User Movements Forecasting by Reservoir Computing Using Signal Streams Produced by Mote-Class Sensors

Claudio Gallicchio[1], Alessio Micheli[1], Paolo Barsocchi[2], and Stefano Chessa[1,2]

[1] Computer Science Department, University of Pisa,
Largo B. Pontecorvo 3, 56127 Pisa, Italy
[2] ISTI-CNR, Pisa Research Area, Via Morruzzi , 56124 Pisa, Italy

Abstract. Real-time, indoor user localization, although limited to the current user position, is of great practical importance in many Ambient Assisted Living (AAL) applications. Moreover, an accurate prediction of the user next position (even with a short advice) may open a number of new AAL applications that could timely provide the right services in the right place even before the user request them. However, the problem of forecasting the user position is complicated due to the intrinsic difficulty of localization in indoor environments, and to the fact that different paths of the user may intersect at a given point, but they may end in different places. We tackle with this problem by modeling the localization information stream obtained from a Wireless Sensor Network (WSN) using Recurrent Neural Networks implemented as efficient Echo State Networks (ESNs), within the Reservoir Computing paradigm. In particular, we have set up an experimental test-bed in which the WSN produces localization information of a user that moves along a number of different paths, and in which the ESN collects localization information to predict a future position of the user at some given mark points. Our results show that, with an appropriate configuration of the ESN, the system reaches a good accuracy of the prediction also with a small WSN, and that the accuracy scales well with the WSN size. Furthermore, the accuracy is also reasonably robust to variations in the deployment of the sensors. For these reasons our solution can be configured to meet the desired trade-off between cost and accuracy.

Keywords: Movement Forecasting, Sensor Stream Analysis, Received Signal Strength, Echo State Networks, Wireless Sensor Networks, Ambient Assisted Living.

1 Introduction

Wireless Sensor Networks (WSN) [7] are a recent development for unattended monitoring, which resulted particularly useful in many different application fields. In a typical deployment, a WSN is composed by a number of wireless sensors: small micro-systems that embed a radio transceiver and a set of transducers

J. Del Ser et al. (Eds.): MOBILIGHT 2011, LNICST 81, pp. 151–168, 2012.

suitable to monitor different environmental parameters. In many applications sensors are battery powered. One of the most promising application is Ambient Assisted Living (AAL) [12], an innovation funding program issued by the European Commission. AAL seeks for solutions integrating different technologies suitable for the improvement of the quality of life of elders and disabled in the environments where these people live (primarily in their houses) and work. In AAL spaces WSN play an important role as they are generally the primary source of context information about the user. For example WSN can monitor physiological parameter of the user, the environmental conditions and his/her movements and activities [34]. In most cases, raw data acquired by the WSN is given in input to software components that refine this information and that forecast the behavior or needs of the user in order to supply the user with appropriate services. In this paper we consider a scenario related to forecasting of user movements. In this scenario the user is localized in real time by a WSN (composed by low cost, low power sensors such as those of mote class [1]), and localization information is used to predict (with a short advance) whether the user will enter in a room or not, in order to timely supply the user with some services available in the room where the user is entering in. To this purpose the user wears a sensor, whose position is computed by a number of static sensors (also called *anchors*) deployed in the house. The sensor on the user and the environmental sensors exchange packets in order to compute the Received Signal Strength (RSS) for each packet, and use this information to evaluate the position of the user in real time. Although simple, this approach faces two main problems. The first is that indoor user localization is not sufficient by itself, since the current user position is not sufficient to predict the future behavior of the user. The second is that RSS measurements in indoor environments are rather noisy and this fact makes localization information imprecise. This latter problem is due both to multipath effects of indoor environments, and to the fact that the body of the user affects the radio signal propagation with irregular patterns, depending on the orientation of the user, orientation of the antenna etc. Overall, the considered scenario requires an approach which is adaptive, efficient and robust to the input noise. For this reasons, we take into consideration Machine Learning models in general, and neural networks in particular. More specifically, we exploit Recurrent Neural Networks (RNNs) [22], a class of dynamical neural network models that can work directly on the streams of RSS produced by the environmental sensors rather than on a localization information. The RNN takes into account also past measurements that reflect the history of previous movements, in order to overcome the fact that the current user position does not provide enough information. In particular, we consider the Reservoir Computing (RC) [26, 37] approach for modeling RNNs. Featured by extreme efficiency, RC models represent ideal candidates for approaching the problem in the considered scenario. Efficiency of the learning model used is in fact a critical factor, in particular in view of its deployment within the sensors themselves. In this work we present the results of a set of experiments in a real indoor environment aimed at producing a sufficiently large dataset to be used for the learning and

evaluation of the RC model, and we evaluate our solution in terms of predictive classification accuracy and cost. In particular, we evaluate the cost in terms of the number of anchors that are necessary to achieve the desired accuracy and in terms of independence from the actual deployment of the anchors (that we evaluate by comparing the accuracy of the predictions depending on the position of the anchors) which has a direct impact on deployment costs. In our experiments we show that our approach provides optimal accuracy with 4 anchors, but it can already provide a good accuracy even with a single anchor. Furthermore our approach scales with the number of anchors, hence it can be easily tuned in order to attain the desired trade-off between accuracy and cost of the solution.

2 Related Work

In the past years, many developed indoor positioning systems extract the location-dependent parameters such as time of arrival, time difference of arrival and angle of arrival [35] from the received radio signal transmitted by the mobile station. Such a measurement needs to be estimated accurately and it requires line of sight (LOS) between the transmitter and the receiver. Furthermore, it requires specialized and expensive hardware integrated into the existing equipments. Due to the high implementation cost, the indoor positioning system based on the use of RSS thus gets more and more interests. Since the deployments of WLAN infrastructures are widespread and the RSS sensor function is available in every 802.11 interface, the RSS-based positioning system is obviously a more cost-effective solution.

