Learning methods for long-term channel gain prediction in wireless networks

Federico Chiariotti, Davide Del Testa, Michele Polese,
Andrea Zanella, Giorgio Maria Di Nunzio, and Michele Zorzi
Department of Information Engineering (DEI), University of Padua, Italy
Email: {chiariot, deltesta, polesemi, zanella, dinunzio, zorzi}@dei.unipd.it

***Abstract*—Efficiently allocating resources and predicting cell handovers is essential in modern wireless networks; however, this is only possible if there is an efficient way to estimate the future state of the network. In order to accomplish this, we investigate two learning techniques to predict the long-term channel gains in a wireless network. Previous works in the literature found efficient methods to perform this prediction with the aid of a GPS signal: in this work, we predict the future channel gains using only past channel samples, without any geographical information.**

I. Introduction

Global mobile data traffic has grown 4,000-fold over the past 10 years [1], and its growth is expected to continue, fueled by the high bandwidth demands of multimedia applications (video alone already accounted for a majority of the total mobile data traffic in 2015).

As mobile multimedia applications have strict Quality of Service (QoS) requirements, the development of optimal resource allocation techniques is a priority for both industry and academia. Applications such as video streaming would greatly benefit from an accurate channel prediction, and prediction-based adaptive streaming systems have already been proposed [2]. Most of the efforts in channel prediction have focused on Multiple Input Multiple Output (MIMO) techniques on short time horizons, but several works [3] [4] have expressed the need for long-term accurate channel gain predictions.

In this paper, we investigate learning methods to predict the wireless channel gain on a long-term scale, without any inputs other than the time-averaged Received Signal Strength Indicator (RSSI). This prediction is performed by the Base Station (BS), which is the only fixed element in the network; the BS can indirectly learn the mobility patterns of the mobile users and the fading characteristics of the channel by observing patterns in their RSSI.

The two machine learning techniques we use are Graphical Bayesian (GB) models and Support Vector Regression machines (SVRs). The GB model can be used as a baseline, as it does not try to generalize its experience, but simply considers each class as a separate classification problem. SVRs are able to find and generalize patterns in the data, making better predictions with fewer data.

The rest of this work is organized as follows: Section II presents the state of the art in channel prediction, and Section III illustrates the learning techniques we used. Section IV presents the model we used to generate the data and the simulation results we obtained, and Section V concludes the paper.

II. State of the art

Wireless links are often modeled as Rayleigh fading channels. Most model-based prediction systems concentrate on short-term predictions of the fading envelope for wideband channels [5], and cannot be directly used for optimization at the higher layers. As mobility was not a central issue, the relatively predictable nature of path loss made long-range channel prediction an easy problem.

Shen *et al.* predict future channel quality from receiver-side Channel State Information (CSI) [6], but the Autoregressive (AR) filters they use are only accurate on a timescale of a few milliseconds. The work in [7] proposes an OFDM-specific prediction method based on time-domain statistics with a slightly longer range, but the timescales for accurate predictions are still far below 100 ms.

Another AR model is proposed by Jarinova [8], but its predictions are extremely short-range and how to choose the order of the filter is an issue.

Several works in the literature have attempted to solve the long-term channel prediction problem with machine learning techniques. A typical example is Ramanan and Walsh's channel prediction algorithm for sensor networks [9]. It is a distributed algorithm that employs message passing techniques to minimize the Kullback-Leibler divergence between the expected prior distribution and the actual posterior.

The problem of channel prediction is central in Cognitive Radio (CR) systems, and Demestichas *et al.* propose a Bayesian Network (BN) [10] to solve this issue. Their BN predicts the future capacity for each possible CR configuration, and adapts to channel conditions online, but it is meant for Modulation and Coding System (MCS) selection rather than for optimization at higher layers.

Flushing *et al.* take an empirical approach [11], combining a probing mechanism with Support Vector Regression (SVR) to predict link quality in dense wireless networks. Thanks to mobility, the probing system can learn about several different topologies and extend this knowledge to larger, denser networks.

Finally, Liao *et al.* perform long-term channel prediction [4] using both spatial and temporal information. The authors propose a Gaussian Process (GP) regression, training the system through a series of routes on a city map. Their prediction method is robust against spatial errors, with better performance than both Support Vector Machines (SVMs) and AR filters. Although their method is sound for large-scale scenarios, it does not deal with smaller cells and needs Global Positioning

System (GPS) information from clients, which might not be available.

