Indoor Positioning based on RSSI of WiFi signals: how accurate can it be?

J. P. Grisales Campeón[†], S. López[*], S. R. de Jesús Meleán[*],
H. Moldovan[*], D. R. Parisi[†‡] and P. I. Fierens[†‡1]

[†]*Instituto Tecnológico de Buenos Aires*
Buenos Aires, Argentina
[*]*Urbix Technologies S.A.*
Buenos Aires, Argentina
[‡]*Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET)*
Buenos Aires, Argentina
[1]`pfierens@itba.edu.ar`

***Abstract*—There is abundant literature on positioning systems based on WiFi signals. Most of the systems with off-the-shelf hardware use RSSI measurements. In this paper, we try to determine the highest achievable accuracy of such systems. Instead of resorting to a theoretical analysis, e.g., Cramér-Rao or Ziv-Zakai bounds, we apply state-of-the-art localization algorithms to measurements in a well-controlled experimental setup. We conclude that an accuracy of less than 1 meter seems unrealistic.**

***Resumen* Hay una abundante literatura acerca de sistemas de posicionamiento usando señales de WiFi. La gran mayoría de los sistemas con hardware comercial no específico, utilizan mediciones de RSSI. En este artículo, intentamos determinar cuál es la máxima exactitud de este tipo de sistemas. En vez de realizar un análisis teórico, por ej., con cotas del estilo de Cramér-Rao o Ziv-Zakai, aplicamos el estado del arte en algoritmos de localización a mediciones propias realizadas en condiciones experimentales bien controladas. A partir de este trabajo, concluimos que es difícil obtener una exactitud menor a 1 metro.**

I. Introduction

Location systems based on Receive Signal Strength Indication (RSSI) have been around for, at least, 50 years [1], and those based on WiFi appeared almost as soon as the technology was in place 20 years ago. There is a vast literature on the subject (see, e.g., [2], [3] and references therein). Most common approaches are based on the Received Signal Strength Indicator RSSI, i.e., the received power, but there are also other alternatives such as the use of the Channel State Information [4]–[6] and Fine Time Measurement [7].

In this work, we focus on positioning systems based on simple RSSI measurements and we try to answer the question posed in the title: how accurate can they be? This question can be tackled from a theoretical point of view using tools such as the Cramér-Rao bound (see [8] and references therein). Another possibility is to try to answer the question experimentally, using state-of-the-art algorithms. Although there are many surveys on the subject, only few of them make a comparison of the state-of-art based on measurements on a single scenario.

Through a careful experimental setup in a quite ideal scenario, we present a comparison of several positioning algorithms based on RSSI of WiFi signals, offering a possible answer to the question of the achievable accuracy.

In Section II we make a quick review of some of the literature which can give us hint on the order of magnitude of the accuracy of localization systems based on RSSI values. Section III gives an overview of the algorithms that we use to estimate the achievable accuracy. Since it is unreasonable to try all algorithms proposed in the literature, we describe and use a few typical localization approaches. Section IV deals with the details of the experimental setup. Results are analyzed in Section V. We close the paper with some conclusions in Section VI.

II. Related work

A complete survey of the state-of-the-art is out of the scope of this paper. There are many surveys than can be consulted, e.g., [3], [9]–[13]. In the literature on the subject, localization errors are estimated through experiments in a diverse set of scenarios and using different algorithms. The diversity of scenarios and algorithms makes a fair comparison impossible and that is the main motivation of this paper. However, we can still get a hint on the order of magnitude of the errors that we might expect. For example, Ref. [3] presents mean localization errors between 0.6 and 10 m, depending on several variables such as the number of access points and the localization algorithm. Correa *et al.* [13] review several papers and records errors of the order of 1-3 m (either mean, median or third quartile).

There are several competitions for positioning algorithms. Some of them are hosted together with the International Conference on Indoor Positioning and Indoor Navigation (IPIN). We take the results of the off-site track of the EvAAL-ETRI Indoor Location Competition at IPIN 2015 [14] as reference values. Teams were given a training and a validation dataset with fingerprints from almost 30 mobile devices and 520 APs in almost 110.000 m^2 (three buildings with up to five floors). A test set was reserved to score the competitors. The best median error was ~ 4.5 m (a 4 m penalty was added for each wrong floor).

Let us review some of the limitations found when positioning using WiFi RSSI measurements. A simple propagation model is described by [15]

$$P_{\text{rx}} = P_{\text{tx}} - 10\gamma \log_{10}\left(\frac{d}{d_0}\right) + S, \quad (1)$$

where P_{tx} and P_{rx} are the transmitted and received power, respectively; γ is the path loss exponent; d is the distance between transmitter and receiver and d_0 is a reference distance. S represents variations due to shadowing caused by obstacles in the path between the transmitter and the receiver, for example, people moving around. It is usual to model S as a Gaussian random variable with zero mean and variance σ_S^2. The effect of shadowing on range estimation can be ameliorated by averaging several measurements. The Cramér-Rao lower bound (CRLB) for distance estimation under this model is given by [16]

$$\sqrt{\text{Var}\left(\hat{d}\right)} \geq \frac{\ln(10)}{10}\frac{\sigma_S}{\gamma}d. \quad (2)$$

Both γ and σ_S depend on several factors. To gain some insight on the order of magnitude of the lowest expected estimation error, let us fix $\gamma = 1.5$ and $\sigma_S = 4$ dB. These figures are similar to those in Ref. [17]. A fit of our own measurements gives $\gamma \approx 1$ and $\sigma_S \approx 4$ dB, although the distribution is clearly non-Gaussian and other factors (some of them explained in the following paragraphs) besides shadowing may have affected our results. We must note that even larger values of σ_S can be found in the literature (see, e.g., [18]). Fixing $d = 1$ m, we get a minimum ranging error $\sqrt{\text{Var}\left(\hat{d}\right)} \geq 0.6$ m. Although the error on the location estimate depends on other factors such as the positions of the access points (see, e.g., [8]), this figure gives us an idea of the order of the error we might expect. Accuracy can be improved by taking several independent measurements with the purpose of averaging out the random shadowing, but this solution leads to a longer positioning delay.

