A Proposal for Path Loss Prediction in Urban Environments using Support Vector Regression

Robson D. A. Timoteo, Daniel C. Cunha
CISG - Communication and Info. Systems Research Group
Centro de Informática - UFPE
Recife - PE - Brazil
rdat@cin.ufpe.br, dcunha@cin.ufpe.br

George D. C. Cavalcanti
VIISAR - Vision and Artificial Intelligence Research Group
Centro de Informática - UFPE
Recife - PE - Brazil
gdcc@cin.ufpe.br

***Abstract*—In the last few years, the mobile data traffic has grown exponentially making evident the importance of wireless networks. To ensure an acceptable level of quality of service for users in a wireless data network, network designers rely on signal propagation path loss models. To provide adaptability, the use of machine learning techniques has been considered to predict characteristics of the wireless channel. In this work, we propose a method for predicting path loss in an urban outdoor environment using support vector regression. Simulation results indicate that, depending on the employed kernel and its parameters, the performance obtained using support vector regression is similar and with lower computational complexity to that obtained by a multilayer perceptron neural network.**

Keywords*—wireless networks, propagation models, machine learning, nonlinear regression.

I. Introduction

Today, being connected is crucially important. High-speed internet access via mobile handsets is the most likely way of achieving digital inclusion [1]. In this context, the importance of wireless and mobile networks grows every day and their demand is also becoming larger even faster.

To ensure an acceptable level of quality of service for users in a wireless data network, network designers rely on signal propagation path loss models. Radio wave propagation models are a series of mathematical calculation developed to predict path characteristics and losses in a given environment [2]. For example, propagation models have traditionally focused on predicting the average received signal strength at a given distance from the transmitter, plus the variability in the signal intensity near a particular location area. Thus, propagation models are mathematical tools used by engineers and scientists to plan and optimize wireless network systems.

Given the problem context, many researchers have turned their attention to the domain of machine learning (ML) [3]. The goal of this class of algorithms is to automatically learn the properties of the environment and to adapt their behavior quickly and easily to them. Artificial neural networks (ANN) are a typical example of a ML algorithm, inspired by the biological neural networks of the brain [4]. In recent time, multilayer perceptron (MLP)-ANN have been shown to successfully perform path loss in rural, urban, suburban and indoor environments [5], [6], [7]. However, a drawback in using MLP-ANN is the required training time to process data, considering the numerous neurons in each layer of the neural network. To handle it, other ML techniques can be used, such as support vector machine (SVM). The main advantages of using SVM are the absence of local minima, the sparseness of the solution and the capacity control obtained by optimising the margin [8]. Aside from that, to the best of our knowledge, there is no similar approach in the literature that considers the use of SVM to perform path loss prediction. Thus, in this work we propose a method for predicting path loss in an urban outdoor environment using support vector regression (SVR). Simulation results indicate that, depending on the employed kernel and its parameters, the performance obtained using support vector regression is similar and with lower computational complexity to that obtained by a MLP neural network.

The remainder of this article is structured as follows. In Section II, two empirical propagation models are presented: Okumura-Hata model and Ericsson 9999 model. Concepts about support vector regression and the measurement setup are described in Section III. Section IV presents the model tuning of the SVR techniques and numerical results. At last, conclusions are drawn in Section V.

II. Empirical Propagation Models

Reliable and accurate propagation models are crucial to the prediction of radio channel characteristics for where the wireless network system is to be deployed. In general, propagation models can be categorised into two types: deterministic and empirical.

Deterministic propagation models consider the physical paths along which the transmitted waves propagate are usually based on ray optical techniques [9]. These models describe wave propagation by different rays that travel from the transmitting to the receiving antenna and are subjected to reflection, scattering and diffraction at walls and edges of buildings and similar objects. Deterministic models offer excellent accuracy and are able to provide additional parameters such as small-scale fading, delay spread, etc. The main disadvantage of the deterministic models is their large computation time.

On the other hand, empirical propagation models are those based only on observations and measurements. In spite of these

 ISBN: 978-1-61208-360-5

models be able to predict rain-fade and multipath [10], they are mainly used to estimate path loss, an important task during the initial deployment of wireless networks and cell planning. Empirical models can be split into two subcategories namely time dispersive and non-time dispersive. Time dispersive models are designed to provide information relating to the time dispersive characteristics of the channel. An example of this type are the Stanford University Interim (SUI) channels models developed under the IEEE 802.16 working group [11]. On the other hand, a non-time dispersive model predicts mean path loss as a function of various parameters as distance, antenna heights, latitude, longitude, etc. Examples of non-time dispersive empirical models are Hata [12] and Ericsson 9999 [13] models. In this work, we consider the Okumura-Hata and Ericsson 9999 models for path loss prediction.