The model-based positioning approach is one of the most widely used technology seen in the literature since it expresses the radio frequency signal attenuation using a path loss model [8, 9]. From an observed RSS, these methods triangulate the person based on a distance calculation from multiple access points. However, the relationship between position and RSS relationship is highly complex due to multipath, metal reflection, and interference noise. Thus, the RSS propagation may not be adequately captured by a fixed invariant model. In contrast to model-based positioning, fingerprinting based RSS approaches are used [6, 23, 31, 39]. Fingerprints are generated during an offline training phase, where RSS data is collected at a set of marked training locations. The most challenging aspect of the fingerprinting based method is to formulate a distance calculation that can measure similarity between the observed RSS and the known RSS fingerprints. Various Machine Learning techniques can be applied to the location estimation problem [21]. Probabilistic method [28], k-nearest-neighbor [6], neural networks [29], and Support Vector Machines [23] are exploited in popular positioning techniques based on the location fingerprinting. Euclidean distance based calculation has been used in [20] to measure the minimum distance between the observed RSS and the mean of the fingerprints collected at each training point. RADAR [6] uses a k-nearest-neighbors method in order to find the closest match between fingerprints and RSS observation. Recently, research efforts have concentrated on developing a better distance measure that can take

into account the variability of the RSS training vectors. These methods estimate probability density for the training RSS and then compute likelihood/a posteriori estimates during the tracking phase using the observed RSS and the estimated densities [39]. User localization is then performed using a maximum-likelihood (ML) or maximum a posteriori (MAP) estimate of position. All these location determination methods do not solve the problem to forecast the user behaviors leveraging on empirical RSS measures. The Machine Learning approach can take advantage of training RSS data to capture characteristics of interest of their unknown underlying probability distribution. In this paper we consider the Echo State Network (ESN) [17, 18] model within the RC paradigm for modeling of RNNs. ESNs are dynamical neural networks used for sequence processing tasks. One of the main characteristics of ESNs is the efficiency of the approach. Learning is indeed restricted to a simple linear output tool, while the dynamical part of the network (the *reservoir*) is left untrained after initialization. The contractive reservoir dynamics provides a fading memory of past inputs, allowing the network to intrinsically discriminate among different input histories [17] in a suffix-based fashion [13, 14, 36], even in the absence of training. Despite the extreme efficiency of the approach, ESNs have been successfully applied to many common tasks in the area of sequence processing, often outperforming other state-of-the-art learning models for sequence domains (e.g. [17, 18]). In particular, in the last years ESN models have shown good potentialities in a range of tasks related to autonomous systems modeling. Examples of such tasks include event detection and localization in autonomous robot navigation [4, 5], multiple robot behavior modeling and switching [3, 38], robot behavior acquisition [16] and robot control [30]. However, such applications are mostly focused on modeling robot behaviors and often use artificial data obtained by simulators (e.g. [3–5, 38]). In this paper we apply the ESN approach to a real-world scenario for user indoor movements forecasting, characterized by real and noisy RSS input data, paving the way for potential applications in the field of AAL.

3 Experimental Setup

We carried out a measurement campaign on the first floor of the the ISTI institute of CNR in the Pisa Research Area, in Italy. The environment is composed of 2 rooms (namely Room 1 and Room 2), which are typical office environments with overall dimensions of approximately 12 m by 5 m divided by an hallway. The rooms contain typical office furniture: desks, chairs, cabinets, monitors that are asymmetrically arranged. This is a harsh environment for wireless communications because of multi-path reflections due to walls and interference due to electronic devices. For the experiments we used a sensor network of 5 IRIS nodes [1] (4 sensors, in the following *anchors*, and one sensor placed on the user, hereafter *mobile*), embedding a Chipcon AT86RF230 radio subsystem that implements the IEEE 802.15.4 standard. The experiments consisted in a set of measures between anchors and mobile. Figure 1 shows the anchors deployed in the environment as well as the movements of the user. The height of the anchors

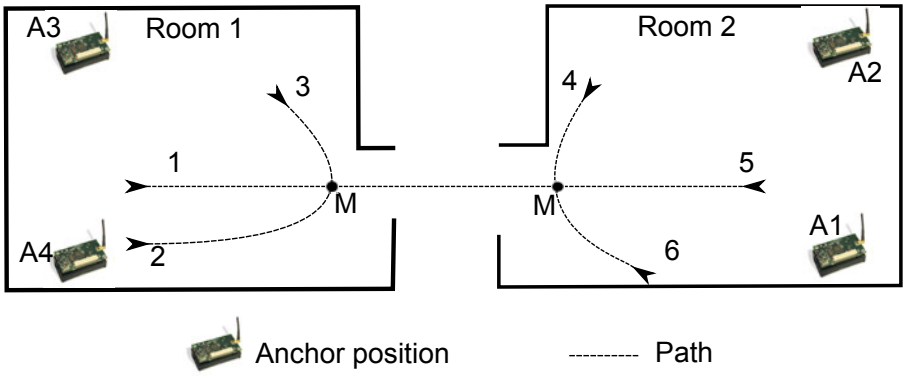

Fig. 1. Test-bed environment where the measurements have been done. The positions of the anchors, and the 6 user movements are shown.

was 1.5 m from the ground and the mobile was worn on the chest. The measurements were carried out in empty rooms to facilitate a constant speed of the user of about 1 m/s. Each measure collected about 200 RSS samples (integer values ranging from 0 to 100), where every sample was obtained by sending a beacon packet from the anchors to the mobile at regular intervals, 10 times per second, using the full transmission power of the IRIS. During the measures the user performs two types of movements: straight and curved, for a total of 6 paths (2 of which straight) that are shown in Fig. 1 with arrows numbered from 1 to 6. The straight movement runs from Room 1 to Room 2 or viceversa (paths 1 and 5 in Fig. 1) for 50 times in total. The curved movement is executed 25 times in Room 1 and 25 times in Room 2 (paths 2, 3, 4 and 6 in Fig. 1). Each path produces a trace of RSS measurements that begins from the corresponding arrow and that is marked when the user reaches a point (denoted with M in Fig. 1) located at 60 cm from the door. Overall, the experiment produced about 5000 RSS samples from each of the 4 anchors. The marker M is the same for all the movements, therefore the different path can not be only distinguished from the RSS values collected in M. The scenario and the collected RSS measures described so far can naturally lead to the definition of a binary classification task on time series for movements forecasting. The RSS samples from the four anchors are organized in 100 input sequences, corresponding to the conducted measures until the marker (M) is reached. The RSS traces can be freely downloaded in [2]. Each single trace is stored in a separate file that contains one row for each RSS measurement. Each row has 4 columns corresponding to: anchor ID, sequence number of the beacon packet, RSS value, and the boolean marker (1 if that measurement is done in point M, 0 otherwise). The resulting input sequences have length varying between 16 and 101. A target classification label is then associated to each input sequence, namely $+1$ for entering movements (paths 1 and 5 in Fig. 1) and -1 for non-entering ones (paths $2, 3, 4$ and 6 in Fig. 1). The constructed dataset is therefore balanced, i.e. the number of sequences with positive classification is equal to the number of sequences with negative classification.

