate School-New Brunswick, 2016.

[8] D. Li, L. Y. Wang, and S. Wu, "Indoor positioning system using wifi fingerprint," http://cs229.stanford.edu/proj2014/DanLi,LeWang,ShiqiWu,IndoorPositioningSystemUsingWifiFingerprint.pdf, 2014, accessed:2019-05-23.

[9] J. Torres-Sospedra, R. Montoliu, A. Martínez-Usó, J. P. Avariento, T. J. Arnau, M. Benedito-Bordonau, and J. Huerta, "UJIIndoorLoc: A new multi-building and multi-floor database for WLAN fingerprint-based indoor localization problems," in *Proceedings of the Fifth Conference on Indoor Positioning and Indoor Navigation*, 2014, pp. 261–270.

[10] "Data analytics and machine learning program website," https://ubiqum.com/programs/data-analytics-and-machine-learning-program/, 2019, accessed:2019-05-23.

[11] J. A. Álvarez-García, P. Barsocchi, S. Chessa, and D. Salvi, "Evaluation of localization and activity recognition systems for ambient assisted living: The experience of the 2012 evaal competition," vol. 5, no. 1, pp. 119–132, Jan. 2013.

[12] P. Barsocchi, S. Chessa, F. Furfari, and F. Potortì, "Evaluating AAL solutions through competitive benchmarking: the localization competition," vol. 12, no. 4, pp. 72–79.

[13] F. Potortì, P. Barsocchi, M. Girolami, J. Torres-Sospedra, and R. Montoliu, "Evaluating indoor localization solutions in large environments through competitive benchmarking: The evaal-etri competition," in *2015 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, Oct 2015, pp. 1–10.

[14] J. Torres-Sospedra, A. J. C. Moreira, S. Knauth, R. Berkvens, R. Montoliu, O. Belmonte, S. Trilles, M. J. Nicolau, F. Meneses, A. Costa, A. Koukofikis, M. Weyn, and H. Peremans, "A realistic evaluation of indoor positioning systems based on wi-fi fingerprinting: The 2015 evaal-etri competition," *JAISE*, vol. 9, no. 2, pp. 263–279, 2017.

[15] D. Dua and C. Graff, "UCI machine learning repository," 2017. [Online]. Available: http://archive.ics.uci.edu/ml

[16] J. Torres-Sospedra, R. Montoliu, A. Martínez-Usó, and J. Huerta, "UJIIndoorLoc data set," https://archive.ics.uci.edu/ml/datasets/UJIIndoorLoc, 2014.

[17] R. Montoliu, E. Sansano-Sansano, J. Torres-Sospedra, and Óscar Belmonte-Fernández, "Chapter 12 - IndoorLoc Platform: A Web Tool to Support the Comparison of Indoor Positioning Systems," in *Geographical and Fingerprinting Data to Create Systems for Indoor Positioning and Indoor/Outdoor Navigation*, ser. Intelligent Data-Centric Systems, J. Conesa, A. Pérez-Navarro, J. Torres-Sospedra, and R. Montoliu, Eds. Academic Press, 2019, pp. 225 – 247.

[18] D. Nüst, C. Granell, B. Hofer, M. Konkol, F. O. Ostermann, R. Sileryte, and V. Cerutti, "Reproducible research and giscience: an evaluation using agile conference papers," *PeerJ*, vol. 6, p. e5072, Jul. 2018.

[19] J. Torres-Sospedra, R. Montoliu, S. Trilles, Óscar Belmonte, and J. Huerta, "Comprehensive analysis of distance and similarity measures for wi-fi fingerprinting indoor positioning systems," *Expert Systems with Applications*, vol. 42, no. 23, pp. 9263 – 9278, 2015.

[20] B. A. Akram, A. H. Akbar, and O. Shafiq, "Hybloc: Hybrid indoor wi-fi localization using soft clustering-based random decision forest ensembles," *IEEE Access*, vol. 6, pp. 38 251–38 272, 2018.