A. Okumura-Hata Model

The Okumura model is one of the most used models for signal prediction in the urban areas. It applies to frequencies in the range of 150 MHz to 1920 MHz and distances from 1 to 100 km [14]. The Okumura model is entirely based on measured data without any analytical explanation. Nevertheless, it is very practical and has become a standard for planning land mobile radio systems in Japan. The Okumura model is a very good model in urban and suburban areas, but not so good in rural areas due to its slow response to rapid changes in the terrain.

The Hata model is an empirical formulation of the path loss data provided by Okumura's model and it is valid from 150 MHz to 1500 MHz. Hata presented the propagation loss in urban area as a standard expression and provided correction factors for applications in other environments. The Okumura-Hata model is the combination of both above models.

The expression for the average path loss in urban areas is given by [12]

$$\begin{aligned} L(\mathrm{dB}) &= 69.55 + 26.26\log f - 13.82\log h_t \\ &- a(h_r) + (44.9 - 6.55\log h_t)\log d \end{aligned} \quad (1)$$

where f is the frequency (in MHz) from 150 MHz to 1500 MHz, h_t is the effective height of the base station antenna (transmitter) in meters, varying from 30 m to 200 m, h_r is the effective height of the mobile station antenna (receiver) in meters, varying from 1 m to 10 m, d is the distance between transmitter and receiver (in km), and $a(h_r)$ is the correction factor to the effective height of the receiver antenna, which is function of the size of the coverage area. For large cities and $f \geq 300$ MHz, the factor $a(h_r)$ is given by

$$a(h_r) = 3.2(\log 11.75h_r)^2 - 4.95 \text{ dB}\,. \quad (2)$$

For suburban and rural areas, the path loss is obtained by other expressions that can be found in [15].

Predictions of the Hata model are quite similar to the Okumura model, whereas d does not exceed 1 km. The Hata model applies to macrocells mobile systems, but not to personal communications service (PCS) systems that have cells in order of 1 km radius.

B. Ericsson 9999 Model

The Ericsson 9999 model is an extension of Hata model, where we can adjust the parameters according to the given scenario [13]. In this model, the path loss is described as

$$\begin{aligned} L(\mathrm{dB}) &= a_0 + a_1\log d + a_2\log h_t + a_3\log h_t\log d \\ &- 3.2(\log 11.75h_r)^2 + g(f) \end{aligned} \quad (3)$$

where $g(f)$ is given by [13]

$$g(f) = 44.49\log f - 4.78(\log f)^2\,. \quad (4)$$

For urban environments, the default values of the parameters a_0, a_1, a_2 and a_3 are, respectively, 36.2, 30.2, 12.0 and 0.1. For suburban and rural environments, the parameters a_0, a_1, a_2 and a_3 assume another values that can be found in [13].

III. Proposed Method

SVM is a popular ML technique that make use of the optimization of a function in its training stage. Lately, SVM have received increasing attention from ML community, since it presents some advantages when compared with other ML techniques, such as the absence of local minima, the sparseness of the solution and the capacity control obtained by optimising the margin [8]. Initially developed for solving classification problems, SVM techniques can be successfully applied in regression, i.e., for function approximation problems.

A. Support Vector Regression

In the regression problems, we estimate the functional dependence of the output variable on an n-dimensional input variable. In other words, we deal with real valued functions and we model an $\mathbb{R}^n$ to $\mathbb{R}$ mapping.

The general regression learning problem is set as follows. The ML algorithm is given a set of training data from which it tries to learn 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 $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_\ell, y_\ell)$, 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 samples in the training data set.