III. Learning techniques

The two learning techniques we used are extremely versatile: they do not assume a specific channel model, so they can be trained and deployed on any wireless channel with only minor adjustments. In order to make the necessary data easy to obtain in a practical scenario, we decided to use the Received Signal Strength Indicator (RSSI) [12] as training data, averaging the data over a long enough window to reduce measurement error and avoid holes in the sampling even in a congested network.

A. Graphical Bayesian Model

The GB model can be represented by the graph shown in Fig. 1. As the Bayesian model only works for discrete attributes, the dynamic interval of the channel gain needs to be discretized into M classes. The memory-n Bayesian model uses the past n samples as features, resulting in M^n possible combinations. The predictor is essentially a classifier, in which the future channel sample is the correct class.

The multimodal classifier is implemented by a Dirichlet distribution [13] over the M-dimensional simplex, which is parameterized by a real non-negative vector α:

$$p(X=(x_1,\dots,x_M)|\alpha)=\frac{1}{B(\alpha)}\prod_{i=1}^{M}x_i^{\alpha_i-1}, \tag{1}$$

where the normalizing constant $B(\alpha)$ is the multivariate Beta function [14]. The random vector $X=(X_1,\dots,X_M)$ is a probability distribution over the M classes, representing the probability that the given sample is in each class (i.e., the probability distribution of the next channel sample). The expected value of X is given by:

$$\mathrm{E}[X_i]=\frac{\alpha_i}{\sum_{j=1}^{M}\alpha_j} \tag{2}$$

$$\mathrm{Var}[X_i]=\frac{\alpha_i\sum_{j\neq i}\alpha_j}{(\sum_{j=1}^{M}\alpha_j)^2(\sum_{j=1}^{M}\alpha_j+1)} \tag{3}$$

Intuitively, the value of α_i relative to the sum of the α vector is a measure of how probable a class is, and the value of the sum measures the uncertainty on that probability. The conjugate distribution of the Dirichlet distribution is the Dirichlet-multinomial distribution; Bayesian inference can be performed by generating a new parameter vector α', defined as

$$\alpha_i'=\alpha_i+n_i, \tag{4}$$

where n_i is the number of observed transitions to class i. The prediction can be performed by taking the expected probability distribution of the next sample, given by

$$p(i)=\frac{\alpha_i'}{\sum_{j=1}^{M}\alpha_j'} \tag{5}$$

The predicted class then corresponds to the maximum probability value.

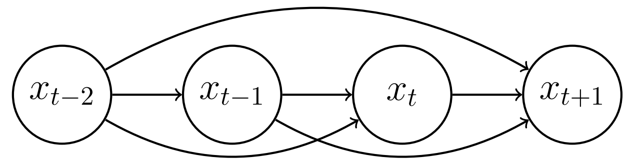


Fig. 1: Representation of the graphical Bayesian model with 3-state memory.

B. Support Vector Regression Machine

Support Vector Regression (SVR) machines [15], [16], minimize the following cost function:

$$C\sum_i E_\epsilon(f_{\mathbf{w}}(\mathbf{x}^{(i)})-x_{t+1}^{(i)})+\frac{1}{2}||\mathbf{w}||^2. \tag{6}$$

In (6), $f_{\mathbf{w}}$ is a function taking as input a memory-n feature vector $\mathbf{x}^{(i)}=(x_{t-n+1},\dots,x_t)$ and predicting a future sample $\hat{x}_{t+1}$, for a given training example i. The error between this predicted sample and the actual sample at time $t+1$, x_{t+1}, is then fed to an ϵ-insensitive error function

$$E_\epsilon(z)=\begin{cases}|z|-\epsilon & \text{if } \ |z|>\epsilon\\ 0 & \text{otherwise}\end{cases} \tag{7}$$

