Title
A machine learning approach for localization in cellular environments

Authors
Abdallah, A.
Saab, S.
Kassas, Z.

Publication Date
2018-04-23

Peer reviewed

A Machine Learning Approach for Localization in Cellular Environments

Ali A. Abdallah and Samer S. Saab
Department of Electrical and Computer Engineering
Lebanese American University, Byblos
Byblos, Lebanon
ali.abdallah04@lau.edu, ssaab@lau.edu

Zaher M. Kassas
Department of Electrical and Computer Engineering
University of California, Riverside
Riverside, U.S.A.
zkassas@ieee.org

***Abstract*—A machine learning approach is developed for localization based on received signal strength (RSS) from cellular towers. The proposed approach only assumes knowledge of RSS fingerprints of the environment, and does not require knowledge of the cellular base transceiver station (BTS) locations, nor uses any RSS mathematical model. The proposed localization scheme integrates a weighted K-nearest neighbor (WKNN) and a multi-layer neural network. The integration takes advantage of the robust clustering ability of WKNN and implements a neural network that could estimate the position within each cluster. Experimental results are presented to demonstrate the proposed approach in two urban environments and one rural environment, achieving a mean distance localization error of 5.9 m and 5.1 m in the urban environments and 8.7 m in the rural environment. This constitutes an improvement of 41%, 45%, and 16%, respectively, over the WKNN-only algorithm.**

I. Introduction

Due to the weakness of received global navigation satellite system (GNSS) signals in urban canyons, substantial work has been devoted towards finding alternative localization methods. Signals of opportunity have been shown to be one of the best alternatives in GNSS-challenged environments [1]–[3]. Over the past two decades, cellular signals have attracted significant attention for localization due to their abundance, geometric diversity, large bandwidth, and high received power [4]. Recent research have demonstrated localization with cellular code-division multiple access (CDMA) and long-term evolution (LTE) signals with meter-level accuracy, by using specialized receivers that exploit the CDMA and LTE forward link channels [5]–[11].

The aforementioned localization approaches with cellular CDMA and LTE signals are based on time-of-arrival (TOA) or time-difference-of-arrival (TDOA), which tend to produce a precise position estimate. An alternative approach that is easier to implement is one that utilizes received signal strength (RSS), but it yields a less precise position estimate. RSS approaches use fingerprinting, where RSS from cellular BTSs is collected *a priori* at specific points, called reference points (RPs), which get saved in an offline database. When the user equipment (UE) enters the same cellular environment, it measures RSS to nearby BTSs in real-time, and matches RSS to the offline database to estimate its position [12], [13].

Several algorithms have been proposed to improve RSS-based localization. Methods to deal with RSS fluctuation issues have been proposed in [14]–[16]. A method to deal with noisy signals and path loss variations (specifically in indoor environments) was proposed in [17]. K-nearest neighbor (KNN) and weighted KNN (WKNN) improve the estimate by averaging different numbers of RPs' positions [18]. The advantage of WKNN is its simplicity. However, to produce an accurate estimate, it requires a very large database, and its performance deteriorates with RSS fluctuations. Neural networks have been considered for indoor localization using RSS from wireless local area networks (WLAN) [19]. Neural network approaches could yield relatively accurate position estimates without the need for a large database [20].

This paper proposes a machine learning localization approach that integrates both WKNN and a multi-layer neural network. This approach takes advantage of WKNN's clustering capability to yield a limited number of inputs that are adequate for the neural network within each cluster. The proposed approach estimates the location of the UE's cluster based on matching the RSS values from an offline RSS database and then refines the position estimate using the multi-layer neural network within the corresponding cluster. The proposed approach assumes no knowledge of the BTS location nor uses any RSS mathematical model. The proposed approach is tested experimentally in two urban environments and a rural environment. RSS data is collected with an Android smart phone and is used to build the offline fingerprinting database as well as to train the neural network. The mean distance localization error is shown to be 5.9 m and 5.1 m in the urban environments and 8.7 m in the rural environment. This constitutes an improvement of 41%, 45%, and 16%, respectively, over the WKNN-only algorithm.

The rest of the paper is organized as follows. Section II gives background about WKNN and multi-layer neural networks. Section III presents the proposed RSS-based localization algorithm that integrates WKNN and neural network. Section IV describes the environmental setup and data collection in the urban and rural environments. Section V analyzes the experimental results. Section VI gives concluding remarks.

This work was supported in part by the National Institute of Standards and Technology (NIST) under Grant 2017-3466.