Starting from the linear regression problem, assume that $h(\mathbf{x}_i, \mathbf{w})$ is a linear regression hyperplane given by

$$h(\mathbf{x}_i, \mathbf{w}) = \left\langle \mathbf{w}^T, \mathbf{x}_i \right\rangle + b\ , \quad (5)$$

where $\mathbf{w} \in \mathbb{R}^n$ is the normal vector to this hyperplane, the scalar $b \in \mathbb{R}$ is called a bias, $\langle \cdot, \cdot \rangle$ is the inner product operator and $(\cdot)^T$ is the transpose operator. In the case of SVR, we measure the error of approximation instead of the margin used in classification. With this in mind, we use a function named Vapnik's linear loss function with ε-insensitivity zone defined as [16]

$$E(e_i) = |e_i|_\varepsilon = \begin{cases} 0 & , \text{ if } |e_i| \leq \varepsilon \\ |e_i| - \varepsilon, & \text{otherwise} \end{cases}\ , \quad (6)$$

 ISBN: 978-1-61208-360-5

where $e_i = y_i - h(\mathbf{x}_i, \mathbf{w})$. The Vapnik's linear loss function $E(e_i)$ is illustrated in Fig. 1(a), where the ε-insensitivity zone is highlighted. Thus, the loss is equal to zero if 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 concerns to find the linear function that approximates the training pairs $(\mathbf{x}_i, y_i)$ with an accuracy ε. In other words, we need to find a vector $\mathbf{w}$ that minimizes the error, which implies to solve the optimization problem given by [17]

$$\min_{\mathbf{w},b} \frac{1}{2} \|\mathbf{w}\|^2 \tag{7}$$

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

To obtain sparse solutions and penalize the large residuals, a penalty term is included in (7), so that

$$\min_{\mathbf{w},b} \left[\frac{1}{2} \|\mathbf{w}\|^2 + C \left(\sum_{i=1}^{\ell} E(e_i) \right) \right] \tag{8}$$

where C is a cost parameter. The function $E(e_i)$ defines an ε-tube as exhibited in Fig. 1(b), where ε is the radius of the tube. The restriction $|e_i| \leq \varepsilon$, i.e., $y_i + \varepsilon \geq h(\mathbf{x}_i, \mathbf{w}) \geq y_i - \varepsilon$ is the condition for a predict point to be within in the ε-tube.

The optimization problem represented by (8) can be relaxed by introducing slack variables, denoted by ξ and $\hat{\xi}$, which allows to deal with points outside the ε-tube. The points above the ε-tube have $\xi > 0$ and $\hat{\xi} = 0$, while the points below the ε-tube have $\xi = 0$ and $\hat{\xi} > 0$. At last, the points inside of ε-tube have $\xi = \hat{\xi} = 0$.

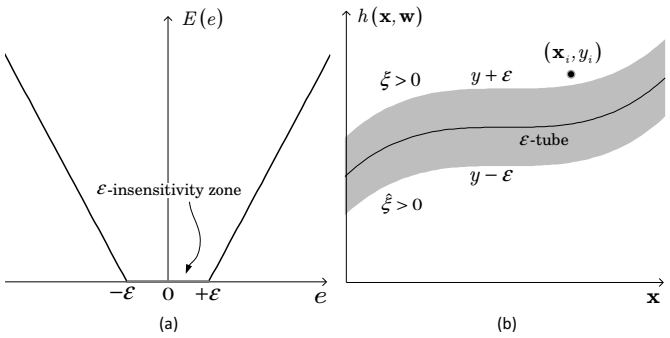

Fig. 1. (a) Vapnik's linear loss function with ε-insensitivity zone versus e. (b) ε-tube defined from the function $E(e)$.

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

$$\min_{\mathbf{w},b} \left[\frac{1}{2} \|\mathbf{w}\|^2 + C \left(\sum_{i=1}^{\ell} (\xi_i + \hat{\xi}_i) \right) \right] \tag{9}$$

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 using Lagrange multipliers, as can be seen in [17]. After calculating the Lagrange multiplier vectors $\boldsymbol{\alpha}$ and $\boldsymbol{\alpha}^*$, the best regression hyperplane obtained is given by

$$h(\mathbf{x}_i, \mathbf{w}) = \sum_{i=1}^{\ell} (\boldsymbol{\alpha} - \boldsymbol{\alpha}^*) \left\langle \mathbf{x}_i^T, \mathbf{x}_i \right\rangle + b \ . \tag{10}$$

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 is reformulated as the maximization of dual Lagrangian with Hessian matrix [8] and the solution is given by

$$h(\mathbf{x}_i, \mathbf{w}) = \sum_{i=1}^{\ell} (\boldsymbol{\alpha} - \boldsymbol{\alpha}^*) \left\langle \Phi^T(\mathbf{x}_i), \Phi(\mathbf{x}_i) \right\rangle + b \ . \tag{11}$$

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*.