so that $f_{\mathbf{w}}$ is constrained to have maximum absolute prediction error lower than a given constant ϵ for all the training data. The second term in (6) accounts for regularization: the trade-off between the minimization of the two terms is governed by the constant C (the reader can refer to [17], [18] for more details). In (6), all the training examples are assumed to lie in an "ϵ-tube" (see Fig. 2). However, this is not verified in general, and (6) can be modified so as to allow for some tolerance in the prediction errors. Therefore, for each training example $\mathbf{x}^{(i)}$, it is possible to introduce *slack variables* ξ_i and ξ_i^*, where $\xi_i>0$ is related to a point for which $(x_{t+1}^{(i)}-f_{\mathbf{w}}(\mathbf{x}^{(i)}))>\epsilon$, and $\xi_i^*>0$ is related to a point for which $(f_{\mathbf{w}}(\mathbf{x}^{(i)})-x_{t+1}^{(i)})<-\epsilon$. Training examples are thus allowed to lie outside the ϵ-tube, as in Fig. 2, provided that the corresponding slack variables are positive: this condition can be formulated as

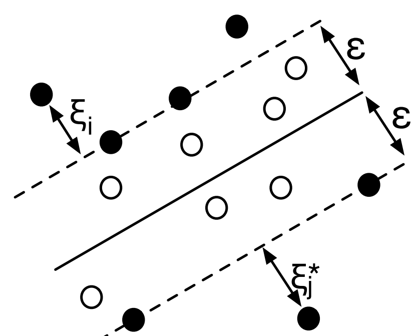


Fig. 2: Graphical representation of an ϵ-tube with slack variables.

$$-\epsilon-\xi_i^*\leq x_{t+1}^{(i)}-f_{\mathbf{w}}(\mathbf{x}^{(i)})\leq+\epsilon+\xi_i \tag{8}$$

The optimization problem becomes

$$\min C\sum_i(\xi_i+\xi_i^*)+\frac{1}{2}||\mathbf{w}||^2, \tag{9}$$

subject to the constraints $\xi_i, \xi_i^* \geq 0$, and (8). It can be seen that only the examples outside the ϵ-tube contribute to the cost, with deviations being linearly penalized. Computing the dual formulation of (9), exploiting the Karush-Kuhn-Tucker conditions [19], [20], and assuming that $f_{\mathbf{w}}$ is simply a linear function of the inputs, *i.e.*, $f_{\mathbf{w}}(\mathbf{x}^{(i)}) =< \mathbf{w}, \mathbf{x}^{(i)} > +b$, it can be found that

$$\mathbf{w} = \sum_i (\lambda_i - \lambda_i^*)\mathbf{x}^{(i)}, \tag{10}$$

were λ_i and λ_i^* are the Lagrange multipliers. The prediction function becomes

$$f_{\mathbf{w}}(\mathbf{x}) = \sum_i (\lambda_i - \lambda_i^*) < \mathbf{x}^{(i)}, \mathbf{x} > +b. \tag{11}$$

In (10), the weight vector $\mathbf{w}$ is a function of the training examples $\mathbf{x}^{(i)}$; however, only those examples such that $\lambda_i - \lambda_i^* \neq 0$, called *Support Vectors* (SVs), have to be evaluated in (10) and (11). Finally, it is possible to allow the prediction function $f_{\mathbf{w}}$ to be non linear in each training example $\mathbf{x}^{(i)}$, so as to allow better generalization over non linear target functions. In fact, in (11), the SVs only appear inside scalar products, and (10) does not need to be calculated explicitly. Therefore, it can be proved that $< \mathbf{x}^{(i)}, \mathbf{x} >$ in (11) can be replaced by particular non linear functions $k(\mathbf{x}^{(i)}, \mathbf{x})$, known as *kernels*, which correspond to scalar products between non linear transformations of $\mathbf{x}^{(i)}$ and $\mathbf{x}$. Substituting $k(\mathbf{x}^{(i)}, \mathbf{x})$ in (11), we thus obtain the optimal prediction function in a non-linear *feature space*, rather than in input space:

$$f_{\mathbf{w}}(\mathbf{x}) = \sum_{i=1}^{m} (\lambda_i - \lambda_i^*) k(\mathbf{x}^{(i)}, \mathbf{x}) + b. \tag{12}$$

IV. Simulation settings and results

The two learning methods were trained on the same RSSI data, generated by simulating a realistic urban scenario. The wireless channel we considered used a 945 MHz downlink carrier frequency (one of the commercial bands used in LTE), and the users moved in a Manhattan grid of 100 buildings.

The grid we used is composed of 20 m wide square buildings, with 10 m wide one-way streets at each corner. The BS is placed at coordinates $(140, 140)$, on top of a building close to the center of the simulation area.