Eq. (1) does not include details such as multipath fading. Although we may approximate P_{rx} by RSSI, the latter is the result of adding up the energy from all paths and, hence, the relation between the received power and the distance cannot be represented a by single path loss exponent [19].

Even in absence of multipath fading, there might be measurement errors. Off-the-shelf wireless chipsets were not devised as calibrated instruments and only approximate RSSI measurements are needed for, e.g., AP selection. Systematic errors may vary from chipset to chipset and, therefore, localization of different devices becomes even more difficult. The authors of Ref. [20] found differences of up to 14 dBm for measurements made by identical models of WiFi cards. In order to gain some intuition on the localization errors that can be expected from measurement errors, we may assume that they are random. Let us further assume that errors are normally distributed with zero mean and variance σ_E^2. In this case, we can modify Eq. (1) to

$$\tilde{P}_{\text{rx}} = P_{\text{tx}} - 10\gamma \log_{10}\left(\frac{d}{d_0}\right) + S + E, \quad (3)$$

where $\tilde{P}_{\text{rx}}$ is the measured received power and E is the measurement error. Since S and E are clearly independent, the CRLB for ranging becomes

$$\sqrt{\text{Var}\left(\hat{d}\right)} \geq \frac{\ln(10)}{10}\frac{\sqrt{\sigma_S^2 + \sigma_E^2}}{\gamma}d. \quad (4)$$

Using the same values as before and letting $\sigma_E = 2.5$ dB, we get $\sqrt{\text{Var}\left(\hat{d}\right)} \geq 0.7$ m. Although we have made many assumptions to compute this value, it may give us an idea of the order of the magnitude of the ranging (and, hence, positioning) errors. If measurement errors are not random, but systematic, then they would not influence most of the localization algorithms we deal with in this work.

Another limitation of a RSSI-based localization approach is the fact that signal propagation conditions change with, for example, furniture re-arrangements or the density of people in the building. Indeed, these variations have been advantageously used to estimate the number of people [21]. We have tried to minimize the effect of these changes in our experimental setup. In particular, there was only one individual in the room and he was seated for most of the measurements.

III. Localization algorithms

From the literature, we have selected several localization methods, trying to include both traditional and newer approaches to the problem. Leaving out some algorithms may generate uncertainty on whether we have forgotten to include a relevant and outperforming approach. However, the fact that the algorithms that we use are diverse and perform similarly, together with their accuracy being comparable, in order of magnitude, to the minimum achievable error predicted by theoretical arguments (Section II), reassure us that a significant loss due to our choice is improbable.

All algorithms have at least two phases. First, a calibration phase takes place. Measurements are taken on R calibration or reference points (RPs). Let us assume that on each RP, the signal strength from A access points is measured N times. Then, let us call $u(n, r, a)$ the n-th measured signal strength ($n \in \{1, 2, \cdots, N\}$) from access point a ($\in \{1, 2, \cdots, A\}$) at reference point r ($\in \{1, 2, \cdots, R\}$). Whenever we deal with the case $N = 1$, we omit n. For each reference point r, we define vectors $\vec{u}_r^n, \vec{u}_r \in \mathbb{R}^A$ as

$$\vec{u}_r^n(a) = u(n, r, a),\ \vec{u}_r = \frac{1}{N}\sum_{n=1}^{N}\vec{u}_r^n,\ \bar{\vec{u}}(a) = \frac{1}{R}\sum_{r=1}^{R}\vec{u}_r(a). \quad (5)$$

The position of reference point r is given by $\vec{x}_r = (x_r, y_r)$ and that of access point a is $\vec{X}_a = (X_a, Y_a)$. Let us define $\mathbf{U}$ as the matrix with vectors $\vec{u}_r^n$ as rows and $\mathbf{X} \in \mathbb{R}^{RN\times 2}$ as the matrix with the corresponding RP's positions.

The second phase is the online or positioning phase. Let us assume that the signal strength from the A access points is measured M times and let us call $s(m, a)$ the m-th measured signal strength from access point a. If $M = 1$, we omit m. We define $\mathbf{S}$ as the matrix $\mathbf{S}(i, j) = s(i, j)$. We define $\vec{s} \in \mathbb{R}^A$ as the vector of average values, i.e.,

$$\vec{s}(a) = \frac{1}{M}\sum_{m=1}^{M}s(m, a). \quad (6)$$

A. Model-based algorithms

Although many localization approaches may be included under the model-based category, we reserve this term for those algorithms for which an experimentally-fit propagation model is its most important component. Based on the implementation in Ref. [22], for each access point (only three with known positions in our experiments), we fit a polynomial to the relation between distance and RSSI:

$$d = \alpha_0 + \alpha_1 \times \text{RSSI} + \alpha_2 \times \text{RSSI}^2 + \cdots . \tag{7}$$

The order of the polynomial is chosen through trial and error in order to get the best results. In the online phase, localization is based on least squares

$$\hat{\vec{x}} = \arg\min_{\vec{x}} \sum_{a=1}^{A} \left(\left\| \vec{x} - \vec{X}_a \right\|_2 - \hat{d}_a \right)^2 , \tag{8}$$

where $\hat{d}_a$ is the distance to AP a estimated with the fitted polynomial. Yang *et. al.* [22] propose further refinements which we do not implement.

