Computer Networks 95 (2016) 51–61

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

An approach using support vector regression for mobile location in cellular networks

Robson D.A. Timoteo[a], Lizandro N. Silva[b,1], Daniel C. Cunha[a,*], George D.C. Cavalcanti[a]

[a] Centro de Informática da UFPE, Av. Jornalista Anibal Fernandes, s/n, 50740-560 Recife, PE, Brazil
[b] Telefonica-Vivo, Avenida Engenheiro Domingos Ferreira, 837, 51011-051 Recife, PE, Brazil

ARTICLE INFO

Article history:
Received 13 February 2015
Revised 10 December 2015
Accepted 15 December 2015
Available online 22 December 2015

Keywords:
Wireless communications
Positioning system
Fingerprinting techniques
Machine learning
Support vector regression

ABSTRACT

Wireless positioning systems have become very popular in recent years. One of the reasons is the fact that the use of a new paradigm named *Internet of Things* has been increasing in the scenario of wireless communications. Since a high demand for accurate positioning in wireless networks has become more intensive, especially for location-based services, the investigation of mobile positioning using radiolocalization techniques is an open research problem. Based on this context, we propose a fingerprinting approach using support vector regression to estimate the position of a mobile terminal in cellular networks. Simulation results indicate the proposed technique has a lower error distance prediction and is less sensitive to a Rayleigh distributed noise than the fingerprinting techniques based on COST-231 and ECC-33 propagation models.

1. Introduction

An extraordinary growth of wireless systems has been observed in recent years. The evolution and dissemination of mobile computing devices, such as tablets, smartphones, sensors and actuators, have motivated this development, primarily in the context of the *Internet of Things*, a new paradigm that is moving forward in the scenario of wireless telecommunications [1].

Considering that wireless information access is widely available, a high demand for accurate positioning in wireless networks has become more intensive, especially for location-based services [2,3]. For this purpose, the positioning techniques can be classified in indoor and outdoor. In case of indoor positioning, the main challenges are: (*i*) inefficiency of global positioning system (GPS)—one of the most traditional localization techniques [4]—inside buildings; (*ii*) different materials (glass, plaster, wood) are used to construct the buildings walls and each material has its specific attenuation values for radio frequency signals; (*iii*) multipath loss, i.e., propagation phenomenon that results in radio signals reaching the receiving antenna by two or more paths. For outdoor localization, GPS is a widely adopted technique that is in use for many years. However, GPS do not solve the outdoor positioning problem for large networks of very small and low power devices. Some factors, such as size, cost, and power constraints inhibit the use of GPS on all nodes. So, it is necessary to explore other alternatives, such as radiolocalization [5–9].

Mobile positioning using radiolocalization techniques can involve different parameters, such as time of arrival (ToA), angle of arrival (AoA), and received signal strength indicator (RSSI) measurements. Various proposals can be found in the literature, but these techniques tend to be

* Corresponding author. Tel.: +55 81 21268430; fax: +55 81 21268438.
E-mail addresses: rdat@cin.ufpe.br (R.D.A. Timoteo), lizandro.silva@telefonica.com (L.N. Silva), dcunha@cin.ufpe.br (D.C. Cunha), gdcc@cin.ufpe.br (G.D.C. Cavalcanti).
[1] Tel.: +55 81 981208194.

http://dx.doi.org/10.1016/j.comnet.2015.12.003
1389-1286/

very inaccurate, in particular because of multipath and non-line-of-sight (NLoS) propagation [10–12]. For the purpose of obtaining a better precision, an interesting approach is to model the geolocalization problem of mobile terminals in wireless networks as a machine learning (ML) problem. ML is a data-driven approach that learns from data. So, based on past observations, ML algorithms aim to automatically extract information from data in order to make accurate predictions. For example, in [13], a least square-support vector machine algorithm is proposed to model the relationship between ToA measurements and the position of the user.

Previous works used different ML approaches to address wireless positioning prediction. Among these techniques, support vector regression (SVR) has been used successfully in indoor location fingerprinting [6–9]. Brunato and Battiti [7] compared five different algorithms: support vector machine (SVM), weighted and non-weighted *k*-nearest neighbors, Bayesian approach, and multi-layer perceptron. Their best results were obtained with SVR. In a recent work, Zou et al. [14] used online sequential extreme learning machine to indoor localization. Their proposal obtained better precision than *k*-nearest neighbors, fuzzy *k*-nearest neighbors, and batch extreme learning machine; however, they did not compare it with SVR. A comparison between extreme learning machine and SVR applied to the problem of range-free localization can be found in [6]. The authors concluded that extreme learning machine is also a good machine learning technique, however, it cannot be preferred to SVR.

In this work, we propose an ML approach using SVR to predict path loss in an urban outdoor environment. From these predictions, coverage maps are used to locate the position of a mobile terminal. Simulation results indicate that the SVR-based approach is a better alternative when compared to fingerprinting location techniques using COST-231 and ECC-33 propagation models. Besides, the proposed approach is less sensitive to a Rayleigh distributed noise than the two fingerprinting location methods considered. It is also important to remark that, to the best of our knowledge, this is the first work to address outdoor localization using ML.

The outline of the paper is as follows. In Section 2, we present the principles of fingerprinting location techniques, as well as some basic concepts about SVR. In Section 3, we propose a fingerprinting SVR-based algorithm to estimate the position of the mobile node based on the path loss predictions. Numerical results comparing the localization methods are shown in Section 4, and finally, conclusions are drawn in Section 5.

2. Basic background

This section provides sufficient background on fingerprinting location techniques and SVR.

2.1. Fingerprinting location techniques

Fingerprinting (FP) location techniques are a group of mobile station (MS) positioning methods that are accurate and suitable for NLoS environments and can be applied in any wireless network [15,16]. There is a huge variety of fingerprinting techniques, but as shown in [15], all share the same basic elements as follows: fingerprint, database, location server, search space reduction technique and pattern matching.

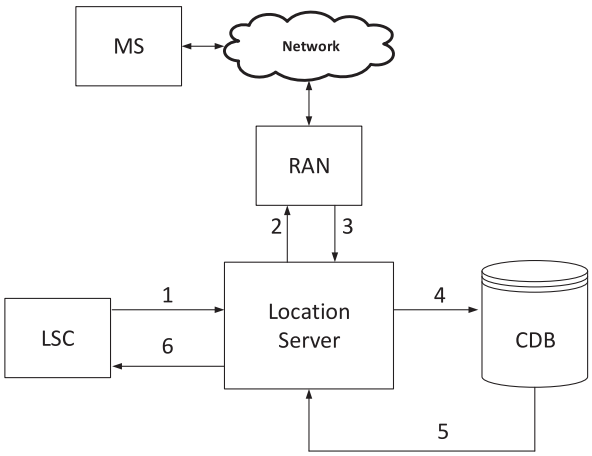

Fig. 1. Simplified diagram of a fingerprinting location (adapted from [15]).

Fingerprint is the vector with all features (observed values) which are used for pattern matching. Among the signal parameters commonly used as features, we can emphasize RSSI, AoA, ToA and time difference of arrival (TDoA) [16]. The database, also known as fingerprint correlation database (CDB), is built from collected field data and predictions made by radio propagation models, such as COST-231 [15]. Each fingerprint stored in the database is linked to a specific position. In this scenario, it is not feasible to make measurements at all positions. Therefore, propagation models are used to generalize the measurements. The location server is a network element that is responsible for receiving location requests, consulting the database and estimating the MS position.