A. Propagation loss and fading

The propagation loss was computed with the open-source system-level network simulator ns–3 [21]. In particular, we used the LTE module [22] and a radio propagation model called Hybrid Buildings Propagation Loss Model, which chooses the correct propagation model based on the reciprocal position of transmitter and receiver (both outdoors, both indoors, only one indoors). This model also takes into account the external wall penetration loss (for different types of buildings, i.e., concrete with windows, concrete without windows, stone blocks, wood), and the internal wall penetration loss.

We used the ns–3 simulation to create a square grid of path loss measures in our urban scenario, with a sampling distance of 33 cm. The path loss was then approximated as a linear combination of the 4 closest points in the grid, weighted by the relative distance. The main parameters of the ns–3 simulation are listed in Table I.

The fading and shadowing processes were both generated in MATLAB, implementing well-known models. We used the log-normal model for shadowing, with a standard deviation of 4 dB and a correlation distance of 8 m.

Doppler fading was modeled with a Rayleigh distribution, using the parameters listed in Annex B.2 of [23] and the MATLAB Welch periodogram method. In the fading calculation, the node speed was assumed to be constant, simplifying the computation significantly with negligible error.

B. Mobility model

We used two mobility models: *pedestrian* and *vehicular*. In both models, the user goes from point A to point B by choosing the direction that takes them closer to point B at each intersection.

In the *pedestrian* model, a person walks at a constant speed of 1.5 m/s along the side of the nearest building at a distance of 0.5 m. Road crossings are placed at each intersection, and the pedestrian waits for a random time between 0 and 5 seconds before crossing to wait for cars.

In the *vehicular* model, the driver keeps a constant speed of 15 m/s while driving straight, switching between the 3 lanes by moving at a 45 degree angle. Before a turn, the driver switches to the correct lane (e.g., they switch to the right lane before turning right), then slows down to 5 m/s with a constant deceleration in the 5 meters before the curve and makes a circular turn. After reaching the destination, the driver stops and reverses to slowly park on the curb, with a semi-circular trajectory.

The channel data was generated by running the urban scenario 5000 times for the pedestrian model and 10000 for the vehicular model, obtaining 3-4 days of data for the vehicular model and 20 hours for the pedestrian model (the car reaches its destination faster, so the traces are shorter). Two example trajectories for both models are shown in Fig. 3.

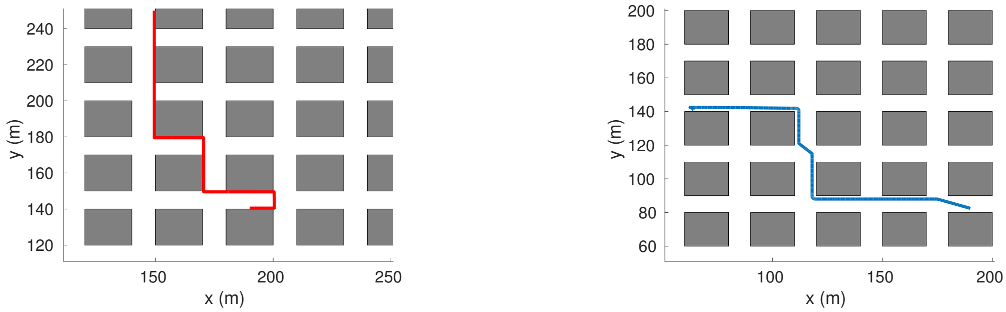
C. Learning parameters and results

Both prediction methods were trained on the full dataset, with two different sampling rates: the channel was averaged over a window of 1 s and 0.5 s.

The Bayesian model used a Gaussian prior, centered on the last known channel sample; the probability vector for all classes was multiplied by a factor k to obtain the Dirichlet parameter vector α. Both the prior weight factor k and the variance σ of the Gaussian distribution were optimized as hyperparameters by cross-validation. The channel quantization step used to divide the data into classes was 2 dB.

TABLE I: Path loss computation parameters

Parameter	Value
Downlink carrier frequency	945 MHz
Uplink carrier frequency	900 MHz
RB bandwidth	180 kHz
Available bandwidth	25 RB
eNB beamwidth	360° (isotropic)
TX power used by eNBs	43 dBm
eNB noise figure	3 dB
Number of buildings	100
Floors for each building	5
Radio Environment Map resolution	9 samples/m^2



(a) Pedestrian (b) Vehicular

Fig. 3: Examples of trajectories for the two mobility models.

