A Hybrid Machine Learning Approach for Mobile User Positioning in Cellular Networks

Robson D. A. Timoteo, Daniel C. Cunha, Lizandro N. Silva and George D. C. Cavalcanti

***Abstract*— The outstanding growth of location-based services and applications for mobile devices has motivated research about wireless positioning techniques for outdoor and indoor environments. In the present paper, a machine learning approach is proposed for finding the mobile user location. More precisely, a hybrid machine learning technique is proposed to obtain the position of a mobile user in an outdoor environment of cellular networks. The proposal employs k-Nearest Neighbors as a regression model to find the distances between the mobile and the base stations, and Genetic Algorithms to estimate mobile position. Simulation results show that the proposed algorithm has better performance than the COST-231/Nelder-Mead trilateration technique. Friedman and Nemenyi tests are used to statistically validate the results.**

***Keywords*— Mobile positioning, machine learning, cellular networks.**

I. Introduction

Wireless positioning systems have received increasing attention in recent years [1], [2]. Position obtaining through wireless and mobile technologies is a key factor to achieve an accurate knowledge of mobile terminal location, which is essential for providing location-based services [3].

Radiolocalization is one of the techniques to derive the positioning of mobile terminals in wireless systems. In cellular networks, the localization of a mobile user (MU) is obtained by measuring physical parameters of the radio frequency signals transmitted between the base transceiver stations (BTSs) and the MU. After, the physical parameters are used to obtain distance information from MU to the BTSs, that are assumed to be reference points. Finally, MU location is estimated from geometrical properties by using processing algorithms.

Machine learning (ML) is a data-driven approach, which means that it extracts information from past observations to make accurate predictions. Recently, many researchers have used ML techniques to improve the accuracy of the MU location in wireless environments [4]. For example, in [5], it is shown that the user location can be inferred by ML techniques using phone's power consumption information.

In this paper, an ML approach is proposed for finding the MU position in an outdoor environment of cellular networks. More precisely, we propose a hybrid ML technique to model the relationship between the received strength signal indicator (RSSI) measurements and the position of the mobile terminal. The proposed technique combines k-Nearest Neighbors and Genetic Algorithms. The former is used as a regression model to find the distances between the MU and the BTSs, while the latter is applied to estimate the mobile position. Simulation results show that the proposed algorithm has better performance than the COST-231/Nelder-Mead (NM) trilateration method. Friedman and Nemenyi tests are used to statistically validate the results.

The remaining of the paper is structured as follows. In Section II-A, lateration-based positioning techniques are introduced. General concepts about k-Nearest Neighbor and genetic algorithms are presented in Section II-B. Also in Section III, a hybrid ML technique for MU positioning in an outdoor environment of cellular networks is described. Numerical results are presented in Section IV and conclusions are drawn in Section V.

Robson D. A. Timoteo, Daniel C. Cunha, Lizandro N. Silva and George D. C. Cavalcanti, Centro de Informática (CIn), Universidade Federal de Pernambuco (UFPE), Recife-PE, Brazil, E-mails: rdat@cin.ufpe.br, dcunha@cin.ufpe.br, lns3@cin.ufpe.br, gdcc@cin.ufpe.br.

II. Background

A. Lateration-Based Positioning Techniques

Lateration-based mobile positioning techniques are a well-known localization method that estimate the position of an object by measuring its distances to multiple reference points called anchors [6]. Using this technique in the context of cellular networks, a solution to locate an MU consists basically on a distance estimation from the MU to the BTSs of the network.

When three BTSs are assumed, we have a trilateration localization technique in which the MU positioning problem can be expressed as a system of quadratic equations such that

$$d_i{}^2 = (x_p - x_i)^2 + (y_p - y_i)^2, i = 1, 2, 3 \ , \qquad (1)$$

where d_i is the estimated distance between the MU and the i-th BTS and the pairs (x_p, y_p) and (x_i, y_i) are, respectively, the latitude and the longitude of the MU and the i-th BTS. The distances d_i can be obtained, for example, using empirical propagation models, as Okumura-Hata and COST-231 [7]. Given d_i, we desire to estimate the coordinates (x_p, y_p) of the MU. The trilateration positioning technique is equivalent to find the solutions to the system defined by (1) using any direct optimization method, such as Nelder-Mead (NM) or Newton-Raphson. More details about lateration-based positioning techniques can be found in [8].