Note that the optimization problem for nonlinear regression, represented by (11), involves the calculation of inner products between vectors of the feature space $\mathfrak{I}$. Since $\mathfrak{I}$ can be very higher dimensional, the calculation of Φ can become infeasible. Therefore, 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 $\mathfrak{I}$ [18].

A kernel is a function that applies to 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 $\mathfrak{I}$ [19], i.e.,

$$K(\mathbf{x}_i, \mathbf{x}_j) = \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) \rangle \ . \tag{12}$$

To ensure the convexity of the optimization problem given by (11) 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 [16]. The more common practice kernels for regression problems are the polynomial kernel and the radial basis functions (RBF) ones. In this paper, we consider the use of the polynomial kernel and two types of RBF kernels: Laplacian and Gaussian. The expressions related to each kernel are given in Table I.

TABLE I
TYPES OF KERNELS CONSIDERED FOR THE PROPOSED METHOD.

Kernel	Expression	Parameters
Polynomial	$K(\mathbf{x}_i, \mathbf{x}_j) = \left(\beta \left\langle \mathbf{x}_i^T, \mathbf{x}_j \right\rangle + c\right)^z$	β, c, z
Gaussian	$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}\right)$	σ
Laplacian	$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|}{\sigma}\right)$	σ

The polynomial kernel is a function that represents the similarity of training samples in the feature space over

polynomials of the original variables and combinations of these. The adjustable parameters are the scale term β, the constant term (off-set) c and the polynomial degree z. Laplacian and Gaussian kernels are examples of RBF kernels. The adjustable parameter σ is very important to the performance of these kernels and should be fit to the problem at hand.

B. Measurement Setup

The work presented in this paper considers mobile radio wave propagation measurements at a carrier frequency of 853.71 MHz. The measurements of the downlink signal strength level were made in an urban environment in the city of Fortaleza-CE, Brazil. Fig. 2 illustrates the urban area of the city where the measurements were taken. The colors used along the indicated streets represent each received signal strength indicator (RSSI) in dBm. In total, 1933 measurements were performed using Agilent E6474A tool as a pilot scanner. The location of the base station (BTS) is also indicated in Fig. 2.

During the drive test, 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. At last, the theoretical path loss of the Okumura-Hata model was also 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 it. The base station was located in a rooftop 90 meters high with a sectored antenna, which had a half-power beam width of 63 °. Also, the effective radiated power (ERP) of the base station was set to 48 dBm.

IV. Training and Evaluation

Many ML algorithms, such SVR and ANN, have important parameters that cannot be set directly from the data. The process of choosing these parameters to obtain the best performance of the model is known as tuning and is described below.

A. Model Tuning

Cross-validation is a model validation technique for evaluating how the results of a statistical analysis will generalize to an independent data set. A common type of cross-validation is the k-fold cross-validation, generally used to evaluate the model accuracy [20]. It is a re-sampling technique where the samples are randomly split into k sets of approximately equal size. These subsets are named folds and they are divided in two groups: the test set with only one fold and the training set with $(k-1)$ folds. Initially, the first fold is established as test set and the model is fit using the others $(k-1)$ folds. The held out sample in the first fold is predicted by the ML algorithm and is utilized to estimate the performance. After that, the first fold is given back to the training set. This procedure is repeated with the second fold held out, and so on. In this paper, we consider $k = 10$ and use the average root mean square error (RMSE) $\bar{\mu}$ defined as

$$\bar{\mu} = \sqrt{\frac{1}{k}\sum_{i=1}^{k}\mu_j} \tag{13}$$

to evaluate the model precision. In (13), μ_j is the RMSE calculated for j-th test set ($j = 1, 2, \ldots, k$), given by

$$\mu_j = \sqrt{\frac{1}{\ell_j}\sum_{i=1}^{\ell_j}\left(y_i - h\left(\mathbf{x}_i, \mathbf{w}\right)\right)^2} \tag{14}$$

where ℓ_j is the number of samples in the j-th test set.