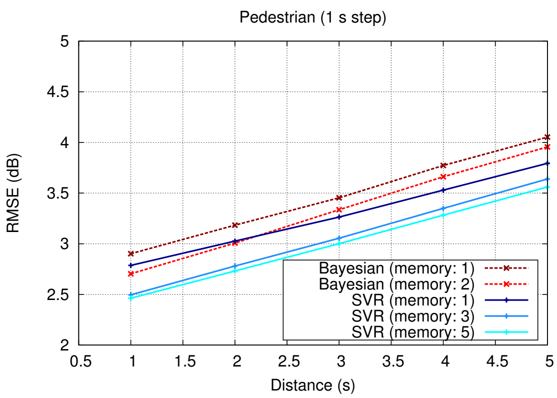


Fig. 4: Prediction error for the pedestrian scenario (1 s window).

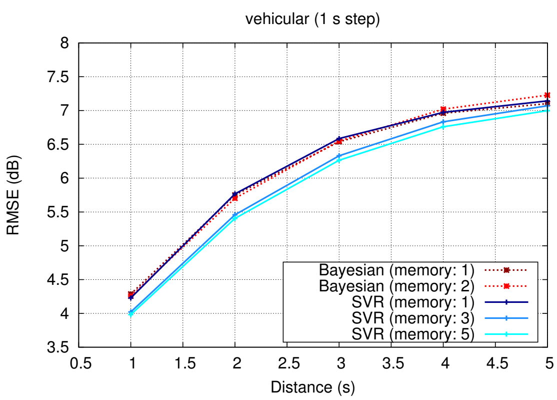


Fig. 5: Prediction error for the vehicular scenario (1 s window).

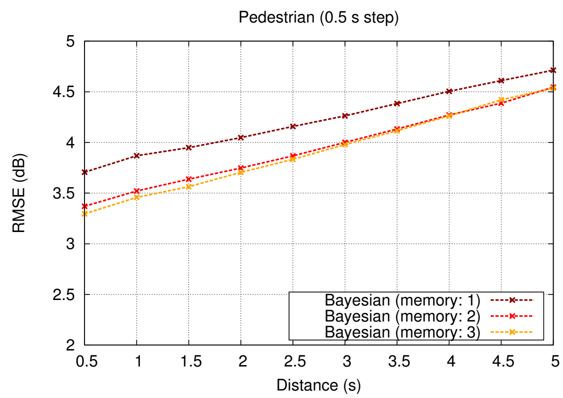


Fig. 6: Prediction error for the pedestrian scenario (0.5 s window).

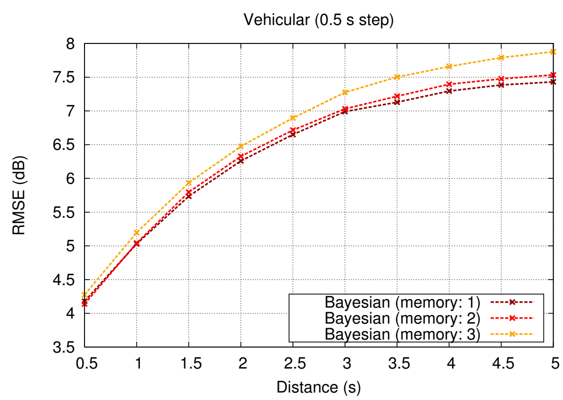


Fig. 7: Prediction error for the vehicular scenario (0.5 s window).

As regards the SVR learning algorithm, we found the Radial Basis Function (RBF) kernel: $k(\mathbf{z}_i, \mathbf{z}_j) = e^{-\gamma||\mathbf{z}_i - \mathbf{z}_j||^2}$ to perform best with respect to other possible kernel choices. In this case, the hyperparameters of the model are γ and C in (9): a grid search on (γ, C) pairs was thus performed, and the one with the best cross-validation RMSE was selected.

After cross-validation, the performance of both prediction methods was measured on a previously unknown test set.