Analyzing all fingerprints in such massive database can be very time-consuming which might not be feasible in practical applications. Therefore, fingerprinting location methods make use of some techniques to reduce the search space within the database. There is a wide range of alternatives, which can be used for reducing the search space, such as genetic algorithms (GA) [15], CDB filtering [15], and timing advanced-based approaches [17].

In order to estimate the MS position, the location server tries to match the measured fingerprint (from sought MS) to a fingerprint stored in the CDB. The key idea is to find the point in the CDB which has the highest similarity or correlation with the measured fingerprint.

Fig. 1 shows a simplified schematic diagram for a generic fingerprinting location technique. This diagram corresponds to a location service client (LSC) according to [18]. In step 1, the LSC sends a request to the location server. Then, the location server requests measurements from MS (target) through the radio access network (RAN). In step 3, RAN sends measurements to the location server. After receiving the measurements, the location server composes the fingerprint and, in step 4, it queries the CDB to

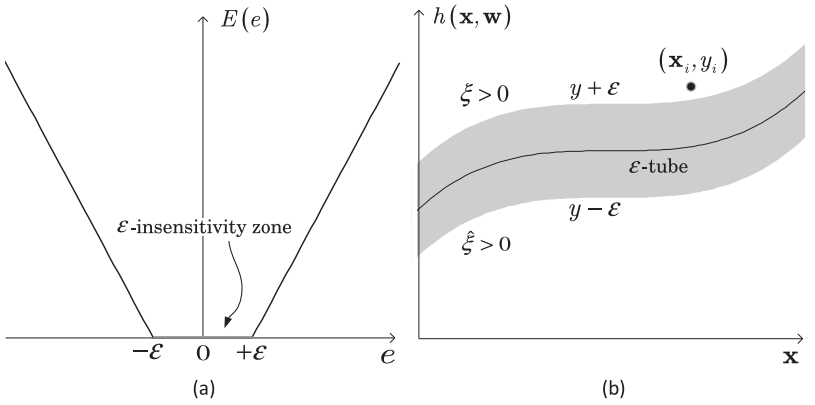

Fig. 2. (a) Vapnik's linear loss function with ε-insensitivity zone versus e. (b) ε-tube defined from the function $E(e)$.

get the reduced search area. In step 5, the location server receives the query results and uses a comparison function to obtain the estimated position of the MS. Finally, in step 6, the location server sends a response to the LSC.

2.2. Support vector regression

SVMs [19] are supervised learning models that were first developed to deal with classification problems which the objective is to find an optimal hyperplane that separates two classes. However, SVM can also be used in regression problems and, in this case, it is called support vector regression (SVR) [20]. SVR aims to construct a hyperplane that lies close to as many of the data points as possible [21]. Thus, the selected hyperplane must have a small norm and must minimize the sum of the distances from the data points to the hyperplane.

The general regression learning problem is defined as follows. Assume that we have a set of observations, defined as training data, and we will use these observations to train, or teach, our algorithm how to estimate the input–output relationship. So, consider a training data set $D = \{((\mathbf{x}_i, y_i) \in \mathbb{R}^n \times \mathbb{R}, i = 1, 2, \ldots, \ell\}$ with ℓ pairs $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_\ell, y_\ell)$, where the inputs are n-dimensional vectors $\mathbf{x}_i \in \mathbb{R}^n$, the outputs $y_i \in \mathbb{R}$ are continuous values and ℓ is the number of instances, or observations, in the training data set D.

We begin by examining the linear regression problem. Accordingly, consider that $h(\mathbf{x}_i, \mathbf{w})$ is a linear regression hyperplane given by

$$h(\mathbf{x}_i, \mathbf{w}) = \langle \mathbf{w}, \mathbf{x}_i \rangle + b, \tag{1}$$

where $\mathbf{w} \in \mathbb{R}^n$ is the normal vector to this hyperplane, the scalar $b \in \mathbb{R}$ is defined as bias, $\langle \cdot, \cdot \rangle$ is the inner product operator and $(\cdot)^T$ is the transpose operator.

In the case of SVR, there are different algorithms such as ε-SVR [19], ν-SVR [22] and ε-Bayesian SVR [23]. In this paper, we consider the ε-SVR algorithm, whose purpose is to find a function that has at most ε deviation from the real target y_i for all training data. Thus, the ε-SVR algorithm does not accept any deviation larger than ε. Therefore, one can use Vapnik's linear loss function with ε-insensitivity zone defined as [19]

$$E(e_i) = |e_i|_\varepsilon = \begin{cases} 0, & \text{if } |e_i| \le \varepsilon \\ |e_i| - \varepsilon, & \text{otherwise} \end{cases}, \tag{2}$$

where $e_i = y_i - h(\mathbf{x}_i, \mathbf{w})$. Fig. 2(a) shows the Vapnik's linear loss function $E(e_i)$, where the ε-insensitivity zone is highlighted. Hence, $E(e_i)$ is zero when the difference between the predicted $h(\mathbf{x}_i, \mathbf{w})$ and the measured value y_i is less than ε.

The solution of the linear regression learning problem is to find the linear function that approximates the training pairs $(\mathbf{x}_i, y_i)$ with an error smaller than ε. In other words, we need to find a vector $\mathbf{w}$ that minimizes the error, and at the same time it, is as flat as possible. Flatness, in this case, means that one seeks for small vectors $\mathbf{w}$ to avoid over-fit [24]. This requirement can be ensured by minimizing the norm of $\mathbf{w}$, which implies to solve the optimization problem given by [25]

$$\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2 \tag{3}$$

restricted to $|e_i| \le \varepsilon$.

A penalty term is included in (3) to obtain sparse solutions and penalize large residuals, so that

$$\min_{\mathbf{w},b} \left[\frac{1}{2}\|\mathbf{w}\|^2 + C\left(\sum_{i=1}^{\ell} E(e_i) \right) \right] \tag{4}$$

where C is a cost parameter establishing the trade-off between the flatness of $h(\mathbf{x}_i, \mathbf{w})$ and the amount up to which deviations larger then ε are tolerated [24]. The function $E(e_i)$ defines an ε-tube as illustrated in Fig. 2(b), where ε is the radius of the tube. The restriction $|e_i| \le \varepsilon$, i.e., $y_i + \varepsilon \ge h(\mathbf{x}_i, \mathbf{w}) \ge y_i - \varepsilon$ is the condition for a predicted point to be within in the ε-tube.

In (4), we assume that the function exists and it approximates all pairs $(\mathbf{x}_i, y_i)$ with accuracy ε. However, sometimes this may not be the case. Therefore, to ensure the existence of such function, we need to allow some errors greater than ε. To cope with otherwise infeasible restriction imposed by (4), Vapnik [19] adapted the *soft margin* loss function presented in [26] to SVR.

Thus, the optimization restriction represented by (4) can be relaxed by introducing slack variables, defined as ξ

and $\hat{\xi}$, which allow to deal with points outside the ε-tube. The points below the ε-tube have $\xi = 0$ and $\hat{\xi} > 0$, while the points above the ε-tube have $\xi > 0$ and $\hat{\xi} = 0$. At last, the points inside of ε-tube have $\xi = \hat{\xi} = 0$.