B. Machine Learning Techniques

1) k-Nearest Neighbors: k-Nearest Neighbors (kNN) is a classifier that belongs to the family of instance-based learning algorithms [9]. In this family, the training instances are stored and no explicit generalization is performed. This strategy

uses a different concept when compared with other methods, such as Artificial Neural Networks and Decision Trees, that construct a general description of the target function based on the training instances. So, in instance-based learning, the generalizing is only performed when a query instance is classified.

The kNN algorithm can be used in classification or regression tasks and it works as follows [10]: given a query instance $\mathbf{X}_i$, the first step is to find the k closest training instances to $\mathbf{X}_i$; these are the neighbors of $\mathbf{X}_i$. More precisely, given that each instance is described by an m-dimensional feature vector $\mathbf{X}_i = [X_{i1}, X_{i2}, \ldots, X_{im}]$, the distance between two instances $\mathbf{X}_i$ and $\mathbf{X}_j$ is defined as $d(\mathbf{X}_i, \mathbf{X}_j)$, such that

$$d(\mathbf{X}_i, \mathbf{X}_j) = \sqrt{\sum_{r=1}^{m} (X_{ir} - X_{jr})^2}\,. \tag{2}$$

After calculating the k neighbors of $\mathbf{X}_i$ using Equation (2), the class of $\mathbf{X}_i$ is assigned as the most common class among its k nearest neighbors for classification problems. However, for regression problems, the predict value of $\mathbf{X}_i$ is given by the average of the values of its k nearest neighbors, according to

$$\hat{f}(\mathbf{X}_i) \leftarrow \frac{\sum_{i=1}^{k} f(\mathbf{X}_i)}{k}\,, \tag{3}$$

where $\mathbf{X}_i$ is a training set instance and $f(\mathbf{X}_i)$ is the target for $\mathbf{X}_i$.

When dealing with very large datasets, kNN is computationally expensive to find the k nearest neighbors. On the other hand, an advantage of the kNN algorithm is that there is no cost associated to the learning process and, besides, kNN is able to learn complex concepts by local approximation using a simple strategy.

2) Genetic Algorithms: Genetic algorithms (GAs) were introduced by Holland in 1975 and have been used in problems involving optimization and search [11]. As they are based on Darwin's Evolutionary Theory, the key issue underlying GAs is natural selection and survival of the most adapted (fittest) individuals.

In GAs, the search for the best solution of a problem is conducted by using a fitness function, which is used to assess the quality of candidate solutions. In GAs, the solutions are represented by chromosomes. Along the evolutionary process, the best chromosomes (which correspond to the best solutions of the problem, according to the fitness function) are selected and submitted to the operations of crossover and mutation, which generate the next offspring (descendants). The process of best chromosomes selection, crossover and mutation is repeated until a stopping criterion is reached. It is important to observe that the search space is given by the set of all possible configurations a chromosome can assume.

Different methods can be used for the purpose of best chromosomes selection, such as roulette wheel selection, elitism and tournament selection. Common stopping criteria are: a fixed number of generations is reached; a fixed percentual of highest fitness chromosome has reached a plateau in a fixed number of successive iterations (that is, better solutions concerning that percentual of chromosomes are no longer produced in successive generations); or a combination of the previous conditions.

III. Proposed ML Approach

In this work, we propose a hybrid ML algorithm to develop models of the relationship between the RSSI measurements and the position of the MU. In the following subsections, a description of the proposal and the experimental setup are presented.