Fig. 4 shows the prediction RMSE for the pedestrian mobility model; the quality of the prediction is very good even with the simpler model, as pedestrians are slow and generally highly predictable. As expected, SVR clearly outperforms the naive Bayesian model, as it is able to generalize its experience and to better capture the features of the model. The gain of the longer memory is less pronounced for the Bayesian model, as it is overshadowed by the small size of the dataset (a longer memory means that a bigger dataset is necessary, and the memory-3 Bayesian model is not plotted, as its performance is not better than that with memory 2).

In the vehicular scenario, the RMSE is higher and the performance gap between the two methods is smaller (see Fig. 5); the Bayesian model even outperforms the SVR if the prediction is more than 3 seconds ahead, but a prediction error of more than 7 dB is only slightly better than no prediction at all (the prediction RMSE when using a memoryless channel model is about 8 dB). This may be due to the high speed of the vehicles (~ 10 times the speed of the pedestrians), which makes accurate generalizations about the evolution of the channel hard.

Fig. 6 and Fig. 7 show the performance of the Bayesian

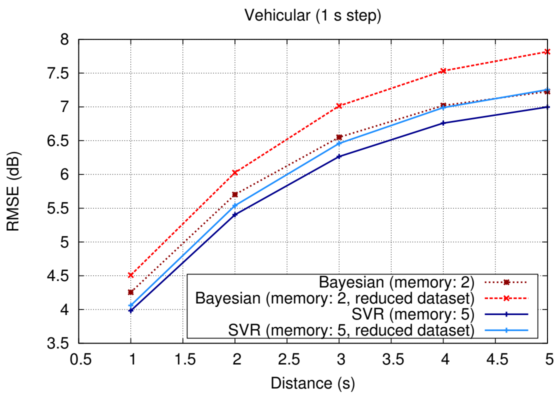


Fig. 8: Prediction error for the vehicular scenario (1 s window) including results using a reduced dataset.

method with a channel sampling window of 0.5 s; due to the computational cost of the SVR training, its performance in this case has not been evaluated. The figures show that the trend in the performance of the Bayesian method is essentially the same, although the error is higher; the performance of the memory-3 Bayesian model shows that a longer memory is beneficial for the pedestrian model, but loses most of its benefits in the more chaotic vehicular scenario unless a bigger dataset is used.

Finally, Fig. 8 shows the performance of the two predictors when they are trained with a reduced dataset: the two predictors were trained on 20% of the available data in the vehicular scenario with a 1 second step. The plot shows how the performance of the SVR degrades far less than that with the Bayesian model, thanks to the former's ability to generalize experience. In fact, the reduced-dataset SVR performs better than the full-dataset Bayesian model when the prediction distance is less than 5 seconds.

V. Conclusions

In this paper, we described and tested two learning-based methods to use past wireless channel information to predict the future channel gain. We compared the performance of the two methods over a synthetically generated dataset with random mobility for both pedestrian and vehicular scenarios.

The training was performed with just a few hours of RSSI data, so a BS with multiple connected users might be able to quickly gather the necessary training data and achieve a high-quality prediction in a very short time. However, the computational cost of the training itself is not negligible; while SVRs show a clear performance gain in both scenarios, the Bayesian model might be enough for applications that need a lower precision. It is worth noting that the SVR can have a satisfactory performance even when trained using a reduced dataset, as shown in Fig. 8; this makes it ideal if the limiting factor is not computational capability, but the size of the available dataset (e.g., in adaptive systems that are trained online to follow a time-varying scenario). The quality of the predictions is generally high, and the RMSE is almost as low as the results shown in [4], but without the use of GPS data, thereby reducing the energy consumption.

Future work may include a refinement of the prediction methods and the training of the predictors on data from real cellular systems. Finally, a promising development might include the creation of a prediction-based resource optimization system like the one presented in [2].

References

[1] Cisco Visual Networking Index, "Global mobile data traffic forecast update, 2015–2020," *Cisco White Paper*, 2016.

[2] S. Mekki and S. Valentin, "Anticipatory quality adaptation for mobile streaming: Fluent video by channel prediction," in *16th International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*. IEEE, 2015, pp. 1–3.

[3] S. Zhou and G. B. Giannakis, "How accurate channel prediction needs to be for transmit-beamforming with adaptive modulation over Rayleigh MIMO channels?" *IEEE Transactions on Wireless Communications*, vol. 3, no. 4, pp. 1285–1294, 2004.