With the slack variables ξ and $\hat{\xi}$, we can redefine the optimization problem as

$$\min_{\mathbf{w},b} \left[\frac{1}{2}\|\mathbf{w}\|^2 + C\left(\sum_{i=1}^{\ell} (\xi_i + \hat{\xi}_i) \right) \right] \tag{5}$$

under the restrictions

$$\begin{cases} |e_i| = \varepsilon + \xi \\ |e_i| = \varepsilon + \hat{\xi}, \\ \xi, \hat{\xi} \geq 0 \end{cases}$$

which can be solved with Lagrange multipliers, as can be found in [25]. After calculating the Lagrange multiplier vectors $\boldsymbol{\alpha}$ and $\boldsymbol{\alpha}^*$, the best regression hyperplane obtained is given by

$$h(\mathbf{x}_i, \mathbf{w}) = \sum_{j=1}^{\ell} (\boldsymbol{\alpha} - \boldsymbol{\alpha}^*) \langle \mathbf{x}_j, \mathbf{x}_i \rangle + b. \tag{6}$$

In the case of nonlinear regression, the basic idea is to map the input vectors $\mathbf{x}_i \in \mathbb{R}^n$ into vectors $\Phi(\mathbf{x}_i)$ of a higher dimensional feature space $\mathfrak{I}$, where Φ represents the mapping. After this transformation, a nonlinear problem in $\mathbb{R}^n$ becomes a linear problem in the feature space $\mathfrak{I}$. So, the optimization problem can be reformulated as the maximization of dual Lagrangian with Hessian matrix [27] and the solution is given by

$$h(\mathbf{x}_i, \mathbf{w}) = \sum_{j=1}^{\ell} (\boldsymbol{\alpha} - \boldsymbol{\alpha}^*) \langle \Phi(\mathbf{x}_j), \Phi(\mathbf{x}_i) \rangle + b, \tag{7}$$

in which the summation is not performed over all training data, but rather over those that have non-zero Lagrange multipliers, which are called *support vectors*.

Moreover, note that the optimization problem for nonlinear regression, represented by (7), involves the calculation of inner products between vectors of the feature space $\mathfrak{I}$. In this approach, the feature space $\mathfrak{I}$ has a very high dimensionality, hence the calculation of Φ can become infeasible. The solution is to resort to the *kernel trick*, i.e., the use of kernels to perform nonlinear regressions without mapping all input vectors $\mathbf{x}_i$ to the feature space $\mathfrak{I}$ [28].

A kernel is a similarity function that takes two vectors $\mathbf{x}_i$ and $\mathbf{x}_j$ in the input space X and returns the inner product of these vectors in the feature space $\mathfrak{I}$ [29], i.e.,

$$K(\mathbf{x}_i, \mathbf{x}_j) = \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) \rangle. \tag{8}$$

To ensure the convexity of the optimization problem given by (7) and that the kernel represents mappings in which it is possible the calculation of the inner products $\langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) \rangle$, kernel functions satisfying the conditions of Mercer are exploited [19]. The most common practical kernels for regression problems are the polynomial kernel and the radial basis functions (RBF) ones. For such regression problem, we use the Laplacian RBF kernel due to its performance showed in [30] for this kind of regression problem.

Table 1
Settings of each BTS in the measurement setup.

Base transceiver station	Elevation (m)	Height (m)
BTS-1	8	41
BTS-2	6	53
BTS-3	8	40

The expression related to Laplacian RBF kernel is given by

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\left\| \mathbf{x}_i - \mathbf{x}_j \right\|}{\sigma} \right) \tag{9}$$

where σ is a tuning parameter defined by the user, which is common to all radial RBF kernels.

3. Algorithm description

In this section, we propose an FP SVR-based algorithm to develop models of the relationship between RSSI measurements and the position of the mobile terminal. The proposed algorithm can be implemented in six steps, which are described in Algorithm 1.

Algorithm 1 Description of the FP SVR-based algorithm.

1: Collect the scanner measurements.
2: Train SVR to predict path loss (one for each BTS).
3: Build the coverage map (CDB).
4: Collect RSSI measurements and timing advance from the sought mobile to the three BTS.
5: Apply CDB filtering by timing advance to reduce the search area.
6: Find the nearest point on the search space.

In the first step, we assume mobile radio wave propagation measurements at 1.8 GHz Global System for Mobile Communications (GSM) frequency band. A drive test was performed to obtain downlink signal strength measurements in an urban environment in the city of Recife-PE, Brazil. Fig. 3 illustrates the urban area of the city where the measurements were taken. In total, 2547 measurements were captured using NEMO FSR1[2] tool as a GSM pilot scanner.

Various field data of each measured point were collected to compose the feature vector of the SVR process. Such field data were antenna-separation distance, terrain elevation, horizontal angle, vertical angle, latitude, longitude, horizontal, and vertical attenuation of the antenna. In addition, the theoretical path loss of the Okumura-Hata model [31] was used as an input of the SVR training algorithm. The terrain elevation was collected using Google Elevation API by a Java client made exclusively for this application. Table 1 illustrates the terrain elevation and the height for three base stations (BTS), namely BTS-1, BTS-2 and BTS-3. The half-power angle and the equivalent

[2] NEMO FSR1 is a modular digital scanning receiver providing accurate RF signal measurements of wireless networks.

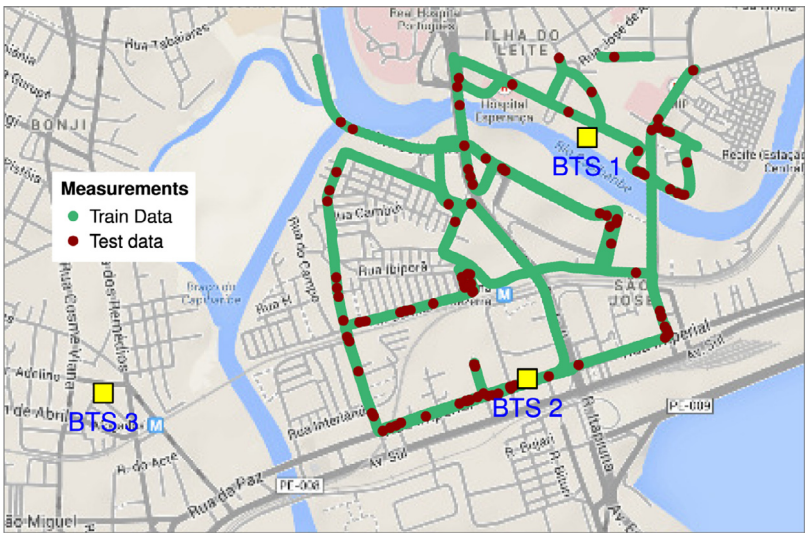

Fig. 3. Urban environment of the city of Recife-PE, Brazil with the indication of the training measurements, testing measurements, and locations of the three BTS.

isotropically radiated power (EIRP) for all BTS are 63 ◦ and 70.1 dBm, respectively. The locations of the three BTS are also indicated in Fig. 3.

