This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

A Fuzzy Logic Based System for Indoor Localisation using WiFi in Ambient Intelligent Environments

Teresa Garcia-Valverde, Alberto Garcia-Sola, Hani Hagras, *Senior Member*, *IEEE*, James Dooley, Victor Callaghan and Juan A. Botia

Abstract—Ambient Intelligence is a new information paradigm where people are empowered through a digital environment that is "aware" of their presence and context, and is sensitive, adaptive and responsive to their needs [23]. Hence, one of the important requirements for Ambient Intelligent Environments (AIEs) is the ability to localise the whereabouts of the user in the AIE to address her/his needs. In order to protect the user privacy, the use of cameras is not desirable in AIEs and hence there is a need to rely on non intrusive sensors. There are various localisation means available for outdoor spaces such as those which rely on satellite signals triangulation. However, these outdoor localisation means cannot be used in indoor environments. The majority of non intrusive and non camera based indoor localisation systems require the installation of extra hardware such as ultra sound emitters/antennas, RFID antennas, etc. In this paper, we will propose a novel indoor localisation system which is based on the WiFi signals which are free to receive and they are available in abundance in the majority of domestic spaces. However, the free WiFi signals are noisy and uncertain and their strengths and availability are continuously changing. Hence, we will present a fuzzy logic based system which employs the free available WiFi signals to localise a given user in AIEs. The proposed system receives WiFi signals from a big number of existing WiFi Access Points (up to 170 Access Points) where no prior knowledge of the access points locations and the environment is required. The system employs an incremental life long learning approach to adjust its behaviour to the varying and changing WiFi signals to provide a zero cost localisation system which can provide high accuracy in real world living spaces. We have compared our system in both simulated and real environments to the other relevant techniques in the literature and we have found that our system outperforms the other systems in the offline learning process, whereas our system was the only system capable of performing online learning and adaptation. The proposed system was tested in real world spaces from a living lab intelligent apartment (iSpace) to a town centre apartment to block of offices. In all these experiments, our system has given high accuracy to detect the user in the given AIEs and the system was able to adapt its behaviour to changes in the AIE or the WiFi signals. We envisage the proposed system to play an important role in AIEs especially for privacy concerned situations like elderly care scenarios.

***Index Terms*—Fuzzy Logic systems, Localisation Systems, Ambient Intelligence, online learning.**

Manuscript received October 3, 2011. This research work is supported by the Spanish Ministry of Science and Innovation under the grant AP2007-04269 of the FPU program and the Séneca Grants program through the 09493/FPI/08 grant.

T. Garcia-Valverde is with the Department of Information and Communications Engineering, University of Murcia, Espinardo, Murcia, 30071, Spain. (corresponding author to provide phone: +34 868 88 78 82; e-mail: mtgarcia@um.es).

A. Garcia-Sola, J.A. Botia are with the Department of Information and Communications Engineering, University of Murcia, Espinardo, Murcia, 30071, Spain. (e-mail: {agarciasola,juanbot}@um.es).

H. Hagras, J. Dooley and V. Callaghan are with the School of Computer Science and Electronic Engineering, Fuzzy Systems Research Group, Computational Intelligence Centre, University of Essex, Colchester CO4 3SQ, U.K. (e-mail: {hani,jpdool,vic}@essex.ac.uk).

I. INTRODUCTION

AMBIENT Intelligence aims to create digital environments that are "aware" of ***our presence and context***, and that are sensitive, adaptive and responsive to our needs [1], [2], [23]. Hence, in order to create an Ambient Intelligent Environment (AIE), there is a need for the environment to be aware of the user presence and context which entails the need to localise the whereabouts of the user. In order to protect the user privacy, there is a need to rely on non intrusive sensors.

There are various localisation systems available for outdoor spaces. The vast majority of outdoor localisation systems rely on triangulation and decoding timing signals like in GSM/UMTS [45] and Satellite positioning systems [24], [9], [37]. However, the outdoor localisation systems cannot be used in indoor environments where their effectiveness is limited [8].

Indoor localisation systems need higher accuracy than outdoor localisation systems due to the relatively narrower spaces available in indoor environments where few meters could make a difference and mean a different room or even a different floor in a building. During the last ten years, there has been a very good progress in indoor localisation systems [39] where there have been big focus on using wearable badges [43], [55] and there have been also commercial products like the UbiSense system [47]. However, the majority of the indoor localisation systems need to go regularly through a time consuming calibration process and they have poor robustness and a high cost as they require new hardware and thus they are less ubiquitous [39]. Hence, there is a need to investigate new indoor localisation systems which are non intrusive (to protect the user privacy), ubiquitous, cheap and do not necessitate new hardware equipment [11].

Nowadays almost every modern dwelling has several WiFi Access Points (APs) accessible nearby, and their signal may be measured (e.g. using RSSI (Received Signal Strength Indicator)) to provide the user location [4], [8], [42]. This alternative is truly ubiquitous and inexpensive as it does not require specialised hardware where it uses WiFi signals that are freely available. Several approaches have shown that the use of wireless signals properties (such as RSSI [31] or AoA (Angle of Arrival) [41]) is suitable to infer location. Some significant examples are RADAR [8] or Active Campus [29]. However, it has been shown that the RSSI is usually more indicative of location than other signal properties [8].

When using RSSI, most of the proposals use stochastic approaches to infer location, like Hidden Markov Models (HMM) [35], [36], Bayesian filtering [54] or Monte Carlo techniques [46]. Other kinds of approaches which have also been applied are clustering techniques [53] and genetic algorithms [16]. However, none of these techniques are easily understandable by human beings and, in some cases, they are not able to deal with inexact and uncertain information. Because of that, some works used fuzzy logic in order to handle the encountered uncertainties and provide systems whose behaviour could be easily understood and analysed by the users [7], [15], [48]. Nonetheless, all the previous work necessitated the knowledge of all the available APs and their location, which is not practical in most environments. Other proposals used hybrid fuzzy logic systems (using neural networks, Kalman filters, etc) to deal with the above problems [5], [52]. However, these approaches need some prior knowledge and these hybrid fuzzy systems reduced the understandability of fuzzy classifiers. Without the need of prior knowledge of the APs, the work of Alonso et al. [6] obtains the position of a robot, but this system was not able to learn online and adapt to the highly dynamic environments.

As described above, the existing works show some weaknesses which made them not suitable to work in the long term within real-world changing environments. Therefore, we present a novel fuzzy logic system which is able to localise the users in an indoor environment using only the RSSI from unknown WiFi APs and a few days of training with no need for prior knowledge, new equipment or extra cost. The technique is able to automatically learn offline and online to adapt in order to deal with the environmental changes. We have compared our proposed technique with the current techniques in the literature and we will show that our system outperforms the other techniques in the offline comparison. In addition, to our knowledge our system is the first system to be able to operate online and adapt its behaviour in an incremental and life long learning approach with no prior knowledge of the given environment. We will show the evaluation of our system in real world spaces ranging from a living lab intelligent apartment (iSpace) to a domestic apartment in the town centre to an office block where we will show the results achieved by our system which has produced a very good performance with a high accuracy.

The presented system is zero cost and it is easy and fast to deploy and it can be embedded in wearable devices. Hence, the proposed system can be used in different ambient intelligence scenarios including ambient assisted living (for elderly people or people with disabilities) for example to notify the carer of unusual situations like if an elderly person did not move for the whole day which can mean that the given person might not feel well. The system could be also used in marketing services, for instance, sending text messages of publicity when the user is near a given shop. In addition, the proposed system can provide an automatic management of devices based on the user's location which could be used to reduce the daily energy-consumption and increase the given user comfort, for example turning off the lights and heating if no user is in the room or adjusting the heating and light levels in the various rooms in the house to maximise the user comfort and reduce the consumed energy.

In the next section, we will present the problems encountered with WiFi signals and the pre-processing and post-processing techniques we will employ in this paper. Section III presents our online lifelong learning fuzzy logic based system for indoor localisation using WiFi signals. Section IV presents the experiments and results in both simulated and real scenarios which demonstrate the feasibility and adaptability of our proposed system. Finally, Section V presents the conclusions and future work.

II. WiFi Signals Measurement Problems and the Employed Pre-Processing and Post-Processing Techniques

Our solution is based on receiving WiFi signals from APs of unknown location and whose existence, signal strength and signal reflection is continuously changing in the given environment. In addition, the APs signals are subject to noise and uncertainty. The proposed system handles all of these problems using a fuzzy logic based classification system which is able to handle the uncertainty, noise and imprecision. Nevertheless, in order to improve the system performance in the long term and handle the changes in the environment an online learning process has been introduced.

However, WiFi RSSI is highly inaccurate and can produce errors in the system. Some of these errors can be detected and minimised using some pre- and post-processing techniques over the received signals. In Section II.A we describe the problems associated with measuring the RSSI of the WiFi signals. Section II.B introduces the employed pre- and post-processing techniques applied to the received signals to improve their characteristics and reduce the encountered noise levels.

A. *The WiFi Measurement Problems*

Received Signal Strength Indicator (RSSI) is a measurement of the power present in a received WiFi signal. Using WiFi RSSI as input to predict location offers several advantages, including a zero cost solution. However, there are many problems encountered when using the WiFi RSSI, these problems include:

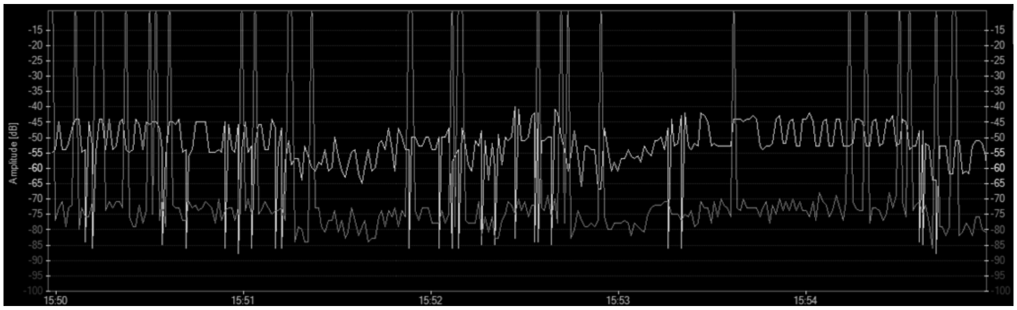


Fig. 1. RSSI measuring.

- **Weather condition variations** such as fog or rain affect the RSSI [19], and therefore, the prediction. In addition, these variations do not affect all of the APs the same way [19].
- **2.4 GHz is an unlicensed spectrum**. This means different hardware and applications apart from WiFi can freely use this spectrum to cause interference, noise and cross talk. The most common instruments which use this frequency and can interfere with WiFi are: microwaves (they were the very first household electronic devices to emit interference in the 2.4GHz band, data throughput can fall by 64% within 25ft of a microwave), Bluetooth devices, Telecommunications Device for the Deaf (TDD) phones, Digital Enhanced Cordless Telecommunications (DECT) phones, baby monitors, wireless speakers and console controllers, wireless security cameras [17], [25], and in the case of the intelligent spaces, location systems such as UbiSense.
- **Artificial variations and interferences**. Ideally, signal variations should only be affected by distance. The longer the distance to an AP, the smaller the RSSI. Walls usually keep this true. However, interferences and some undesired effects such as reflection can make the system detect a signal at a smaller RSSI being closer to the AP. These may be caused by different set of conditions[1] such as analogue video senders (they can even hide the SSID of an AP), building materials and reflecting structures (specially metals) can block and degrade signals, and mirrors cause signals reflection, the weather (as described above), Christmas lights (which can reduce WiFi performance by 25%), power cables (because of the electromagnetic radiation) or even old CRT televisions [17], [25].
- **APs are distributed throughout different channels** where a standard WiFi card is not able to listen at all the channels at the same time. It has to constantly hop over all the channels to discover new APs and update the RSSI of the already discovered APs. This means that a complete update of all the visible APs at a specific moment in time lasts few seconds, and this takes place in rather small steps. In every step a new channel is updated. Furthermore there are two available bands (2.4 GHz and 5 GHz). In the 2.4GHz band there are up to 14 channels available, and up to 42 in the 5GHz band. When a router transmits on each channel, the effective width of its signal is about 20MHz, which, in the 2.4GHz space, means it can overlap up to eight neighbouring channels. Therefore, if more than three networks are in close proximity to one another, co-channel and adjacent channel interference become a problem[2].
- **RSSI measuring in WiFi is highly inaccurate**. Fig. 1 shows the high variability of the RSSI during 5 minutes. Only two selected APs are shown for the clarity of the image. Not only changes in the signal are visible but also peaks and drops (meaning a disappearance of the AP).
- **Variability of available APs**. Fixed installed APs usually remain in the same position for long periods. However, people can create their own APs using laptops or mobile phones. Especially in crowded places there are a lot of users who do this, which tricks the system leading to a wrong prediction. Not only the APs can change their location, but they can also appear and disappear (be turned off). Some routers do not only broadcast one AP, but also have **virtual APs** where the APs have different MAC addresses, but they share common hardware and therefore, common RSSI.