II. Relevant Background

This section gives relevant background on WKNN and multi-layer perceptron neural networks.

A. Clustering and Weighted K-Nearest Neighbor

WKNN involves an offline stage and an online stage, which are described next.

1) Offline Stage: During the offline stage, several RPs $\mathbf{r}_i = [x_i,\, y_i]^\mathsf{T}$ are selected at which the RSS to neighboring cellular BTSs are measured and stored in an offline database, i.e.,

$$^{i}\mathbf{RSS}_{\text{off}} = [\text{RSS}_{\text{off},1}, \ldots, \text{RSS}_{\text{off},M}]^\mathsf{T}, \quad i = 1, \ldots, L, \quad (1)$$

where L is the number of RPs, $^{i}\mathbf{RSS}_{\text{off}}$ is the vector of RSS values at the i^{th} RP to all M neighboring cellular BTSs, and $\text{RSS}_{\text{off},m}$ is the RSS from the m^{th} cellular BTS.

The next step is to divide the environment into small clusters, where the probability of identifying the right cluster is used as a constraint in determining the number of clusters. The clustering algorithm is summarized in Algorithm 1.

Algorithm 1: Environment Clustering

Input: p = Probability of correctly identifying the right cluster, $\{\mathbf{r}_i\}_{i=1}^{L}$, and $\{^{i}\mathbf{RSS}_{\text{off}}\}_{i=1}^{L}$

Output: Clustered environment into j clusters

```
1  Set match = 0
2  Set j = 2
3  Divide the environment into j clusters, each containing
     the same number of RPs per cluster
4  For i = 1 to L
5      Calculate virtual distances between the i^th RP and the
6      other L − 1 RPs, i.e.,
7      d_i = ||^i RSS_off − ^ι RSS_off||_2,  ι ∈ {1, ..., L} \ i
8      Sort the virtual distances in ascending order
9      Find ^i r̂ = (Σ_{p=1}^{K} r_p/d_p) / (Σ_{p=1}^{K} 1/d_p)
10     If ^i r̂ and r_desired belong to the same cluster
11         Set match ← match + 1
12     end if
13 end for
14 If match/L > p
15     Set j ← j + 1
16     Go to step 3
17 else
18     Set j ← j − 1
19 end
```

---

2) Online Stage: During the online stage, the WKNN calculates the RSS "virtual distances" between the online measurements

$$\mathbf{RSS}_{\text{on}} = [\text{RSS}_{\text{on},1}, \ldots, \text{RSS}_{\text{on},M}]^\mathsf{T}. \quad (2)$$

and the offline RPs' values according to

$$d_i = ||\mathbf{RSS}_{on} - {}^{i}\mathbf{RSS}_{off}||_2, \quad i = 1, \ldots, L. \quad (3)$$

The estimate of the cluster coordinates is calculated according to

$$\hat{\mathbf{r}} = \frac{\sum_{i=1}^{K} \mathbf{r}_i/d_i}{\sum_{i=1}^{K} 1/d_i}. \quad (4)$$

Subsequently, K RPs are chosen according to the best RSS match, and each chosen RP is assigned a weight. The K in WKNN could be chosen based on several approaches, one of which is the graphical elbow method [21]. However, a rule of thumb in localization is to choose K to be equal the square root of number of clusters.

At this point, the WKNN has identified the corresponding cluster of the UE. Then, for each one of clusters, an approximation multi layer perceptron (MLP) is employed to estimate the final coordinates within the cluster itself. In the next subsection, this is discussed in details.

B. Multi-Layer Perceptron Neural Network

A neural network is an information processing paradigm that has been applied in many fields like pattern recognition, nonlinear mapping, approximation models, and data fusion. The MLP trained by a back-propagation algorithm is a sturdy tool for such applications. The MLP consists of an input layer, an output layer, and at least one hidden layer, where each layer consists of at least one neuron. The neurons are connected via weights that are considered the bulk of the trained neural network in order to estimate the desired output. Moreover, a three-layer neural network has the capability to approximate any nonlinear function [22]. This paper employs a three-layer MLP, depicted in Fig. 1.

