# CSI Estimation Using Artificial Neural Network

Viraj Gajjar and Kurt Kosbar
Missouri University of Science and Technology
Rolla, MO, 65409
vgf4c@umsystem.edu, kosbar@mst.edu

## ABSTRACT

We propose using machine learning to estimate channel state information (CSI) for MIMO communication links. The goal is to use information such as atmospheric conditions, amount of path loss, and Doppler shift to improve the accuracy of CSI estimates. We start by designing an algorithm which estimates the CSI based on previously mentioned factors. Using this algorithm, we simulate a dataset of CSI over varying atmospheric conditions, receiver position, and receiver velocity. We then use this dataset to train an artificial neural network, which is able to estimate the CSI by using the current atmospheric condition, receiver position, and velocity.

## INTRODUCTION

In wireless communications, CSI describes how a signal propagates through the medium. Accuracy of the CSI estimates is one of the major factors which limits bit-error-rate (BER) and choice of modulation schemes. Techniques that can accurately estimate CSI are essential for many high-performance systems. CSI incorporates the effects of large-scale path loss, small-scale fading, and multipath [1], which on more fundamental level, depends on factors such as the distance between the transmitter and the receiver, weather conditions, velocity of the transmitter and receiver, carrier frequency, size of the scattering objects, type of reflecting surfaces. This paper investigates how a function can be obtained, that maps the factors affecting the wireless channel and the CSI using machine learning tools.

The recent growth in computational power has provided an opportunity to process large amounts of data, and learn complex relations from it, using machine learning (ML) algorithms. A class of ML algorithms known as supervised learning have the ability to learn a function that maps an input to the output based on available data, which consists of input-output pairs [2]. This method has been applied in the wireless communications to estimate channel noise statistics in multiple-input-multiple-output (MIMO) wireless networks [3]. In another application, it learned a mobile terminal's usage pattern in a cellular system to provide optimal handover solutions [4]. Other

applications include improvement in pilot contamination in cellular massive MIMO by learning channel parameters of the desired link in the target cell and undesired link in the adjacent cell [5], and to estimate CSI for 5G cellular communications using convolutional neural networks [6]. Artificial neural networks (ANNs) [7] are useful in solving problems related to function fitting, classification, time series prediction, and clustering. In regards to function fitting problems, ANNs have the ability to approximate any continuous function under certain assumptions [8]. Thus, machine learning (ML) techniques such as ANNs offer a possibility to find a complex function that can map the factors affecting the wireless channel to its CSI.

This paper presents an ANN based CSI estimation technique which uses simulated data to train an ANN for CSI estimation. Specifically, CSI data is simulated for a MIMO network across varying atmospheric conditions, and varying receiver position and velocity. Approximately two-thirds of the simulated data is used to train an ANN, which uses the distance between the transmitter and the receiver, velocity of the receiver, and the atmospheric conditions as features. The remainder of the data is used to test and validate the efficiency of the trained network.

## THE DATASET

For simulating the dataset, certain parameters that affect the propagation of a signal were considered, and are as follows:

### Free Space Path Loss

Free space path loss (PL) represents signal attenuation in dB for line-of-sight communications, and can be calculated using the following equation [1]:

$$PL\ (dB) = -10log\left[\frac{G_t G_r \lambda^2}{(4\pi)^2 d^2}\right], \tag{1}$$

where $G_t$ and $G_r$ are the transmitter and the receiver antenna gain respectively, $\lambda$ is the wavelength of the signal, and $d$ is the distance between the transmitting and receiving antenna.

### Doppler Shift

When the receiver is moving with respect to the transmitter, there will be an apparent change in frequency receiver known as Doppler shift $f_d$. The Doppler shift can be calculated using the following equation [1]:

$$f_d = \frac{v}{\lambda} cos\theta, \tag{2}$$

where $v$ is the velocity of the receiver, $\lambda$ is the wavelength, and $\theta$ is the angle of elevation from the receiver to the source. The Doppler shift is positive if the receiver is moving towards the transmitter and it is negative if the receiver is moving away from the transmitter.

**Rainfall Attenuation**

The attenuation of electromagnetic signals due to rain for frequencies up to 1,000 GHz can be approximated using the following equation [9]:

$$\gamma_R = kR^{\alpha}, \tag{3}$$

where $\gamma_R(dB/km)$ is the specific attenuation based on rain-rate $R(mm/hr)$, and coefficients $k$ and $\alpha$ depend on the frequency, the polarization state, and the elevation angle of the of the signal path.