[21] M. T. Hoang, Y. Zhu, B. Yuen, T. Reese, X. Dong, T. Lu, R. Westendorp, and M. Xie, "A soft range limited k-nearest neighbors algorithm for indoor localization enhancement," *IEEE Sensors Journal*, vol. 18, no. 24, pp. 10 208–10 216, Dec 2018.

[22] H. Gan, M. H. B. M. Khir, G. Witjaksono Bin Djaswadi, and N. Ramli, "A hybrid model based on constraint oselm, adaptive weighted src and knn for large-scale indoor localization," *IEEE Access*, vol. 7, pp. 6971–6989, 2019.

[23] R. Berkvens, M. Weyn, and H. Peremans, "Localization Performance Quantification by Conditional Entropy," in *Proceedings of the Sixth International Conference on Indoor Positioning and Indoor Navigation (IPIN)*.

[24] J. Wietrzykowski, M. Nowicki, and P. Skrzypczyński, "Adopting the fab-map algorithm for indoor localization with wifi fingerprints," in *Automation 2017*, R. Szewczyk, C. Zieliński, and M. Kaliczyńska, Eds. Cham: Springer International Publishing, 2017, pp. 585–594.

[25] K. S. Kim, S. Lee, and K. Huang, "A scalable deep neural network architecture for multi-building and multi-floor indoor localization based on wi-fi fingerprinting," *Big Data Analytics*, vol. 3, no. 1, p. 4, Apr 2018.

[26] B. Jia, B. Huang, H. Gao, and W. Li, "Dimension reduction in radio maps based on the supervised kernel principal component analysis," *Soft Computing*, vol. 22, no. 23, pp. 7697–7703, Dec 2018.

[27] W. Njima, I. Ahriz, R. Zayani, M. Terre, and R. Bouallegue, "Smart probabilistic approach with rssi fingerprinting for indoor localization," in *2017 25th International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, Sep. 2017, pp. 1–6.

[28] M. T. Uddin and M. M. Islam, "Extremely randomized trees for wi-fi fingerprint-based indoor positioning," in *2015 18th International Conference on Computer and Information Technology (ICCIT)*, Dec 2015, pp. 105–110.

[29] L. Xiao, A. Behboodi, and R. Mathar, "A deep learning approach to fingerprinting indoor localization solutions," in *2017 27th International Telecommunication Networks and Applications Conference (ITNAC)*, Nov 2017, pp. 1–7.

[30] C. Zhou and A. Wieser, "Cdm: Compound dissimilarity measure and an application to fingerprinting-based positioning," in *2018 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, Sep. 2018, pp. 1–7.

[31] F. Potortì, A. Crivello, and F. Palumbo, "Chapter 11 - The EvAAL Evaluation Framework and the IPIN Competitions," in *Geographical and Fingerprinting Data to Create Systems for Indoor Positioning and Indoor/Outdoor Navigation*, ser. Intelligent Data-Centric Systems, J. Conesa, A. Pérez-Navarro, J. Torres-Sospedra, and R. Montoliu, Eds. Academic Press, 2019, pp. 209 – 224.

[32] N. Hernández, J. M. Alonso, and M. Ocaña, "Fuzzy classifier ensembles for hierarchical wifi-based semantic indoor localization," *Expert Systems with Applications*, vol. 90, pp. 394 – 404, 2017.

[33] M. A. Nuño-Maganda, H. Herrera-Rivas, C. Torres-Huitzil, H. Marisol Marín-Castro, and Y. Coronado-Pérez, "On-device learning of indoor location for wifi fingerprint approach," *Sensors*, vol. 18, no. 7, 2018.

[34] Y. Li, Z. Gao, Z. He, Y. Zhuang, A. Radi, R. Chen, and N. El-Sheimy, "Wireless fingerprinting uncertainty prediction based on machine learning," *Sensors*, vol. 19, no. 2, 2019.

[35] T. Cover and P. Hart, "Nearest neighbor pattern classification," vol. 13, no. 1, pp. 21–27.

[36] T. K. Ho, "Random decision forests," in *Proceedings of 3rd international conference on document analysis and recognition*, vol. 1. IEEE, 1995, pp. 278–282.