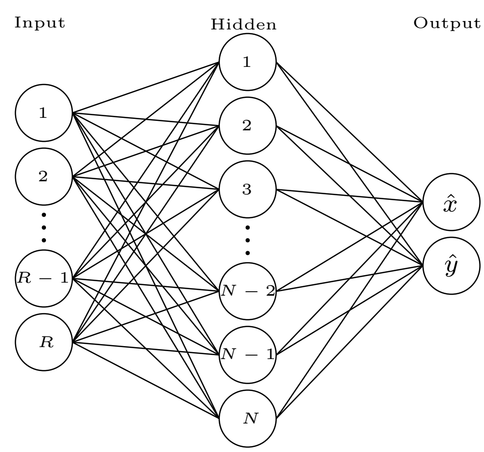

Fig. 1. The structure of proposed neural network for UE coordinate estimation

The number of neurons in the hidden layers is chosen to be the difference between the number of inputs and the number of outputs. The training of the neural network will be based on the back-propagation presented in [23], which achieves fast convergence. Back-propagation optimizes its weights and thresholds in order to minimize the sum of squared errors of the neurons in the output layer, i.e., minimizing the cost

$$C = \frac{1}{S}\sum_{i=1}^{S}({}^{i}\text{Output}_{\text{target}} - {}^{i}\text{Output}_{\text{est}})^2, \quad (5)$$

where ${}^{i}\mathrm{Output}_{\mathrm{target}}$ and ${}^{i}\mathrm{Output}_{\mathrm{est}}$ are the desired output (ground truth) and the estimated output (by the neural network) for each training sample, respectively, and S is the total number of RPs in the cluster. Also, the weights and thresholds of the neural network are updated using the 4 standard equations of back-propagation that are explained in details in [24]. These equations calculate the error at the level of the output layer according to (5) and then back-propagate this error to adjust the weights and thresholds based on a learning rate. The learning rates are generally set from 0.01 to 0.1 in order to sustain the stability of the network. The algorithm employs an adaptive learning rate to reduce the computations, while improving the overall performance.

III. Proposed Localization Approach

The proposed localization approach learns the environment offline, storing RSS fingerprints of RPs in an offline database. The area is divided into several clusters and a few RPs are strategically selected within each cluster, where each RP is expected to have a unique fingerprint. The RPs' positions and RSS average values over several time samples are recorded. The weights and biases of the neural network in each cluster are trained using the back-propagation approach by going over several locations at different times within each cluster, while feeding the neural network the locations and their associated RSS values. In the online phase, the WKNN is first used to identify the specific cluster and also estimate the location of the UE. Based on the identified cluster by WKNN, the neural network is loaded with the corresponding training weights that correspond to the identified cluster. Subsequently, the neural network estimates the UE's position based on the corresponding RSS values. A flowchart of the proposed algorithm is depicted in Fig. 2.

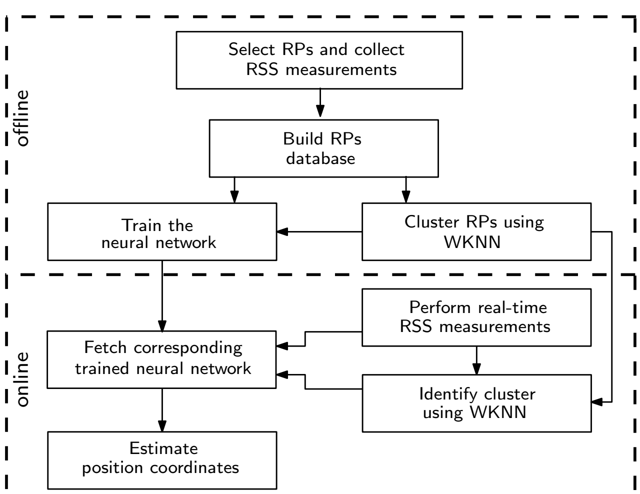

Fig. 2. Flowchart of proposed RSS-based localization algorithm with WKNN and neural network

As shown in Fig. 2, the process starts by selecting several RPs and measuring their respective RSS values. These measurements are used to build the offline database that will be used later for clustering. The clustering is based on the WKNN algorithm. Tuning is employed to divide the large cluster into smaller ones, taking into consideration that the smaller the cluster, the better is the performance, while ensuring the highest probability of identifying the corresponding cluster in the online stage (see Algorithm 1). In order to improve the confidence in the produced estimates, after the targeted area is clustered, RSS measurements are taken as training samples to the neural network. By using the training samples and based on the back-propagation algorithm, the neural network will be trained at the level of each small cluster. The next step involves online measurements, where RSS measurements are used to find the desired small cluster. As a result, the location estimate within the cluster depends on the final weights and thresholds of the trained neural network of the targeted cluster to finally estimate the position coordinates.

