20th International Symposium INFOTEH-JAHORINA, 17-19 March 2021

Kernel based Extreme Learning Machines for Wireless Channel Prediction

Miloš Stojanović[a], Nikola Sekulović[a], Aleksandra Panajotović[b], Predrag Popović[c] and Milan Protić[a]
[a]College of Applied Technical Sciences, Niš, Serbia
[b]Faculty of Electronic Engineering, University of Niš, Serbia
[c]Faculty of Civil Engineering and Architecture, University of Niš, Serbia
milos.stojanovic@vtsnis.edu.rs, nikola.sekulovic@vtsnis.edu.rs, aleksandra.panajotovic@elfak.ni.ac.rs, predrag.popovic@gaf.ni.ac.rs, milan.protic@vtsnis.edu.rs

Abstract — **This paper proposes approach for prediction of wireless channel conditions of single-input single-output (SISO) systems in microcellular and picocellular environments using Kernel based Extreme Learning Machines (K-ELM). For evaluation of prediction quality normalized mean squared error (NMSE) and execution time are used. The simulation results show high prediction quality and short computing time of the K-ELM prediction model on measured values for the signal-to-noise ratio (SNR).**

Keywords – channel prediction; kernel based extreme learning machines; microcellular environment; picocellular environment

I. INTRODUCTION

Nowadays, wireless channel state prediction is even more important, because of increasing demands for high-data services and limited wireless spectrum. The enhancement of system performance can be achieved by using the channel prediction, instead of using the channel estimation [1]. Moreover, wireless channel state rapidly changes, so the channel state obtained by channel estimation can easily become outdated.

Many papers recently using prediction techniques instead estimation for channel states prediction. Autoregressive (AR) model, support vector machine (SVM), discrete wavelet transform (DWT) method in combination with AR and linear regression (LR) algorithm (DWT-AR-LR), echo state network (ESN) and extreme learning machines (ELM) are widely used in [2]-[8].

In this paper we propose Kernel based Extreme Learning Machines (K-ELM) model [9]-[10] for prediction of wireless channel conditions single-input single-output (SISO) systems in microcellular and picocellular environments, as an alternative to the often-used SVM model. The SVM training is based on solving the quadric programming problem, which is time consuming when dealing with large training set. Beside that SVMs are initially proposed for binary classification, and then reformulated for regression. On the other hand, K-ELM has the unique formulation for binary classification, multi class classification and regression. The model often shows better generalization performances and reduced training time, compared with SVM [10].

This paper proposes prediction model based on K-ELM for microcellular and picocellular environments. Data used for experiments consist of measured signal-to-noise ratio (SNR) samples used in [11]. Normalized mean squared error (NMSE) and the execution time are used for model evaluation.

The rest of the paper is organized as follows. Section II presents theory of K-ELM. Section III presents communication scenario, data sets and K-ELM model. The simulation results are discussed in Section IV, while Section V presents conclusion.

II. KERNEL BASED EXTREME LEARNING MACHINES (K-ELM)

Let us define N training examples as $(\mathbf{x}_j, \mathbf{y}_j)$ where $\mathbf{x}_j = [x_{j1}, x_{j2}, ..., x_{jn}]^T \in \mathbf{R}^n$ denotes *j-th* training instance of dimension n and $\mathbf{y}_j = [y_{j1}, y_{j2}, ..., y_{jm}]^T \in \mathbf{R}^m$ represents *j-th* training label of dimension m, where m is the number of classes. K-ELM has the unified solutions for regression, binary and multiclass classification. In the case of regression, which is of interest for problem considered in this paper, it holds that $m=1$ [10]. Single hidden layer feedforward neural network (SLFN) with the activation function $h(x)$ and L hidden neurons could be defined as:

$$\sum_{i=1}^{L} \beta_i h(\mathbf{w}_i \cdot \mathbf{x}_j + b_i) = \mathbf{f}_j,\ j = 1,\ldots,N \tag{1}$$

where $\mathbf{w}_i = [w_{i1}, w_{i2}, ..., w_{in}]^T$ denotes the vector of weights which connects the i^{th} hidden neuron and all input neurons, $\beta_i = [\beta_{i1}, \beta_{i2}, ..., \beta_{im}]^T$ is the weight vector which connects i^{th} hidden neuron and all output neurons, and b_i is the bias of the i^{th} hidden neuron. By ELM theory [9], $\mathbf{w}_i$ and b_i can be assigned in advance randomly and independently, without a priori knowledge of the input data. The ELM network structure is presented in Figure 1.