**Atmospheric Gas Attenuation**

The attenuation of signal at frequencies up to 1,000 GHz due to dry air and water vapor, can be evaluated for given pressure, temperature and humidity using the following model [10]:

$$\gamma = \gamma_o + \gamma_w = 0.1820f\ N''(f), \tag{4}$$

where $\gamma\ (db/km)$ is the attenuation due to gases, $\gamma_o$ and $\gamma_w$ are the specific attenuations in $db/km$ due to dry air (oxygen, pressure-induced nitrogen, and non-resonant Debye attenuation) and water vapor, respectively, $f\ (GHz)$ is the frequency, and $N''(f)$ is the imaginary part of the complex atmospheric refractivity [10].

**Fog and Cloud Attenuation**

Attenuation within a cloud or fog is given as [11]:

$$\gamma_c(f,T) = K_l(f,T)M, \tag{5}$$

where $\gamma_c\ (dB/km)$ is the attenuation within the cloud, $K_l\ ((dB/km)/(g/m^3))$ cloud liquid water attenuation coefficient, $M(g/m^3)$ is the liquid water density, $f\ (GHz)$ is the frequency, and $T(K)$ is the cloud liquid water temperature.

**Dataset Generation**

A dataset of 1.5 million samples was generated considering these attenuation models for a 2-by-2 MIMO system as shown in Figure 1, using the following equation:

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{21} \\ h_{12} & h_{22} \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \end{bmatrix} \tag{6}$$

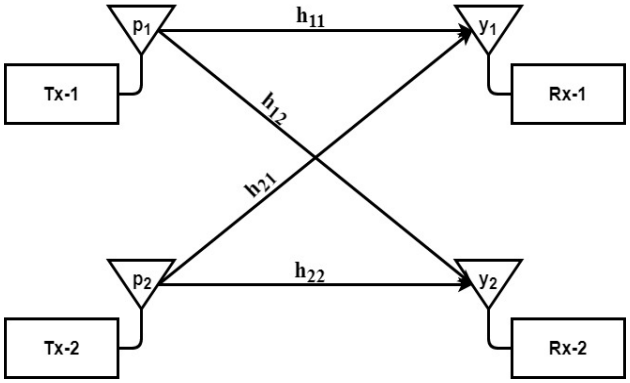


**Figure 1 A 2-by-2 MIMO System for Simulating the Dataset**

Where $p_1$ and $p_2$ are the pilot signals transmitted from transmitter-1 and transmitter-2 respectively; $h_{ij}$ is the channel coefficient from $i^{th}$ transmitter to $j^{th}$ receiver, where $i = 1,2, and\ j = 1,2$; $y_1$ and $y_2$ are the received signals at receiver-1 and receiver-2 respectively after being subjected to the attenuation models discussed earlier. In order to cover a range of possibilities, the parameters that affect the CSI were varied as stated in Table 1.

**Table1 Dataset Parameter Range**

| Parameter | Range |
|---|---|
| Distance between transmitter and receiver | $1\ km\ to\ 3\ km$ |
| Velocity of the receivers | $-10\ (m/s)\ to\ 10\ (m/s)$ |
| Temperature | $-30℃\ to\ 30℃$ |
| Carrier frequency | $1\ MHz\ to\ 30\ MHz$ |
| Rain-rate | $0\ (mm/hr)\ to\ 50\ (mm/hr)$ |
| Fog | $0.05\ (g/m^3\ )to\ 0.5\ (g/m^3\ )$ |
| Humidity | changed with respect to the temperature [12] |

## ARTIFICIAL NEURAL NETWORK

The dataset consists of two matrices, an input feature matrix and an output CSI matrix. The input feature matrix consists of 1.5 million samples of the eight following features: carrier frequency, the distance between the transmitter and receiver, velocity of receiver, angle of elevation/depression from transmitting antenna to the receiving antenna, temperature, rain-rate, fog-density, and humidity. The output CSI matrix also has 1.5 million samples of CSI, where the four channel coefficients of each sample are split into their respective real and imaginary parts; therefore, each sample in the CSI matrix is a vector of length eight. Thus, the neural network should have eight neurons at the input which takes a vector of features as input, and eight neurons at the output which estimates real and imaginary part of four channel coefficients.