IV. Experimental Setup

A. Areas Selection

The performance of the proposed method is tested in two urban environments in the city of Beirut, Lebanon: (1) Hamra and (2) Beirut Central District (BCD), where each area correspond to one cluster. Hamra is a dense urban area where building heights range between 18–36 m, whereas the buildings in BCD are not as dense with heights between 9–18 m. The buildings in the rural area are sparse with heights ranging between 4–9 m. The following describes the RPs' selection in these environments:

1) Urban Environments:
   a) Cluster 1, Hamra: The covered area was 348 $\times 10^3\,\mathrm{m}^2$, containing 3133 RPs with 7 m spacing.
   b) Cluster 2, Beirut Central District (BCD): The covered area was 307 $\times 10^3\,\mathrm{m}^2$, containg ,890 RPs with 7 m spacing.
2) Rural Environment:
   a) Cluster 3: The covered area was 420 $\times 10^3\,\mathrm{m}^2$, containg 3300 RPs with 11 m spacing.

B. Data Collection

In the offline stage, an adequate number of RPs was selected for each cluster. Then, the RSS values of cellular signals were recorded from BTSs of the two cellular providers in Lebanon: Alfa and Touch. An Android smart phone was used to collect the raw RSS measurements. For each location, the RSS values were averaged over 30 s and 5 s in the urban and rural environments, respectively. The data was stored to train the neural network. A snapshot of the collected RPs is illustrated in Fig. 3.

C. Neural Network Training

The Neural Designer software was used to train the neural network [25]. In both urban environments and the rural environment, 60% of the data was used for training, 20% of the data was used for selection, and 20% of the data was used for testing. The training data was used to construct

(a)

(b)

(c)

Fig. 3. Snapshot of RPs in: (a) rural environment, (b) Hamra, and (c) BCD

different models. The selection data was used for choosing the predictive model with best generalization properties (remove the most incoherent samples that could confuse the network). The testing data was used to validate the model.The normalization method used for inputs was the mean standard deviation scaling. The activation functions for all neurons was the hyperbolic tangent function, and the loss index on which the network convergence was based was the root mean squared error. The number of input neurons for the MLPs were 57, 49, and 34, for Hamra, BCD, and rural environment, respectively. The number of neurons in the hidden layer was the number of input neurons minus the number of outputs, namely 2. Finally, MLPs were trained using the back-propagation approach and were tested to approximate the desired outputs using the testing samples. The performance of the proposed algorithm is compared with the WKNN-only algorithm.

D. Clustering

After finishing the data collection stage, one hundred RPs were selected from each cluster. In the online stage, RSS measurements were substituted in (3) and (4) along with the hundred chosen RPs from each cluster. Also, K was set to be constant in (4) as the nearest RP was chosen. Hence, the best matching RP'd coordinates determine the targeted cluster (see Algorithm 1). Experimental results and corresponding analysis are discussed in the next section.

V. Experimental Results

Three neural networks were trained with the collected data, one for each cluster covering the two urban environments and the rural environment. Table I shows the total number of RPs, number of training samples used for training (60%), area covered, number of iterations for training, and the gradient loss. The linear regression between the desired output and the estimated output is given in Table II, showing the regression coefficients in each cluster (in the form of ($\hat{x}$, x_{desired}) and ($\hat{y}$, y_{desired})). A strong relationship between the estimated outputs and the desired ones can be noticed in the urban environments, but such relationship is slightly weaker in the rural environment due to the large spacing used and lack of the unique RSS fingerprints.

TABLE I
Training Data for the Three Environments

Cluster	Number of RPs	Number of training samples	Covered area [$\times 10^3$ m^2]	Number of iterations	Final network loss
Hamra	3133	1880	348	6159	0.051
BCD	2890	1734	307	2657	0.042
Rural	3300	1980	420	15956	0.097

TABLE II
Output Linear Regression of the 3 Neural Networks Associated with the 3 Environments

Environment	Hamra	BCD	Rural
Regression Coefficients	(0.998, 0.996)	(0.997, 0.992)	(0.966, 0.952)

To test the trained neural networks in the Hamra, BCD, and rural environment, a total of 627, 578, and 660 RPs were used, respectively, which constitute 20% of the RPs in the data set. Fig. 4 shows the cumulative distribution function (CDF) of the localization error in the two urban environments and the rural environment according to the proposed approach versus that of the WKNN-only approach. It can be seen that the proposed algorithm outperforms the WKNN-only approach in all environments. The mean distance errors due to the proposed approach versus the WKNN-only approach are summarized in Table III. Note that the WKNN in Hamra and BCD show that the smallest area that attains a probability of 99% in identifying the correct cluster was $392\,\mathrm{m}^2$.