SLFN in (1) should satisfy $\sum_{i=1}^{L}\|\mathbf{f}_i - \mathbf{y}_i\| = 0$, i.e., there exist β_i, $\mathbf{w}_i$ and b_i such that:

$$\sum_{i=1}^{L}\beta_i h(\mathbf{w}_i\cdot\mathbf{x}_j+b_i)=\mathbf{y}_j,\ j=1,\ldots,N \tag{2}$$

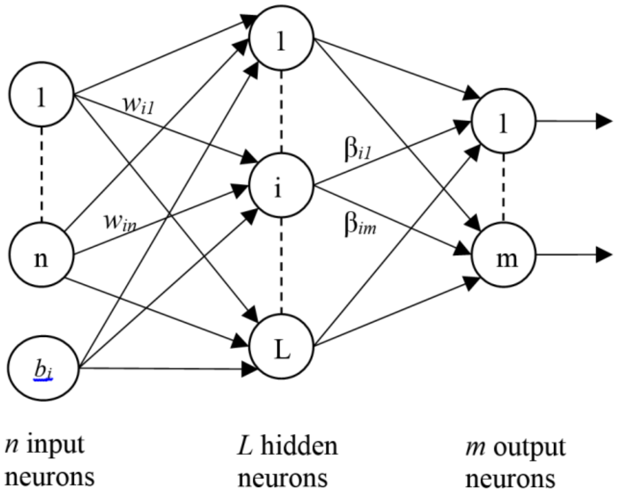

Fig. 1. Structure of an ELM network.

The equivalent compact matrix form of (2) can be written as

$$\mathbf{H}\boldsymbol{\beta}=\mathbf{Y} \tag{3}$$

where **H** in (3) represents the hidden layer output matrix of the neural network; the i^{th} column of **H** represents the i^{th} hidden neuron's output vector in regard to inputs $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N$.

$$\mathbf{H}=\begin{bmatrix} h(\mathbf{w}_1\cdot\mathbf{x}_1+b_1) & \cdots & h(\mathbf{w}_L\cdot\mathbf{x}_1+b_L) \\ \vdots & \cdots & \vdots \\ h(\mathbf{w}_1\cdot\mathbf{x}_N+b_1) & \cdots & h(\mathbf{w}_L\cdot\mathbf{x}_N+b_L)\end{bmatrix}_{N\times L} \tag{4}$$

and

$$\boldsymbol{\beta}=\begin{bmatrix}\boldsymbol{\beta}_1^T\\ \vdots \\ \boldsymbol{\beta}_L^T\end{bmatrix}_{L\times m} \text{ and } \mathbf{Y}=\begin{bmatrix}\mathbf{y}_1^T\\ \vdots \\ \mathbf{y}_N^T\end{bmatrix}_{N\times m} \tag{5}$$

Although the output weights can be analytically determined by finding the unique smallest norm least-squares solution of the linear system (3), the constrained optimization problem can be formed as in [10]:

$$Minimize: L_P=\frac{1}{2}\|\boldsymbol{\beta}\|^2+C\frac{1}{2}\sum_{i=1}^{N}\|\xi_j\|^2 \tag{6}$$

$$Subject\,to: h(\mathbf{x}_j)\boldsymbol{\beta}=\mathbf{y}_j^{\,T}-\xi_j^{\,T},\ j=1,\ldots,N$$

where $\xi_j=\left[\xi_{j1},\ldots,\xi_{jm}\right]^T$ is the training vector of the *m* output nodes with respect to the training sample x_i, while *C* represents the tradeoff parameter between the model complexity and allowed errors ξ_j during the training. Based on *Karush - Kuhn -Tucker* (*KKT*) theorem, the optimization problem defined in (6) is equivalent of solving the dual optimization problem:

$$\begin{aligned} L_D &= \frac{1}{2}\|\boldsymbol{\beta}\|^2+C\frac{1}{2}\sum_{j=1}^{N}\|\xi_i\|^2 \\ &-\sum_{j=1}^{N}\sum_{i=1}^{m}\alpha_{ji}\left(h(\mathbf{x}_j)\boldsymbol{\beta}_i-\mathbf{y}_{ji}+\boldsymbol{\xi}_{ji}\right)\end{aligned} \tag{7}$$

where $\alpha_j=\left[\alpha_{j1},\ldots,\alpha_{jm}\right]^T$ are Lagrange multipliers.

After solving (7) based on *KKT* conditions, which can be found in detail in [10], the following solution is obtained:

$$\beta=\left(\frac{\mathbf{I}}{C}+\mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{H}^T\mathbf{Y} \tag{8}$$

and the decision function of ELM is:

$$f(\mathbf{x})=h(\mathbf{x})\beta=h(\mathbf{x})\mathbf{H}^T\left(\frac{\mathbf{I}}{C}+\mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{Y} \tag{9}$$

If feature mapping $h(\boldsymbol{x})$ is unknown, we can apply Mercer's condition on ELM. We can define the kernel matrix for ELM as:

$$\begin{aligned}\boldsymbol{\Omega}_{ELM} &= \mathbf{H}\mathbf{H}^{\mathrm{T}} \\ \boldsymbol{\Omega}_{ELM\,i,j} &= h(\boldsymbol{x}_i)h(\boldsymbol{x}_j)=K(\boldsymbol{x}_i,\boldsymbol{x}_j)\end{aligned} \tag{10}$$

In ELM $\mathbf{H}=\left[h(\boldsymbol{x}_1)^{\mathrm{T}}\ \cdots\ h(\boldsymbol{x}_N)^{\mathrm{T}}\right]^{\mathrm{T}}$ represents hidden layer output matrix which maps data $\boldsymbol{x}_i$ from the input space to the hidden layer feature space and it is irrelevant to target values y_i and number of output nodes *m*. The kernel matrix $\boldsymbol{\Omega}_{ELM}=\mathbf{H}\mathbf{H}^{\mathrm{T}}$ is related only to input data $\boldsymbol{x}_i$ and the number of training samples *N*, for regression, binary classification and multi class classification.

Then, the output function of K-ELM (9) can be written compactly as:

$$f(\mathbf{x}) = \left(\begin{bmatrix} K(\boldsymbol{x}, \boldsymbol{x}_1) \\ \vdots \\ K(\boldsymbol{x}, \boldsymbol{x}_N) \end{bmatrix}^{\mathrm{T}} \left(\frac{\mathbf{I}}{C} + \Omega_{ELM} \right)^{-1} \mathbf{Y} \right) \quad (11)$$

In this case the feature mapping $h(\boldsymbol{x})$ does not need to be defined by users, as well as the dimensionality of the feature space L (number of hidden nodes), just its kernel $K(\boldsymbol{u},\boldsymbol{v})$. In our experiments Radial basis function (RBF) is used as a kernel, defined as:

$$K(\boldsymbol{u},\boldsymbol{v}) = \exp(-\gamma \|\boldsymbol{u} - \boldsymbol{v}\|^2) \quad (12)$$

where γ represents parameter of Gaussian kernel. It can be noted from (11) and (12) that optimal combination of parameters C and γ have to be obtained in order to achieve good generalization performance.

III. K-ELM PREDICTION MODEL

The evaluation of the proposed K-ELM prediction model is performed for wireless communication system with a single transmit antenna and a single receive antenna (SISO) with two different channels, B and E.

B channel is related to a microcell environment where distance between a mobile station (MS) and a base station (BS) is in the order of 30 m. It assumes indoor-to-outdoor propagation with BS located outside and indoor environment usually consisted of several small offices.

E channel is related to indoor-to-indoor scenario. It represents a picocell environment in modern open office with windows metallically shielded.

Let T and N denote sampling interval and total number of samples, respectively. Data used for simulation contain values of SNR obtained based on measurement campaigns described in details in [11]. A series of SNR samples $x(k) = x(kT),\ k = \overline{1,N}$, from [11], are used for K-ELM model formation.