3.1 Slow Fading Analysis

The wireless channel is affected by multipath fading that causes fluctuations in the receiver signals amplitude and phase. The sum of the signals can be constructive or destructive. This phenomenon, together with the shadowing effect, may strongly limit the performance of wireless communication systems and makes the RSS values unstable. Most of the recent research works in wireless sensor networks, modeled wireless channel with Rayleigh fading channel model [11,25], which is suitable channel model for wireless communications in urban areas where dense and large buildings act as rich scatterers. In indoor environments Nakagami or Ricean fading channel model works well, because it contains both non-LOS and LOS components. But Nakagami-m distribution function, proposed by Nakagami [27], is a more versatile statistical representation that can model a variety of fading scenarios including those modeled by Rayleigh and one-sided Gaussian distributions. Furthermore, in [33] the authors demonstrated that Nakagami-m distribution is more flexible and fits more accurately with experimental data for many propagation channels than the other distributions. We observe that the received signal

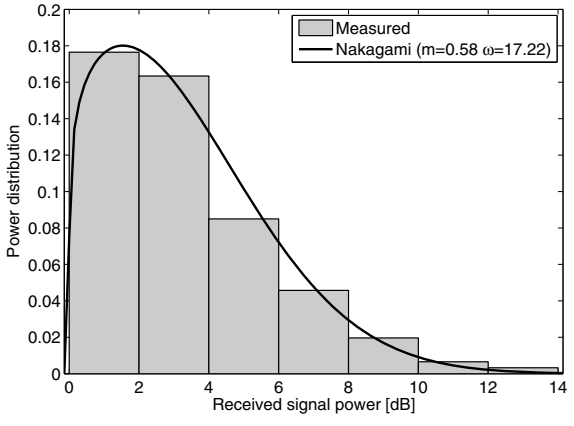

Fig. 2. Distribution of received power level from the anchor A4 and Nakagami-m distribution with $\mu = 0.58$ and $\omega = 17.22$

envelope is modulated by a slow fading process that produce an oscillation of RSS with respect its mean. This is due to multipath effects caused by scattering of the radio waves on office furniture. In order to verify this hypothesis, we look for the signature of multipath fading, by considering the distribution of the received power. The fading distribution approximates a Nakagami-m distribution around the mean received power. The Nakagami-m distribution has two parameters: a shape parameter m and a controlling spread ω. ω and m, lie in the range from 17 to 50 and from 0.5 to 0.8 for most of the measurements, respectively. The distribution observed on measured data (Fig. 2 shows an example for the same measurement

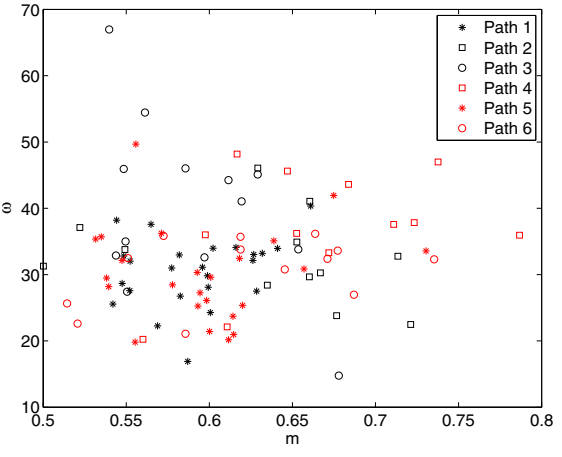

Fig. 3. Nakagami parameters of the different user paths

as above and for the anchor A4) are consistent with what is observed in [33]. As far as the dependence of the fading statistics on measurement parameters is concerned, we observe that the power spectrum is similar for all the measurements. In fact, m and ω are similar for all the measurements. For the arc movements, the Nakagami parameters are more widely spread, as shown in Fig. 3 with respect to the straight ones. As highlighted in Fig. 3 all the paths produce similar RSS traces, making it hard to forecasting the user behavior. Despite the similar RSS distributions, in Section 5 we will show that the proposed system is able to forecast the user behavior. It is also interesting to note that the traces collected in Room 1 can be modeled with m parameter values more close together with respect to the traces collected in Room 2. Consequently, we expect that the proposed system will mis-classify these paths more frequently.

4 Model

An ESN is a RNN with an input layer of N_U units, a large and sparsely connected hidden *reservoir* layer of N_R recurrent non-linear units and a *readout* layer of N_Y feed-forward linear units (see Fig. 4).

The untrained reservoir acts as a *fixed* non-linear temporal expansion function, implementing an encoding process of the input sequence into a state space where the trained linear readout is applied. More formally, given an input sequence $\mathbf{s} = [\mathbf{u}(1), \ldots, \mathbf{u}(n)]$ over the input space $\mathbb{R}^{N_U}$, at each time step $t = 1, \ldots, n$ the reservoir computes the following state transition function:

$$\mathbf{x}(t) = f(\mathbf{W}_{in}\mathbf{u}(t) + \hat{\mathbf{W}}\mathbf{x}(t-1)) \tag{1}$$