B. k-nearest neighbors

Given an online measurement $\vec{s}$, the k-nearest neighbors (kNN) algorithm looks for the k closest examples $\vec{u}_{r_1}$, $\vec{u}_{r_2}$, $\cdots$, $\vec{u}_{r_k}$ recorded in calibration phase and estimates the current position of the device as

$$\hat{\vec{x}} = \sum_{i=1}^{k} w_i \vec{x}_i, \tag{9}$$

where w_i are suitably chosen weights. This is one of the simplest and earliest positioning algorithms based on RSSI of WiFi signals. It was the core of the RADAR system [23]. Algorithms vary in the metrics used to measure closeness and the way in which weights are computed. A usual choice for the metric is the simple Euclidean distance, although other alternatives such as cosine similarity can be found in the literature [24]. The simplest alternative is to use use uniform weights. However, sometimes it is convenient to use weights which are proportional to the inverse of the distance.

In this paper, we use Euclidean distance and uniform weights. The number of neighbors is chosen on a trial and error basis to get the best results.

C. HORUS

HORUS [25] is a positioning system which is commonly used as a benchmark for comparison with new algorithms. HORUS can be considered an extension of the k-nearest neighbors algorithm in which the distance is inversely proportional to the posterior probability $\mathrm{p}\left(\vec{x}_r \middle| \vec{s}\right)$ and the weights are given by

$$w_i = \frac{\mathrm{p}\left(\vec{x}_{r_i} \middle| \vec{s}\right)}{\sum\limits_{j=1}^{k} \mathrm{p}\left(\vec{x}_{r_j} \middle| \vec{s}\right)} . \tag{10}$$

Using Bayes' theorem and assuming a uniform distribution on all possible locations, posterior probabilities can be replaced by prior probabilities in this equation. Calculations are further simplified if independence between different APs is assumed, i.e.,

$$\mathrm{p}\left(\vec{s} \middle| \vec{x}_r\right) = \prod_{a=1}^{A} \mathrm{p}\left(\vec{s}(a) \middle| \vec{x}_r\right) . \tag{11}$$

Prior probabilities have to be estimated from the measurements and several variants of this approach differ on the way that such probabilities are estimated. Ref. [25] suggest the use of both parametric and non-parametric distributions. While for the parametric case HORUS uses a simple Gaussian approximation, for the non-parametric case it estimates the probabilities by means of several histograms. We use simple Gaussian fits in our experiments.

D. Kernel density estimation

HORUS' estimator can be considered an approximation to the conditional expectation of the position given current measurements. Kushki *et al.* [26] propose to estimate this expectation using kernel density estimation (KDE). Assume that $k = R$ in Eq. (9). Then, the weights are computed as

$$w_r = \frac{1}{n\sigma^A} \sum_{n=1}^{N} K\left(\frac{\vec{s} - \vec{u}_r^n}{\sigma}\right) , \tag{12}$$

where K is a suitably chosen kernel function and σ is a parameter. In this work, we used a simple Gaussian kernel and estimated σ as suggested in [26]. We also tried variable kernel density estimators, but we did not find significant differences in performance.

E. General regression algorithms

A straightforward approach to localization is to fit a linear model using least squares

$$\hat{\vec{\alpha}} = \arg\min_{\vec{\alpha}} \left\| \mathbf{U}\vec{\alpha} - \mathbf{X} \right\|_2 . \tag{13}$$

It is well-known that this type of fit tends to overfit training data. An usual alternative is to add a penalty (regularization) term

$$\hat{\vec{\alpha}} = \arg\min_{\vec{\alpha}} \left\| \mathbf{U}\vec{\alpha} - \mathbf{X} \right\|_2 + \lambda \left\| \vec{\alpha} \right\|_2 . \tag{14}$$

The resulting approach is known as ridge regression [27] and favors 'smaller' solutions. λ is a parameter that must be chosen, e.g., by cross-validation. An alternative is to use an l_1 penalty

$$\hat{\vec{\alpha}} = \arg\min_{\vec{\alpha}} \left\| \mathbf{U}\vec{\alpha} - \mathbf{X} \right\|_2 \qquad \text{subject to } \left\| \vec{\alpha} \right\|_1 \leq t, \tag{15}$$

where t is a parameter that needs to be tuned. This is the LASSO regression approach [28] which favors 'sparser' solutions. Elastic net regression [29] uses both types of penalties

$$\hat{\vec{\alpha}} = \arg\min_{\vec{\alpha}} \left\| \mathbf{U}\vec{\alpha} - \mathbf{X} \right\|_2 + \lambda \left\| \vec{\alpha} \right\|_1 + \lambda_2 \left\| \vec{\alpha} \right\|_2 . \tag{16}$$

These regression methodologies have already been used for localization [30]. Further details can be found, e.g., in Ref. [31].

A limitation of all these approaches is that they use linear models. Nonlinear models can be incorporated by resorting to the kernel trick [31], [32]. Indeed, through the kernel trick, (linear) ridge regression is done in a higher dimensional feature space.

In this work, we choose t, λ, λ_1 and λ_2 through cross-validation. We use a Gaussian kernel for kernel ridge regression.