B. Pre-processing and Post-Processing of the WiFi Signals

1) Pre-processing of the WiFi signals

As seen in Section II.A, WiFi RSSI measurement can be highly inaccurate, difficult to interpret, and prone to produce errors in prediction. Some of these errors are intrinsic to the WiFi technology and are very difficult to avoid by software, but some other errors could be detected and minimised using some pre-processing techniques.

After examining the raw data obtained from different days at different places we found out very interesting behaviours of the data (caused by the problems described in the previous subsection) where there is big drop in RSSI caused by temporal blackouts, i.e. an AP which is not seen in a particular time lapse. We have also noticed peaks where sometimes the AP returns a very high level of RSSI (unreachable in normal conditions).

In an ideal situation with no noise and perfect and accurate hardware when the user stands still, the signal might remain constant. When the user moves backward from an AP the

[1] http://www.pcpro.co.uk/features/367681/what-can-get-in-WiFis-way

[2] http://www.pcpro.co.uk/features/367675/how-WiFi-works

RSSI with respect to that AP should decrease smoothly as the user moves, and when the user gets closer to the AP the RSSI should increase. Hence, we have implemented a pre-processing strategy which aims to imitate this ideal world to a get a smoother signal while trying to minimise the noise, imprecision and uncertainty associated with the RSSI measurement.

We have made the pre-processing using two different alternatives: the first one is based on a heuristic pre-processing (based on the expertise gained from monitoring the RSSI in the tests) which is only valid for WiFi signals. The second pre-processing approach is based on more universal and automatic techniques valid for any signal of this kind, therefore other antennas such as WiMaX and Bluetooth could be pre-processed in the same manner. Section II.B.1.a will present the heuristic pre-processing approach while Section II.B.1.b will present the responsive universal pre-processing approach.

a) Heuristic Pre-processing

The heuristic pre-processing routines were hardcoded for WiFi RSSI to enable the API to distinguish between a real loss of signal (the AP is not visible anymore) and a temporary one (the reader has not detected it, but the AP is still in the reader's range). Depending on the age (i.e. the last time seen) and the last RSSI of the AP we proceed as follows:

- If the age is short (recently seen) and the last RSSI was high (user close to the AP), a miss is assumed to be. But the AP is still there (i.e. the card missed it) and hence the last RSSI value is returned.
- If the age is short but the last seen RSSI was low (far for the AP), the miss could be a real miss or a failure; therefore, the last seen value is just decreased (rather than considered as "No Detected") by a constant value, up to a minimum limit corresponding with the "No Detected" value.
- If the age is high the AP can be assumed to be gone, therefore it is considered as "No Detected".
- If the difference of the current RSSI value with the last value is high, it is limited to a fixed value. If there is a higher difference there might be an error.

b) Responsive Universal Pre-processing

The purpose of the pre-processing is getting the real signal closer to the noiseless signal achieved under ideal conditions. Using high frequency measuring readers (i.e. getting many reads per second) the measured RSSI should be a curve which decreases when the reader gets further from the AP and increases as the reader gets closer to the AP. Besides, the RSSI should be smooth with no spikes or jumps. Having this in mind, we have carried tests using the EMA (Exponential Moving Average) smoothing filter [38] to smooth the incoming RSSI. However, the EMA techniques were not promising where comparing the results with no pre-processing the improvement was really small and compared to the heuristic pre-processing explained above the performance was even worse.

Hence, we have developed another technique which aims to be responsive, since a new increase in the RSSI value should be reflected the very same moment it arrives. The pre-processing should be adaptive and it should remove single erroneous peaks, retain big isolated changes but if the RSSI changes start getting bigger it should allow big quick changes. Since different APs and different types of antennas can be used, the pre-processing should also be independent of the technology.

To satisfy the above aims, first, a trim like operation [20] is performed to remove peaks and drops. We then employ a modified *standard deviation* based technique to be applied to the last received RSSI values for every AP. Let *RS* be the set of the last ξ RSSI received values for a specific AP sorted out in an ascending order according to their strength:

$$RS = \{RSSI_1, RSSI_2, \dots, RSSI_{\xi-1}, RSSI_\xi\} \mid RSSI_1 < RSSI_2 < \dots < RSSI_{\xi-1} < RSSI_\xi \quad (1)$$

In order to remove the erroneous peaks and drops, 35% of the highest and lowest values of *RS* are removed. Different tests have been performed while experimenting with removing various percentages of the highest and lowest values. Smaller values than 35% tend to keep some peaks or drops, whereas bigger values leave too few remaining information values. Thus we have used the value 35% in our tests as it has shown to be effective to remove most of the outliers, whereas the remaining values will be useful to find out the residual difference of the RSSI, which tells how the difference under normal conditions is, hence we will have a reduced set of RS which we will call MRS.

Once peaks and drops have been removed, the remaining values must represent the difference determined by the movement of the user with respect to the AP with some noise. In such way, the standard deviation σ of the ξ last values is calculated as follows:

$$\sigma_\xi = \sqrt{\frac{\sum_{\Delta=0}^{\kappa}(\gamma_\Delta - \zeta)^2}{\kappa}} \text{ where } \kappa = |MRS| \text{ and } \zeta = \frac{\sum_{\Delta=0}^{\kappa}\gamma_\Delta}{\kappa} \quad (2)$$

where κ is the size of the set *MRS*, γ_Δ is the Δ-th value of *MRS*, and ζ is the average of the values of *MRS*.

Now the output of the Responsive Universal Pre-processing for a given RSSI as input γ_ξ is defined as:

$$\begin{cases} \gamma_\xi & \text{if} \quad \gamma_{\xi-1} - \sigma_\xi \le \gamma_\xi \le \gamma_{\xi-1} + \sigma_\xi \\ \gamma_{\xi-1} + \sigma_\xi & \text{if} \quad \gamma_{\xi-1} + \sigma_\xi < \gamma_\xi \\ \gamma_{\xi-1} - \sigma_\xi & \text{if} \quad \gamma_{\xi-1} - \sigma_\xi > \gamma_\xi \end{cases} \quad (3)$$

where γ_ξ is the incoming value and $\gamma_{\xi-1}$ the previous value in time. If the difference between γ_ξ and $\gamma_{\xi-1}$ is greater than the standard deviation σ_ξ, the signal is truncated to a maximum (by adding the standard deviation to the old signal) to avoid peaks and drops and vice versa. The standard deviation self-adapts to changes in the input.

2) Post-processing

In spite of the pre-processing, the classification output is not 100% accurate, as sometimes there is no way to differentiate between a movement and a difference caused by external factors (noise); this leads to sporadic misclassifications. Also, the lack of a comprehensive training which includes all the available APs or failures in the training process can lead to misclassifications. This kind of errors takes the form of an isolated error followed by a right value. To prevent this behaviour, since the user usually stays a (relatively) long time in every room we have used a post-processing routine based on a sliding window. This is only used when the system is classifying, not in the training phase. This post-processing mechanism returns the most common value of the last ϑ outputs (i.e. the *mode* from the last ϑ values). In our system ϑ is set to 10. A bigger value than 10 has shown in the tests to delay the output, whereas using a smaller value the sliding window was not big enough to filter the erroneous values.

III. An Online Life Long Learning Fuzzy Logic Based System for Indoor Localisation using WiFi Signals

In this paper, we present a novel online life long learning fuzzy logic system for indoor localisation using WiFi signals. The proposed technique is capable of handling the encountered uncertainties associated with WiFi signals in indoor environments to produce good accuracy for localising people in indoor AIEs. Furthermore, the proposed system adapts in an online life long learning approach to the encountered changes in the WiFi signals, APs and the environment. The proposed fuzzy logic based system provides the ability of fuzzy rules to approximate independent local models for mapping a set of inputs to a set of outputs to provide transparent and flexible representations that are easily adaptable with a high degree of interpretability [22].

As discussed in the following subsection, the approach presented in this work is able to automatically learn offline and online to adapt to the changes of the highly dynamic intelligent environments.

A. The Fuzzy Offline Learning Approach

1) Capturing Input/Output Data

The offline approach starts by monitoring the user physical location in the AIE (i.e. the location of the user in the AIE, i.e which room, corridor, etc) and collects the signal strength readings from all the accessible WiFi APs (whose physical locations are unknown). Hence the system records a 'snapshot' of the current inputs WiFi RSSI ($x^{(t)}$) and the associated physical location in the AIE ($C_q^{(t)}$). These 'snapshots' are accumulated over a period of time (less than a minute in each location is enough) so that the system observes as much of the user's presence within the environment as possible. From the accumulated data, the offline approach then learns a descriptive model for the user localisation. Therefore given a set of data pairs:

$$\left(x^{(t)}, C_q^{(t)}\right), \mathrm{t} = 1, 2, \ldots, \mathrm{N} \quad (4)$$

where N is the number of data instances, $x^{(t)} \in R^n$ and $C_q^{(t)} \in R^k$. The offline system then extracts rules which describe how the k output locations are related to the n input RSSI variables $x = (x_1, \ldots, x_n)^T \in R^n$ based on the sampled data. In our experiments, the number of WiFi APs has reached 170 APs. The fuzzy rules which are extracted represent local models that map a set of inputs to the set of outputs without the need of formulating any mathematical model. Individual rules can therefore be adapted online influencing only specific parts of the descriptive model learnt by the offline system.

Fig. 2 shows the software program used to capture input data, which allows the system to perform the offline and online training. It also shows the output data which permits the user to interact with the system (in the right part the user can notify the right location if there was an error, or even introduce new locations). Finally, the output is used to validate the system.

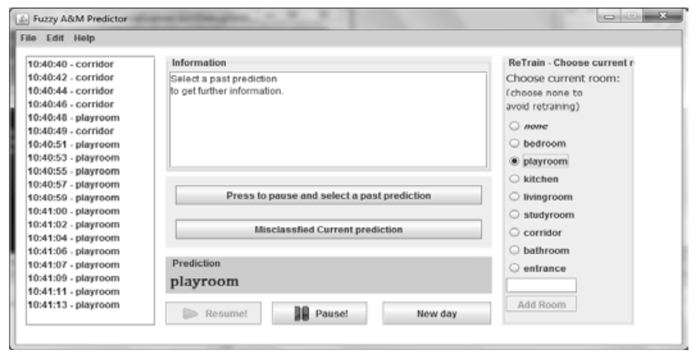


Fig. 2. Desktop predictor software.

2) Fuzzy Membership Functions Generation

Four fuzzy sets are used to categorize the input crisp values of the RSSI of the APs in the scenario. The fuzzy sets are defined by $A = \{“Close”, “Medium”, “Far”, “No\ Detected”\}$. Another additional fuzzy set is also used which is *"Don´t Care"*, to deal with the problem of incomplete IF rules which can be caused by the variability of the number of APs over time (this problem is described deeply in the next subsection).

The fuzzy sets *"Close"*, *"Medium"* and *"Far"* provide an appropriate granularity of the fuzzy partition of each attribute when the APs have been detected. An AP is detected if its crisp value x_i is within the interval U, where U defines the range where the RSSI of the AP can be detected. Otherwise, the AP is not detected in that place, i.e. $x_i \notin U$.