Many ML algorithms, such ε-SVR, have important parameters that cannot be set directly from the data. The process of setting these parameters to obtain the best performance of the model is known as tuning. In other words, we train the FP SVR-based algorithm varying the parameters to find the best fit model, characterizing the second step of Algorithm 1. For instance, Eq. (9) has σ as a specific parameter for the Laplacian RBF kernel. The definition of σ follows the analytical approach presented in [32], where it is shown that the optimal values of σ are in the range of the 10th and the 90th percentile of $\|\mathbf{x}_i - \mathbf{x}_j\|^2$. In addition to that, it is suggested in [28] that the midpoint of these two percentiles should be used.

The cost C, common parameter to all kernels, is fundamental for adjusting the complexity of the model. When the cost is large, the model is more flexible, but it becomes more likely to over-fit. With a small cost, the flexibility of the model decreases, but the over-fit is less likely. However, a small cost can lead to poor predictions due to under-fit [33]. In the model tuning process, 18 values were tested for C, from 2^{-2} to 2^{15}, being each value a power of 2.

The validation technique denominated *cross-validation* is applied in order to find the best fit. A common type of this technique is the k-fold cross-validation, generally used to evaluate the model accuracy [33]. It is a re-sampling technique in which the samples are randomly split into k sets of approximately equal size. These subsets are named folds and they are divided into two groups: the test set with only one fold and the training set with $(k-1)$ folds. The pseudo-code of k-fold cross-validation can be seen in Algorithm 2.

In this paper, we consider $k = 10$ and use the average root mean square error (RMSE) $\bar{\mu}$ and μ_σ, its standard deviation, as cross-validation metrics for choosing the best model. The average RMSE $\bar{\mu}$ is defined as

Algorithm 2 Description of k-fold cross-validation tuning process.

1: Partition data into k disjoint sets T_1 ...T_k
2: **for** $i = 1$ to k **do**
3: Use T_i for validation and the remaining for training.
4: Calculate error on validation set T_i.
5: **end for**
6: Return average error on validation sets.

Table 2
Results of the training stage of each SVR using 10-fold cross-validation.

BTS	σ	C	ε	$\bar{\mu}$ (dB)	μ_σ
BTS-1	0.258	16	0.1	3.66	0.271
BTS-2	0.207	32	0.1	3.83	0.240
BTS-3	0.215	16	0.1	3.49	0.259

$$\bar{\mu} = \sqrt{\frac{1}{k}\sum_{j=1}^{k}\mu_j} \tag{10}$$

in which μ_j is the RMSE calculated for the jth test set ($j = 1, 2, \ldots, k$), that is given by

$$\mu_j = \sqrt{\frac{1}{\ell_j}\sum_{i=1}^{\ell_j}(y_i - h(\mathbf{x}_i, \mathbf{w}))^2}, \tag{11}$$

where ℓ_j is the number of samples in the jth test set. In the training stage, it was assumed $\varepsilon = 0.1$ and $\varepsilon = 0.05$ in combination with the range of C previously specified. Fig. 4 shows the 10-cross validated RMSE $\bar{\mu}$ over the training dataset, where we can find the best values of C for each BTS. Table 2 shows the optimized values of SVR parameters

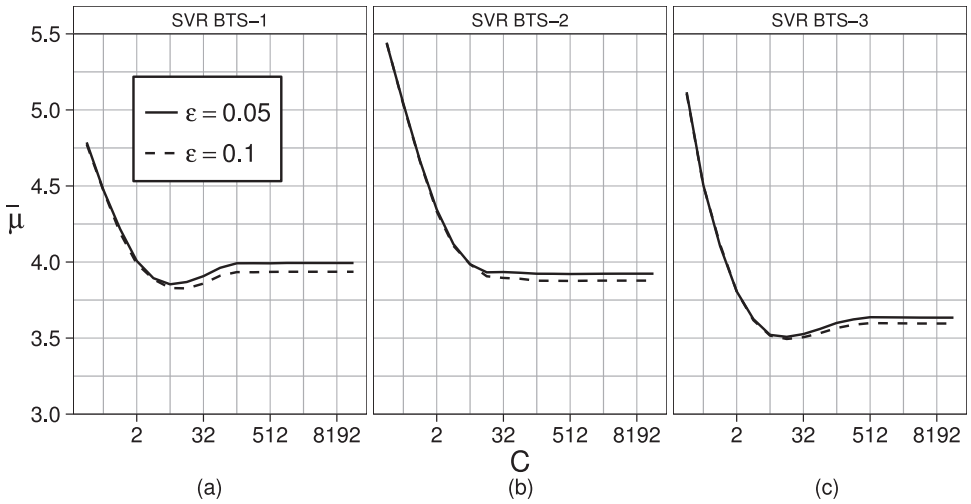

Fig. 4. The 10-fold cross-validated RMSE in function of C and ε: (a) BTS-1. (b) BTS-2. (c) BTS-3.

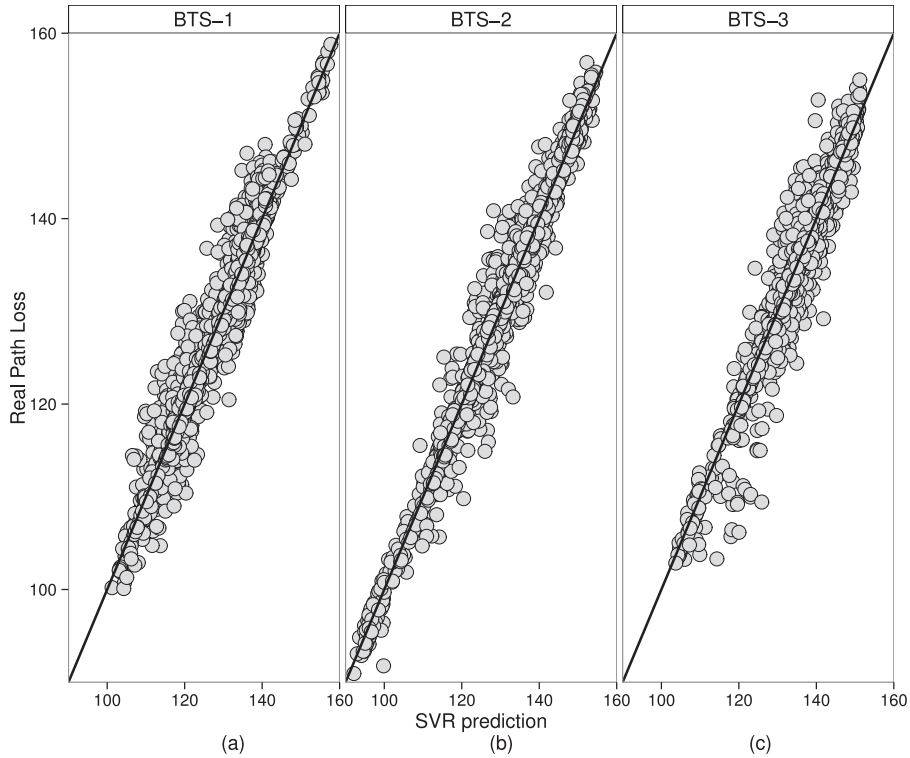

Fig. 5. Predicted path loss by the FP SVR-based algorithm versus real path loss for training dataset: (a) BTS-1. (b) BTS-2. (c) BTS-3.

for the best fit, as well as the values of $\bar{\mu}$ and μ_σ obtained by the 10-fold cross-validation tuning process.