F. Support vector machines

Support vector machines (SVMs) have also been used for positioning [33]–[36]. One of the alternatives to use support vector machines for regression is through what is known as ε-insensitivity [31], [37]. The problem solved, in the linear case, is

$$\hat{\vec{\alpha}}, \hat{\alpha_0} = \arg\min_{\vec{\alpha},\alpha_0} \sum_{r=1}^{R} \sum_{n=1}^{N} L_\varepsilon \left(\vec{\alpha}^T \vec{u}_r^n + \alpha_0 - \vec{x}_r \right) + \frac{\lambda}{2} \|\vec{\alpha}\|_2 , \tag{17}$$

$$L_\varepsilon(r) = \begin{cases} 0 & \text{if } |r| < \varepsilon \\ |r| - \varepsilon & \text{otherwise.} \end{cases} \tag{18}$$

Note that small errors ($< \varepsilon$) are not taken into account. This approach can be extended by using the kernel trick. For more details, see the [31], [37] and references therein. Let us just note that there is a connection between variants of support vector regression (SVR) and ridge regression in the previous section [38]. In this paper, we choose hyperparameters through trial and error and we use a Gaussian kernel.

G. Neural networks

Neural networks have been extensively used for localization (see, e.g., [39]–[43]). In this work, we focus on multilayer perceptron (MLP) neural nets. In particular, we use a single hidden layer network with logistic nodes. The size of the network was chosen on trial and error basis.

H. Access point selection and outlier detection

Power measurements from different APs are not necessarily independent. Therefore, there are measurements from certain access points which may be left out without significant loss of information. Moreover, the use of weak signals from distant APs may degrade positioning accuracy [3]. For these reasons, it makes sense to select a subset of all observable access points. From the many selection methodologies, in this work we use two of the most common ones [3]:

1) Strongest APs: Only those access points with highest mean power are selected.
2) Fisher criterion: Received power from an access point may be high, but it may also exhibit a high variance. The Fisher criterion selects those APs with high values of a metric that takes into account the stability of the measured power from each access point. The score for each AP is given by

$$F_a = \frac{\sum_{r=1}^{R} \left(\vec{u}_r(a) - \bar{\vec{u}}(a) \right)^2}{\frac{1}{N-1} \sum_{n=1}^{N} \sum_{r=1}^{R} \left(u(n,r,a) - \vec{u}_r(a) \right)^2}. \tag{19}$$

We study the influence of the AP selection algorithm and the number of chosen APs in our experiments.

Erroneous measurements may lead to errors in the training of the positioning algorithm. For this reason, it may be useful to leave out suspicious training samples or outliers. We used Hampel filter (see [3] and references therein). For each measurement, we defined

$$\mathrm{MAD}(n,r,a) = \frac{|u(n,r,a) - \mathrm{median}(\vec{u}_r^a)|}{\mathrm{median}\left\{ |u(n,r,a) - \mathrm{median}(\vec{u}_r^a)| \right\}}. \tag{20}$$

Those measurements for which $\mathrm{MAD}(n,r,a) > \eta$, where η is a suitably chosen threshold, are discarded as outliers. In this paper, we analyze the influence of outlier detection in the offline/calibration phase, but we did not implement outlier detection in the online phase.

IV. Experimental setup

We used a Lenovo Yoga Tablet 2 and we developed a simple Android application to handle measurements from several sensors, including WiFi signals. The application asked the operating system to continuously scan for wireless access points (APs) and asynchronously received information on all WiFi networks detected in the area. Although the time between information updates was not controlled, the application received one update every 5 seconds, approximately. We must observe that RSSI measurements reported by the operating system were not calibrated and their accuracy depended completely on the underlying hardware. Disassembly of the tablet [44] reveals that it uses a Broadcom BCM43241 chip. Its datasheet [45] summarized the accuracy of the RSSI measurements at a 95 % confidence level as shown in Table I. Although it is not clear the nature of the measurement errors, say, random errors in a single device or measurement differences among devices, a standard deviation of ~ 2.5 dB is found for the lower power range under a normality assumption.

Measurements were taken in a large room, often used for motion capture experiments, at one of ITBA's buildings. As seen in Fig. 1, the room was almost empty. We set up a measurement grid on the floor. A coarse, 1×1-meter measurement grid, was refined with several intermediate points, as shown in Fig. 2. An even finer grid was deemed unnecessary in view of the order of magnitude of the theoretical accuracy bounds. Exactly 20 measurements were taken at each location. The Lenovo tablet was placed on a short stool (see Fig. 2). In order to get LOS signals, we also placed three APs on the same room, at heights similar to that of the tablet or slightly higher. These APs were based on RaspBerry Pi running a Linux variant. We must note that an analysis of the resulting radiomaps suggested that the antennas on these access points were far from isotropic.

Besides the three APs placed inside the room, the tablet recorded information from other APs. A set of nine access points was detected at all measurement locations. These APs transmitted beacons on the following frequencies: 2412, 2417, 2437, 2462 and 5745 MHz. RSSI measurements ranged from -85 to -18 dBm.

TABLE I
RSSI accuracy.

Range [dBm]	Accuracy [dB]	
	Minimum	Maximum
-98 to -30	-5	5
> -30	-8	8

Fig. 1. Experimental setup.

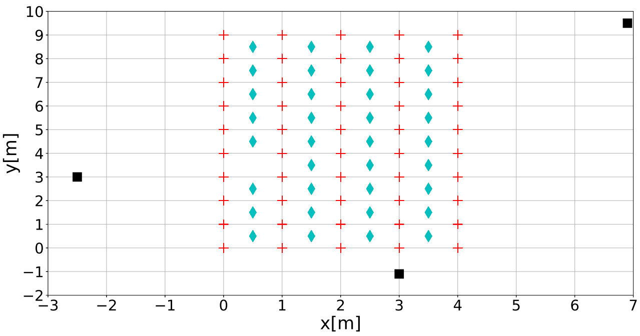

Fig. 2. Simplified measurement setup. Crosses and circles mark measurement locations. Crosses form a 1×1-meter grid. Filled squares mark the location of the three access points placed inside the room.