We have used left and right shoulder membership functions respectively for the *"Close"* and *"Far"* fuzzy sets and a triangular membership function for the *"Medium" fuzzy set*. The defined membership functions are drawn in Fig. 3.

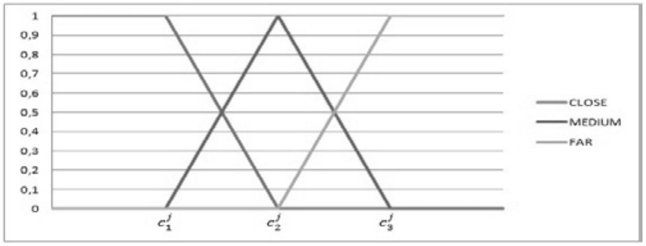


Fig. 3 Membership functions.

The fuzzy sets are defined over the *U* interval. In order to partition the *U* interval in the three fuzzy sets for a detected AP, the Fuzzy c-means clustering algorithm (FCM) [10] is used. The FCM defines a set of *p* clustered regions over the sampled data. Hence, there are *p* centres $\overline{c_1}, \overline{c_2}, \ldots, \overline{c_p}$ defined for these clustered regions. The number of clusters *p* is predefined and, in our case, *p* was set to 3 corresponding to *"Close"*, *"Medium"* and *"Far"*. Using these centres, the distances of each instance from each cluster centre is calculated and used to assign each instance with a degree of membership to each cluster.

On the other hand, the fuzzy set *"No Detected"* is used when the AP could not be detected in a specific place (but had been detected previously). Note that this fuzzy set is different from the special fuzzy set "Don´t Care", which is used when no information about the AP is obtained in any place that day. In such way, the shape of *"Don´t Care"* membership function, $\mu_{A_{Don't\,Care}}(x)$, is the same as the unit interval, where $\mu_{A_{Don't\,Care}}(x) = 1\ for\ x\ \in U$, whereas the shape of *"No Detected"* membership function, $\mu_{A_{No\ Detected}}$, is the zero-one membership function defined as: $\mu_{A_{No\ Detected}}(x) = 0\ if\ x\ \in U\ and\ \mu_{A_{No\ Detected}}(x) = 1\ if\ x\ \notin U.$

3) *Fuzzy Rule Extraction*

The defined set of membership functions are combined with the existing input/output data to extract the rules defining the user localisation behaviour based on the WiFi RSSI. The fuzzy sets for the antecedents and consequents of the rules divide the input and output space into fuzzy regions.

In such way, the used fuzzy rules are defined for the *n*-dimensional pattern classification problem as follows:

$$\begin{aligned} &Rule\ R_q\colon IF\ x_1\ is\ A_1^{(q)}\ and \ldots and\ x_n\ is\ A_n^{(q)} \\ &\qquad then\ Class\ C_q\ with\ CF_q \end{aligned} \tag{5}$$

where $q = (1, 2, \ldots, M)$, *M* is the number of rules and *q* is the index of the rules and $x = (x_1, \ldots, x_n)^T$ is a *n*-dimensional pattern vector (the *n*-dimension of the problem is defined by the number of APs in the scenario) representing the RSSI from the WiFi APs. $A_i^{(q)}, i = 1, \ldots, n$ is an antecedent fuzzy set for the *i*-th WiFi attribute (the *i*-th attribute of the rule R_q is the RSSI of the *i*-th AP in the scenario), C_q is a consequent location class. The consequent class in this work is one of the possible locations in the scenario, for example, *kitchen*, *bathroom* or *bedroom* in a home scenario or *office1* and *office2* in an office building scenario. In such way, a room level granularity of location is used to infer the location of the user. A room level granularity has been shown to provide sufficient location information for many AIEs applications [30]. The consequent classes are updated automatically from the training set and if a new location is added after the training phase, the online learning algorithm is able to update this information in the system. Therefore, no prior information about the structure or map of the building is required in this approach. CF_q is rule confidence [34] which is explained in Step 2) below.

The different steps involved in the offline approach for rule extraction and location classification are as follows now described:

Step 1) For an input-output pair $\left(x^{(t)}, C_q^{(t)}\right)$, $t = 1, 2, \ldots, N$, compute the membership values $\mu_{A_i^k}(x_i{}^{(t)})$ for each membership function $k = 1, \ldots, V$ and for each input variable *i* $(i = 1, \ldots, n)$ find $k^* \in \{1, \ldots, V\}$, such that

$$\mu_{A_i^{k^*}}\left(x_i^{(t)}\right) \geq \mu_{A_i^k}(x_i{}^{(t)}) \tag{6}$$

for all $k = 1, \ldots, V$. Let the following rule be called the rule generated by $(x^{(t)}, C_q^{(t)})$:

$$\begin{aligned} &Rule\ R_q\colon IF\ x_1^t\ is\ A_1^{k^*}\ and \ldots and\ x_n^t\ is\ A_n^{k^*} \\ &\qquad then\ Class\ C_q^{(t)} \end{aligned} \tag{7}$$

For each input variable x_i there are V fuzzy sets A_i^k, $k = 1, \ldots V$, to characterise it; so that the maximum number of possible rules that can be generated is V^n. However given the dataset only those rules among the V^n possibilities whose dominant region contains at least one data point will be generated. In Step 1) one rule is generated for each input–class pair, where for each input the fuzzy set that achieves the maximum membership value at the data point is selected as the one in the "IF" part of the rule, as explained in (6),(7). The firing strength of the generated rule is calculated as follows:

$$w^{(t)} = \prod_{i=1}^{n} \mu_{A_i^q}(x_i\ (t)) \tag{8}$$

The firing strength of a rule $w^{(t)}$ is a measure of the strength of the points $x^{(t)}$ belonging to the fuzzy region covered by the rule.

Step 2) Step 1) is repeated for all the *t* data points from 1 to *N* to obtain the *N* data rules in the form of (5). Since the size of the training set can be quite large, the rule base could also be quite large. As many rules share the same antecedents but they have different consequents, in order to resolve these conflicting rules, each rule R_q is weighted with a confidence grade CF_q. The confidence grade of a rule can be thought as the degree of belief for that rule or a measure of the validity of the rule [40]. Accordingly, when a new input causes several rules to be fired with different consequent classes, the confidence of the rule is used to determine which rule is the most reliable to classify this input. In this step, the confidence of each rule is measured.

The N rules are divided into groups, with rules in each group sharing the same "IF" part. If we assume that there are H such groups, let group l have N_l rules. Then, each group l is divided into groups, with rules in each group sharing the same consequent C_q. Hence, if we assume that there are G such groups, let group b have N_b rules. Thus, each rule in a group b shares same "IF" part and same consequent. In such way, the confidence of the fuzzy rule $A_q \Rightarrow C_q$ is defined as [33], [34]:

$$CF_q = c\left(A_q \Rightarrow C_q\right) = \frac{\sum_{e=1}^{N_b} w(t_e^b)}{\sum_{u=1}^{N_l} w(t_u^l)} \tag{9}$$

where the consequent of the group b is C_q, $e = 1, \ldots, N_b$, and t_e^b is the index for the data points in the group b. And $u = 1, \ldots, N_l$, and t_u^l is the index for the data points in the group l.

Step 3) Step 2) finishes the creation of the rule base over the training data. Hence when a new input $(x_1, \ldots, x_n)^T$ arrives from the testing data, the system is able to classify the incoming input to map it to a given location. For each input $x^{(t)}$ its compatibility grade with each rule R_q in the rule base is measured by the firing of the rule, $w_q^{(t)}$. If $w_q^{(t)} > 0$ the rule q will be fired. Then, we employ the single winner method mentioned in [33] for classifying the new input. Hence, for a new input $x^{(t)}$, $t = 1, 2, \ldots, N$, compute the values $w_q^{(t)}$, for each rule $q = 1, \ldots, M$ and find $q^* \in \{1, \ldots, M\}$, such that

$$w_{q^*}^{(t)} \cdot CF_{q^*} \geq w_q^{(t)} \cdot CF_q \text{ for all } q = 1, \ldots, M \tag{10}$$

Consequently, the new input is classified with the consequent class of the winner rule R_{q^*}. If no rule is fired, the input will be rejected and, therefore, it will be not classified.

Step 4) The total number of combinations of antecedent fuzzy sets in this problem is 4^n for an n-dimensional pattern classification scenario, i.e. a scenario where n APs have been detected. Besides, knowing that the number of APs in a small scenario is around 50 APs, it is unfeasible to have all the fuzzy rules corresponding to all the combinations of the antecedent fuzzy sets even in this small scenario. Therefore, it is possible that in Step 3) none of the existing rules in the rule base may be fired and then the input is rejected. In order to deal with this problem, every time that there is a misclassification, a process based on similarity of fuzzy rules is performed, rather than rejecting the input (because no rule is fired).

Other works have used the similarity approach to determine the location using the idea that closer devices have more similar signals [14]. However we do not have any idea about the physical locations of the APs. Hence, in this paper we present a new approach where the similarity between the incoming inputs and the existing rules in the rule base is measured as a function distance. When an input $x^{(t)}$ is rejected in Step 3), its compatibility grade with each rule R_q is measured using similarity. The most similar rule is chosen to classify the input.

Hence, in this step, for each misclassified input $x^{(t)}$ in Step 3), its membership values are computed according to (6), where the fuzzy set that achieves the maximum membership value at the data point is selected: $A_i^{k^*}(x_i\,(t))$, where $k = 1, \ldots, V$ and $i = 1, \ldots, n$. Hence, we obtain a rule for the input $x^{(t)}$ where this rule antecedent part is written as follows:

$$IF\ x_1^t\ is\ A_1^{k^*}\ and \ldots and\ x_n^t\ is\ A_n^{k^*} \tag{11}$$

In order to calculate the similarity in the antecedent parts between the rule generated by the input $x^{(t)}$ and each rule R_q, a function distance, defined as $\mathfrak{D}\,(A_i^{k^*}, A_i^{(q)})$, is used between $A_i^{k^*}(x_i\,(t))$ and the i-th antecedent of the "IF" part of that rule. With this aim, we define a distance that find the difference between the linguistic labels which are coded. Thus, the distance between "*Close*" and "*Medium*" is $\mathfrak{D}\,(\text{"}Close\text{"}, \text{"}Medium\text{"}) = 1$, as well as, $\mathfrak{D}\,(\text{"}Medium\text{"}, \text{"}Far\text{"}) = 1$, $\mathfrak{D}\,(\text{"}Far\text{"}, \text{"}No\ Detected\text{"}) = 1$ and $\mathfrak{D}\,(\text{"}No\ Detected\text{"}, \text{"}Don\acute{}t\ Care\text{"}) = 1$, $\mathfrak{D}\,(\text{"}Close\text{"}, \text{"}Far\text{"}) = 2$, etc .

Finally, using this distance, the similarity between the rule created by the input $x^{(t)}$ with each rule R_q is calculated as:

$$\mathcal{S}\left(x^{(t)}, R_q\right) = \frac{\sum_{i=1}^{n}\left(1 - \frac{\mathfrak{D}\left(A_i^{(t)}, A_i^{(q)}\right)}{V-1}\right)}{n} \tag{12}$$

where $\mathcal{S}\left(x^{(t)}, R_q\right) \in [0,1]$, V is the number of fuzzy sets and $i = 1, \ldots, n$, where n is the number of RSSI values of the inputs which is the number of antecedents of the rule, i.e. n-dimensional problem.

The higher the similarity distance value is, the more similar the rule generated by the incoming inputs and the existing rule in the rule base is. Therefore, for each misclassified input $x^{(t)}$ compute the similarity values $\mathcal{S}\left(x^{(t)}, R_q\right)$ for each rule R_q, $q = 1, \ldots, M$ and find $R_{q^*}, q^* \in \{1, \ldots, M\}$, such that

$$\mathcal{S}\left(x^{(t)}, R_{q^*}\right) \geq \mathcal{S}\left(x^{(t)}, R_q\right) \quad \text{for all } q = 1, \ldots, M. \tag{13}$$

Consequently, the new input is classified with the consequent class of the rule R_{q^*}.