A. Proposal Description

The proposed algorithm employs kNN as a regression model to find the MU-BTS distances d_i. In addition, our proposal uses GA as an optimization tool to find the solutions to the system defined by (1). Based on this, the hybrid ML technique proposed here is denoted as the kNN/GAq algorithm, where q is the number of BTSs used in the regression model. Table I shows the five steps of the kNN/GAq algorithm. The first step of the proposed algorithm is referred to database building. To do this, several radio frequency signals measurements of the cellular network are collected using a scanner. More details about the experimental setup considered in this work are presented in Subsection III-B.

TABLE I
Description of the kNN/GAq algorithm.

Step	Description
1	Collect the scanner measurements (database building).
2	Store the training instances to obtain hypothesis functions with kNN for predicting the MU-BTS distances (one for each BTS).
3	Collect the RSSI measurements from the sought mobile to all BTSs.
4	Use kNN to predict the distances between the sought mobile and the BTSs.
5	Use GA to estimate the position of sought mobile using the distances predicted in Step 4.

The measurements obtained in the first step are used to build the training and test datasets. The training dataset is used to adjust the kNN regression model. For the regression model, the features are the path loss for each BTS, and the target is the MU-BTS distance. In other words, the kNN regression model provides q hypothesis functions $f_i(\cdot), i = 1, 2, \ldots, q$, that are utilized to derive the distances between the MU and each one of the q BTSs. Figure 1(a) illustrates the training stage of the hybrid ML technique. Since q BTSs are considered, q distances $d_i, i = 1, 2, \ldots, q$, are obtained at the end of the training stage.

Many ML algorithms, such as kNN, 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. To put it in another way, we evaluate the kNN algorithm varying the parameter k (k is an integer) in the interval $[1; 45]$ to find the best fit model, characterizing the second step of the algorithm. The 10-fold cross-validation re-sampling technique [12] was used to find the best k for each BTS. Table II shows the best values of k for each BTS, the cross-validation distance error $\bar{\mu}$, and its standard deviation μ_σ. We assume that $q = 6$ BTSs are

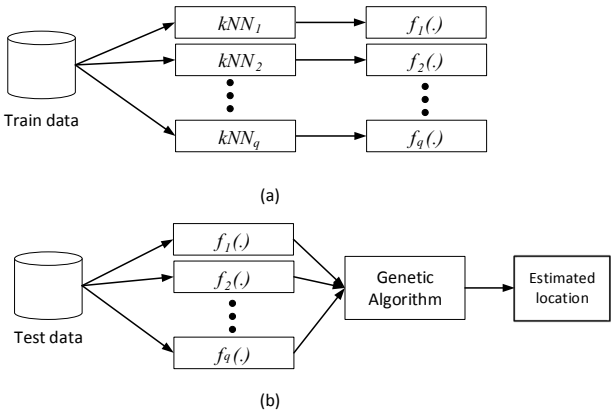

Fig. 1. Diagram of the kNN/GAq algorithm: (a) Training stage to obtain q hypothesis functions. (b) Validation using test stage (q hypothesis functions followed by optimization using GA).

employed in the tuning process. The motivation for using six BTSs is a promising trend to increase the density of BTSs in urban areas. In accordance to [13], this tendency indicates an expansion of the system capacity in the near future cellular networks using small cells.

After the evaluation of the kNN regression model, the proposed hybrid ML technique is ready to be employed. For that, new measurements should be acquired using the scanner in a practical situation (third step). In our experiment setup, we use a test dataset from the measurements obtained in the first step. Then, the last three steps of the kNN/GAq algorithm consist in validating its accuracy. For each of the q RSSIs used as input of the hypothesis functions $f_i(\cdot), i = 1, 2, \ldots, q$, an MU-BTS distance d_i is obtained (fourth step). Then, these q distances are applied to the GA in attempt to achieve the localization of the MU, characterizing the fifth step. Figure 1(b) illustrates the fourth and fifth steps of the proposed ML technique.