The definition of σ, common to RBF kernels, was made using the analytical approach presented in [21], where it is shown that the optimal values of σ are in the range of the 10^{th} and the 90^{th} percentile of $\|\mathbf{x}_i - \mathbf{x}_j\|^2$. In addition to that, it is suggested in [18] that the midpoint of these two

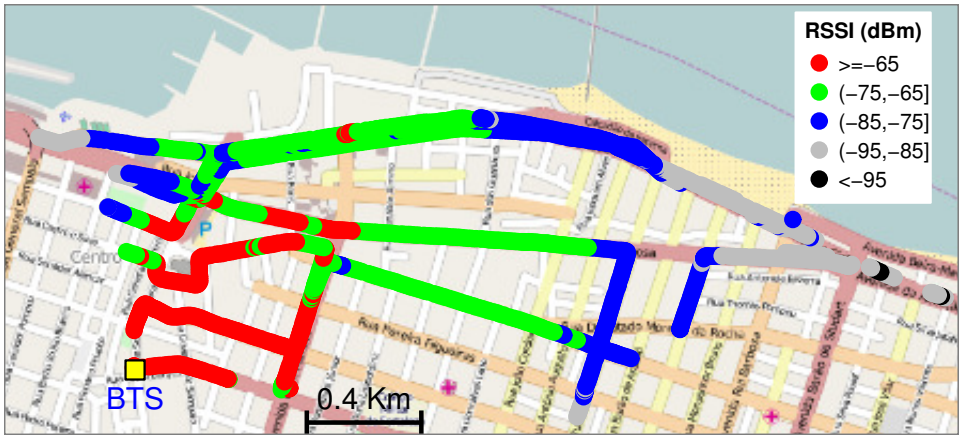

Fig. 2. Drive test with measurements in an urban area in the city of Fortaleza. The colors indicate the radio signal strength indicator (RSSI) in dBm and the yellow square represents the location of the base station (BTS).

 ISBN: 978-1-61208-360-5

percentiles should be used. Thus, this kernel parameter was estimated to be $\sigma = 0.244$.

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 [20]. In the tuning process, 18 values were tested for C, from 2^{-2} to 2^{15}, being each value a power of 2.

In the tuning process considered in this work, it was tested $\varepsilon = 0.1$ and $\varepsilon = 0.05$ in combination with the range of C specified previously. The best fit was found for $\varepsilon = 0.05$ to the Laplace kernel, and $\varepsilon = 0.1$ to the Polynomial and the Gaussian ones.

B. Numerical Results

All models considered in this paper are implemented using the R language. The performance of the SVR algorithms is evaluated via computer simulations for the three kernels mentioned in Subsection III-A. For their implementation, the kernlab package is employed [18].

According to what has been explained about SVR for nonlinear learning problems in Subsection III-A, Fig. 3 shows the support vectors and the respective regression line when the Laplace kernel is adopted. Note that, for the sample set in evidence, the number of support vectors is inferior to the number of measurements, but it is sufficient to obtain the regression line.

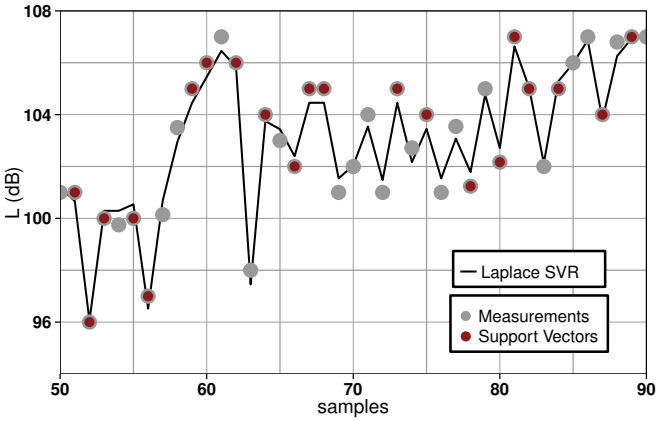

Fig. 3. Support vectors used to obtain the regression line for path loss in dB when Laplacian kernel is adopted.

Fig. 4 shows a comparison of measurements corresponding to 10% of the sample set obtained from the drive test and the predictions using SVR algorithms for polynomial, Gaussian and Laplacian kernels. One can see that the Laplacian kernel is the best option among the three kernels.

For comparison of the SVR algorithms with other ML techniques, it is implemented a MLP-ANN with a weight decay w_d, having a input layer with nine neurons, a hidden layer with M neurons and a output layer with one neuron. The backpropagation algorithm is used to train the MLP-ANN. In the tuning process, some MLP-ANN configurations were investigated for $M = 9, 12, 15, 18, 21, 24, 27$ and $w_d = 0.01, 0.05, 0.1$. The best fit found for the MLP-ANN was $M = 27$ and $w_d = 0.01$.