B. *Incremental Online Learning*

Despite the offline algorithm described above is able to use the RSSI to predict locations over different kinds of scenarios, there are still some issues and limitations which prohibit wide availability and reliability of the system over time. This section describes the problems which limit a high performance of the system over time and the proposed incremental online learning to deal with such situations. The online incremental life long learning approach provides the capability of adapting the system over time which has not been approached in previous works.

In order to tolerate this dynamism, the presented fuzzy rule-based classification system is modified to be able to adapt the model to face the variations of the environment over time. In such way, the system is able to face up variations over number of APs in the environment. Fig. 4 shows an overview of the overall proposed localisation system which starts with the offline learning which was described in the previous subsection. After the offline learning a fuzzy rule base $S = \{R_1, \dots, R_M\}$ is available in the system to classify the location. When a new day starts this fuzzy rule base is loaded into the system and new inputs are received to predict their locations. However, the conditions of the new day may have changed and the system should adapt to this by triggering the online adaptation module.

In such way, when a new input $x^{(t)}$ arrives to the system the set of APs could have changed. If so, on the one hand, the rule base should be updated to reflect the new information and the system should learn from these changes. In that case, a new fuzzy rule base *S'* is created based on the original rule base *S*. On the other hand, the new input should be fit to the rule base. The different steps involved in the online approach to create the new rule base and location classification with online learning is as follows:

Step 1) For an input data $x = (x_1, \dots, x_g) \in R^g$, *g* input WiFi RSSI variables are received based on the new input. Notice that now a *g*-dimensional pattern vector is received and the rule base is *n*-dimensional. This is because the number and set of APs can have changed. Therefore, some changes are required in order to allow the system to classify the incoming input to map it to a given location using the process described in Step 3) of the offline learning.

Firstly, a new fuzzy rule base *S'* is created to represent the current information in the environment. Then, for each rule $R_q \in S$, a new equal rule R_z is created in *S'*, where $z = (1, 2, \dots, Y)$, *Y* is the number of rules in *S'* and *z* is the index of the rules. Nevertheless, some changes are made over the rule R_z. Let $A \in R^n$ be the set of AP identifiers in the rule base *S'* and $B \in R^g$ is the set of AP identifiers in the input. In such way, for each rule in *S'* some changes are made. For each AP identifier in *B* not included in *A*, a new antecedent is added to each rule with the value *"Don´t Care"*. This modification is made with the aim of dealing with incomplete rules (they are incomplete regarding the new APs discovered). Finally, when all the rules have been updated, the confidence of the rules in *S'* is calculated following (9).

In a similar way, a change process is performed over the input $x = (x_1, \dots, x_g)$. Firstly, its membership values are computed according to (6), where the fuzzy set that achieves the maximum membership value at the data point is selected: $A_i^{k^*}(x_i(t))$, where $k = 1, \dots, V$ and $i = 1, \dots, n$. Then, for each AP identifier in *A* not included in *B*, a new fuzzy set is added to the input with the value *"Don´t Care"*.

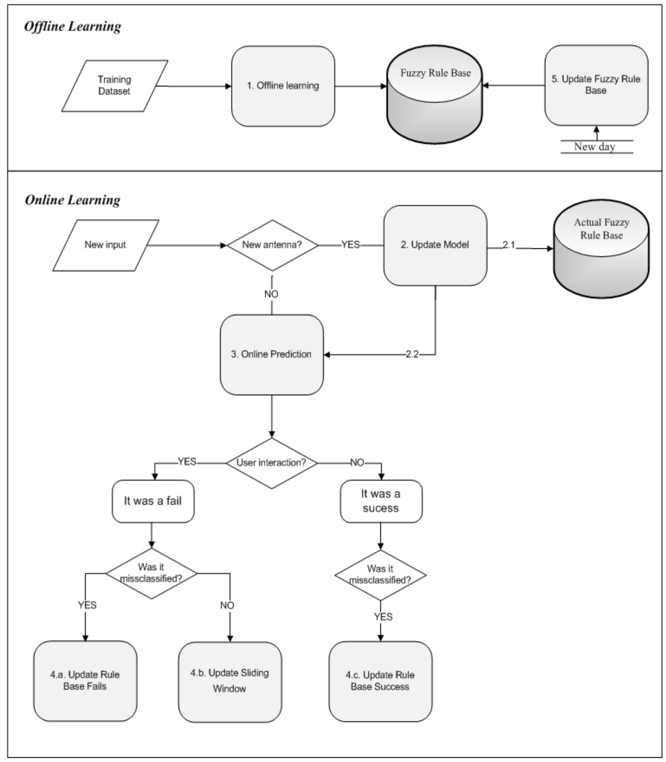


Fig. 4 Overview of the overall proposed localisation system.

After this process, both the rule base *S'* and the input have the same dimension, a *f*-dimension, where *f* is given by the new number of APs. Then, the system is able to classify the incoming input to map it to a given location as described in Step 3) of the offline learning, but using the rule base *S'*. This step of the fuzzy rule base updating is labelled as *Process 2* in Fig. 4.

Step 2) Step 1) updates the rule base when new APs are contained in an input, i.e., new AP were discovered in the scenario. Since there is new information in the scenario from the new APs, the system may learn it. Thus now, when the system predicts a new input, two kinds of learning are possible depending on the intervention of the user where if the user interacts with the system to notify an error of the prediction, a non-automatic learning will be produced (*Processes 4.a.* and *4.b* in Fig. 4). However, even if the user does not interact with the system, an automatic learning will be produced (*Process 4.c.* in Fig. 4).

Three new structures are used in order to do the two kinds of learning: two temporal fuzzy rule bases (fuzzy rule base of fails ($S_{Fails} = \{R_{Fails_1}, \dots, R_{Fails_a}\}$, a = 1,..., Q, Q is the number of rules in S_{Fails} and fuzzy rule base of success ($S_{Sucess} = \{R_{Success_1}, \dots, R_{Success_d}\}$, $d = 1, \dots, E$, E is the number of rules is S_{Sucess} and *d* is the index of the rules) and a sliding window (*W*).

Step 2.1) **Non-automatic learning.** The non-automatic learning is produced when the user notifies an error to the system. This error could be produced by an error in the fuzzy rule based classification system or by an error in the process of similarity caused by a misclassification as described in the previous section. In that second case, the error is produced because there was not enough information in the rule base and because of that, no rule was fired and the similarity process was applied. This situation is different from the first case; therefore, both cases are handled in different ways.

In such way, if an input $x^{(t)}$ is classified by the similarity process and the user notifies an error in this classification, the system performs a non-automatic learning about this input. Let the rule which was used to classify the input $x^{(t)}$ be called R_z (i.e. the most similar rule to the input):

$$Rule\ R_z\colon IF\ x_1\ is\ \ A_1^{(z)}\ and\ \dots and\ x_f\ is\ A_f^{(z)} \\ then\ Class\ C_z\ with\ CF_z \tag{14}$$

where $R_z\ \in S'$, the consequent class C_z was wrong and the user notified that the right class should be C_c. Then the system creates a new rule in the rule base S_{Fails} to reflect this new information. Following Step 1) of the offline learning for rule regeneration, a new rule is generated by $(x^{(t)}, C_c)$:

$$Rule\ R_a\colon IF\ x_1\ is\ A_1^{k^*}\ and\ \dots and\ x_f\ \ is\ A_f^{k^*} \\ then\ Class\ C_c\ with\ CF_a \tag{15}$$

where $a = (1, 2, \dots, Q)$, Q is the number of rules in S_{Fails}, a is the index of the rules. Furthermore, $R_a\ \in S_{Fails}$, $A_h^{k^*}\left(x_h\ (t)\right)$ is the maximum membership value achieved at the data point, $k = 1, \dots, V$ and $h = (1, \dots, f)$, where f is the dimension of the pattern vector which represents the RSSI from the WiFi APs. The consequent of the rule is the class which was notified by the user as right for this input, C_c. Notice that C_c can be one of the existing locations, where $C_c\ \in\ R^k$, or a new location defined by the user (e.g. the structure of the building has changed) and, then, the new location will be included in R^k.

Finally, the confidence of the rules in S_{Fails} is recalculated as in (9) for each rule in S_{Fails}. This process is always performed when an input is classified wrongly by the similarity process and the user notifies the error. Then, when another new input arrives, the system uses the rule base S_{Fails} to classify and if no rule is fired, the rule base *S'* will be used as usual. This process enables the system to correct immediately the error and use the new information for the next classifications. This process is labelled as *Process 4.a.* in Fig. 4.

On the other hand, if an input $x^{(t)}$ is classified by a fired rule R_z in the rule base *S'* and the user notifies an error in this classification, the system also performs a non-automatic learning about this input, but in a different way than in the previous situation described above. In this case, the fired rule R_z is introduced in the sliding window *W*. We take a sliding window of [illegible] fired rules which produced wrong classifications. Each rule in the window is recorded as a triple $\{R_z, C_c, \mathscr{v}\}$, where $R_z\ \in S'$, C_c is the right consequent notified by the user and $\mathscr{v}$ is the number of times that the rule has produced this error. In such way, if the user notifies an error with the right consequent C_c and the error was produced by a fired rule $R_z\ \in S'$, the system will check if the pair (R_z, C_c) was already in *W*. If it was, $\mathscr{v}$ will be incremented by one unit. Otherwise, the pair (R_z, C_c) will be introduced in W with $\mathscr{v} = 1$.

Each time that a new triple is introduced in *W*, a process of window updating is performed. The oldest triple $\{\mathrm{R}_z, \mathrm{C}_c, \mathscr{v}\}^*$ in W is checked. If its $\mathscr{v}$ value is higher than $\mathscr{p}$, the consequent of R_z is updated for C_c in *S'*[3]. Otherwise, if the $\mathscr{v}$ value is lower, no updating will be performed and the error will be forgotten. This means that the system delays adapting its learnt rules until the error has reoccurred several times. This process is *Process 4.b* in Fig. 4 and it is called "learning inertia" [22]. It prevents the system from adapting its rules in response to "one off" user actions that do not reflect a marked change in the environment.

Step 2.2) Automatic learning. In the automatic learning the user does not notify anything to the system. If an input $x = \left(x_1, \dots, x_g\right) \in\ R^g$ is classified and the user does not notify an error in this classification, the system considers that the classification was right, if the classification was performed by a fired rule as well as if it was made by the similarity process. However, if the classification was made by the similarity process, new information can be obtained. Thus, the system will add a new rule in the rule base $S_{Success}$ when the input is undefined by the existing rules in the rule base, i.e., none of the existing rules was fired and the system classified with the similarity process.

Let the rule which was used to classify the input $x = \left(x_1, \dots, x_g\right)$ be called R_z (the most similar rule to the input):

$$Rule\ R_z\colon IF\ x_1\ is\ \ A_1^{(z)}\ and\ \dots and\ x_f\ is\ A_f^{(z)} \\ then\ Class\ C_z\ with\ CF_z \tag{16}$$

where $R_z\ \in S'$. Then, if the user does not notify an error for this classification, the system will create a new rule in the rule base $S_{Success}$. Following Step 1) of the offline learning for rule regeneration, a new rule is generated by $(x^{(t)}, C_z)$:

$$Rule\ R_d\colon IF\ x_1\ is\ A_1^{k^*}\ and\ \dots and\ x_f\ \ is\ A_g^{k^*} \\ then\ Class\ C_z\ with\ CF_d \tag{17}$$

where $d = (1, 2, \dots, E)$, E is the number of rules in $S_{Success}$ and d is the index of the rules. Furthermore, $R_d\ \in S_{Success}$, $A_j^{k^*}\left(x_j\ (t)\right)$ is the maximum membership value achieved at the data point, $k = 1, \dots, V$, $j = (1, \dots, g)$ and the consequent of the rule is the consequent of R_z. Finally, the confidence of the rules in $S_{Success}$ is recalculated as in (9) for each rule in S_{Sucess}. This process is *Process 4.c* in Fig. 4 and enables the system to learn situations not captured by the existing rules

[3] In our experiments $\mathscr{p} = 3$, because we consider that three notifications by the user means a real change. However, this value can be changed depending on how much relevance we want to give to the user.

and create new rules that reflect these conditions to be used in the following classifications.