In this step, the GA individual is a vector with the geographical position of the MU (latitude and longitude). Thus, in the GA context, the MU estimated position will be given by the individual with the highest fitness function value, which as given by:

$$f_g(x_g, y_g) = \min_{1 \le i \le q} \left(d_i - \sqrt{(x_i - x_g)^2 + (y_i - y_g)^2} \right), \quad (4)$$

where (x_g, y_g) is the GA individual, (x_i, y_i) is the position of i-th BTS, q is the number of BTS and d_i is the distance obtained using (1).

TABLE II
RESULTS OF THE TRAINING STAGE OF EACH kNN MODEL USING 10-FOLD-CROSS-VALIDATION FOR $q = 6$ BTSS.

BTS	best k	$\bar{\mu}$ (m)	μ_σ (m)
BTS-1	7	86.0	10.4
BTS-2	5	88.2	13.2
BTS-3	7	90.4	9.2
BTS-4	5	88.2	13.2
BTS-5	5	104.0	13.4
BTS-6	5	86.5	5.4

The GA is set with an initial population of 250 individuals. To evolve the population, uniform random mutation and local arithmetic crossover are utilized [14].

B. Measurement Setup

We assume mobile radio wave propagation measurements at 1.8 GHz Global System for Mobile Communications (GSM) frequency band. A drive test where measurements of the downlink signal strength level were made in an urban environment in the city of Recife-PE, Brazil.

Figure 2 illustrates the urban area of the city where the measurements were taken, as well as the locations of all BTSs. We should notice that the antenna of each BTS is set at a given azimuth related to the true north. For BTSs-3 and 5, the azimuth is 220°, while for BTS-6, the azimuth is 60°. For BTSs-2 and 4, the azimuths are 10° and 120°, respectively. In total, 2956 measurements were performed using NEMO FSR1[1] tool as a GSM pilot scanner. In the database building, the training dataset consisted of 2756 measurements and the test set with 200 measurements was considered.

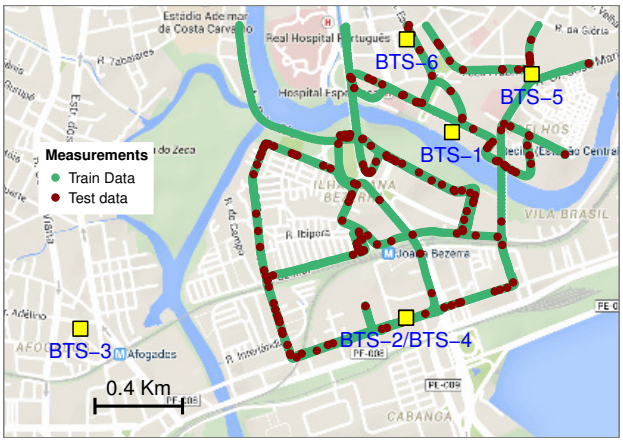

Fig. 2. Urban environment of the city of Recife-PE, Brazil with the indication of the training measurements, testing measurements, and the locations of the BTSs.

IV. NUMERICAL RESULTS

In this work, four positioning techniques are implemented. In the first technique (C-231/NM), the COST-231 model is used to predict the MU-BTS distances and NM optimization method is applied to estimate the MU position. The second technique (kNN/NM) consists on using the kNN algorithm and the NM method to estimate the MU-BTS distances and the MU position, respectively. In both techniques, the three BTSs with lower distance prediction error are chosen. The third and fourth techniques, named as kNN/GA3 (with three BTSs) and kNN/GA6 (with six BTSs), refer to the hybrid proposal. For all techniques considering three BTSs, the BTS-1, BTS-4 and BTS-6 are used. The performance of the positioning techniques is evaluated via computer simulations using a test dataset with 200 samples. The ML algorithms are implemented using R programming language, with emphasis on the packages caret [15] and genetic algorithms [14].