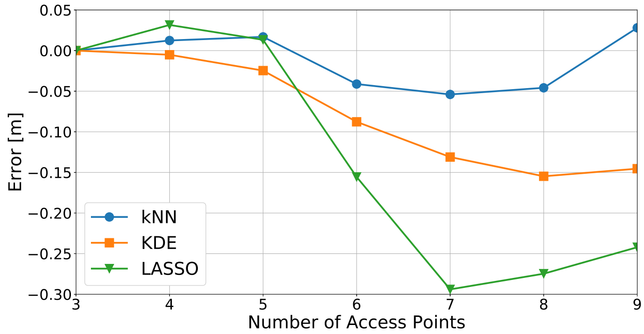

Fig. 3. Effect of the maximum-mean-power criterion for the selection of APs. Only deviations of the median errors from the 3-APs case are shown.

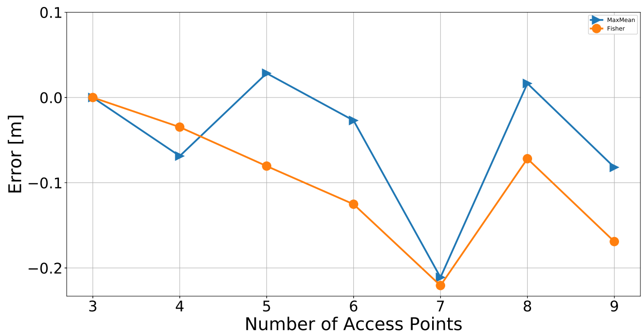

Fig. 4. Effect of the maximum-mean-power (triangles) and the Fisher (circles) criteria for the selection of APs when the neural network algorithm is used. Only deviations of the median errors from the 3-APs case are shown.

V. RESULTS

We randomly chose 80% of the reference points for training and the remaining 20% for testing. Empirical cumulative distribution functions and quartiles are based on 100 hundred random partitions in order to average out the effects of particular training and testing datasets. We implemented all algorithms in Python, using Scikit-learn [46], SciPy [47], NumPy [48], Pandas [49] and Matplotlib [50].

Fig. 3 presents the effect of the maximum-mean-power criterion for the selection of access points when kNN, KDE and LASSO are used. The performance of kernel-density estimation and LASSO improves significantly when more access points are taken into account. The behavior of all other algorithms (not shown for in the figure for the sake of clarity), with the exception of kNN and neural networks, is similar to that of KDE. Indeed, a larger number of APs provides more information and, at least in our experiments, it seems that there are no access points providing highly erroneous or misleading data. K-nearest neighbors, however, does not show a significant accuracy improvement with increasing number of access points. A possible explanation is that, if there are no APs that provide highly erroneous information, no matter the number of access points chosen, the same nearest neighbors are always selected. Fig. 4 shows the results for the neural network algorithm. In this case, the performance is highly dependent on the number of APs, not showing any increasing or decreasing tendency, and we have not been able to come up with an explanation for such a puzzling behavior.

Fig. 5 presents the effect of the Fisher criterion for the selection of access points when kNN, KDE and ridge regression are used. The conclusions are similar to those from Fig. 3.

Sometimes, selecting a subset of APs for localization may help to lower the complexity of the algorithms. In this sense, it is interesting to study which criterion performs better at the task of access point selection. Fig. 6 shows the median error when 5 APs are selected according to each criterion and reveals that the Fisher criterion performs consistently better for all localization algorithms.

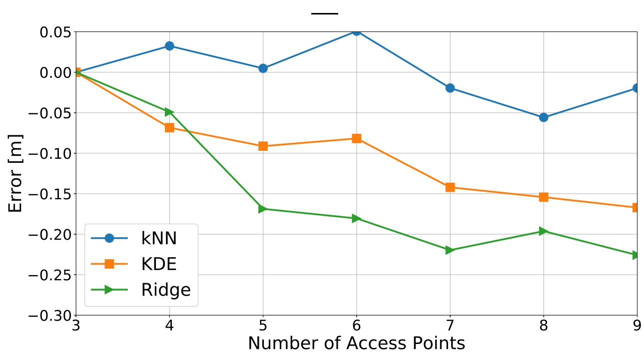

Fig. 5. Effect of the Fisher criterion for the selection of APs. Only deviations of the median errors from the 3-APs case are shown.

Fig. 7 presents the effect of the Hampel filter on kNN, support vector regression, KDE and the model-based algorithm. Results are shown for η (see Eq. (20)) ranging 3.5 (many examples are discarded as outliers) to 5 (no example is discarded). The performance of all algorithms, including those not shown in the figure for the sake of clarity, but with

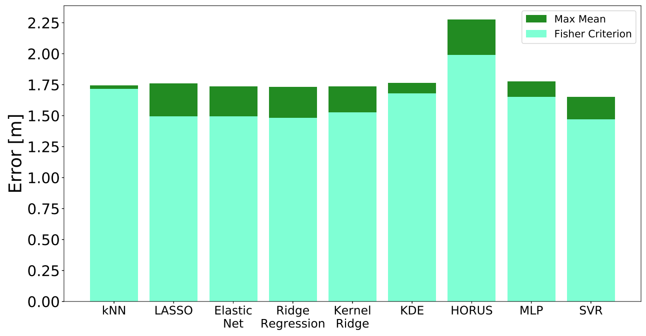

Fig. 6. Comparison of the maximum-mean-power and Fisher criteria for the selection of APs. Bars present the median errors for the 5-APs case.

the exception of the model-based methodology, increases as more data is included (less “outliers” are discarded). In the case of kernel-density estimation, the improvement is significant. A larger number of examples allows a better density estimation. In the case of the model-based algorithm, a few “bad” training examples may harm the model fitted according to Eq. (7).