Step 3) At the end of the day, Step 2) has added rules to the rule base of fails, S_{Fails} , and to the rule base of successes, S_{Sucess}. Besides, it could have modified the rule base *S'* by the sliding window *W* and the discovery of new APs. Hence, at the end of the day, this new information is updated in the original rule base *S*. All the rules in *S'*, S_{Fails} and S_{Sucess} are combined into *S* and the temporal structures S_{Fails}, $S_{Success}$ and *S'* are cleaned.

This process, *Process 5* in Fig. 4, extends the original rule base *S*. As the number of rules increases the required computation and memory requirements will rise and this could hinder the real time operation of the system. Hence, a rule capping mechanism is performed [22]. We have a limit on the rules that can be stored; this limit is related to the memory and processing capabilities of the computer which is using the system. When the number of rules exceeds this limit, we keep the most important rules for the system. The degree of importance of each rule is related to how frequently the rule is being used. This is determined by a frequency counter associated with each rule which is incremented every time the rule is fired.

With the incremental online learning described, the system adopts life-long learning, where it adapts its rules as the state of the environment changes over a significantly long period of time. As the previous steps described, the system is able to change both existing rules as adding new rules. This allows rules to continue operating even if there are changes in the APs or in the environment conditions. Besides, even if there is a situation in the environment which is not captured by the existing rules, the system will automatically create new rules that satisfy the current conditions. The system will, therefore, unobtrusively and incrementally extend the rule base to be adapted to the conditions of the environment while a zero cost and easy deployment of the infrastructure in the scenario is kept.

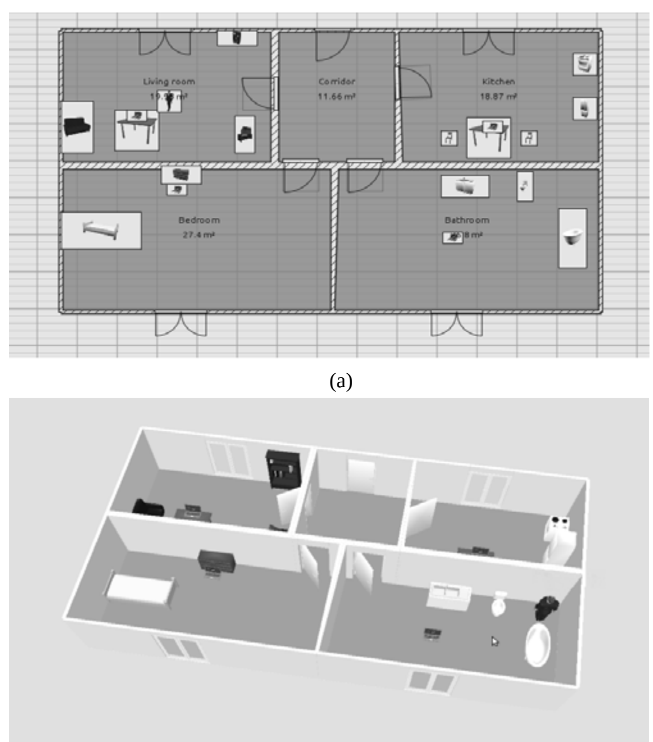
IV. Experiments and Results

In this section, we will present the experiments and results we have performed in order to validate the proposed system. The experiments were made in two phases, where in the first phase we have used a simulator of intelligent and ubiquitous environments. The use of a simulator in the first stages enables to validate the proposed system before the real world deployment in the second phase of the experiments. The following subsections describe both phases of experiments and show the feasibility of the presented approach.

A. Simulation Experiments

We have used the simulator UbikSim (shown in Fig. 5) [13] in our experiments. UbikSim is used as a means to test and validate the system from a programming perspective and also to demonstrate the effectiveness of the classification algorithm. The simulator includes a simulation model for the physical environment (i.e. indoor elements like walls, doors, furniture, WiFi APs, RFID sensors or any other type of sensor) and simulation models for the humans involved in the simulation (i.e. in this case, humans are the users of the system, equipped with a mobile phone which runs the location service inside and moves through the physical environment testing the location service). The simulator also includes the ubiquitous computing middleware software needed as an infrastructure to run services (e.g. the location service held at the mobile) and a context-awareness middleware, OCP [26].

Note that, in particular, in this simulation, the middleware software and the software implementing the location service are the only elements that would be used as such in production mode. UbikSim process provides an easy and low cost solution to test and validate the location service in an initial phase. One of the interesting tools which come with the simulator is a world editor based on a palette with different elements to recreate indoor environments (i.e. walls, furniture, doors) equipped with ubiquitous computing systems (i.e. sensors and actuators like RFID and bluetooth antennas and tags or badges, QR codes, etc), humans (i.e. elders, caregivers, workers in a office building, etc.). The world is edited in 2D (Fig. 5a)) and it is simultaneously seen in 3D (Fig. 5 b)). The simulator is an adapted version for ubiquitous computing of SweetHome3D editor. The simulation of UbikSim is based on the social simulation paradigm and the simulation engine is based on MASON social simulator [26].



(a)

(b)

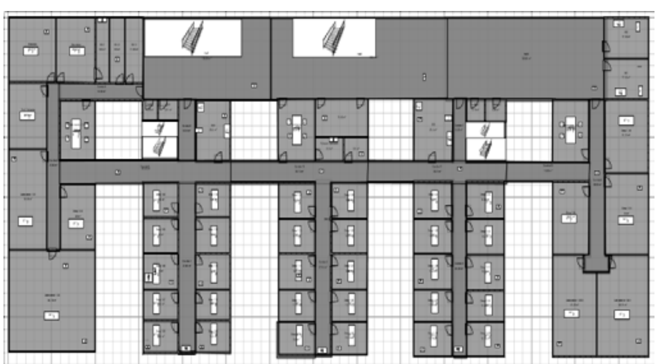
Fig. 5 (a) Floor 2D Home. (b) Floor 3D Home.

Two different scenarios have been simulated using UbikSim. The first one represents a typical house with one living room, one bedroom, one kitchen, one bathroom and a corridor which connects all the rooms. A 2D view of the plan of the house is shown in the Fig. 5 (a) and a 3D view is shown in Fig. 5 (b). In this scenario, an AP is located in each room and the corridor does not have any AP. Hence, there are four APs in this first scenario.

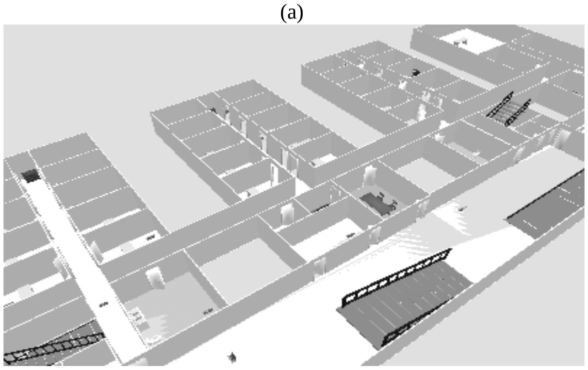
Fig. 6 (a) and Fig. 6 (b) show the second simulated scenario in a 2D and 3D view, respectively. This scenario represents an office block. The scenario contains several rooms, offices, toilets and facility rooms. In such a way, there are 73 places to predict in this scenario where only 41 APs are placed in 41 of these places.

While the first scenario is a small example which enables simple tests on the system and on the ability of learning of the algorithm, the second one is more complex. It allows testing the system at the learning level in an ideal world. In the next subsection, the tests over the real world will demonstrate the effectiveness in real settings.

A default user behaviour is set in both scenarios. The default behaviour implies that the user walks through all the rooms. He goes to the nearest unvisited room, then, randomly he walks inside the room during a random period of time and then he changes to another room. In order to simulate these paths, the simulator builds an abstract model based on an undirected graph which models the structure of the building [27]. In the graph, rooms, corridors and stairs are the nodes whereas doors and crossings are edges. Hence, the agent of the user is able to calculate the best path between a source and a destination by using the Dijkstra algorithm [21].



(a)



(b)

Fig. 6 (a) Floor 2D Building. (b) Floor 3D Building.

Finally, a model of the signal is set to each AP in order to simulate the signal propagation over the scenario. The used simulated signal propagation is a simplification of a real signal. There are many work which tried to simulate signals from different devices (Bluetooth, WiFi, Zigbee, etc.) taking into account features like the signal-loss, the propagation distance, signal interferences, deflection, and other complex radio features ([51], [28]). However, none of the models is able to represent the unpredictable and unexpected features which affect to the behaviour of the signal. With these premises, a simple general model of the signal is simulated in this work. The RSSI of the AP decreases by a factor randomly selected from an interval which depends on the distance between the AP and the receiver. Therefore, the RSSI is smaller when the distance between the AP and receiver is bigger. However, in order to achieve a more realistic model, a gaussian noise is added to the signal to simulate the variability of the strength of the signal. Furthermore, the RSSI is reduced by a degree value if there are walls between the receiver and the AP.

In the experiments, noise is also introduced to the signal model in order to check the performance of the location prediction in situations with different levels of noise. The noise is introduced by removing some values of the RSSI randomly to simulate signal-loss. According to the level of noise, the degree of randomness is fitted. Hence, the noise model is able to simulate in a simple way the signal-loss caused by different ambient factors. Notice that, even with the degrees of randomness introduced in the models of the signal and noise, a simulation is not able to completely reproduce a real behaviour of a signal. Nevertheless, the simulation of the signal and the noise are not the focus of this work and the described models are good enough to perform this phase of the experiments.

In the simulation the conditions of the environment do not change, meaning that the number of APs always remains the same. Furthermore, in this stage no online learning was needed.

With the described features, the simulations of both scenarios provide the needed data to train and test the offline system and compare the various techniques to evaluate its performance.

We have started by the experiments in the home simulated scenario where we have compared our proposal with two decision tree algorithms (J48 [44] and REPTree [50]) and with two decision rule algorithms (JRIP [18] and PART [49]), which use crisp values in the rules. These algorithms are chosen due to the fact that they are classical techniques where it has been proven that they show good results in this kind of problems. Furthermore, these algorithms share similar features with our approach, like the rules and a higher understability than other kind of techniques. Other techniques were tested in the experiments, as IBk (an implementation of the k-nearest neighbours classifier) [3], SMO (Sequential minimal optimization), support vector machines [12], Naïve Bayes, Bayesian Network, OneR [32], etc. However, J48, REPTree, JRIP and PART gave better results than the other techniques and hence we will show only their results in the paper due to the space constraints.

Table I summarises the results for the simulated home experiments where the table describes the percentages of success, errors, not classified instances, size of the model (number of rules in our approach and in the association rule algorithms and number of nodes in the decision tree algorithms) and finally, the time taken to build the model for each technique. In these experiments a 6% of noise was introduced.

TABLE I RESULTS OF SIMULATED HOME

Algorithm	Success	Errors	No classifies	Size	Time (ms to built model)
Fuzzy Based System	**81,21 %**	18,79 %	0 %	665	1,05
Fuzzy Based System without Similarity	79,96 %	15,04 %	5 %	665	1,03
J48	80,86 %	19,14 %	0 %	53	0.13
REPTree	79,71 %	20,29 %	0 %	43	0.15
JRIP	80,30 %	19,70 %	0 %	12	2.24
PART	80,56 %	19,44 %	0 %	27	2.04

The same experiments were performed over the simulated building and the results are described in Table II.

TABLE II RESULTS OF SIMULATED BUILDING

Algorithm	Success	Errors	No classifies	Size	Time (ms to built model)
Fuzzy Based System	**82,06 %**	17,94 %	0 %	1065	3.55
Fuzzy Based System without Similarity	77,30 %	15,92 %	6,78 %	1065	3.51
J48	81,71 %	18,29 %	0 %	661	1.33
REPTree	78,51 %	21,49 %	0 %	215	70.4
JRIP	77,07 %	22,93 %	0 %	419	37.99
PART	81,38 %	18,62 %	0 %	9185	23.98

As shown in the Table I and Table II, the best performances are obtained using the fuzzy based system (using the idea of similarity described in Section III.A.3), J48 and PART, while the other techniques show worse results and take longer times to build their models. A longer time to build the model is also taken by PART algorithm.