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

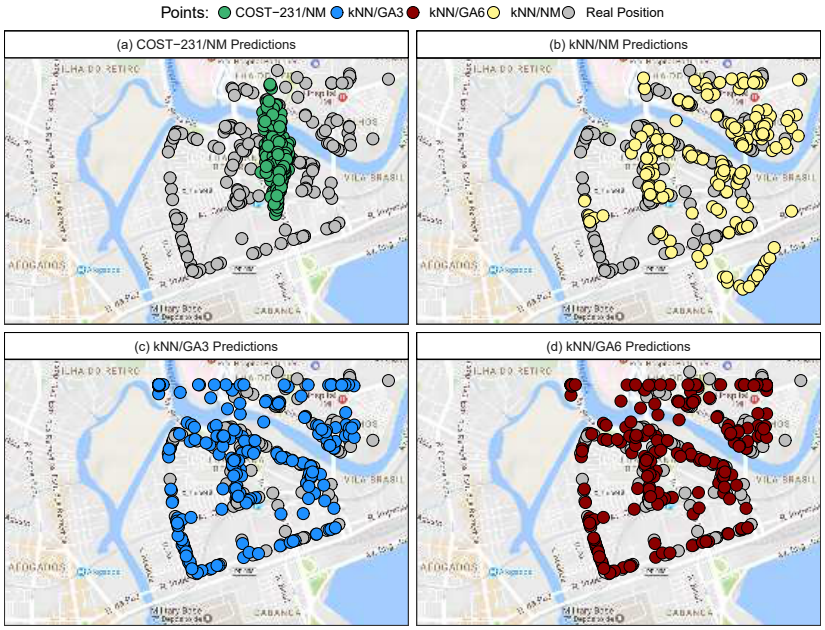

Fig. 3. Prediction maps for each MU positioning technique.

Table III presents root mean square error (RMSE) of the MU-BTS distances predicted using COST-231 and kNN models for the following positioning techniques: C-231/NM, kNN/GA3, and kNN/GA6. The BTSs 2, 3 and 5 were not used for C-231/NM and kNN/GA3 techniques, since both techniques employ three BTSs to estimate the mobile localization, thus these RMSE values were omitted in Table III. With this in mind, we can observe that the lower RMSE values were obtained for kNN/GA6, while the higher ones are related to the C-231/NM technique. These output values are used as inputs for NM and GA optimization methods in order to find the MU position. As we shall see, predicted MU-BTS distances with low RMSE will increase the localization accuracy.

To compare the positioning techniques mentioned previously, we define the localization prediction error η as the distance difference between the real and the predicted positions, measured in meters. Table IV provides a statistical analysis of the localization prediction errors for each positioning technique. The average localization prediction error is represented by $\bar{\eta}$, its standard deviation by η_σ, and the maximum and minimum errors by η_{max} and η_{min}, respectively. Also in Table IV, we can see that the kNN/GA6 algorithm presents

TABLE III
RMSE OF THE ESTIMATED MU-BTS DISTANCES.

BTS	C-231/NM	kNN/GA3	kNN/GA6
BTS-1	650.6 m	252.6 m	72.8 m
BTS-2	-	-	71.3 m
BTS-3	-	-	72.5 m
BTS-4	350.7 m	228.2 m	71.3 m
BTS-5	-	-	96.3 m
BTS-6	388.0 m	220.8 m	75.3 m

TABLE IV
STATISTICAL ANALYSIS OF THE LOCALIZATION PREDICTION ERRORS FOR THE CONSIDERED POSITIONING TECHNIQUES.

Pos. Tech.	$\bar{\eta}$	η_σ	η_{max}	η_{min}
C-231/NM	550.0 m	352.3 m	1731.6 m	6.0 m
kNN/NM	352.7 m	473.1 m	2363.7 m	1.3 m
kNN/GA3	228.9 m	426.5 m	1777.1 m	0.9 m
kNN/GA6	132.4 m	168.8 m	890.3 m	1.5 m

an average error $\bar{\eta} = 132.4$ m, but the accuracy decreases when using fewer BTSs, which can be observed for the kNN/GA3 algorithm ($\bar{\eta} = 228.9$ m). Finally, the kNN/NM technique presents $\bar{\eta} = 352.7$ m, while the C-231/NM exhibits $\bar{\eta} = 550.0$ m.