Given the tuning process, each FP SVR-based algorithm is set with the parameters specified in Table 2. Fig. 5 illustrates the predicted path loss versus the real path loss for each BTS, considering the training dataset. If a gray point is exactly in the diagonal line, it means that the predicted path loss is equal to the real path loss. In other words, the error is zero for these points.

After the training stage of the SVR for each BTS, the next step is to plot the coverage map (CDB, as defined in Section 2) for the proposed algorithm. To do this, a localization area should be defined. In this paper, a localization area of 1.38 km $\times$ 1.38 km with a grid resolution at 20 m $\times$ 20 m is considered and it can be seen in Fig. 6. In addition, the expected timing advance (TA), one for each BTS, is set for each position on the coverage map. TA corresponds to a propagation delay of the transmitted signal, which is expressed as an integer number from 0 to 63. Each value corresponds to a distance between the MS and the BTS, in steps of 550 m, that is, $0 = 0$ m, $1 = 550$ m, $\ldots$, $63 = 34,650$ m [17].

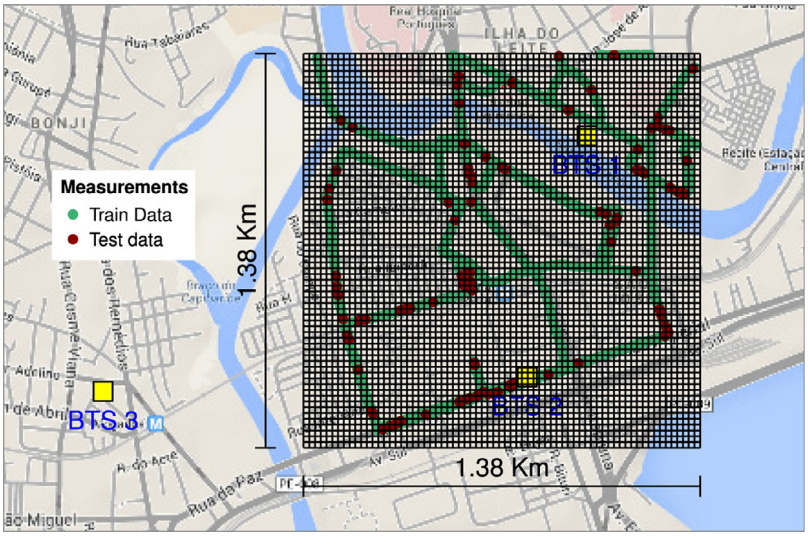

Fig. 6. Localization area of 1.38 km × 1.38 km with a grid of resolution 20 m × 20 m.

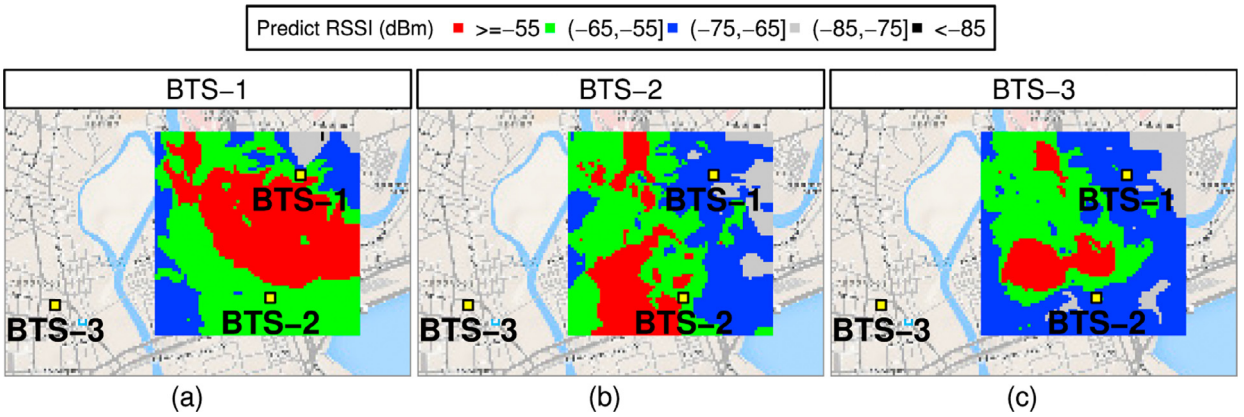

Fig. 7. Coverage maps obtained from the predictions of the FP SVR-based algorithm: (a) BTS-1. (b) BTS-2. (c) BTS-3.

Considering that the localization area is a grid of q positions, the coverage map can be defined as a data set $S = \{(\mathbf{p}_i, \mathbf{s}_i, \mathbf{t}_i) \in \mathbb{R}^2 \times \mathbb{R}^3 \times \mathbb{R}^3, i = 1, 2, \ldots, q\}$, where $\mathbf{p}_i = [p_i^{(1)}, p_i^{(2)}]$ is the position vector for the ith square 20 m × 20 m in the grid, being $p_i^{(1)}$ the longitude and $p_i^{(2)}$ the latitude of the center of the ith square, while $\mathbf{s}_i = [s_i^{(1)}, s_i^{(2)}, s_i^{(3)}]$ represents the SVR predictions of RSSI for BTS-1, BTS-2, and BTS-3, respectively. At last, $\mathbf{t}_i = [t_i^{(1)}, t_i^{(2)}, t_i^{(3)}]$ is the expected TA, for each BTS, in the center of ith square in the grid. Thus, the trained SVR of each BTS is used to predict RSSI for all position vectors $\mathbf{p}_i$ in the location grid. In Fig. 7, we can see the coverage map for each BTS.

Given the coverage map S, the fourth step is to measure RSSI and TA of the sought mobile from BTS-1, BTS-2 and BTS-3. Consider $\mathbf{m} = [m_1, m_2, m_3]$ being a vector representing RSSI measurements, while $\mathbf{a} = [a_1, a_2, a_3]$ is the measured TA vector. After taking the measurements, the fifth step is to apply a CDB filtering by TA method to reduce the search area. The filter is defined in Algorithm 3, which has the measured TA vector $\mathbf{a}$ as input and a reduced coverage map S_R, a subset of S, as output. The key idea of Algorithm 3 is to maximize the number of matches between the expected and measured TAs. To do that, Algorithm 3 works as follows. First, it tries to get all points with three matches (line 3), which is the maximum (one for each BTS). Second, if there is no matches, it goes for points with two matches (from line 7 to 12). Next, if S_R is still empty, it tries to get points with at least one match (from line 13 to 15). Lastly, if S_R remains empty, then the entire map is returned.

Given the reduced coverage map S_R, the last step of the FP SVR-based algorithm is to estimate the mobile position using Euclidean distance as similarity function. Let d_i' be the Euclidean distance between $\mathbf{m}$ and $\mathbf{s}_i$ for the ith position in the reduced coverage map S_R, which can be expressed by

$$d_i' = \sqrt{(s_i^{(1)} - m_1)^2 + (s_i^{(2)} - m_2)^2 + (s_i^{(3)} - m_3)^2}, \qquad (12)$$

$i = 1, 2, \ldots, q$. The best estimated position $\mathbf{p}_i$ is that whose vector $\mathbf{s}_i$ has the smallest Euclidean distance d_i'.