From the above results, it is clear that the use of similarity when the fuzzy based system is not able to classify some instances improves the results. In fact, the results show that the combination between fuzzy based system and the similarity provides a competitive algorithm which obtains as good results as the best techniques.

Among the best three algorithms, our proposed fuzzy system with similarity produces the highest success rate. However, although J48 and PART algorithms have as good performance as our approach, they are less understandable than our fuzzy logic based system.

In addition, in the case of PART, the time to build the model makes PART impossible to be used in real environments. Another disadvantage of both J48 and PART is that they are not able to learn and adapt online to changes of the environment. However, our proposed technique is able to perform an incremental online learning (see Section III.B) which provides adaptation to the environment. Without online learning and adaptation, the performance of the system gets worse over time and the system is not useful in a long term. Therefore, the other techniques are not able to deal with these requirements.

The above results show that our proposed technique is able to classify the location and obtain as good results as the other techniques while being build in a relatively fast time and maintaining the understandability of the generated model. In the next subsection, we will show the real world experiments.

B. *Real World Experiments*

In simulation there are behaviours which are taken for granted that are not always true, whereas in real world things are failing and behaving unexpectedly all the time. So, real world experiments were needed to validate the technique thoroughly.

In the real world there is no common way to collect WiFi RSSI. Even within a single Operating System there are different APIs and methods to do this, and none of them is standard. In this work we have used the Microsoft Native WiFi[4], which is widely used. However, this API gives little information. For example, there is no way to differentiate between a disappeared AP and an AP which has not been detected in a very specific scan hop, which could lead the system to think that the user is far away from the AP (out of the range) while the user is actually within the range. Among the information given by the API, the only information of interest is the following: service set identifier (SSID*)*, basic service set identifier (BSSID) and RSSI. In some cases the SSID is not unique to differentiate an AP (e.g. in routers). So the only way to differentiate a physical device from another is using the BSSID, which correspond with the MAC address and is unique. Finally, the RSSI is given in dB.

Given this information, collected data is defined as follows. Every time the system is launched, a new set *A* to contain the different APs and their RSSIs is created and initialised. Every time new information is retrieved from the system, a 'snapshot' of all the APs with their RSSI seen so far is recorded. Let the snapshot recorded at certain time *t* be called β_t:

$$\beta_t = \{(BSSID, RSSI)_1, \dots, (BSSID, RSSI)_\varpi\} \quad (18)$$

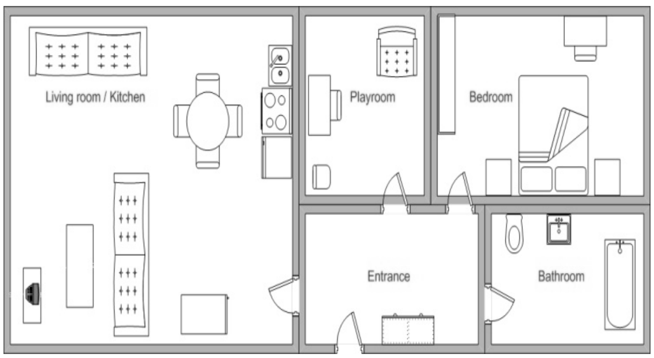
where *BSSID* is the Basic Service Set Identifier which identifies uniquely every AP, *RSSI* is the received signal strength indicator in dB and ϖ the number of APs in the set *A*. Thus, β_t represents the data arrived on the timestamp *t*. Furthermore, the function $\partial(\beta_t)$ is defined as $\partial(\beta_t) = \{(BSSID)_1, \dots, (BSSID)_\varpi\}$ where:

[4] http://msdn.microsoft.com/en-us/library/ms706556(v=vs.85).aspx

$$\partial(\beta_{t_1}) \subseteq \partial(\beta_{t_2}), \ \ t_1 < t_2 \qquad (19)$$

where $\partial(\beta_t)$ is the set of BSSIDs for a timestamp t. In this way, in every step new information is gathered. Since we cannot guarantee that a 'lost' AP is really lost, we record it but with RSSI equivalent to zero. Therefore, to the system, APs can completely disappear only when the application is re-launched, which is done once a day and there is no other way do differentiate between a lost AP and an AP which is running but the signal is too low to be discovered.

Using this information, in this second phase of the experiments, real tests over different scenarios of the real world were performed. In such way, the experiments have been done in three different scenarios, trying to cover the widest variety of scenarios and validate the system under different circumstances.



(a)

(b)

Fig. 7 (a) Map iSpace. (b) iSpace.

The first experiments were carried out in the iSpace which is an intelligent apartment in the University of Essex acting as a test bed for AIEs [22]. The map of the iSpace is shown in Fig. 7 (a) and the living room photo is shown in Fig. 7 (b). The iSpace has several APs deployed. As shown in Fig. 7 (a), the rooms are small and very close to each other, which leave little distance between each room. Furthermore, there are a lot of interferences where there are a big number of APs (ranging from 70 to 90 APs) working in the 2.4GHz frequency and there is also the UbiSense localisation system which causes more interference with the APs signals. Moreover, there are some metallic walls around the iSpace which cause signal reflection.

The second experiments were carried out in the rooms and corridors of an office block. At Fig. 8 (a) a plan of the corridors is shown and Fig. 8 (b) shows a picture of these corridors. As it is shown in the map, two floors are used. There are 22 rooms and 5 corridors between these two floors. Furthermore, in the scenario the number of APs ranges from 110 up to 170. There are some difficulties in predicting the location in this office block as all the rooms head in the same direction. Therefore, the RSSI difference from one room to the other is tiny, because all of them have a window heading outside the building and a door which leads to the corridor.

(a)

(b)

Fig. 8 (a) Map Office block. (b) Office block.

Finally, the last experiments were conducted in a small two storey house. The house is located in a residential area with no big buildings around, but just small houses like this one. The house plan is shown in Fig. 9 (a) and its photo is shown in Fig. 9 (b). The house is a rather small typical house, where not only the rooms are small and very close to each other and the number of APs smaller, but also there is this problem of the second floor, which tends to have similar RSSI to those in the first floor. In this case, the number of APs is lower than in the previous scenarios, ranging from 34 to 50 APs.

(a)

(b)

Fig. 9 (a) Map Home. (b) Home.

1) *Offline Experiments*

The first part of the real world experiments were conducted using the offline technique. In such way, the collected data for the experiments were collected in the same day. Validating the offline technique without the online learning allows determining if the technique is able to handle the features of the real world and the existing real noise. Furthermore, in this part, the offline technique is evaluated in the three described scenarios against other techniques. Therefore, the influence of the pre-processing of the data (see Section II.A) over the different techniques and scenarios is analysed.

Since the experiments are offline, the data set collected from the real world is split using 2/3 for training and the rest 1/3 for testing. Table III describes the obtained results. Like the simulated experiments, our proposal was compared with other traditional techniques, however J48 is the only technique which obtained good results in a reasonable time. From the results, it is possible to see how the pre-processing technique highly improves the results over all the scenarios and techniques. It is clear that the proposed technique gets a higher benefit from the pre-processing and it obtains the highest performance. Our technique outperforms the J48 in all the scenarios and as the level of available APs increase from the home to the iSpace to the Office Block, the margin between our system and J48 increases, which shows the power of our system in handling the high uncertainty levels associated with a big number of APs.

TABLE III OFFLINE RESULTS WITHOUT AND WITH PRE-PROCESSING

		% Success	
Technique	**Scenario**	**Raw data**	**Pre-processing data**
J48	**iSpace**	75.57 %	85.59 %
	Office block	66.83 %	74.45 %
	Home	81.17 %	87.47 %
Fuzzy Based System	**iSpace**	76.40 %	91.49 %
	Office block	75.12 %	92.42 %
	Home	81.29 %	87.92 %

The above results are obtained when the training and testing data is obtained from the data set of one day. However, experiments using a different data set from different days to train and test the system show that the results get worse down to 10% between each consecutive day. This worsening happens due to the fact that the offline technique can not take new APs different from the APs learnt from the training and it can not adapt to the changes of the environment. In fact, after one month the performance got only over 35% of success. Because of that, the online learning is needed in order to provide a good performance over a long period.

2) *Online Experiments*

We have conducted experiments over real world test beds to validate if the whole technique was able to learn over time and adapt to different APs and environmental changes. Fig. 10 shows the results from these experiments in the iSpace with and without post-processing. The experiments were carried out using different data sets from 8 different days obtained over a period of one month. After these 8 days the system gets stable and no more training data is needed.

A fixed offline learning data set (Day 0) is used for all the experiments, and different online learning data sets are formed from different days (from Day 1 to Day 5). All the results are tested using the same validation set (Day 6), except for the last one (Test 7), which is validated using Day 7. Day 1 data set is taken 19 days after Day 0 in order to check the performance after a period from the day of the offline learning (Day 0). Then, from Day 2 to Day 6, online learning is performed. Finally, Day 7 is taken two weeks after Day 6 to check the performance of the system after a long period without any update in the system. Between these days no modification is made in the system at all. The different Tests shown in axis X of Fig. 10 are described in Table IV.

TABLE IV TESTS ONLINE LEARNING

Tests	**Offline training data set**	**Online learning data set**	**Validation data set**
Test 1	Day 0	-	Day 6
Test 2	Day 0	Day 1	Day 6
Test 3	Day 0	Day 1 + Day 2	Day 6
Test 4	Day 0	Day 1 + Day 2 + Day 3	Day 6
Test 5	Day 0	Day 1 + Day 2 + Day 3 + Day 4	Day 6
Test 6	Day 0	Day 1 + Day 2 + Day 3 + Day 4 + Day 5	Day 6
Test 7	Day 0	Day 1 + Day 2 + Day 3 + Day 4 + Day 5	Day 7

Each test uses *Day 0* as offline training and incrementally adds one more online learning data set. *Day 6* is used as validation in all of them except in *Test 7* where *Day 7* is used. In these experiments, the aforementioned pre-processing and post-processing techniques are used. As shown in Fig. 10, using just *Day 0* as training the results after more than a month (using *Day 6* as validation) reach only a 34.91% of success. However, the system continues learning and improving its performance using online learning (from *Days 1* to *5*) until it gets 76.47% without post-processing and 81.01% with post-processing (*Test 6*).

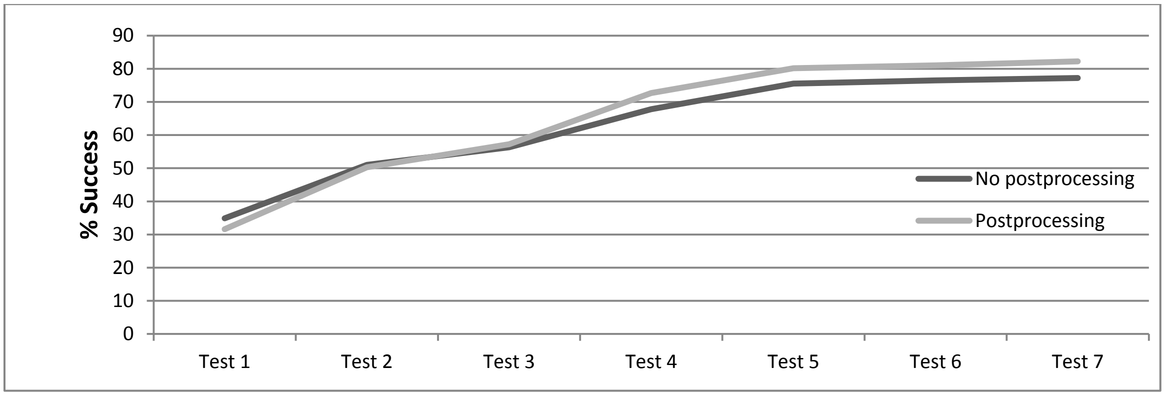


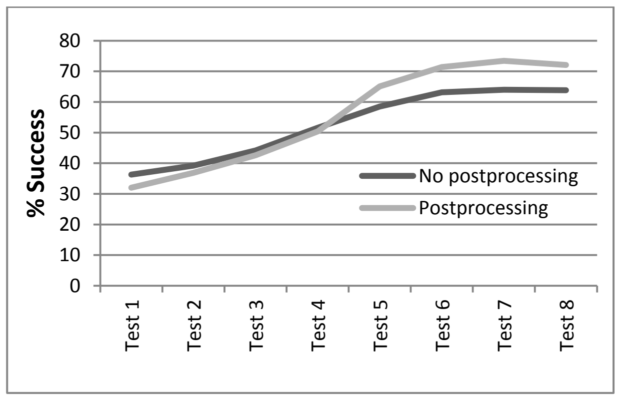
Fig. 10 Online learning – iSpace.