For providing a graphical comparison, prediction maps can be built for each positioning technique such as illustrated in Figure 3. To obtain each map, we should distribute the test points collected in the field and overlap them to those evaluated by one of the considered positioning techniques. For all maps, the actual MU positions are represented by gray dots.

Figure 3(a) shows the distribution of points acquired from the C-231/NM technique, where the estimated positions correspond to the green dots. It can be observed a concentration of green dots that do not cover the entire region of the gray ones. A possible justification for this concentration is the fact that the three selected BTSs (1, 2, and 4) were not optimized for the C-231 model, but for k-NN one.

In Figure 3(b), the estimated positions are related to the kNN/NM technique and they are represented by yellow dots. These points are obtained similarly as in the previous map and we can notice that the yellow dots are less concentrated than the green ones in Figure 3(a). Thus, if we compare the C-231/NM and kNN/NM models, we verify that the kNN

improved the prediction of the MU-BTS distances.

Figure 3(c) indicates the estimated positions given by the kNN/GA3 algorithm using blue dots. We wish to highlight the convergence of the estimated points and the real ones (gray dots), i.e., they are very close to each other. This means that the kNN/GA3 model is more accurate than the previous ones (C-231/NM and kNN/NM). At last, Figure 3(d) indicates the estimated positions given by the kNN/GA6 technique using red dots. Comparing kNN/GA6 with kNN/GA3, we observe that there are fewer real MU positions not covered by the estimated positions in the former.

Another way to compare the positioning techniques addressed in this paper is by using histograms. Figure 4 shows a histogram for each technique considering the test dataset, in which the x-axis represents the localization prediction error η, while the y-axis corresponds to the count of samples having the same η. Analyzing the four histograms, it is possible to

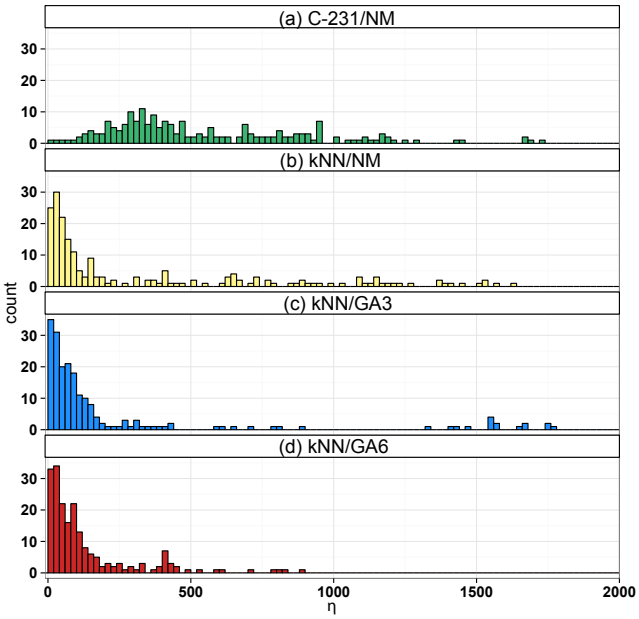

Fig. 4. Histograms of the localization prediction error η (in meters) for each MU positioning technique.

verify that the kNN/GA6 technique is the best one, because the majority of the samples are accumulated at the beginning of the histogram, i.e., the errors remain around 250 m. The second best technique is kNN/GA3, which had the accuracy decreased by the outlier points with errors up to 1000 m. For the other two techniques, we recognize that the errors are more evenly distributed across the adopted range (0 – 2000 m).

Lastly, to verify if the differences between the performances of the positioning techniques are statistically relevant, Friedman test with the Nemenyi post-hoc test are applied [16].