Before training the ANN, the input feature matrix is normalized, and both the input and the output datasets are divided into a training set, a validation set, and a test set. The output dataset was scaled up by a factor of $10^4$, as the values of the channel coefficients were in the range of $10^{-2} - 10^{-10}$. Approximately 70% samples of the entire dataset and is used to train the ANN. The validation set forms 15% of the dataset, and is used to provide an unbiased evaluation of the training data fit while tuning model parameters. The remaining 15% of the dataset is the test set, which is used to evaluate the final model.

The overall architecture of the ANN trained is shown in Figure 2. The network consists of five layers - the input layer and the output layer with eight neurons each, and three hidden layers with 17 neurons each including one bias neuron.

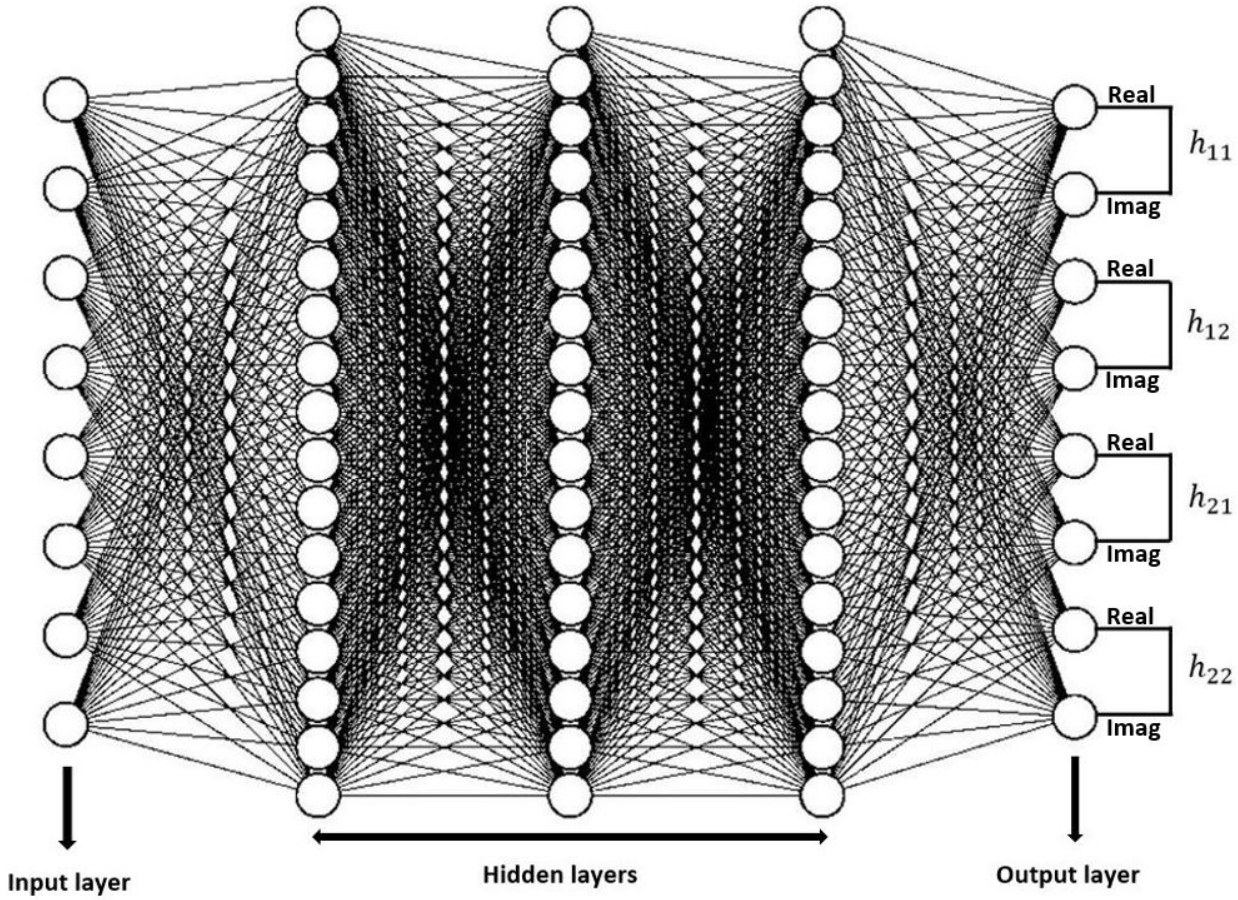


**Figure 2 ANN with an Input Layer, Three Hidden Layers, and an Output Layer**

All the weights (connections between each neuron) are randomly initialized before the training process starts. The training process starts by feeding a feature vector to the input layer and based