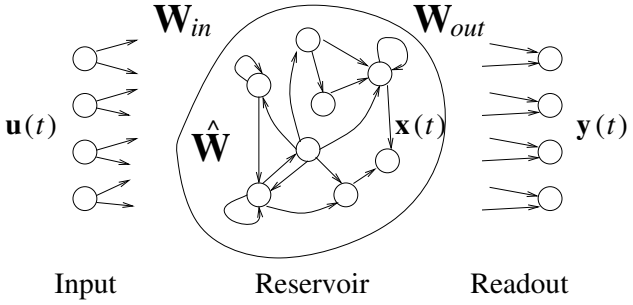

Fig. 4. The architecture of an ESN

where $\mathbf{x}(t) \in \mathbb{R}^{N_R}$ denotes the reservoir state (i.e. the output of the reservoir units) at time step t, $\mathbf{W}_{in} \in \mathbb{R}^{N_R \times N_U}$ is the input-to-reservoir weight matrix (possibly including a bias term), $\hat{\mathbf{W}} \in \mathbb{R}^{N_R \times N_R}$ is the (sparse) recurrent reservoir weight matrix and f is the component-wise applied activation function of the reservoir units (we use $f \equiv tanh$). A null initial state is used, i.e. $\mathbf{x}(0) = \mathbf{0} \in \mathbb{R}^{N_R}$. For the case of sequence binary classification tasks, which is of interest for this paper, the linear readout is applied only after the encoding process computed by the reservoir is terminated, according to the equation:

$$\mathbf{y}(\mathbf{s}) = sgn(\mathbf{W}_{out}\mathbf{x}(n)) \tag{2}$$

where sgn is a sign threshold function returning $+1$ for non-negative arguments and -1 otherwise, $\mathbf{y}(\mathbf{s}) \in \{-1, +1\}$ is the output classification computed for the input sequence $\mathbf{s}$ and $\mathbf{W}_{out} \in \mathbb{R}^{N_Y \times N_R}$ is the reservoir-to-output weight matrix (possibly including a bias term).

In this paper we also consider leaky integrator ESNs (LI-ESNs) [19], in which leaky integrator reservoir units are used. In this case, the state transition function of equation 1 is replaced by the following:

$$\mathbf{x}(t) = (1 - a)\mathbf{x}(t - 1) + af(\mathbf{W}_{in}\mathbf{u}(t) + \hat{\mathbf{W}}\mathbf{x}(t - 1)) \tag{3}$$

where $a \in [0, 1]$ is a *leaking rate* parameter, which is used to control the speed of the reservoir dynamics, with small values of a resulting in reservoirs that react slowly to the input [19, 26]. Compared to the standard ESN model, LI-ESN applies an exponential moving average to the state values produced by the reservoir units (i.e. $\mathbf{x}(t)$), resulting in a low-pass filter of the reservoir activations that allows the network to better handle input signals that change slowly with respect to the sampling frequency [5, 26]. LI-ESN state dynamics are therefore more suitable for representing the history of input signals. Note that for $a = 1$ equation 3 reduces to equation 1 and standard ESNs are obtained. In the following, we thereby use equation 3 to refer the reservoir computation for both LI-ESN and ESN (with $a = 1$).

The reservoir is initialized to satisfy the so called *Echo State Property* (ESP) [17]. The ESP simply states that the reservoir state of an ESN driven by a long

input sequence does only depend on the input sequence itself. Dependencies on the initial states are progressively forgotten after an initial *transient* (the reservoir provides an echo of the input signal). A sufficient and a necessary condition for the reservoir initialization are given in [17]. Usually, only the necessary condition is used for reservoir initialization, whereas the sufficient condition is often too restrictive [10, 17]. The necessary condition for the ESP is that the system governing the reservoir dynamics of equation 3 is locally asymptotically stable around the zero state $\mathbf{0} \in \mathbb{R}^{N_R}$. Setting $\tilde{\mathbf{W}} = (1-a)\mathbf{I} + a\hat{\mathbf{W}}$, where a is the leaking rate parameter of equation 3, the necessary condition is satisfied whenever the following constraint holds:

$$\rho(\tilde{\mathbf{W}}) < 1 \tag{4}$$

where $\rho(\tilde{\mathbf{W}})$ is the *spectral radius* of $\tilde{\mathbf{W}}$. In the following, for the ease of notation, we simply use ρ to refer the spectral radius of matrix $\tilde{\mathbf{W}}$. Matrices $\mathbf{W}_{in}$ and $\hat{\mathbf{W}}$ are therefore randomly initialized from a uniform distribution, and then $\hat{\mathbf{W}}$ is scaled such that equation 4 holds. Values of ρ close to 1 are commonly used in practice, leading to reservoir dynamics close to the edge of chaos [24], often resulting in the best performance in applications (e.g. [17]).

For sequence classification tasks, each training sequence is presented to the reservoir a number of $N_{transient}$ consecutive times, to account for the initial transient. The final reservoir states corresponding to the training sequences are collected in the columns of matrix $\mathbf{X}$, while the vector $\mathbf{y}_{target}$ contains the corresponding target classifications. The linear readout is therefore trained to solve the least squares linear regression problem

$$\min \|\mathbf{W}_{out}\mathbf{X} - \mathbf{y}_{target}\|_2^2 \tag{5}$$

Usually, Moore-Penrose pseudo-inversion of matrix $\mathbf{X}$ or ridge regression are used to train the readout [26].

The most striking feature of ESNs is efficiency. Indeed, training is restricted only to a linear output part and is very efficient, whereas the dynamic part of the network is fixed and the cost of its encoding procedure scales linearly with the length of the input for both training and test. In this regard, the ESN approach compares extremely well with competitive state-of-the-art learning models for sequence domains, including RNNs (in which the dynamic recurrent part is trained, e.g. [22]), Hidden Markov Models (with the additional cost for the inference also at test time, e.g. [32]) and Kernel Methods for sequences (whose cost can scale quadratically or more with the length of the input, e.g. [15]).

5 Computational Experiments

5.1 Experimental Settings

Accordingly to the classification task defined in Section 3, the t-th element $\mathbf{u}(t)$ of an input sequence $\mathbf{s}$ consists in the t-th set of RSS samples from the different

anchors considered in the corresponding measure, rescaled into the real interval $[-1, 1]$. Each input sequence was presented $N_{transient} = 3$ times to account for the initial transient. For our purposes, we considered different experimental settings in which the RSS from one or more of the four anchors is non-available. Therefore, the dimension of each $\mathbf{u}(t)$ can vary from 1 to 4 depending on the setting considered, i.e. on the number $N_{anchors}$ of anchors used.