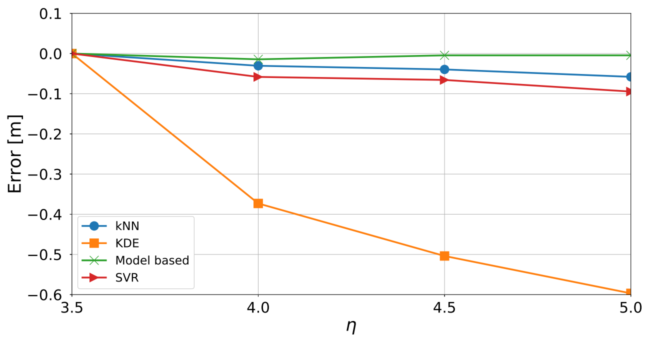

Fig. 7. Effect of Hampel filter. Only deviations of the median errors from the $\eta = 3.5$ case are shown.

As we have already mentioned, one factor that affects localization accuracy based on power measurements is shadowing (see Eq. (1)). By averaging several measurements, we may reduce the effect of shadowing, at the expense of increasing the localization time. Fig. 8 shows that averaging can indeed increase the localization accuracy. This figure presents the cumulative distribution function (CDF) of the errors when SVR is used and either the average of 20 measurements or no averaging is carried out. The first two rows of Table II present the same result in a different way. The median error when measurements are averaged is $\sim 10\%$ less than when there is no averaging.

Finally, Fig. 9 and Table II present the performance results for the localization algorithms, using the best hyperparameters, number of APs and Hampel filter threshold. Some algorithms were not included in the figure for the sake of clarity. Indeed, LASSO, elastic net, ridge regression and kernel ridge regression have a similar performance and, thus, only the latter is shown. As it can be observed, support vector regression gives the best results, although they are not significantly better than those of many other algorithms. kNN and HORUS give poorer results. The model-based methodology gives the worst results, but it has to be taken into account that only three access points were considered in this case.

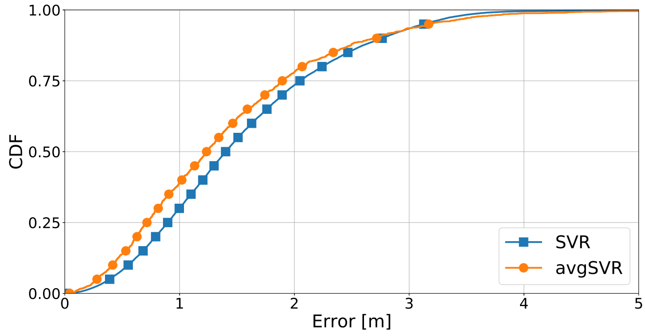

Fig. 8. Effect of averaging several RSSI measurements for Support Vector Regression. Empirical cumulative distribution functions of the error when 20 measurements (circles) are averaged and when no averaging takes place (squares).

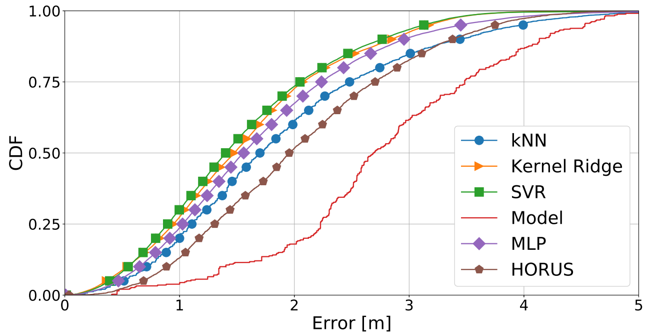

Fig. 9. Empirical cumulative distribution functions of the error for different algorithms.

TABLE II
LOCALIZATION ACCURACY.

Algorithm	1st Quartile [m]	Median [m]	3rd Quartile [m]
SVR w/averaged meas.	0.72	1.24	1.89
SVR	0.90	1.40	2.05
Elastic Net	0.92	1.45	2.11
LASSO	0.92	1.45	2.11
Ridge Regression	0.91	1.45	2.12
Kernel Ridge Regr.	0.95	1.48	2.08
Neural Net	1.03	1.56	2.24
KDE	1.06	1.61	2.26
kNN	1.11	1.70	2.48
HORUS	1.31	1.96	2.70
Model-based	2.23	2.69	3.48

VI. CONCLUSION

In this paper, we focused on the question of the attainable accuracy of WiFi RSSI-based localization algorithms using off-the-shelf hardware. In order to answer this question, we obtained measurements from a carefully planned experiment and we tried several state-of-the-art localization algorithms. We found improbable to obtain median positioning errors below 1 m. Indeed, the best performing algorithm gave a median error of 1.4 m.

Under our experimental setup, we found convenient, in general, to use the information from as many access points as possible. In the case where there is a need to restrict the set of APs, the Fisher criterion performs better than the simpler maximum-mean-power criterion.

We also found that discarding plausible outliers does not seem to help. However, averaging several measurements may help to diminish the effect of signal shadowing.

Although localization algorithms based on RSSI measurements have dominated the commercial products, new hardware allows the precise measurement of round-trip signal traveling times. In particular, the latest standards allow measurements with a precision of up to 1 ns [7], equivalent to a ranging precision of $\sim$ 30 cm. This new technology will certainly allow greater accuracy in coming localization developments.

ACKNOWLEDGMENT

We gratefully acknowledge financial support from ANPCyT under Project PID 2015-003.

REFERENCES

[1] W. G. Figel, N. H. Shepherd, and W. F. Trammell, "Vehicle location by a signal attenuation method," in *19th IEEE Vehicular Technology Conference*, vol. 19, 1968, pp. 105–109.