[4] Q. Liao, S. Valentin, and S. Stanczak, "Channel gain prediction in wireless networks based on spatial-temporal correlation," in *16th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*. IEEE, 2015, pp. 400–404.

[5] L. Dong, G. Xu, and H. Ling, "Prediction of fast fading mobile radio channels in wideband communication systems," in *Global Telecommunications Conference (GLOBECOM)*, vol. 6. IEEE, 2001, pp. 3287–3291.

[6] Z. Shen, J. G. Andrews, and B. L. Evans, "Short range wireless channel prediction using local information," in *37th Asilomar Conference on Signals, Systems and Computers*, vol. 1. IEEE, 2003, pp. 1147–1151.

[7] I. C. Wong, A. Forenza, R. W. Heath, and B. L. Evans, "Long range channel prediction for adaptive OFDM systems," in *38th Asilomar Conference on Signals, Systems and Computers*, vol. 1. IEEE, 2004, pp. 732–736.

[8] D. Jarinová, "On autoregressive model order for long-range prediction of fast fading wireless channel," *Telecommunication Systems*, vol. 52, no. 3, pp. 1533–1539, 2013.

[9] S. Ramanan and J. M. Walsh, "Distributed estimation of channel gains in wireless sensor networks," *IEEE Transactions on Signal Processing*, vol. 58, no. 6, pp. 3097–3107, 2010.

[10] P. Demestichas, A. Katidiotis, K. A. Tsagkaris, E. F. Adamopoulou, and K. P. Demestichas, "Enhancing channel estimation in cognitive radio systems by means of bayesian networks," *Wireless personal communications*, vol. 49, no. 1, pp. 87–105, 2009.

[11] E. F. Flushing, J. Nagi, and G. A. Di Caro, "A mobility-assisted protocol for supervised learning of link quality estimates in wireless networks," in *International Conference on Computing, Networking and Communications (ICNC)*. IEEE, 2012, pp. 137–143.

[12] R.-H. Wu, Y.-H. Lee, H.-W. Tseng, Y.-G. Jan, and M.-H. Chuang, "Study of characteristics of RSSI signal," in *International Conference on Industrial Technology (ICIT)*. IEEE, 2008, pp. 1–3.

[13] D. Geiger and D. Heckerman, "A characterization of the Dirichlet distribution with application to Learning Bayesian Networks," in *11th Conference on Uncertainty in Artificial Intelligence*, ser. UAI'95. ACM, 1995, pp. 196–207.

[14] F. W. J. Olver, A. B. Olde Daalhuis, D. W. Lozier, B. I. Schneider, R. F. Boisvert, C. W. Clark, B. R. Miller, and B. V. Saunders, "Beta function," in *NIST Digital Library of Mathematical Functions*.

[15] V. N. Vapnik, *The Nature of Statistical Learning Theory*. New York, NY, USA: Springer-Verlag New York, Inc., 1995.

[16] H. Drucker, C. J. C. Burges, L. Kaufman, A. J. Smola, and V. Vapnik, "Support vector regression machines," *Advances in Neural Information Processing Systems*, pp. 155–161, 1997.

[17] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statistics and computing*, vol. 14, no. 3, pp. 199–222, Aug. 2004.

[18] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, Sep. 1995.

[19] W. Karush, "Minima of functions of several variables with inequalities as side constraints," Ph.D. dissertation, Masters thesis, Dept. of Mathematics, Univ. of Chicago, 1939.

[20] H. W. Kuhn and A. W. Tucker, *Nonlinear Programming*. University of California Press, 1951.

[21] G. F. Riley and T. R. Henderson, "The ns-3 network simulator," in *Modeling and Tools for Network Simulation*. Springer, 2010, pp. 15–34.

[22] N. Baldo, M. Miozzo, M. Requena-Esteso, and J. Nin-Guerrero, "An open source product-oriented LTE network simulator based on ns-3," in *14th international conference on modeling, analysis and simulation of wireless and mobile systems*. ACM, 2011, pp. 293–298.

[23] LTE ETSI, "Evolved Universal Terrestrial Radio Access (E-UTRA); Base Station (BS) radio transmission and reception (3GPP TS 36.104 version 8.6. 0 Release 8), July 2009," *ETSI TS*, vol. 136, no. 104, p. V8.