In our experiments, we used reservoirs with $N_R = 500$ units and 10% of connectivity, spectral radius $\rho = 0.99$ and input weights scaled in the interval $[-1, 1]$. For LI-ESNs, we used the leaking rate $a = 0.1$. A number of 10 independent (random guessed) reservoirs was considered for each experiment (and the results presented are averaged over the 10 guesses). The performances of ESNs and LI-ESNs were evaluated by 5-fold cross validation, with stratification on the movement types, resulting in a test set of 20 sequences for each fold. For model selection, in each fold the training sequences were split into a training and a (33%) validation set. To train the readout, we considered both pseudo-inversion and ridge regression with regularization parameter $\lambda \in \{10^{-3}, 10^{-5}, 10^{-7}\}$. The readout regularization was chosen by model selection on the validation set.

5.2 Experimental Results

Number of Anchors. In this subsection we present the performance results obtained by ESNs and LI-ESNs corresponding to the different experimental settings considered, with a number of anchors $N_{anchors}$ varying from 1 to 4. For every value of $N_{anchors}$, the results are averaged (and standard deviations are computed) over the possible configurations of the anchors. The accuracies on the test set achieved by ESNs and LI-ESNs are graphically shown in Fig. 5. It is evident that the performances of both ESNs and LI-ESNs scale gracefully (and almost linearly) with the number of anchors used, i.e. with the cost of the WSN. The accuracy of ESNs varies from 0.53 (for $N_{anchors} = 1$) to 0.66 (for $N_{anchors} = 4$), whereas the accuracy of LI-ESNs varies from 0.81 (for $N_{anchors} = 1$) to 0.96 (for $N_{anchors} = 4$). Thus, the performance of the LI-ESN model is excellent for $N_{anchors} = 4$, scaling to acceptable values even for $N_{anchors} = 1$. In this regard it is also interesting that ESNs are consistently outperformed by LI-ESNs for every value of $N_{anchors}$. This result enlightens the better suitability of LI-ESNs for appropriately emphasizing the overall input history of the RSS signals considered with respect to the noise. The ROC plot in Fig. 6 provides a further graphical comparison of the test performances of ESNs and LI-ESNs.

Tables 1, 2, 3 and 4 detail the mean accuracy, sensitivity and specificity of ESNs and LI-ESNs, respectively, on the training and test sets, for increasing $N_{anchors}$. For both ESNs and LI-ESNs, sensitivity is slightly higher than specificity on the test set.

The nice scaling behavior of the performance with the decreasing number of anchors used, thus with the decreasing cost of the WSN, is also apparent from Tables 5 and 6, which provide the confusion matrices for ESNs and LI-ESNs, respectively, averaged over all the test set folds, with 20 sequences each (10 with positive target, 10 with negative target).

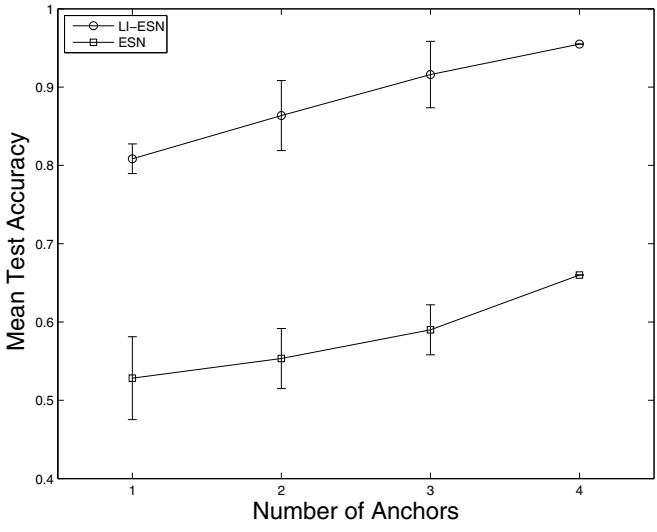

Fig. 5. Mean accuracy of ESNs and LI-ESNs on the test set, varying the number of anchors considered

Table 1. Mean training accuracy, sensitivity and specificity of ESNs, varying the number of anchors considered

$N_{anchors}$	Accuracy	Sensitivity	Specificity
1	0.96(±0.01)	0.96(±0.02)	0.97(±0.01)
2	1.00(±0.00)	1.00(±0.00)	1.00(±0.00)
3	1.00(±0.00)	1.00(±0.00)	1.00(±0.00)
4	1.00(±0.00)	1.00(±0.00)	1.00(±0.00)

Table 2. Mean test accuracy, sensitivity and specificity of ESNs, varying the number of anchors considered

$N_{anchors}$	Accuracy	Sensitivity	Specificity
1	0.53(±0.05)	0.53(±0.05)	0.53(±0.06)
2	0.55(±0.04)	0.55(±0.06)	0.56(±0.02)
3	0.59(±0.03)	0.59(±0.04)	0.58(±0.02)
4	0.66(±0.00)	0.69(±0.00)	0.63(±0.00)

The distribution of LI-ESN classification errors occurring in correspondence of each of the path types (see Fig. 1) is provided in Table 7, for the case of $N_{anchors} = 4$. Interestingly, the classification errors mainly occur for input sequences which correspond to movements in the Room 1, i.e. paths 1, 2 and 3 in Fig. 1. This actually confirms the coherence of the LI-ESN model with respect

Table 3. Mean training accuracy, sensitivity and specificity of LI-ESNs, varying the number of anchors considered

$N_{anchors}$	Accuracy	Sensitivity	Specificity
1	0.92(±0.02)	0.97(±0.01)	0.87(±0.04)
2	0.99(±0.01)	1.00(±0.00)	0.98(±0.02)
3	1.00(±0.00)	1.00(±0.00)	1.00(±0.00)
4	1.00(±0.00)	1.00(±0.00)	1.00(±0.00)

Table 4. Mean test accuracy, sensitivity and specificity of LI-ESNs, varying the number of anchors considered