on this input the activations of the next layer are calculated. The network calculates the activations of each subsequent layer using the following equation:

$$A^{(1)} = \sigma(WA^{(0)} + b) \tag{7}$$

where $A^{(1)}$ and $A^{(0)}$ are activations of the output layer and the input layer respectively, $W$ is the matrix of weights connecting the input and the subsequent layer, $b$ is the bias vector, and $\sigma$ is the activation function. Equation (8) and Equation (9) are the activation functions used for the hidden layers and the output layer respectively. The linear function (Eq. 9) at the output layer is better at fitting a function compared to the tan-sigmoid function (Eq. 8).

After calculating the activations of each neuron, the cost function of the entire network is calculated by comparing the network outputs and actual outputs. This cost function is then used to calculate the gradient using Levenberg-Marquardt backpropagation [13], then the calculated gradient is used to perform gradient descent to update the weights. The network iterates through the entire dataset several times until either expected accuracy is achieved or the performance plateaus.

$$f(x) = \frac{2}{1 + e^{-2x}} - 1 \tag{8}$$

$$f(x) = x \tag{9}$$

## RESULTS AND DISCUSSION

The ANN was trained for more than 8,000 epochs, and then the network was tested on the test set which consisted of completely new samples which the network wasn't exposed to during the training stage. Before analyzing the performance of the algorithm, it should be noted that the outputs range in the order of $10^2$ to $10^{-6}$. Figure 3 shows the Mean Squared Error (MSE) of the entire training set over several epochs.

Before the training starts, the performance of the network is poor with the MSE being approximately 3,650. But once the network starts training the performance improves rapidly till 1,000 epochs, and the MSE drops to approximately 0.09.

However, the improvement in the performance isn't substantial after 1,000 epochs, and it only drops by roughly 0.06 over $6{,}000$ iterations. After that point, the improvement in the performance is negligible. The final performance of approximately 0.02 over the entire test set shows the ANN was able to closely fit a function that is able to map the input features to the CSI.

Figure 4 shows the error histogram with 20 bins, and illustrates how far the outputs deviate from the target values. As the histogram suggests, the majority of the samples are close to the zero error

for all the training, validation and test sets. The spread of the histogram, just like MSE, narrowed rapidly till 1,000 epochs, but then the error decreased gradually for the next 6,000 iterations.