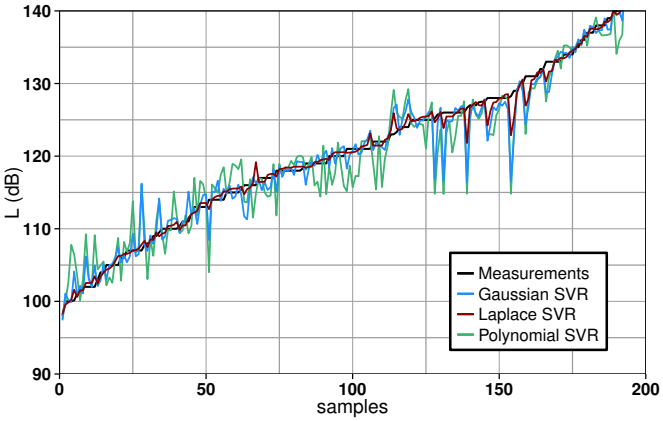

Fig. 4. Comparison of path loss predictions using SVR algorithms and measurements obtained from drive test for 10% of the sample set.

Table II provides a statistical analysis of the SVR algorithms, the MLP-ANN and the two empirical propagation models mentioned in Section II, where the average RMSE $\bar{\mu}$ and the standard deviation of the RMSE, denoted by σ_μ are presented. The best configuration parameters of the SVR algorithms (C and ε) and the parameters of each kernel are also shown in Table II. We can see that the Laplacian SVR presents an average RMSE $\bar{\mu} = 1.76$ dB, while the polynomial and the Gaussian SVRs presents an average RMSE of 3.47 dB and 4.55 dB, respectively. Both empirical propagation models have inferior performance when compared to all considered ML techniques.

The MLP-ANN performance, with $\bar{\mu} = 1.89$ dB, can be considered similar to the Laplacian SVR performance. In spite of analogous performance, MLP-ANN and Laplacian SVR have significant differences in their implementation, which are discussed below.

At first, as the MLP-ANN has local minimal, it is necessary to initialize the weight matrix with different values in the attempt to test more points (in this work, we initialize the MLP-ANN three times), whereas such problem do not exist in SVR algorithms. In case of SVR, a convex optimization problem is solved resulting in a global minimum. Therefore, when using SVR there is no problem with initializations and checking for convergence [8].

Secondly, as there is no feature extraction in the MLP-ANN, sometimes it is necessary to use another ML algorithm to do this task [18]. In the SVR algorithm, the data can be applied directly without the need for feature extraction, because the SVR algorithm already do this function [22]. Furthermore, when the number of features increases, the MLP-ANN complexity demands more computational cost,

 ISBN: 978-1-61208-360-5

TABLE II
STATISTICAL ANALYSIS OF THE SVR ALGORITHMS WITH THEIR BEST PARAMETERS C AND ε, MLP-ANN AND THE EMPIRICAL MODELS CONSIDERED IN THIS WORK.

ML Algorithm/Model	$\bar{\mu}$ (dB)	σ_μ (dB)	C	ε	Kernel parameters
Polynomial SVR	3.47	0.54	1024	0.1	$\beta = 0.1, c = 1, z = 3$
Gaussian SVR	4.55	0.15	8192	0.1	$\sigma = 0.244$
Laplacian SVR	1.76	0.12	1024	0.05	$\sigma = 0.244$
MLP-ANN	1.89	0.17	-	-	-
Okumura-Hata	7.13	5.08	-	-	-
Ericsson 9999	21.59	6.22	-	-	-

whereas in the SVR algorithm, once that a valid kernel has been selected, one can practically work in spaces of any dimension without any significant additional computational cost [22].

Finally, the MLP-ANN training time is normally longer than SVR one. We can mention two reasons for that: first, the MLP-ANN usually needs to be initialized more than one time; second, in the SVR algorithm, the training is executed considering only the support vectors, while in the MLP-ANN the training is performed on the entire data set.