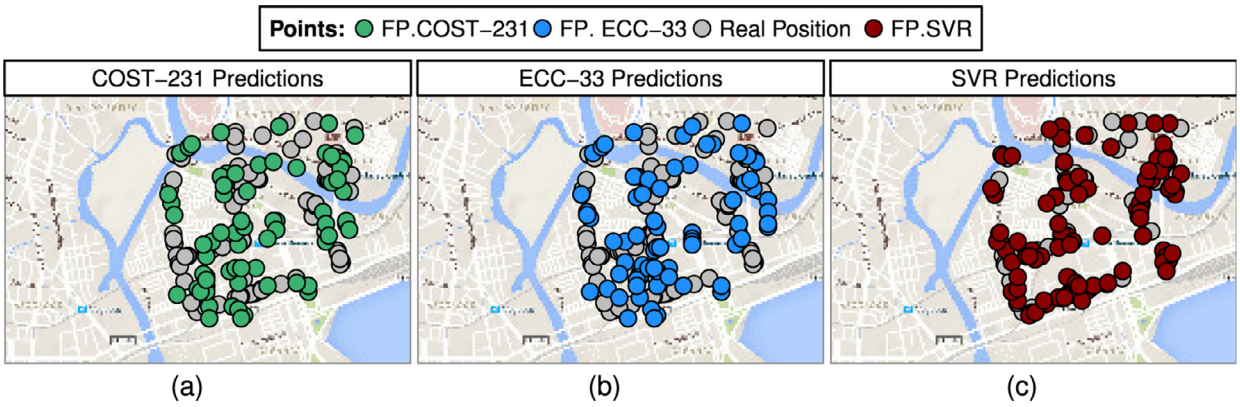

Fig. 8. Prediction maps for each location method: (a) FP COST-231. (b) FP ECC-33. (c) FP SVR-based approach. (For interpretation of the references to colour in the text, the reader is referred to the web version of this article.)

Algorithm 3 Filter to reduce the search area in the coverage map.

```
1: function FILTER(a1,a2,a3)
2:     S ← Coverage map
3:     S_R ← query (Select * from S where a1 = t1 and a2 = t2 and a3 = t3)
4:     if S_R is empty then
5:         S_R ← query (Select * from S where a1 = t1 and a2 = t2)
6:     end if
7:     if S_R is empty then
8:         S_R ← query (Select * from S where a1 = t1 and a3 = t3)
9:     end if
10:    if S_R is empty then
11:        S_R ← query (Select * from S where a2 = t2 and a3 = t3)
12:    end if
13:    if S_R is empty then
14:        S_R ← query (Select * from S where a1 = t1 or a2 = t2 or a3 = t3)
15:    end if
16:    if S_R is empty then
17:        S_R ← S
18:    end if
19:    return S_R
```

4. Numerical results

The performance of the three localization methods (FP with ECC-33 model, FP with COST-231 model and FP SVR-based approach) is evaluated via computer simulations using a test position dataset with 100 observation points. The training data set is employed to define the multislope parameters, used to optimize COST-231 and ECC-33 models, as shown in [17]. The FP SVR-based algorithm is implemented using R language, with emphasis on kernlab [28] and caret packages [33]. To compare the localization methods previously mentioned, we define the distance prediction error η as the difference (in meters) between the real and the predicted points.

Table 3
Statistical analysis of the distance prediction errors for the three localization methods.

Loc. method	$\bar{\eta}$ (m)	η_σ (m)	$\eta_{\max}$ (m)	$\eta_{\min}$ (m)
FP SVR	54.0	56.8	383.3	0.4
FP COST-231	211.9	142.1	807.9	16.2
FP ECC-33	226.3	150.1	807.9	16.2

Table 3 provides a statistical analysis of the distance prediction errors for each localization method. The average distance prediction error is represented by $\bar{\eta}$, its standard deviation η_σ, and the maximum and minimum errors η_{max} and η_{min}, respectively. Also in Table 3, we can see that the FP SVR-based algorithm presents an average error $\bar{\eta} = 54.0$ m, while the other FP methods present $\bar{\eta} = 211.9$ m (FP COST-231), and $\bar{\eta} = 226.3$ m (FP ECC-33).

In order to compare the average errors above, prediction maps can be created for each localization method as it is illustrated in Fig. 8. To obtain each map, we should distribute the test points collected in the field and overlap them to those evaluated by the localization methods. For all maps, the positions related to the measured levels of the radio signals collected in the field are represented by gray dots.

Fig. 8 (a) shows the distribution of points acquired from the FP COST-231 model, where the estimated positions correspond to the green dots. A concentration of green dots can be observed, but there are no overlapping gray ones across the region. In Fig. 8(b), the estimated positions are related to the FP ECC-33 model and they are represented by the blue dots. As in the previous case (FP COST-231), there are real positions (gray dots) that are not overlapped by the estimated positions (blue dots). Furthermore, Fig. 8(c) indicates the estimated positions given by the FP SVR-based algorithm using red dots. By observing the convergence of the estimated positions and the real ones (gray dots), we can conclude that the FP SVR-based localization method is more accurate than FP COST-231 and FP ECC-33 methods.

Another way to compare the localization methods addressed in this paper is by using histograms. Fig. 9 shows one histogram for each localization technique considering

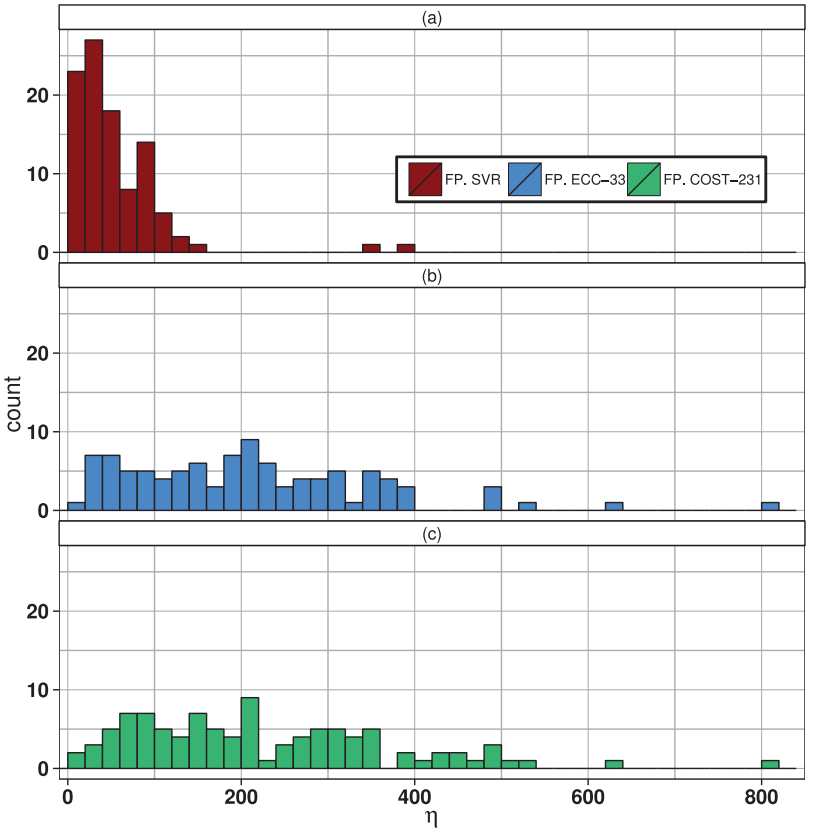

Fig. 9. Histograms of the average distance prediction error (in meters) for test position dataset: (a) FP SVR-based approach. (b) FP ECC-33. (c) FP COST-231.