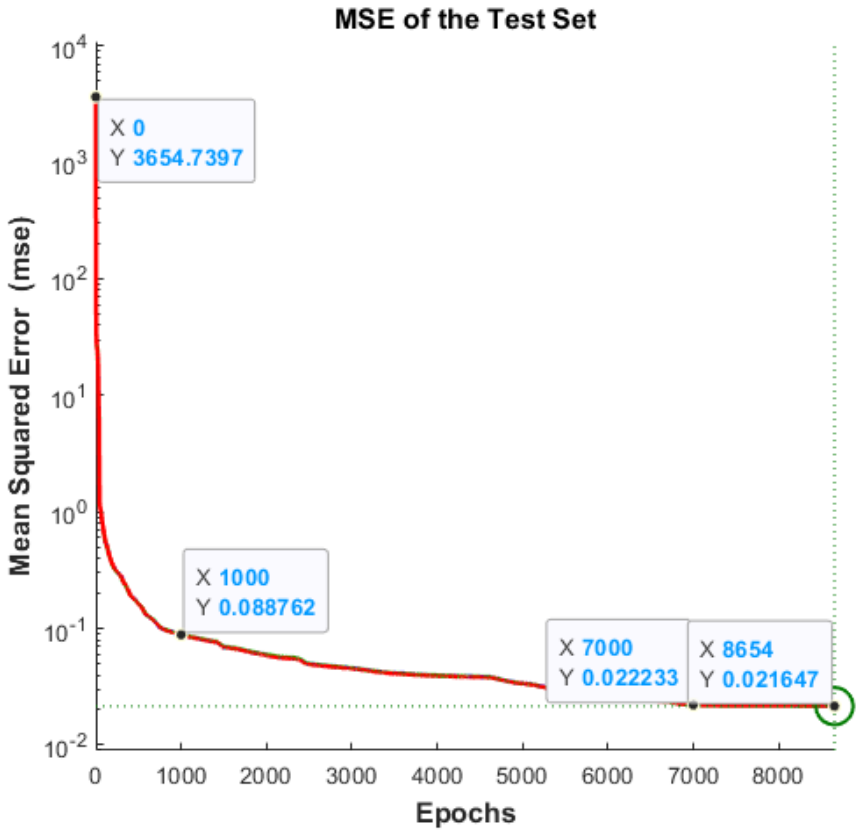


**Figure 3 MSE of the Test Plotted Against the Number of Epochs**

The amount of data used to train the ANN considerably affected the performance of the algorithm. The accuracy of the system increased with the size of the dataset till one million samples, but after that, there was only a marginal improvement in the performance. With smaller datasets the performance of the network was perfect for the training set, but the network struggled to perform on the test dataset. This meant the network did not have enough data to train, and was overfitting. Figure 5 demonstrates the effect of overfitting, as it can be seen that the neural network is able to fit a function across the test set, but as the function is specific to the training samples it fails to recognize test and validation samples.

The ANN was successful in learning a function that mapped the features to CSI. This was expected for a simulated dataset as the CSI was generated using several functions defining the amount of path loss, weather conditions, and the amount of Doppler shift. Therefore, the ANN was able to come up with a function that incorporated the effect of all the functions used to generate the dataset.