$N_{anchors}$	Accuracy	Sensitivity	Specificity
1	0.81(±0.02)	0.86(±0.04)	0.76(±0.02)
2	0.86(±0.04)	0.88(±0.05)	0.85(±0.04)
3	0.92(±0.04)	0.93(±0.04)	0.90(±0.04)
4	0.96(±0.00)	0.98(±0.00)	0.93(±0.00)

Table 5. Averaged confusion matrix on the test set (with 10 positive samples and 10 negative samples for each fold) for ESNs, varying the number of anchors considered

$N_{anchors}$	True Positives	True Negatives	False Positives	False Negatives
1	5.29(±0.52)	5.28(±0.57)	4.73(±0.57)	4.71(±0.52)
2	5.46(±0.59)	5.60(±0.24)	4.40(±0.24)	4.54(±0.59)
3	5.90(±0.42)	5.84(±0.22)	4.17(±0.22)	4.10(±0.42)
4	6.90(±0.00)	6.30(±0.00)	3.70(±0.00)	3.10(±0.00)

Table 6. Averaged confusion matrix on the test set (with 10 positive samples and 10 negative samples for each fold) for LI-ESNs, varying the number of anchors considered

$N_{anchors}$	True Positives	True Negatives	False Positives	False Negatives
1	8.57(±0.40)	7.60(±0.15)	2.40(±0.15)	1.43(±0.40)
2	8.80(±0.52)	8.47(±0.43)	1.53(±0.43)	1.20(±0.52)
3	9.31(±0.45)	9.01(±0.40)	0.99(±0.40)	0.69(±0.45)
4	9.84(±0.00)	9.26(±0.00)	0.74(±0.00)	0.16(±0.00)

to the RSS input signals. Indeed (see Section 3), the movement paths in Room 1 are very similar and more hardly distinguishable among each other (in particular paths 1 and 2, see Fig. 1) than the path types in Room 2.

Actual Deployment of the Anchors. In this sub-section, we detail the performance results of ESNs and LI-ESNs for each possible configuration of the set of anchors used, with $N_{anchors}$ varying from 1 to 4. For each configuration

Table 7. Distribution of test errors for LI-ESNs in the case $N_{anchors} = 4$ (with a total test error of 4%) occurring for each of the path types in Fig. 1

Test Error (%)					
Path 1	Path 2	Path 3	Path 4	Path 5	Path 6
10.76%	51.86%	32.52%	1.43%	3.43%	0%

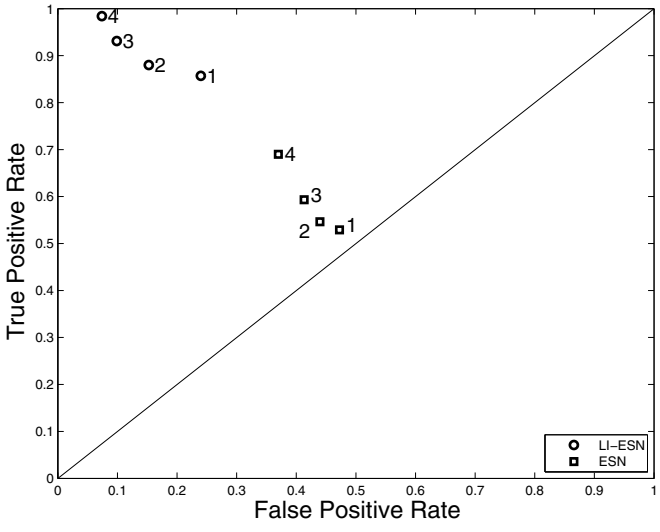

Fig. 6. ROC plot of ESNs and LI-ESNs on the test set, varying the number of anchors considered (indicated beside each point in the graph)

considered, the results are averaged (and the standard deviations are computed) over the 10 reservoir guesses. Tables 8 and 9 show the mean test accuracy, sensitivity and specificity for ESNs and LI-ESNs, respectively, in correspondence of every configuration of the anchors. Although the performances achieved in correspondence of the different choices for the same value of $N_{anchors}$ are quite similar, Tables 8 and 9 indicate that specific configurations can result in better performances. Despite the fact that different combination of anchors could give different performance was expected (since the disturbance and quality of signal is clearly affected by the position of the anchors in the environment), we observe from the tables that the accuracy of the prediction is reasonably robust to the position of the available anchors, which means that the deployment of the anchors does not need to be extremely accurate (thus reducing deployment costs).

On the other hand, the results show clearly that it is better to distribute the anchors as much as possible, e.g. in Table 9 the worse results are obtained when the available anchors are in the same room, while with two anchors displaced in different rooms the system already achieves accuracy in the range 87% - 93%.

Table 8. Mean test accuracy, sensitivity and specificity of ESNs for the possible configurations of the considered anchors

Anchors	Accuracy	Sensitivity	Specificity
A1	0.49(±0.06)	0.51(±0.10)	0.47(±0.09)
A2	0.47(±0.05)	0.45(±0.08)	0.48(±0.08)
A3	0.59(±0.06)	0.58(±0.10)	0.61(±0.07)
A4	0.57(±0.06)	0.57(±0.08)	0.56(±0.10)
A1, A2	0.52(±0.06)	0.47(±0.10)	0.56(±0.08)
A1, A3	0.51(±0.07)	0.50(±0.11)	0.52(±0.08)
A1, A4	0.58(±0.07)	0.60(±0.08)	0.57(±0.09)
A2, A3	0.61(±0.07)	0.62(±0.09)	0.60(±0.08)
A2, A4	0.53(±0.06)	0.50(±0.11)	0.55(±0.08)
A3, A4	0.58(±0.06)	0.60(±0.08)	0.56(±0.10)
A1, A2, A3	0.57(±0.07)	0.57(±0.10)	0.56(±0.10)
A1, A2, A4	0.58(±0.06)	0.57(±0.08)	0.59(±0.08)
A1, A3, A4	0.57(±0.06)	0.56(±0.09)	0.57(±0.08)
A2, A3, A4	0.64(±0.06)	0.66(±0.08)	0.62(±0.08)
A1, A2, A3, A4	0.66(±0.07)	0.69(±0.10)	0.63(±0.09)