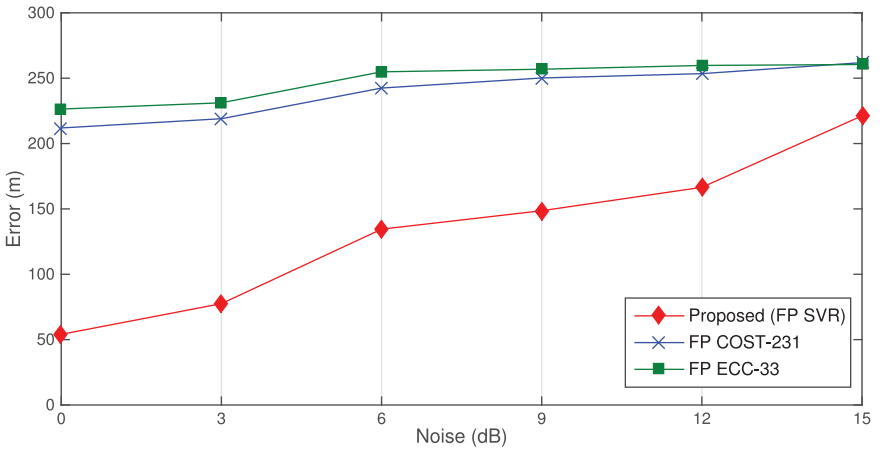

Fig. 10. Average error location for outliers due to presence of noise (with Rayleigh distribution) in the measurements from the sought mobile.

the test position dataset, in which the x-axis represents average distance prediction error $\bar{\eta}$, while the y-axis corresponds to the count of samples that have the same $\bar{\eta}$. By analyzing the three histograms, it can be verified that the FP SVR-based algorithm is the best method, given that most of its samples is concentrated at the beginning of the histogram and their values are less than 200 m. For the other two methods, we can see that the errors are more evenly distributed across the range between 0 and 600 m.

In order to verify the behavior of the proposed approach in the presence of outliers, we used Rayleigh distribution to simulate different levels of noise in the measurements from the sought mobile. Fig. 10 presents the average error location for all location methods in the presence of noise with the following mean values: 3, 6, 9, 12, and 15 dB. We can see that, at all levels of noise, the accuracy of the proposed method (FP SVR) overcomes the other two approaches (FP COST-231 and FP ECC-33).

Table 4
Average computational cost for each location method.

Task/loc. method	FP SVR	FP COST-231	FP ECC-33
Model training	80.90	1.00	1.00
Coverage map building	1.00	1.20	1.25
Searching for 100 mobile terminals	1.00	1.00	1.00

Table 5
Pairwise comparisons using Nemenyi post-hoc test.

Technique 1	Technique 2	p-value
FP SVR	FP COST-231	0.00022
FP SVR	FP ECC-33	3.8×10^{-6}
FP COST-231	FP ECC-33	0.65

Another important aspect to compare the methods used is the computational cost. For this purpose, we established three tasks to measure the time consumed: model training, coverage map building, and searching for 100 mobile terminals. For all location methods, it was assumed the use of the same hardware. Table 4 shows average computational cost for each task (rows) in each location method (columns). For each task, the lowest time consumed was made equal to one (reference value) and the other times were compared to it. For example, considering the model training task, the time consumed by FP SVR-based method is almost 81 times larger than the time consumed by the other methods. For the second task (building the coverage map), the FP SVR-based approach obtained a slightly better performance. Finally, in the task of searching for 100 mobile terminals, all methods have similar performance.

Lastly, to verify if the differences between the performances of the positioning techniques are statistically relevant, Friedman and Nemenyi post-hoc tests are applied [34]. Friedman test is a non-parametric statistical test with two hypothesis: H_0 (null-hypothesis) and H_1 (alternative hypothesis). The hypothesis H_0 is that all techniques are equivalent, thus it is rejected if there are some difference among the approaches. If the hypothesis H_0 is rejected, we can proceed with the Nemenyi post-hoc test.

In the Friedman test, the hypothesis H_0 can be rejected if the p-value is less than the confidence level α [34]. The p-value is defined as the probability of rejecting the hypothesis H_0 when it is true. In this work, the Friedman test was performed over the test dataset for a confidence level $\alpha = 0.05$, and resulted in $p = 3.34 \times 10^{-7}$. Thus, H_0 can be rejected because $p < <\alpha$, which means that at least two techniques differ.

After multiple comparisons made by the Friedman test, the Nemenyi post-hoc test is used to make a pairwise comparison. Table 5 shows the p-value obtained for each pairwise comparison in the Nemenyi post-hoc test. We see that the FP SVR-based technique differs from the other two, because the p-values are smaller than the confidence level ($\alpha = 0.05$). On the other hand, FP COST-231 and FP ECC-33 cannot be considered different, because the p-value is greater than the confidence level α.

5. Conclusions

In this study, a fingerprinting location algorithm using support vector regression was proposed to estimate the position of mobile terminals in a cellular network. In addition, fingerprinting location methods with COST-231 and ECC-33 propagation models were used as reference for comparison. In all methods, mobile radio wave propagation measurements at a carrier frequency of 1.8 GHz GSM were obtained in an urban environment in the city of Recife-PE, Brazil. Some field data, like antenna-separation distance, terrain elevation, and the theoretical path loss of the Okumura-Hata model were used as input of the SVR training algorithm, while the Laplacian kernel was adopted. In spite of the increased computational cost in the model training procedure, numerical results, represented by statistical analysis, prediction maps and histograms, showed that the fingerprinting SVR-based approach had a lower error distance prediction and was less sensitive to a Rayleigh distributed noise than the other fingerprinting techniques. A work is in progress to investigate whether a combination of SVRs can improve the target positioning, not only in cellular networks, but also in vehicular networks.