K-ELM prediction model is formed as follows:

Training

(a) Obtain optimal (C, σ) pair on training set, based on grid search and *k*-fold cross validation using eq. (11) and eq. (12);

(b) Afterwards, compute Ω_{ELM} using eq. (10);

Testing

(a) Compute the prediction *f*(x) using eq. (11) and eq. (12).

IV. EXPERIMENTAL RESULTS

NMSE is used as a prediction error metrics which is defined as

$$NMSE = \frac{\sum_i \left(y_{measured}(i) - y_{predicted}(i) \right)^2}{\sum_i \left(y_{measured}(i) \right)^2}. \quad (7)$$

Data sets used to test the proposed method contain the measured instantaneous SNR values at the receiver side for both B and E channel model for the case when SNR at the transmitter side is 20 dB. Evaluation is performed using N=4000 samples. The data are divided into two equal sets for training and testing (N_{tr}=N_{te}=2000). Number of previous values used for predictions (lags) is set to 3, as proposed in [7]. RBF and linear kernel is used as kernel function. For the tests, K-ELM library is used, publicly available in [12].

In Fig. 1 and Fig. 2 target signal and prediction curve for both E and B channel are presented, respectively. Besides NMSE, which gives “one number” goodness of the fit, Fig. 1 and Fig. 2 show that predicted values strictly follow shape and trend of the time series, for both E and B channel.

The NMSE for test set is evaluated, as well as the training and the testing times measured in seconds, on an *Ryzen 3 3200G* CPU with 16GB of RAM. Table I contains NMSE, the training time and the testing times in seconds for K-ELM model trained with RBF and the linear kernel.

The obtained values of NMSE presented in Table I are in range of 0.01 to 0.003. In Table I, the training time is divided in two columns. Column denoted with *Training time 1* represents time needed for optimization of parameters (C, σ) on the training set by using the grid search and the *k*-fold cross validation procedure. Column denoted with *Training time 2* represents the time needed for computing Ω_{ELM}, once we know optimal (C, σ) pair. It should be noticed that in the most studies only *Training time 2* is given, while *Training time 1* is omitted, which gives optimistic assessment of training time. Test time is given in last column of Table I and it is in range of 0.007 to 0.06 seconds.

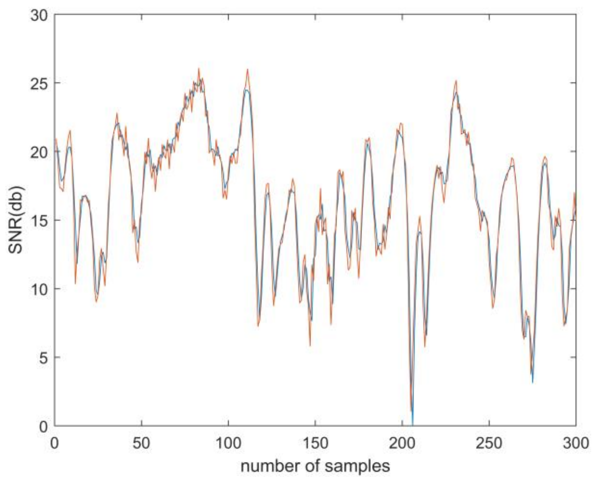

Figure 1. Target signal and prediction curve for E channel

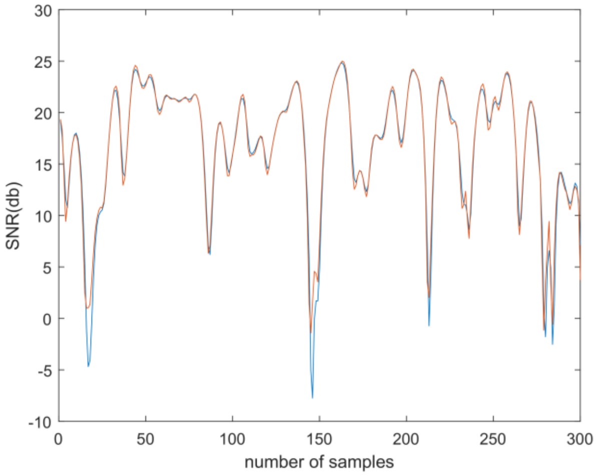

Figure 2. Target signal and prediction curve for B channel

The obtained values of the NMSE for K-ELM, regardless of the kernel used, are within the expected range of precision and comparable to the results obtained in other studies [5]-[8].