Table 9. Mean test accuracy, sensitivity and specificity of LI-ESNs for the possible configurations of the considered anchors

Anchors	Accuracy	Sensitivity	Specificity
A1	0.84(±0.01)	0.90(±0.01)	0.77(±0.01)
A2	0.80(±0.03)	0.85(±0.06)	0.75(±0.03)
A3	0.81(±0.03)	0.88(±0.03)	0.74(±0.02)
A4	0.78(±0.03)	0.79(±0.06)	0.78(±0.03)
A1, A2	0.83(±0.03)	0.87(±0.06)	0.78(±0.02)
A1, A3	0.87(±0.03)	0.89(±0.04)	0.85(±0.05)
A1, A4	0.88(±0.02)	0.91(±0.02)	0.85(±0.02)
A2, A3	0.89(±0.02)	0.89(±0.03)	0.88(±0.03)
A2, A4	0.93(±0.01)	0.95(±0.02)	0.91(±0.02)
A3, A4	0.79(±0.03)	0.78(±0.04)	0.80(±0.05)
A1, A2, A3	0.87(±0.03)	0.89(±0.04)	0.86(±0.04)
A1, A2, A4	0.94(±0.03)	0.96(±0.04)	0.92(±0.03)
A1, A3, A4	0.88(±0.03)	0.89(±0.04)	0.87(±0.03)
A2, A3, A4	0.98(±0.02)	0.99(±0.01)	0.96(±0.03)
A1, A2, A3, A4	0.96(±0.03)	0.98(±0.03)	0.93(±0.03)

6 Conclusions

We have discussed the problem of forecasting the user movements in indoor environments. This problem cannot be tackled with by using mere user localization, since the information about the current position of the user does not

necessary provide an indication about his future position. In our approach we have combined localization information obtained by a wireless sensor network of MicaZ sensors with a RNN (implemented as an ESN) that takes in input a stream of RSS data produced by the sensors and signals when the user is about to enter in/exit from a given room. The problem is also made complex due to the intrinsic difficulty of localization in indoor environments, since presence of walls and objects disturb the radio propagation and makes RSS data imprecise. We have considered a scenario in which a user enters in and exits from two rooms according to different paths, which intersect in a marker point at the time in which the ESN is requested to make the prediction. We also have considered different numbers and combinations of anchors in order to investigate the trade-off between the cost of the WSN, the cost of deployment (that is dependent on the sensibility of the solution to the position of the anchors) and accuracy of prediction. The results confirmed the potentiality of our appproach. In particular we have observed that our solution obtains good precisions also with a single anchor, and it scales gracefully with the number of anchors (with 4 anchors it reaches a test accuracy of 96%). Furthermore it is reasonably robust to the position of the anchors, although the experiments gave a clear indication that the anchors should be placed as distant from each other as possible, in order to guarantee a better coverage. Concerning the ESN models considered, results in Section 5.2 have shown that LI-ESNs consistently lead to better performances than standard ESNs for every experimental setting (i.e. for varying the number and the deployment of the used anchors). The bias of in the LI-ESN model, acting as a low-pass filter of the reservoir states, has therefore revealed to be suitable for approaching the characteristics of the problem considered. LI-ESNs have indeed shown a good ability to appropriately represent the history of the noisy RSS input signals used in experiments. Standard ESNs, on the other hand, would need a larger dataset and less noise in the input signals in order to achieve better generalization performances. Moreover, the experimental results have enlightened the coherence of the learning models used with respect to the known difficulties of the problem. In fact, the great part of the classification errors of LI-ESNs (for the 4 anchors setting) has occurred in correspondence of movements in Room 1, where the possible path types are much more similar among each other than the corresponding paths in Room 2. Finally, we observe that the ESN-based solution is potentially suitable for its embedding in wireless sensors of the mote class, due to its efficiency and low requirements of memory and processing. This embedding is matter of ongoing work. Future work also include the investigation of trade-off between accuracy and energy spent by the sensors to produce localization information, and the extension of our approach to environments of different nature (e.g. public buildings, building with different composition of walls, room size etc.). As final remark, we stress that the dataset used in our experiments is openly available for download in our website [2].

Acknowledgments. This work is partially supported by the EU FP7 RUBICON project (contract n. 269914). The authors also wish to thank their colleague Alberto Gotta for his patience and help during the measurements.

References

1. Crossbow technology inc., `http://www.xbow.com`
2. Experimental dataset (February 2011), `http://wnlab.isti.cnr.it/paolo/index.php/dataset/forecasting`
3. Antonelo, E.A., Schrauwen, B., Stroobandt, D.: Modeling multiple autonomous robot behaviors and behavior switching with a single reservoir computing network. In: Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, pp. 1843–1848 (October 2008)
4. Antonelo, E.A., Schrauwen, B., Campenhout, J.M.V.: Generative modeling of autonomous robots and their environments using reservoir computing. Neural Processing Letters 26(3), 233–249 (2007)
5. Antonelo, E.A., Schrauwen, B., Stroobandt, D.: Event detection and localization for small mobile robots using reservoir computing. Neural Networks 21(6), 862–871 (2008)
6. Bahl, P., Padmanabhan, V.: Radar: an in-building rf-based user location and tracking system. In: Proceedings of the Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM 2000, vol. 2, pp. 775–784. IEEE (2000)
7. Baronti, P., Pillai, P., Chook, V.W.C., Chessa, S., Gotta, A., Hu, Y.F.: Wireless sensor networks: A survey on the state of the art and the 802.15.4 and zigbee standards. Comput. Commun. 30(7), 1655–1695 (2007)
8. Barsocchi, P., Lenzi, S., Chessa, S., Giunta, G.: A novel approach to indoor rssi localization by automatic calibration of the wireless propagation model. In: IEEE 69th Vehicular Technology Conference, VTC Spring 2009, pp. 1–5 (April 2009)
9. Barsocchi, P., Lenzi, S., Chessa, S., Giunta, G.: Virtual calibration for rssi-based indoor localization with ieee 802.15.4. In: IEEE International Conference on Communications, ICC 2009, pp. 1–5 (June 2009)
10. Buehner, M., Young, P.: A tighter bound for the echo state property. IEEE Transactions on Neural Networks 17(3), 820–824 (2006)
11. Cui, S., Goldsmith, A., Bahai, A.: Energy-efficiency of mimo and cooperative mimo techniques in sensor networks. IEEE Journal on Selected Areas in Communications 22(6), 1089–1098 (2004)
12. Ducatel, K., Bogdanowicz, M., Scapolo, F., Leijten, J., Burgelman, J.C.: Scenarios for Ambient Intelligence in 2010. Tech. rep., IST Advisory Group (February 2001)
13. Gallicchio, C., Micheli, A.: A markovian characterization of redundancy in echo state networks by PCA. In: Proceedings of the ESANN 2010, pp. 321–326. d-side (2010)
14. Gallicchio, C., Micheli, A.: Architectural and markovian factors of echo state networks. Neural Networks 24(5), 440–456 (2011)
15. Gärtner, T.: A survey of kernels for structured data. SIGKDD Explorations Newsletter 5, 49–58 (2003)
16. Hartland, C., Bredeche, N.: Using echo state networks for robot navigation behavior acquisition. In: IEEE International Conference on Robotics and Biomimetics (ROBIO 2007), pp. 201–206. IEEE Computer Society Press (2007)