[37] J. R. Quinlan, "Induction of decision trees," *Machine Learning*, vol. 1, no. 1, pp. 81–106, Mar 1986. [Online]. Available: https://doi.org/10.1007/BF00116251

[38] V. Vapnik, I. Guyon, and T. Hastie, "Support vector machines," *Mach. Learn*, vol. 20, no. 3, pp. 273–297, 1995.

[39] D. E. Rumelhart, G. E. Hinton, R. J. Williams *et al.*, "Learning representations by back-propagating errors," *Cognitive modeling*, vol. 5, no. 3, p. 1, 1988.

[40] I. Sutskever, G. E. Hinton, and A. Krizhevsky, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, pp. 1097–1105, 2012.

[41] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: theory and applications," *Neurocomputing*, vol. 70, no. 1-3, pp. 489–501, 2006.

[42] N. Friedman, D. Geiger, and M. Goldszmidt, "Bayesian network classifiers," *Machine learning*, vol. 29, no. 2-3, pp. 131–163, 1997.

[43] S. Wager and G. Walther, "Adaptive concentration of regression trees, with application to random forests," in *arXiv preprint, arXiv:1503.06388*, 2015.

[44] F. Potortì, S. Park, A. R. Jiménez Ruiz, P. Barsocchi, M. Girolami, A. Crivello, S. Y. Lee, J. H. Lim, J. Torres-Sospedra, F. Seco, R. Montoliu, G. M. Mendoza-Silva, M. D. C. Pérez Rubio, C. Losada-Gutiérrez, F. Espinosa, and J. Macias-Guarasa, "Comparing the performance of indoor localization systems through the evaal framework," *Sensors*, vol. 17, no. 10, 2017.

[45] "Gbm package in cran (the comprehensive r archive network)," https://cran.r-project.org/web/packages/gbm/index.html, 2019, accessed:2019-05-23.

[46] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *The Annals of Statistics*, vol. 29, no. 5, pp. 1189–1232, 2001.

[47] ——, "Stochastic gradient boosting," *Computational Statistics & Data Analysis*, vol. 38, no. 4, pp. 367 – 378, 2002, nonlinear Methods and Data Mining.

[48] "Xgb package in cran (the comprehensive r archive network)," https://cran.r-project.org/web/packages/xgboost/index.html, 2019, accessed:2019-05-23.

[49] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '16. New York, NY, USA: ACM, 2016, pp. 785–794.

[50] S. Knauth, M. Storz, H. Dastageeri, A. Koukofikis, and N. Mähser-Hipp, "Fingerprint calibrated centroid and scalar product correlation rssi positioning in large environments," in *Proceedings of the Sixth International Conference on Indoor Positioning and Indoor Navigation (IPIN)*.

[51] S. Choi, J. Yoo, and H. J. Kim, "Machine learning for indoor localization: Deep learning and semi-supervised learning," in *USB On-Site Proceedings of the Sixth International Conference on Indoor Positioning and Indoor Navigation (IPIN)*.

[52] A. Moreira, M. J. Nicolau, F. Meneses, and A. Costa, "Wi-fi fingerprinting in the real world - RTLS@UM at the EvAAL competition," in *Proceedings of the Sixth Conference on Indoor Positioning and Indoor Navigation*, 2015.

[53] J. Torres-Sospedra, G. M. Mendoza-Silva, R. Montoliu, O. Belmonte, F. Benitez, and J. Huerta, "Ensembles of indoor positioning systems based on fingerprinting: Simplifying parameter selection and obtaining robust systems," in *2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, Oct 2016, pp. 1–8.

[54] "Keras: The python deep learning library," https://keras.io/, 2019, accessed:2019-05-23.

[55] "scikit-learn: Machine learning in python," https://scikit-learn.org/stable/, 2019, accessed:2019-05-23.

[56] "Weka 3: Machine learning software in java," https://www.cs.waikato.ac.nz/ml/weka/, 2019, accessed:2019-05-23.