[2] S. He and S. H. G. Chan, "Wi-fi fingerprint-based indoor positioning: Recent advances and comparisons," *IEEE Communications Surveys Tutorials*, vol. 18, no. 1, pp. 466–490, Firstquarter 2016.

[3] A. Khalajmehrabadi, N. Gatsis, and D. Akopian, "Modern wlan fingerprinting indoor positioning methods and deployment challenges," *IEEE Communications Surveys Tutorials*, vol. 19, no. 3, pp. 1974–2002, thirdquarter 2017.

[4] S. Sen, R. R. Choudhury, B. Radunovic, and T. Minka, "Precise indoor localization using PHY layer information," in *Proceedings of the 10th ACM Workshop on Hot Topics in Networks*, ser. HotNets-X, ACM. New York, NY, USA: ACM, 2011, p. 18.

[5] J. Xiao, K. Wu, Y. Yi, and L. M. Ni, "Fifs: Fine-grained indoor fingerprinting system," in *2012 21st International Conference on Computer Communications and Networks (ICCCN)*, Jul. 2012, pp. 1–7.

[6] K. Wu, J. Xiao, Y. Yi, D. Chen, X. Luo, and L. M. Ni, "CSI-based indoor localization," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 7, pp. 1300–1309, Jul. 2013.

[7] L. Banin, U. Schatzberg, and Y. Amizur, "Wifi ftm and map information fusion for accurate positioning," in *2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN),*, 2016.

[8] M. Laaraiedh, S. Avrillon, and B. Uguen, "Cramer–rao lower bounds for nonhybrid and hybrid localisation techniques in wireless networks," *Transactions on Emerging Telecommunications Technologies*, vol. 23, no. 3, pp. 268–280, 2012. [Online]. Available: http://dx.doi.org/10.1002/ett.1530

[9] H. Liu, H. Darabi, P. Banerjee, and J. Liu, "Survey of wireless indoor positioning techniques and systems," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 37, no. 6, pp. 1067–1080, Nov. 2007.

[10] V. Honkavirta, T. Perala, S. Ali-Loytty, and R. Piche, "A comparative survey of wlan location fingerprinting methods," in *2009 6th Workshop on Positioning, Navigation and Communication*, March 2009, pp. 243–251.

[11] Y. Gu, A. Lo, and I. Niemegeers, "A survey of indoor positioning systems for wireless personal networks," *IEEE Communications Surveys Tutorials*, vol. 11, no. 1, pp. 13–32, First 2009.

[12] K. Al Nuaimi and H. Kamel, "A survey of indoor positioning systems and algorithms," in *Innovations in Information Technology (IIT), 2011 International Conference on*, Apr. 2011, pp. 185–190.

[13] A. Correa, M. Barcelo, A. Morell, and J. L. Vicario, "A review of pedestrian indoor positioning systems for mass market applications," *Sensors*, vol. 17, no. 8, 2017. [Online]. Available: http://www.mdpi.com/1424-8220/17/8/1927

[14] J. Torres-Sospedra, A. Moreira, S. Knauth, R. Berkvens, R. Montoliu, O. Belmonte, S. Trilles, M. João Nicolau, F. Meneses, A. Costa, A. Koukofikis, M. Weyn, and H. Peremans, "A realistic evaluation of indoor positioning systems based on wi-fi fingerprinting: The 2015 evaal–etri competition," vol. 9, pp. 263–279, 2017.

[15] A. Goldsmith, *Wireless Communications*. Cambridge University Press, 2005.

[16] Y. Qi and H. Kobayashi, "On relation among time delay and signal strength based geolocation methods," in *Global Telecommunications Conference, 2003. GLOBECOM '03. IEEE*, vol. 7, Dec 2003, pp. 4079–4083 vol.7.

[17] R. Akl, D. Tummala, and X. Li, "Indoor propagation modeling at 2.4 ghz for ieee 802.11 networks." in *Wireless and Optical Communications*, 2006.

[18] T. Chrysikos, P. Georgakopoulos, I. Oikonomou, S. Kotsopoulos, and D. Zevgolis, "Channel measurement and characterization for a complex industrial and office topology at 2.4 ghz," in *2017 11th International Conference on Software, Knowledge, Information Management and Applications (SKIMA)*, Dec 2017, pp. 1–8.

[19] S. Sen, J. Lee, K.-H. Kim, and P. Congdon, "Avoiding multipath to revive inbuilding WiFi localization," in *Proceeding of the 11th Annual International Conference on Mobile Systems, Applications, and Services*, ser. MobiSys '13. New York, NY, USA: ACM, 2013, pp. 249–262.

[20] G. Lui, T. Gallagher, B. Li, A. G. Dempster, and C. Rizos, "Differences in rssi readings made by different wi-fi chipsets: A limitation of wlan localization," in *2011 International Conference on Localization and GNSS (ICL-GNSS)*, June 2011, pp. 53–57.

[21] S. Depatla, A. Muralidharan, and Y. Mostofi, "Occupancy estimation using only WiFi power measurements," *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 7, pp. 1381–1393, Jul. 2015.

[22] J. Yang and Y. Chen, "Indoor localization using improved rss-based lateration methods," in *GLOBECOM 2009 - 2009 IEEE Global Telecommunications Conference*, Nov 2009, pp. 1–6.

[23] P. Bahl and V. N. Padmanabhan, "Radar: an in-building rf-based user location and tracking system," in *Proceedings IEEE INFOCOM 2000. Conference on Computer Communications. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (Cat. No.00CH37064)*, vol. 2, 2000, pp. 775–784 vol.2.