17. Jaeger, H.: The "echo state" approach to analysing and training recurrent neural networks. Tech. rep., GMD - German National Research Institute for Computer Science (2001)
18. Jaeger, H., Haas, H.: Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication. Science 304(5667), 78–80 (2004)
19. Jaeger, H., Lukosevicius, M., Popovici, D., Siewert, U.: Optimization and applications of echo state networks with leaky- integrator neurons. Neural Networks 20(3), 335–352 (2007)
20. Kaemarungsi, K., Krishnamurthy, P.: Modeling of indoor positioning systems based on location fingerprinting. In: Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM 2004, vol. 2, pp. 1012–1022 (March 2004)
21. Kjærgaard, M.B.: A Taxonomy for Radio Location Fingerprinting. In: Hightower, J., Schiele, B., Strang, T. (eds.) LoCA 2007. LNCS, vol. 4718, pp. 139–156. Springer, Heidelberg (2007)
22. Kolen, J., Kremer, S. (eds.): A Field Guide to Dynamical Recurrent Networks. IEEE Press (2001)
23. Kushki, A., Plataniotis, K.N., Venetsanopoulos, A.N.: Kernel-based positioning in wireless local area networks. IEEE Transactions on Mobile Computing 6(6), 689–705 (2007)
24. Legenstein, R.A., Maass, W.: Edge of chaos and prediction of computational performance for neural circuit models. Neural Networks 20(3), 323–334 (2007)
25. Liu, W., Li, X., Chen, M.: Energy efficiency of mimo transmissions in wireless sensor networks with diversity and multiplexing gains. In: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2005), vol. 4, pp. iv/897–iv/900 (March 2005)
26. Lukosevicius, M., Jaeger, H.: Reservoir computing approaches to recurrent neural network training. Computer Science Review 3(3), 127–149 (2009)
27. Nakagmi, M.: The m-distribution. a general formula of intensity distribution of rapid fading. Statistical methods in radio wave propagation. Pergamon, Oxford (1960)
28. Madigan, D., Einahrawy, E., Martin, R., Ju, W.H., Krishnan, P., Krishnakumar, A.: Bayesian indoor positioning systems. In: INFOCOM 2005, vol. 2, pp. 1217–1227 (March 2005)
29. Martínez, E.A., Cruz, R., Favela, J.: Estimating User Location in a WLAN Using Backpropagation Neural Networks. In: Lemaître, C., Reyes, C.A., González, J.A. (eds.) IBERAMIA 2004. LNCS (LNAI), vol. 3315, pp. 737–746. Springer, Heidelberg (2004)
30. Oubbati, M., Kord, B., Palm, G.: Learning Robot-Environment Interaction Using Echo State Networks. In: Doncieux, S., Girard, B., Guillot, A., Hallam, J., Meyer, J.-A., Mouret, J.-B. (eds.) SAB 2010. LNCS, vol. 6226, pp. 501–510. Springer, Heidelberg (2010)
31. Pan, J.J., Kwok, J., Yang, Q., Chen, Y.: Multidimensional vector regression for accurate and low-cost location estimation in pervasive computing. IEEE Transactions on Knowledge and Data Engineering 18(9), 1181–1193 (2006)
32. Rabiner, L.R.: A tutorial on hidden markov models ans selected applications in speech recognition. Proceedings of the IEEE 77(2) (1989)
33. Rubio, L., Reig, J., Cardona, N.: Evaluation of nakagami fading behaviour based on measurements in urban scenarios. AEU - International Journal of Electronics and Communications 61(2), 135–138 (2007)

34. Rui, C., Yi-bin, H., Zhang-qin, H., Jian, H.: Modeling the ambient intelligence application system: Concept, software, data, and network. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews 39(3), 299–314 (2009)
35. Sun, G., Chen, J., Guo, W., Liu, K.: Signal processing techniques in network-aided positioning: a survey of state-of-the-art positioning designs. IEEE Signal Processing Magazine 22(4), 12–23 (2005)
36. Tiňo, P., Hammer, B., Bodén, M.: Markovian Bias of Neural-based Architectures with Feedback Connections. In: Hammer, B., Hitzler, P. (eds.) Perspectives of Neural-Symbolic Integration. SCI, vol. 77, pp. 95–133. Springer, Heidelberg (2007)
37. Verstraeten, D., Schrauwen, B., D'Haene, M., Stroobandt, D.: An experimental unification of reservoir computing methods. Neural Networks 20(3), 391–403 (2007)
38. Waegeman, T., Antonelo, E.A., Wyffels, F., Schrauwen, B.: Modular reservoir computing networks for imitation learning of multiple robot behaviors. In: Proceedings of the 8th IEEE International Symposium on Computational Intelligence in Robotics and Automation, pp. 27–32. IEEE (2009)
39. Youssef, M., Agrawala, A.: The horus wlan location determination system. In: MobiSys 2005: Proceedings of the 3rd International Conference on Mobile Systems, Applications, and Services, pp. 205–218. ACM, New York (2005)