References

[1] D. Miorandi, S. Sicari, I. Chlamtac, Internet of Things: vision, applications and research challenges, Ad Hoc Networks 10 (7) (2012) 1497–1516.
[2] R. Barnes, J. Winterbottom, M. Dawson, Internet geolocation and location-based services, IEEE Commun. Mag. 49 (4) (2011) 102–108.
[3] M.B. Kjaergaard, Location-based services on mobile phones: minimizing power consumption, IEEE Pervasive Comput. 11 (1) (2012) 67–73.
[4] L.M. Ni, Y. Liu, Y.C. Lau, A.P. Patil, LANDMARC: indoor location sensing using active RFID, Wireless Networks 10 (6) (2004) 701–710.
[5] N. Bulusu, J. Heidermann, D. Estrin, GPS-less low cost outdoor localization for very small devices, IEEE Personal Commun. 7 (5) (2000) 28–34.
[6] H. Liu, H. Darabi, P. Banerjee, J. Liu, Survey of wireless indoor positioning techniques and systems, IEEE Trans. Syst. Man Cybern. Part C: Appl. Rev. 37 (6) (2007) 1067–1080.
[7] M. Brunato, R. Battiti, Statistical learning theory for location fingerprinting in wireless LANs, Comput. Networks 47 (6) (2005) 825–845.
[8] C.L. Wu, L.C. Fu, F.L. Lian, WLAN location determination in e-home via support vector classification, Int. Conf. Network Sens. Control 2 (2004) 1026–1031.
[9] J. Lee, B. Choi, E. Kim, Novel range-free localization based on multidimensional support vector regression trained in the primal space, IEEE Trans. Neural Networks Learn. Syst. 24 (7) (2013) 1099–1113.
[10] M.P. Wylie, J. Holtzman, The non-line of sight problem in mobile location estimation, in: Proceedings of the Fifth IEEE International Conference on Universal Personal Communication, Cambridge—MA, USA, 1996, pp. 827–831.
[11] P.C. Chen, A non-line-of-sight error mitigation algorithm in location estimation, in: Proceedings of the IEEE Wireless Communication and Networking Conference, New Orleans—LA, USA, 1999, pp. 316–320.
[12] M. Wylie-Green, S. Wang, Robust range estimation in the presence of the non-line-of-sight error, in: Proceedings of the IEEE Vehicular Tech. Conf. Fall 2001, Atlantic City—NJ, USA, 2001, pp. 101–105.
[13] G. Sun, W. Guo, Robust mobile geo-location algorithm based on LS-SVM, IEEE Trans. Veh. Technol. 54 (3) (2005) 1037–1041.
[14] H. Zou, X. Lu, H. Jiang, L. Xie, A fast and precise indoor localization algorithm based on an online sequential extreme learning machine, Sensors 15 (2015) 1804–1824.
[15] J. Benikovsky, P. Brida, J. Machaj, RF fingerprinting location techniques, Handbook of Position Location: Theory, Practice, and Advances, Wiley Online Library, 2011, pp. 487–520.
[16] R.S. Campos, L. Lovisolo, Localization in Real GSM network with fingerprinting utilization, Mobile Lightweight Wireless Systems, Springer, Berlin, Heidelberg, 2010, pp. 699–709.

[17] L. Klozar, P. Jan, Wireless network localization: optimization processing, in: Proceedings of the Seventh International Conference on Digital Telecommunications (ICDT 2012), Chamonix, FRA, 2012, pp. 45–49.
[18] European Telecommunications Standard Institute, ETSI TS 101724 v8.9.0 (2004 06) Digital Telecommunications System (Phase 2 +); Location Services (LCS); Functional Description; Stage 2 (3GPP TS 03.71 version 8.9.0 Release 1999), 2004.
[19] V.N. Vapnik, The Nature of Statistical Learning Theory, Springer-Verlag, New York, 1995.
[20] V. Vapnik, S.E. Golowich, A. Smola, Support vector method for function approximation, regression estimation, and signal processing, in: Advances in Neural Information Processing Systems, 1997, pp. 281–287.
[21] T.B. Trafalis, H. Ince, Support vector machine for regression and applications to financial forecasting, in: Neural Networks, IEEE-INNS-ENNS International Joint Conference, 2000, p. 6348.
[22] A.J. Smola, Regression Estimation with Support Vector Learning Machines, in: Master's dissertation, Technische Universit at Munchen, 1996.
[23] J. Gao, S. Gunn, C. Harris, M. Brown, A probabilistic framework for SVM regression and error bar estimation, Mach. Learn. 46 (2002) 71–89.
[24] A. Smola, B. Schölkopf, A tutorial on support vector regression, Stat. Comput. 14 (3) (2004) 199–222.
[25] C. Bishop, Pattern Recognition and Machine Learning, Springer, New York, 2006.
[26] K.P. Bennett, O.L. Mangasarian, Robust linear programming discrimination of two linearly inseparable sets, Optim. Methods Softw. 1 (1) (1992) 23–24.
[27] V. Kecman, Support vector machines: an introduction, Support Vector Machines: Theory and Applications 47 (2005) 1–47.
[28] A. Karatzoglou, A. Smola, K. Hornik, A. Zeilis, Kernlab: an S4 package for kernel methods in R, J. Stat. Softw. 11 (9) (2004) 1–20.
[29] R. Herbrich, Learning Kernel Classifiers: Theory and Algorithms, MIT Press, 2001.
[30] R.D.A. Timoteo, D.C. Cunha, G.D.C. Cavalcanti, A proposal for path loss prediction in urban environments using support vector regression, in: Proceedings of the 10th Advanced International Conference on Telecommunications (AICT 2014), Paris, FRA, 2014, pp. 1–5.
[31] M. Hata, Empirical formula for propagation loss in land mobile radio services, IEEE Trans. Vehic. Technol. 29 (3) (1980) 317–325.
[32] A.J. Smola, B. Caputo, K. Sim, F. Furesjo, Appearance-based object recognition using SVMs: which kernel should I use? Proceedings of the NIPS Workshop on Statist. Methods for Comput. Exp. in Visual Process. and Comp. Vision, Whistler, CAN, 2002.
[33] M. Kuhn, K. Johnson, Applied Predictive Modeling, Springer, New York, 2013.
[34] J. Demšar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (2006) 1–30.

Robson D. A. Timoteo received the B.Sc. degree in electrical engineering from University of Pernambuco (2000) and the M.Sc. degree in computer science (2014) from Federal University of Pernambuco, all in Recife-PE, Brazil. He worked as radio frequency (RF) engineer and senior RF engineer at BCP—Bell South Brazil (2000–2004), and Claro (2004–2007), respectively. He is currently a Ph.D. candidate in the Graduate Program in Computer Science at Centro de Informática of the Federal University of Pernambuco (CIn-UFPE), Recife-PE, Brazil. His current research interests are in mobile positioning, cellular networks, and machine learning techniques.

Lizandro N. Silva received the B.Sc. degree in electronic engineering and the M.Sc. degree in systems engineering, both from the University of Pernambuco (UPE) in 2007 and 2013, respectively. He is currently a Ph.D. candidate in the Graduate Program in Computer Science at Centro de Informática of the Federal University of Pernambuco (CIn-UFPE), Recife-PE, Brazil. He is also with Telefonica-VIVO S/A as a telecommunications engineer since 2008. His current research interests are in Internet of Things, radio frequency identification, e-healthcare, location-based services, and mobile networks.

Daniel C. Cunha received the M.Sc. and Ph.D. degrees in electrical engineering from the State University of Campinas (UNICAMP), in 2003 and 2006, respectively. He is currently an associate professor of Computer Science and Computer Engineering at the Centro de Informática of the Federal University of Pernambuco (CIn/UFPE), Recife-PE, Brazil. Formerly, he was an associate professor of Telecommunications Engineering at University of Pernambuco (UPE), Recife-PE, from 2007 to 2012. His contributions are in the area of error control coding, cooperative communications and machine learning techniques applied to telecommunications systems. His current research interests are in mobile positioning systems and power consumption of mobile devices.

George D.C. Cavalcanti received the D.Sc. degree in Computer Science from Centro de Informática, Federal University of Pernambuco, Brazil. He is currently an associate professor with the Centro de Informática, Federal University of Pernambuco, Brazil. His research interests include machine learning, pattern recognition, computer vision, and biometrics.