Notice that the post-processing is only improving the performance when the results are good enough, while the post-processing is degrading the results when the system is not working properly. This is due to the fact that the post-processing technique benefits the last previous predictions, in such way if the predictions are wrong, the next prediction will have more probability of being wrong.

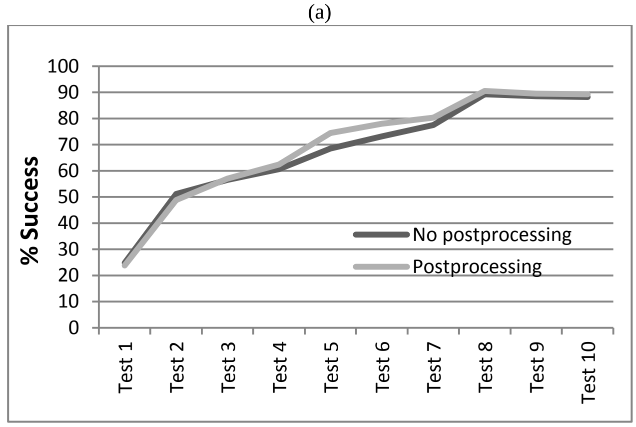
Finally, two weeks after these experiments (*Day 7*), a new test (i.e. *Test 7*) was made in order to check the performance after a long period of time, and the system reached an 82.22% and a 77.22%, with and without post-processing respectively. In fact, the system is able to stabilize, with a very good accuracy over an 80% of success, after only 5 days collecting new data as online learning. Therefore, it is clear that the system has learnt and adapted to medium-term changes over time and is able to provide a good performance in a long-term when more abrupt changes occur using online-learning.

Similar results were obtained in the other scenarios. Fig. 11 shows the results for the office block and the home scenarios. In both scenarios, as the rooms are so close, this makes the system getting confused between close rooms. However, despite these structures, the results are still pretty good and the system shows its ability to learn online and adapt to the changes in the different environments.

Hence, real world experiments reinforce the strength of the technique, which has shown to be valid in all the evaluated scenarios. Real world experiments have needed harder work and improvement of the technique (i.e. pre and post-processing or online learning) because of some difficulties such as big noise, disappearance of some APs or stationery behaviours (different RSSI in the morning/afternoon). However, the obtained results are very promising, obtaining a good success rate as well as a zero cost solution in real world situation.



(a)



(b)

Fig. 11 Online learning (a) Office block. (b) Home.

V. Conclusions and Future Work

In this paper, we presented a novel fuzzy online learning and adaptation localisation system for indoor environments which provides a zero cost solution. Our system is able to handle the short and long term uncertainties and noises from different indoor environments and operate over long durations of time in a lifelong learning mode to incrementally adapt and accommodate the uncertainties and environmental changes.

The proposed system uses WiFi RSSI of already deployed WiFi environments and the system only needs a few hours over a few days of training for a new scenario to be learnt with high accuracy and good resilience to changes. Furthermore, the proposed system provides an easy and cheap deployment which can be realized in any kind of indoor environment. In this paper has been shown that the system is able to obtain high success rates in three very different real buildings: a test bed (iSpace), a big office block and a typical residential house.

Responsive universal pre-processing and post-processing techniques have been successfully developed and applied to improve the overall results and deal with the instable and dynamic nature of WiFi RSSI.

We have presented a novel similarity process which is essential to make the system useful when dealing with the problem of having a high number of APs. Furthermore, the similarity process allows the system to continue classifying when the environmental conditions change, even if there are no rules corresponding to this new situation thus improving the results of the fuzzy based approach. We have also presented novel online learning and adaptation to provide a life-long localisation solution which, unobtrusively and incrementally, can extend the rule base to be adapted to environment conditions. The proposed solution can provide high accuracy at zero cost in real world living spaces.

In order to evaluate our proposed system, we have carried different experiments, over both simulated and real environments, testing the performance of our system with regard to other approaches and we have shown that our proposal outperforms the other techniques while being able to operate online in a long term mode to learn and adapt to changes of the environmental conditions.

We envisage that the proposed system will have a high impact for the realization of truly ubiquitous and intelligent environments. The high accuracy, low cost and easy deployment of the system will allow its use in very different scenarios of ambient intelligence, from living assistance systems for elderly people (e.g. notifying unusual situations like being a long time in a corridor, which could imply a fall) to marketing services for private companies (e.g. sending an offer alert when the user is near a commerce).

In the future, we intent to perform more advanced post-processing techniques which will provide a higher accuracy maintaining the zero cost and non-intrusive solution. These techniques will be based on the user movements through the indoor environments which provide useful information to the system with no prior knowledge. We also intent to extend the current system, which is based on a type-1 fuzzy systems to type-2 fuzzy systems. This extension provides a more appropriate framework for modelling and handling the short and long uncertainties arising in WiFi environments.

References

[1] G. Acampora and V. Loia. "Fuzzy control interoperability and scalability for adaptive domotic framework." IEEE Transactions on Industrial Informatics, vol. 1, no. 2, pp. 97–111, 2005.

[2] G. Acampora, V. Loia, and A. Vitiello. "Distributing emotional services in Ambient Intelligence through cognitive agents." Service Oriented Computing and Applications, vol. 5, no. 1, pp. 17–35, 2011.

[3] D. Aha. "Tolerating noisy, irrelevant, and novel attributes in instance-based learning algorithms." Int J Man-Machine Studies, Vol. 36, 267–287, 1992.

[4] U. Ahmad, A. Gavrilov, S. Lee, and Y. Lee, "Modular multilayer perceptron for WLAN based localization," in Neural Networks, 2006. IJCNN'06. International Joint Conference on. IEEE, Vancouver, BC, 2006, pp. 3465–3471.

[5] U. Ahmad, A. Gavrilov, Y. Lee, and S. Lee, "Context-aware, self-scaling Fuzzy ArtMap for received signal strength based location systems," Soft Computing-A Fusion of Foundations, Methodologies and Applications, vol. 12, no. 7, pp. 699–713, 2008.

[6] J. Alonso, M. Ocaña, M. Sotelo, L. Bergasa, and L. Magdalena, "WiFi localization system using fuzzy rule-based classification," Computer Aided Systems Theory-EUROCAST 2009, Las Palmas de Gran Canaria, Spain, pp. 383–390, 2009.

[7] J. Astrain, J. Villadangos, J. Garitagoitia, J. González de Mendívil, and V. Cholvi, "Fuzzy location and tracking on wireless networks," in Proceedings of the 4th ACM international workshop on Mobility management and wireless access. ACM, New York, NY, 2006, pp. 84–91.

[8] P. Bahl, and V. Padmanabhan, "RADAR: An in-building RF-based user location and tracking system," in INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies, Tel Aviv , Israel, 2000, pp. 775–784.

[9] J. Benedicto, S. Dinwiddy, G. Gatti, R. Lucas, and M. Lugert, "GALILEO: Satellite System Design," European Space Agency, 2000.

[10] J.Bezdek, Pattern recognition with fuzzy objective function algorithms. New York: Plenum, 1981.

[11] J. Botia, A. Villa, J. Palma, D. Perez, and E. Iborra, "Detecting domestic problems of elderly people: simple and unobstrusive sensors to generate the context of the attended," Distributed Computing, Artificial Intelligence, Bioinformatics, Soft Computing, and Ambient Assisted Living, Salamanca, Spain, pp. 819–826, 2009.

[12] C. Burges. "A tutorial on support vector machines for pattern recognition." Data Mining and Knowledge Discovery, vol. 2, no. 1, pp. 121-167, 1998.

[13] F. Campuzano, T. Garcia-Valverde, A. Garcia-Sola, and J. Botia. "Flexible simulation of ubiquitous computing environments." International Symposium of Ambient Intelligence, Salamanca, Spain, 2011, pp. 189–196.

[14] A. Carlotto, M. Parodi, C. Bonamico, F. Lavagetto, and M. Valla, "Proximity classification for mobile devices using wi-fi environment similarity," in Proceedings of the first ACM international workshop on Mobile entity localization and tracking in GPS-less environments. ACM, San Francisco, CA, 2008, pp. 43–48.

[15] C. Chen, J. Yang, G. Tseng, Y. Wu, and R. Hwang, "An Indoor Positioning Technique Based on Fuzzy Logic," Proceedings of the International MultiConference of Engineers and Computer Scientists, Hong Kong, 2010, vol. 2, pp. 854–857.

[16] K. Chintalapudi, A. Padmanabha Iyer, and V. Padmanabhan, "Indoor localization without the pain," in Proceedings of the sixteenth annual international conference on Mobile computing and networking. ACM, Chicago, Illinois, 2010, pp. 173–184.

[17] Cisco Borderless Mobility with CleanAir Walt Shaw. Product Line Manager, WNBU Marketing May 2010. [Online]. Available: http://www.cisco.com/web/DK/assets/docs/presentations/BN_Roadshow_CleanAir.pdf

[18] W. Cohen. "Fast effective rule induction" in Proceedings of the 12th International Conference on Machine Learning, Lake Tahoe City, California, pp. 115-123, 1995.

[19] R. Crane, "Prediction of attenuation by rain," IEEE Transactions on Communications, vol. COM-28, pp. 1727–1732, September 1980.

[20] T. Dasu and T. Johnson, "Exploratory data mining and data cleaning," John Wiley & Sons, pp. 29-30, 2003.

[21] E. Dijkstra. "A note on two problems in connexion with graphs." Numerische Mathematik, pp. 269–271, 1959.