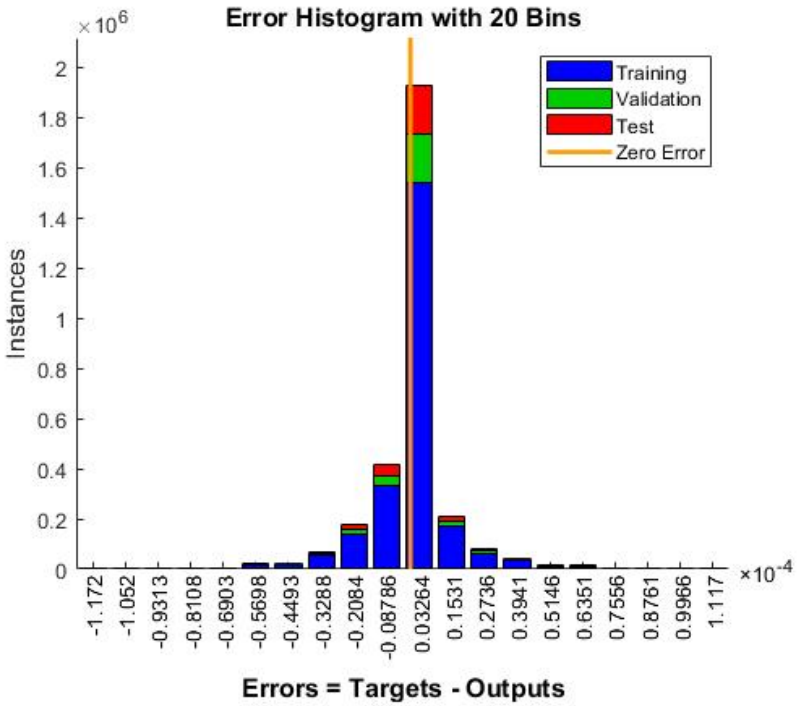


**Figure 4 Error Histogram**

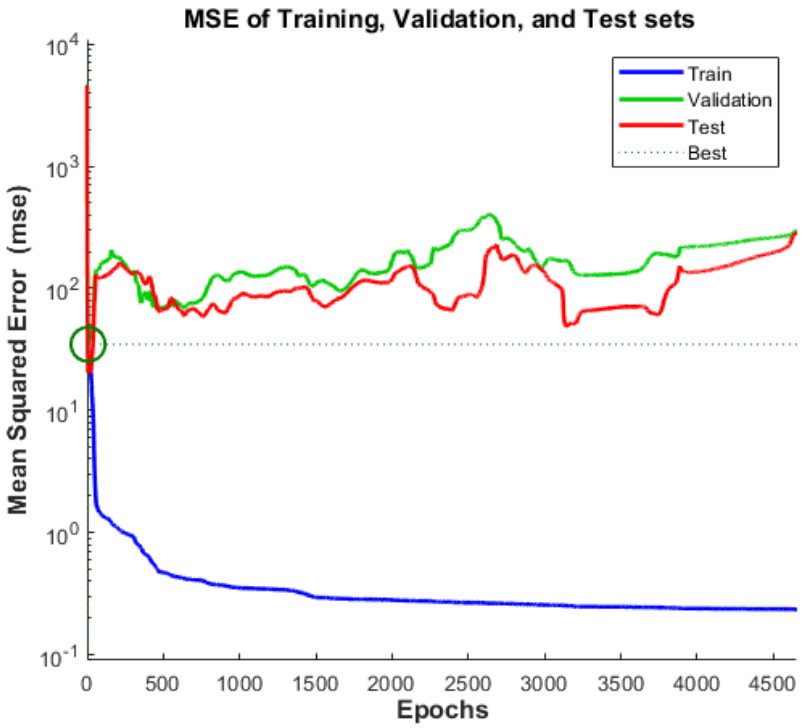


**Figure 5 MSE with a Small Dataset Demonstrates the Effect of Overfitting**

## CONCLUSIONS

Our simulations suggest that the relation between the CSI of a MIMO communication link and the parameters affecting it can be exploited using machine learning tools. We demonstrated that an ANN was able to fit a function that mapped features affecting CSI - atmospheric conditions, the distance between the transmitters and receivers, the amount of Doppler shift - to its estimates. Although the ANN was only tested on the simulated dataset, with sufficient data and a deep enough network, this method may be useful for real-world applications.

## REFERENCES


[1] Rappaport, T. S. (2002). *Wireless communications: Principles and practice*. (2nd ed.) (Prentice Hall communications engineering and emerging technologies series). Upper Saddle River, N.J: Prentice Hall.

[2] Stuart J. Russel, Peter Norvig (2010) *artificial Intelligence: A Modern Approach, Third Edition, Prentice Hall.*

[3] P. Zhou, Y. Chang, and J. A. Copeland, "Determination of Wireless Networks Parameters through Parallel Hierarchical Support Vector Machines," *IEEE Trans. Parallel Distrib. Syst.*, vol. 23, no. 3, Mar. 2012, pp. 505–12.

[4] B. K. Donohoo *et al.*, "Context-Aware Energy Enhancementsfor Smart Mobile Devices, *IEEE Trans. Mobile Comput.*, vol. 13, no. 8, Aug. 2014, pp. 1720–32.

[5] C.-K. Wen *et al.*, "Channel Estimation for Massive MIMO Using Gaussian-Mixture Bayesian Learning," *IEEE Trans. Wireless Commun.*, vol. 14, no. 3, Mar. 2015, pp. 1356–68.

[6] C. Luo, J. Ji, Q. Wang, X. Chen and P. Li, "Channel State Information Prediction for 5G Wireless Communications: A Deep Learning Approach," in *IEEE Transactions on Network Science and Engineering*.

[7] D. F. Specht, "A general regression neural network," in *IEEE Transactions on Neural Networks*, vol. 2, no. 6, pp. 568-576, Nov. 1991.

[8] Balázs Csanád Csáji (2001) Approximation with Artificial Neural Networks; Faculty of Sciences; Eötvös Loránd University, Hungary.

[9] Radiocommunication Sector of the International Telecommunication Union. *Recommendation ITU-R P.838-3: Specific attenuation model for rain for use in prediction methods*. 2005.

[10] Radiocommunication Sector of the International Telecommunication Union. *Recommendation ITU-R P.676-10: Attenuation by atmospheric gases*. 2013.

[11] Radiocommunication Sector of the International Telecommunication Union. *Recommendation ITU-R P.840-6: Attenuation due to clouds and fog*. 2013.

[12] Climate/humidity table. Retrieved from http:// www.tis-gdv.de/tis_e/misc/klima.htm

[13] G. Lera and M. Pinzolas, "Neighborhood based Levenberg-Marquardt algorithm for neural network training," in *IEEE Transactions on Neural Networks*, vol. 13, no. 5, pp. 1200-1203, Sept. 2002.