In order to compare the results of the K-ELM with other commonly used prediction techniques, we have measured accuracy of the ANNs and random forest (RF) on the same dataset. NMSE for ANNs was 0.0104 for E channel and 0.0058 for B channel, while NMSE for RF reached 0.0093 for E channel and 0.0049 for B channel. It can be noticed that RBF kernel K-ELM slightly outperforms ANNs and RF in terms of NMSE, having the similar algorithm complexity.

TABLE I. TIME CONSUMPTION OF THE K-ELM MODEL

Kernel	Channel	NMSE	Training time 1 (s)	Training time 2 (s)	Test time (s)
RBF	E	0.0085	12.5667	0.1322	0.0624
Linear	E	0.0126	0.9290	0.0929	0.0077
RBF	B	0.0038	12.6149	0.1325	0.0629
Linear	B	0.0086	1.0530	0.1053	0.0116

V. CONCLUSION

In this paper, the application of K-ELM prediction model for SISO systems in microcellular and picocellular environments has been proposed. The effectiveness of the approach has been confirmed using NMSE along with execution time. The obtained results are in rank with the results obtained in other studies.

Some future research could include detail analysis on how the training set selection and the feature selection influence on the prediction quality.

REFERENCES

[1] A. Panajotović, F. Riera-Palou, and G. Femenias, "Limited feedback MU-MIMO-OFDM system," *International ITG Workshop on Smart Antennas WSA 2016*, Munich, Germany, 9-11 March 2016, pp. 348-355.

[2] P. Sharma and K. Chandra, "Prediction of state transitions in Rayleigh fading channels," IEEE Trans. Veh. Technol., vol. 56, pp. 416-425, 2007.

[3] X. Zhao, C. Hou, and Q. Wang, "A new SVM-based modeling method of cabin path loss prediction," Int. J. Antennas and propagation, vol. 2013, pp. 1-7, April 2013.

[4] X. Long and B. Sikdar, "A wavelet based long range signal strength prediction in wireless networks," *IEEE Int. Conf. Commun.*, Bejing, China, 19-23 May 2008, pp. 2043-2047.

[5] Y. Zhao, H. Gao, N. C. Beaulieu, and Z. Chen, "Echo state network for fast channel prediction in Rician fading scenarios," IEEE Comm. Lett., vol. 21, pp. 672-675, 2017.

[6] N. Sekulović, A. Panajotović, M. Banđur, Đ. Banđur, and D. Milović, "Channel prediction in wireless microcell and picocell systems using echo state network," AJER, vol. 7, no. 12, pp. 134-138, 2018.

[7] N. Sekulović, M. Stojanović, A. Panajotović, and M. Banđur, "Extreme learning machines for wireless channel prediction in microcell and picocell environments," *ICEST 2019*, Ohrid, North Macedonia, 27-29 June 2019, pp. 82-85.

[8] M. Stojanović, N. Sekulović, and A. Panajotović, "A comparative performance analysis of extreme learning machine and echo state network for wireless channel prediction," TELSIKS 2019, Niš, Serbia, 23-25 October 2019, in press.

[9] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: Theory and applications," Neurocomputing, vol. 70, no. 1–3, pp. 489–501, 2006.

[10] G.-B. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme Learning Machine for Regression and Multiclass Classification," IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics, vol. 42, no. 2, pp. 513-529, 2012.

[11] J. P. Kermoal, L. Schumacher, K. I. Pedersen, P. E. Mogensen, and F. Frederiksen, "A Stochastic MIMO radio channel model with experimental validation," IEEE J. Select. Areas Commun., vol. 20, pp. 1211-1226, 2002.

[12] https://www3.ntu.edu.sg/home/egbhuang/elm_codes.html