[24] S. Han, C. Zhao, W. Meng, and C. Li, "Cosine similarity based fingerprinting algorithm in wlan indoor positioning against device diversity," in *2015 IEEE International Conference on Communications (ICC)*, June 2015, pp. 2710–2714.

[25] M. Youssef and A. Agrawala, "The horus wlan location determination system," in *Proceedings of the 3rd International Conference on Mobile Systems, Applications, and Services*, ser. MobiSys '05. New York, NY, USA: ACM, 2005, pp. 205–218. [Online]. Available: http://doi.acm.org/10.1145/1067170.1067193

[26] A. Kushki, K. N. Plataniotis, and A. N. Venetsanopoulos, "Kernel-based positioning in wireless local area networks," *IEEE Transactions on Mobile Computing*, vol. 6, no. 6, pp. 689–705, June 2007.

[27] A. E. Hoerl and R. W. Kennard, "Ridge regression: Biased estimation for nonorthogonal problems," *Technometrics*, vol. 12, no. 1, pp. 55–67, 1970. [Online]. Available: https://amstat.tandfonline.com/doi/abs/10.1080/00401706.1970.10488634

[28] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996. [Online]. Available: http://www.jstor.org/stable/2346178

[29] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, vol. 67, no. 2, pp. 301–320, 2005. [Online]. Available: http://www.jstor.org/stable/3647580

[30] A. Khalajmehrabadi, N. Gatsis, D. J. Pack, and D. Akopian, "A joint indoor wlan localization and outlier detection scheme using lasso and elastic-net optimization techniques," *IEEE Transactions on Mobile Computing*, vol. 16, no. 8, pp. 2079–2092, Aug 2017.

[31] J. F. Trevor Hastie, Robert Tibshirani, *The Elements of Statistical Learning*. Springer, 2017.

[32] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. MIT Press, 2012.

[33] Z. l. Wu, C. h. Li, J. K. y. Ng, and K. R. p. h. Leung, "Location estimation via support vector regression," *IEEE Transactions on Mobile Computing*, vol. 6, no. 3, pp. 311–321, March 2007.

[34] K. Shi, Z. Ma, R. Zhang, and H. C. Wenbiao Hu, "Support vector regression based indoor location in ieee 802.11 environments," vol. 2015, no. 295652, 2015.

[35] A. S. Abdou, M. A. Aziem, and A. Aboshosha, "An efficient indoor localization system based on affinity propagation and support vector regression," in *2016 Sixth International Conference on Digital Information Processing and Communications (ICDIPC)*, April 2016, pp. 1–7.

[36] N. Hernández, M. Ocaña, J. M. Alonso, and E. Kim, "Continuous space estimation: Increasing wifi-based indoor localization resolution without increasing the site-survey effort," *Sensors*, vol. 17, no. 1, 2017. [Online]. Available: http://www.mdpi.com/1424-8220/17/1/147

[37] H. Drucker, C. J. C. Burges, L. Kaufman, A. Smola, and V. Vapnik, “Support vector regression machines,” in *Proceedings of the 9th International Conference on Neural Information Processing Systems*, ser. NIPS’96. Cambridge, MA, USA: MIT Press, 1996, pp. 155–161. [Online]. Available: http://dl.acm.org/citation.cfm?id=2998981.2999003

[38] C. Saunders, A. Gammerman, and V. Vovk, “Ridge regression learning algorithm in dual variables,” in *Proceedings of the Fifteenth International Conference on Machine Learning*, ser. ICML ’98. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1998, pp. 515–521. [Online]. Available: http://dl.acm.org/citation.cfm?id=645527.657464

[39] T. L. N. Roberto Battiti and A. Villani, “Location-aware computing: a neural network model for determining location in wireless lans,” Universit‘a di Trento, Dipartimento di Informatica e Telecomunicazioni, Tech. Rep. DIT-02-0083, 2002.

[40] S. Saha, K. Chaudhuri, D. Sanghi, and P. Bhagwat, “Location determination of a mobile device using IEEE 802.11b access point signals,” in *2003 IEEE Wireless Communications and Networking, 2003. WCNC 2003.*, vol. 3, Mar. 2003, pp. 1987–19 923.

[41] U. Ahmad, A. Gavrilov, U. Nasir, M. Iqbal, S. J. Cho, and S. Lee, “In-building localization using neural networks,” in *2006 IEEE International Conference on Engineering of Intelligent Systems*, 2006, pp. 1–6.

[42] M. S. Rahman, Y. Park, and K. D. Kim, “Localization of wireless sensor network using artificial neural network,” in *2009 9th International Symposium on Communications and Information Technology*, Sept 2009, pp. 639–642.

[43] Y. Gu, Y. Chen, J. Liu, and X. Jiang, “Semi-supervised deep extreme learning machine for wi-fi based localization,” *Neurocomputing*, vol. 166, pp. 282 – 293, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S092523121500418X

[44] David. (2018) Lenovo yoga tablet 2 teardown. [Online]. Available: http://www.myfixguide.com/manual/lenovo-yoga-tablet-2-disassembly/

[45] Broadcom, *BCM43241 Advance Data Sheet*, 2012.

[46] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, “Scikit-learn: Machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[47] E. Jones, T. Oliphant, P. Peterson *et al.*, “SciPy: Open source scientific tools for Python,” 2001–, [Online; accessed ¡today¿]. [Online]. Available: http://www.scipy.org/

[48] T. E. Oliphant, *A guide to NumPy*. Trelgol Publishing, 2006.

[49] W. McKinney, “Data structures for statistical computing in python,” in *Proceedings of the 9th Python in Science Conference*, S. van der Walt and J. Millman, Eds., 2010, pp. 51 – 56.

[50] J. D. Hunter, “Matplotlib: A 2d graphics environment,” *Computing In Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007.