In this work, the Friedman test was performed over the test dataset, for a confidence level $\alpha = 0.05$, and resulted in $p = 2.2{\cdot}10^{-16}$. Thus, H_0 (null hypothesis) 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. The p-value are smaller than the confidence level ($\alpha = 0.05$) for all pairwise comparison in the Nemenyi post-hoc test, which imply that the techniques are different.

V. Conclusions

In this paper, a hybrid machine learning approach was proposed for finding the mobile user position in an outdoor environment of cellular networks. The proposal focused on modeling the relationship between the radio strength signal indicator measurements and the position of the mobile terminal. For doing this, the hybrid technique combined k-Nearest Neighbors (kNN) and Genetic Algorithms (GA). The prior was used as regression model to obtain the distances between the mobile user and the base stations, while the second was employed to estimate the mobile localization as a solution of an optimization problem.

Regarding the estimation of the distances between mobile user and base stations, kNN was a better option when compared to COST-231 propagation model. Concerning the assessment of the mobile user position, the use of GA reduced the distance prediction error when compared to the NM method. Also, it was verified that the increase of base stations in the hybrid technique diminishes the distance prediction error. The price for this improvement in performance is a higher computational complexity.

References

[1] M. Veletic and M. Sunjevaric, "On the Cramer-Rao lower bound for RSS-based positioning in wireless cellular networks," *Int. Journal of Electronics and Communications (AEU)*, vol. 68, pp. 730-736, 2014.

[2] S. Yiu and K. Yang, "Gaussian process assisted fingerprinting localization," *IEEE Internet of Things Journal*, vol. 3, n. 5, pp. 683-690, 2016.

[3] M. Xin, M. Lu, and W. Li, "An adaptive collaboration evaluation model and its algorithm oriented to multi-domain location-based services," *Expert Systems with Apps*, vol. 42, pp. 2798-2807, 2015.

[4] R. D. A. Timoteo et. al., "An approach using support vector regression for mobile location in cellular networks," *Computer Networks*, v. 95, pp.51-61, 2016.

[5] Y. Michalevsky et. al. "PowerSpy: location tracking using mobile device power analysis," arXiv preprint arXiv:1502.03182, 2015.

[6] R. Zekavat and R. M. Buehrer, *Handbook of Position Location: Theory, Practice and Advances*. John Wiley & Sons, 2011.

[7] P. E. Mogensen and J. Wigard, COST action 231-digital mobile radio towards future generation systems: Tech. Report, European Cooperation in Science and Technology, 1999.

[8] L. N. Silva et. al., "Calibragem de modelos de propagação aplicados à localização em telefonia móvel celular," In Proc. *XXXI Brazilian Symposium on Telecommunications (SBrT 2013)*, pp. 1-5, 2013.

[9] D. W. Aha, D. Kibler, and M. K. Albert, "Instance-based learning algorithms," *Machine Learning*, vol. 6, n. 1, pp. 37-66, 1991.

[10] T. Mitchell, *Machine Learning*. McGraw-Hill, 1997.

[11] D. E. Goldberg and J. H. Holland, "Genetic algorithms and machine learning," *Machine Learning*, vol. 3, n.2, pp. 95-99, 1988.

[12] J. Demšar, "A study of cross-validation and bootstrap for accuracy estimation and model selection," *The Journal of Machine Learning Research*, vol. 14, n.2, pp. 1137-1145, 1995.

[13] X. Gelabert, P. Legg, and C. Qvarfordt, "Small cell densification requirements in high capacity future cellular networks ," *In Proc. IEEE ICC 2013 2nd Int. Workshop on Small Cell Wireless Networks (SmallNets)*, pp. 1-5, 2013.

[14] L. Scrucca, "GA: a package for genetic algorithms in R," *Journal of Statistical Software*, vol. 53, n. 4, pp. 1-36, 2013.

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

[16] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *The Journal of Machine Learning Research*, vol. 7, pp. 1-30, 2006.