[22] F. Doctor, H. Hagras, and V. Callaghan. “A fuzzy embedded agent-based approach for realizing ambient intelligence in intelligent inhabited environments.” IEEE Transactions on Systems, Man, and Cybernetics, Part A, vol. 35, no. 1, pp. 55–65, 2005.
[23] K. Ducatel, M. Bogdanowicz, F. Scapolo, and J.-C. Burgelman, “Scenarios for Ambient Intelligence in 2010, Tech. Rep.,” Information Soc.Technol., Advisory Group (ISTAG), Inst. Prospective Technol. Studies (IPTS), Seville, Spain, Feb. 2001.
[24] P. Enge, and P. Misra, “Special issue on global positioning system,” Proceedings of the IEEE, vol. 87, no. 1, pp. 3–15, 1999.
[25] Farpoint Group. “The Invisible Threat: Interference and Wireless LANs”. White Paper. Document FPG 2006-321.1. Oct, 2006.
[26] A. Garcia-Sola, T. Garcia-Valverde, and J. Botia. “Reasoning on a Semantic Web Based Context-Awareness Middleware” in Trends in Practical Applications of Agents and Multiagent Systems: 8th International Conference on Practical Applications of Agents and Multiagent Systems, Springer, Salamanca, Spain, 2010, pp. 147-153.
[27] T. Garcia-Valverde, A. Garcia-Sola, and J. Botia. Improving RFID’s location based services by means of Hidden Markov Models. In European Conference of Artificial Intelligence (ECAI’ 10), Lisbon, Portugal, 2010.
[28] J. Gorce, K. Jaffres-Runser, and G. de la Roche, “Deterministic approach for fast simulations of indoor radio wave propagation,” IEEE Transactions on Antennas and Propagation, vol. 55, no. 3, pp. 938–948, Mar. 2007.
[29] W. Griswold, R. Boyer, S. Brown, T. Truong, E. Bhasker, G. Jay, and R. Shapiro, “Activecampus-sustaining educational communities through mobile technology,” University of California, San Diego, Department of Computer Science and Engineering, Technical Report, 2002.
[30] A. Haeberlen, E. Flannery, A. Ladd, A. Rudys, D. Wallach, and L. Kavraki, “Practical robust localization over large-scale 802.11 wireless networks,” in Proceedings of the 10th annual international conference on Mobile computing and networking. ACM, Philadelphia, PA, 2004, pp. 70–84.
[31] T. He, C. Huang, B. Blum, J. Stankovic, and T. Abdelzaher. “Range-free localization schemes for large scale sensor networks.” in Proceedings of the 9th annual international conference on Mobile computing and networking. San Diego, California, 2003.
[32] R. Holte. “Very simple classification rules perform well on most commonly used datasets.” Machine Learning, vol. 11, pp. 63–91, 1993.
[33] H. Ishibuchi, T. Nakashima, and T. Morisawa, “Voting in fuzzy rule-based systems for pattern classification problems,” Fuzzy Sets and Systems, vol. 103, no. 2, pp. 223-238, April, 1999.
[34] H. Ishibuchi, and T. Yamamoto. “Rule weight specification in fuzzy rule-based classification systems”, IEEE Transactions on Fuzzy Systems, vol 13, no. 4, pp. 428-435, 2005.
[35] J. Krumm, and E. Horvitz, “Locadio: Inferring motion and location from wi-fi signal strengths,” in First Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services (Mobiquitous 2004). Boston, Massachusetts, 2004, pp. 4–13.
[36] A. Ladd, K. Berkis, A. Rudys, L. Kavraki, and D. Wallach, “Robotics-based location sensing using wireless Ethernet,” Wireless Networks, vol. 11, no. 1-2, pp. 189–204, 2005.
[37] J. LaMance, J. DeSalas, and J. Jarvinen, “INNOVATION-Assisted GPS: A Low Infrastructure Approach-Assisted GPS offers higher sensitivities and accuracies than stand-alone GPS and requires only low infrastructure costs. A reference network,” GPS World, vol. 13, no. 3, pp. 46–51, 2002.
[38] A. Lawrance, and P. Lewis. “An Exponential Moving Average sequence and point process, EMA(1)”. J. Appl. Prob. 14, pp. 98-113.
[39] H. Liu, H. Darabi, P. Banerjee, and J. Liu, “Survey of wireless indoor positioning techniques and systems,” IEEE Transactions on Systems, Man, and Cybernetics, Part C, vol. 37, no. 6, pp. 1067–1080, 2007.
[40] J. Mendel, “Fuzzy logic systems for engineering: A tutorial,” Proc. IEEE, vol. 83, pp. 345–377, Mar. 1995.
[41] D. Niculescu, and B. Nath, “Ad hoc positioning system (APS) using AOA,” in INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies, vol.3. IEEE, San Francisco, 2003, pp. 1734–1743.
[42] T. Ogawa, S. Yoshino, and M. Shimizu, “Location determination method for wireless systems based on learning vector quantization,” NTT Technical Review, vol. 1, no. 9, pp. 27–36, 2003.
[43] S. Polito, D. Biondo, A. Iera, M. Mattei, and A. Molinaro, “Performance evaluation of active RFID location systems based on RF power measures,” in Personal, Indoor and Mobile Radio Communications, 2007. PIMRC 2007. IEEE 18th International Symposium on. IEEE, Athens, Greece, 2007, pp. 1–5.
[44] R. Quinlan. “C4.5: Programs for Machine Learning”. Morgan Kaufman, San Mateo, 1993.
[45] W. Shu, M. Jungwon, and K. Byung. “Location Based Services for Mobiles: Technologies and Standards”. In IEEE International Conference on Communication (ICC), Beijing, China, 2008.
[46] S. Siddiqi, G. Sukhatme, and A. Howard, “Experiments in montecarlo localization using wifi signal strength,” in The 11th International conference on advanced robotics, Coimbra, Portugal, 2003.
[47] P. Steggles and S. Gschwind, “The Ubisense smart space platform,” in Adjunct Proceedings of the Third International Conference on Pervasive Computing, vol. 191, Munich, Germany, 2005, pp. 73–76.
[48] A. Teuber, B. Eissfeller, and T. Pany, “A two-stage fuzzy logic approach for wireless LAN indoor positioning,” in Proc. IEEE/ION Position, Location, and Navigation Symposium, San Diego, California, 2006, pp. 730–738.
[49] I. Witten, and E. Frank. “Generating accurate rule sets without global optimization” in Proceedings of the 15th International Conference on Machine Learning, Madison, Wisconsin, 1998, pp. 144-151.
[50] I. Witten, and E. Frank. “Data Mining – Practical Machine Learning Tools and Techniques”. Elsevier Inc., 2005.
[51] [51] G. Wolfle, P. Wertz, and F. Landstorfer, “Performance, accuracy and generalization capability of indoor propagation models in different types of buildings,” in IEEE Int. Symposium on Personal, Indoor, and Mobile Radio Communications, Osaka, Japan, Sept. 1999.
[52] B. Wu, C. Jen, and K. Chang, “Neural fuzzy based indoor localization by Kalman filtering with propagation channel modeling,” in Systems, Man and Cybernetics, 2007. ISIC. IEEE International Conference on. IEEE, Montreal, QC, Canada, 2007, pp. 812–817.
[53] M. Youssef and A. Agrawala, “The Horus WLAN location determination system,” in Proceedings of the 3rd international conference on Mobile systems, applications, and services. ACM, Seatle, WA, 2005, pp. 205–218.
[54] G. Zaruba, M. Huber, F. Kamangar, and I. Chlamtac, “Indoor location tracking using RSSI readings from a single Wi-Fi access point,” Wireless networks, vol. 13, no. 2, pp. 221–235, 2007.
[55] J. Zhao, Y. Zhang, and M. Ye, “Research on the received signal strength indication location algorithm for RFID system,” in Communications and Information Technologies, 2006. ISCIT’06. International Symposium on. IEEE, Bangkok, 2006, pp. 881–885.

Teresa Garcia-Valverde is a senior research officer in the Department of Information and Communications Engineering (DIIC) at the University of Murcia, Spain, where she had previously obtained her B.Sc. and M.Sc. degrees (Computer Science) in 2005 and 2007. She obtained her PhD in 2012 at the same university. She has worked on several projects in both academia and industry and she is a member of the IEEE Computational Intelligence Society (CIS). She develops her research activity in the Murcia University. Her main interests include ambient intelligence and intelligent data analysis. Teresa has co-authored several publications in workshops, conferences, journals and books.

Alberto Garcia-Sola received the B.Sc. degree in Computer Science from the School of Computer Science, the University of Murcia, Murcia, Spain, in 2007, and the M.Sc. degree in Computer Science from the Department of Computer Science, the University of Murcia, Murcia, Spain, in 2008. He is currently working toward the Ph.D. degree in ambient intelligent systems at the University of Murcia. His research interests include ambient intelligence, computational intelligence, fuzzy logic, indoor localisation, pervasive computing, and intelligent buildings.

Hani Hagras (M'03–SM'05) received the B.Sc. and M.Sc. degrees in electrical engineering from Alexandria University, Alexandria, Egypt, and the Ph.D. degree in computer science from the University of Essex, Colchester, U.K.

He is a Professor in the School of Computer Science and Electronic Engineering, Director of the Computational Intelligence Centre and the Head of the Fuzzy Systems Research Group in the University of Essex, UK. His major research interests are in computational intelligence, notably type-2 fuzzy systems, fuzzy logic, neural networks, genetic algorithms, and evolutionary computation. His research interests also include ambient intelligence, pervasive computing and intelligent buildings. He is also interested in embedded agents, robotics and intelligent control. He has authored more than 200 papers in international journals, conferences and books. He is a Fellow of the Institution of Engineering and Technology (IET (IEE)) and a Senior Member of the Institute of Electrical and Electronics Engineers (IEEE). He was the Chair of IEEE Computational Intelligence Society (CIS) Senior Members Sub-Committee. His research has won numerous prestigious international awards where most recently he was awarded by the IEEE Computational Intelligence Society (CIS), the 2013 Outstanding Paper Award in the IEEE Transactions on Fuzzy Systems and he was also awarded the 2006 Outstanding Paper Award in the IEEE Transactions on Fuzzy Systems. He is an Associate Editor of the IEEE Transactions on Fuzzy Systems. He is also an Associate Editor of the International Journal of Robotics and Automation, the Journal of Cognitive Computation and the Journal of Ambient Computing and Intelligence. He is a member of the IEEE Computational Intelligence Society (CIS) Fuzzy Systems Technical Committee. Prof. Hagras chaired several international conferences where he served as the General Co-Chair of the 2007 IEEE International Conference on Fuzzy systems London, July 2007 and he served as the Chair of 2009, 2011 and 2013 IEEE Symposium on Intelligent Agents, he has also served as the Chair of 2011 IEEE Symposium on Advanced to Type-2 Fuzzy Logic Systems, Paris 2011.

James Dooley is a senior research officer in the School of Computer Science and Electronic Engineering (CSEE) at the University of Essex, UK where he had previously obtained his BSc (Computer Science with Robotics and Intelligent Machines) in 2003 and his PhD in 2011. James is an honorary lecturer at King Abdulaziz University (Saudi Arabia) and has worked on several projects in both academia and industry - most notably at British Telecom and City University. As a member of the Intelligent Environments Group (IEG), he manages and develops several "living labs" for educational and research purposes whilst also developing ideas into projects and supervising undergraduate and postgraduate students. His research interests center around Robotics, Ubiquitous-Computing and Ambient-Intelligence – the essential ingredients for an envisioned world of "Intelligent Environments" where technology is embedded in the objects and spaces that surround us. James is a member of the IET and the IEEE, has co-organized conference workshops, co-edited several journal publications and has co-authored publications in workshops, conferences, journals and books.

Victor Callaghan is an Emeritus Professor of Computer Science, deputy-director of the Digital Lifestyles Centre and co-director of the Immersive Education group at Essex University. He established and directed mobile robotics research at Essex before founding the Intelligent Environment Group. These areas now host world-class teams and facilities including the Robot-Arena (a purpose build mobile robots lab), the iSpace (a full-size digital home), the iClassroom (a smart classroom) and the iCampus (a Living Lab based on an instrumented campus). Professor Callaghan holds a B.Eng and PhD in Electronics and Computing, respectively, from Sheffield University. In general terms, his main expertise concerns technology for the creation of intelligent environments involving research into affective intelligent agents, adjustable autonomy agents, quantum robotic controllers, end-user programming, fuzzy logic control, mixed reality and methods for ensuring stability of distributed coordinating agent systems. He has authored over 200 papers in international journals, conferences and has been principal investigator on numerous research projects attracting over 2 million pounds in funding and involving collaborations with researchers in many countries such as Germany, France, Greece, Mexico, USA, China and Taiwan. He is a founder and principal organiser of the international annual IEEE sponsored conference, Intelligent Environments (www.intenv.org - being program chair for the first 7 years), founder member of the iCampus conference and past program chair of Active Media Technology, Pervasive Computing & Applications and Ubiquitous Intelligence & Computing conferences He is a founder member of the Creative Science Foundation (www.creative-science.org) and principal organiser of the associated conferences plus a member of the editorial teams of the Intelligent Buildings International, Pervasive Computing & Communications, Ambient Intelligence and Smart Environments journals. He is also a founder member of the Faculty Cooperative initiative (www.FacultyCopperative.org) and CEO of the educational technology company FortiTo.

Juan A. Botia obtained his Ph.D. degree in Computer Science from the Universidad de Murcia in 2002. He received the B.Eng. in the same university in 1996. From 1996 to 1997 he was a Senior Engineer with Tecnatom S.A. From 1997 to 1999 he was an Associate Professor with the Universidad de Alcala, Madrid. Since 1999 he has been an Assistant Professor in the Universidad de Murcia. Juan A. has co-authored several publications in workshops, conferences, journals and books. His main research interests include software agents and multi-agent systems, intelligent data analysis and social simulation. Currently, he focuses the application of his research to ubiquitous computing and ambient intelligence.