V. CONCLUSIONS

In this study, a method to predict path loss in an urban outdoor environment using SVR was proposed. To do that, mobile radio wave propagation measurements at a carrier frequency of 853.71 MHz obtained in an urban environment in the city of Fortaleza-CE, Brazil were considered. Various field data of each measured point like antenna-separation distance, terrain elevation, the theoretical path loss of the Okumura-Hata model among others were collected as input of the SVR process. Polynomial, Gaussian and Laplacian kernels were adopted for SVR algorithms. For comparison, we considered two empirical propagation models (Okumura-Hata and Ericsson 9999) and a MLP-ANN optimized for our prediction problem. In case of SVR, it was verified that the Laplacian kernel was the best option among the investigated kernels. In addition, the SVR algorithm using Laplacian kernel and the MLP-ANN had similar performance, being the former an alternative of lower computational complexity. The authors conjecture that the lower computational complexity of the SVR technique is due to the use of support vectors and the kernel trick which reduce the training time, naturally perform the feature extraction and increase the capacity of working with higher dimensional spaces.

ACKNOWLEDGMENTS

The authors would like to thank Evandro Uchoa and Gustavo Raulino for their assistance in the acquisition of field data.

REFERENCES

[1] R. Want, "When cell phones become computers," *IEEE Pervasive Computing*, vol. 8, n. 2, 2009, pp. 2-5.

[2] T. S. Rappaport. *Wireless Communications: Principles and Practice*. Prentice-Hall PTR, 2 ed., 2009.

[3] T. Mitchell. *Machine Learning*. McGraw-Hill, 1 ed., 1997.

[4] S. Haykin. *Neural Networks: A Comprehensive Foundation*. Prentice-Hall, NJ, USA, 2 ed., 1999.

[5] E. Östlin, H.-J. Zepernick and H. Suzuki, "Macrocell path-loss prediction using artificial neural networks," *IEEE Vehic. Tech.*, vol. 59, n. 6, 2010, pp. 2735-2747.

[6] P. S. Sotiroudis et. al., "A neural network approach to the prediction of the propagation path-loss for mobile communications systems in urban environments," in *Proc. of the Progress in Electromag. Research Symp. (PIERS)*, Prague, CZ, 2007.

[7] A. Neskovic et. al., "Indoor electric field level prediction model based on artificial neural networks," *IEEE Commun. Lett.*, vol. 4, n. 6, 2000, pp. 190-192.

[8] V. Kecman, *Support Vector Machines: An Introduction*. Springer, New York, 2005.

[9] G. E. Athanasiadou et. al., "A microcellular ray-tracing propagation model and evaluation of its narrowband and wideband predictions," *IEEE J. Sel. Areas Commun.*, vol. 18, n. 3, 2000, pp. 322-335.

[10] R. K. Crane, "Prediction of attenuation by rain," *IEEE Trans. Commun.*, vol. COM-28, 1980, pp. 1727-1732.

[11] V. Erceg et. al., "Channel models for fixed wireless applications," *Tech. Rep.*, IEEE 802.16 Broadband Wireless Access Working Group, Jan 2001.

[12] M. Hata, "Empirical formula for propagation loss in land mobile radio services," *IEEE Trans. Veh. Tech.*, vol. 29, n. 3, 1981, pp. 317-325.

[13] S. S. Kale and A. N. Jadhav, "Performance analysis of empirical propagation models for WiMAX in urban environment," *IOSR Journal of Electronics and Communication Engineering*, 2013, pp 24-28.

[14] Y. Okumura et. al., "Field strength and its variability in VHF and UHF land-mobile radio-services," *Rev. Elec. Comm. Lab.*, vol. 16, Sep-Oct 1968.

[15] T. K. Sarkar et. al., "A survey of various propagation models for mobile communication," *IEEE Ant. and Propag. Mag.*, vol. 45, n. 3, 2003, pp. 51-82.

[16] V. N. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, New York, 1995.

[17] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, New York, 2006.

[18] A. J. Smola, K. Hornik and A. Karatzoglou. "An S4 Package for Kernel Methods in R," *Journal of Statistical Software*, vol.11, n.9, 2006, pp 1-20.

[19] R. Herbrich. *Learning Kernel Classifiers: Theory and Algorithms*. MIT Press, 2001.

[20] M. Kuhn and K. Johnson. *Applied Predictive Modeling*. Springer, New York, 2013.

[21] B. Caputo, K. Sim, F. Furesjo and A. Smola. Appearance-based Object Recognition using SVMs: Which Kernel Should I Use?, in *Proc. of NIPS Workshop on Statist. Methods for Comput. Exp. in Visual Process. and Comp. Vision*, Whistler, 2002.

[22] K. Hornik, D. Meyer and K. Hornik. "Support vector machines in R," *Journal of Statistical Software*, vol. 15, n. 9, 2006, pp 1-28.

 ISBN: 978-1-61208-360-5