TABLE III
Mean Distance Localization Error (m)

Environment	Hamra	BCD	Rural
WKNN-only	8.3	7.4	10.1
Proposed	5.9	5.1	8.7

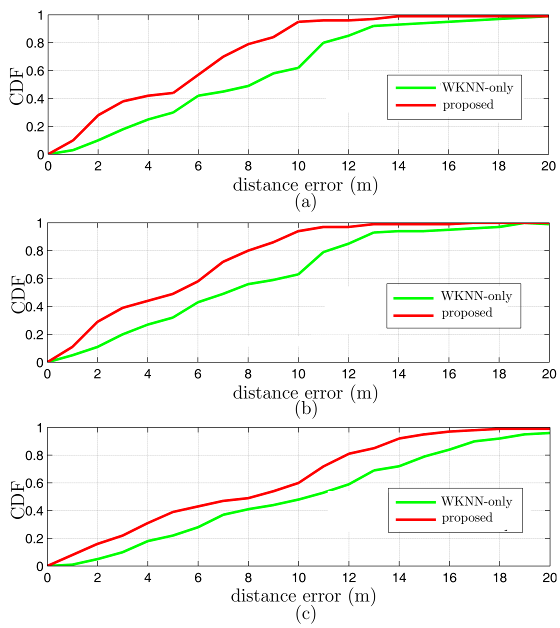

Fig. 4. CDF of localization error in three environments: (a) Hamra, (b) BCD, and (c) rural environment

VI. Conclusion

This paper presented an RSS-based localization approach that only requires RSS fingerprinting of RPs in the environment. The proposed approach employs an integrated WKNN algorithm and a multi-layer neural network in order to estimate the location of the UE. RSS measurements from cellular BTS signals were collected along with the respective locations of several RPs. A percentage of the collected RPs was used to train using the back-propagation algorithm, which approximates the nonlinear relationship between location coordinates and RSS values. Experimental results of the proposed approach yielded a mean localization error of 5.9 m and 5.1 m in two urban environments, and 8.7 m in a rural environment, which constitute an improvement of 41%, 45%, and 16%, respectively, over the WKNN-only approach.

VII. Acknowledgment

The authors would like to thank Hussein Hazime for his help in developing the Android application that was used in data collection. The authors would also like to thank Sarah Abu Sahyoun and Bashar Deeb for their help in data collection.

References

[1] L. Merry, R. Faragher, and S. Schedin, “Comparison of opportunistic signals for localisation,” in *Proceedings of IFAC Symposium on Intelligent Autonomous Vehicles*, September 2010, pp. 109–114.

[2] K. Pesyna, Z. Kassas, J. Bhatti, and T. Humphreys, “Tightly-coupled opportunistic navigation for deep urban and indoor positioning,” in *Proceedings of ION GNSS Conference*, September 2011, pp. 3605–3617.

[3] Z. Kassas, “Collaborative opportunistic navigation,” *IEEE Aerospace and Electronic Systems Magazine*, vol. 28, no. 6, pp. 38–41, 2013.

[4] Z. Kassas, J. Khalife, K. Shamaei, and J. Morales, “I hear, therefore I know where I am: Compensating for GNSS limitations with cellular signals,” *IEEE Signal Processing Magazine*, pp. 111–124, September 2017.

[5] C. Yang, T. Nguyen, and E. Blasch, “Mobile positioning via fusion of mixed signals of opportunity,” *IEEE Aerospace and Electronic Systems Magazine*, vol. 29, no. 4, pp. 34–46, April 2014.

[6] J. del Peral-Rosado, J. Parro-Jimenez, J. Lopez-Salcedo, G. Seco-Granados, P. Crosta, F. Zanier, and M. Crisci, “Comparative results analysis on positioning with real LTE signals and low-cost hardware platforms,” in *Proceedings of Satellite Navigation Technologies and European Workshop on GNSS Signals and Signal Processing*, December 2014, pp. 1–8.

[7] F. Knutti, M. Sabathy, M. Driusso, H. Mathis, and C. Marshall, “Positioning using LTE signals,” in *Proceedings of Navigation Conference in Europe*, April 2015, pp. 1–8.

[8] M. Ulmschneider and C. Gentner, “Multipath assisted positioning for pedestrians using LTE signals,” in *Proceedings of IEEE/ION Position, Location, and Navigation Symposium*, April 2016, pp. 386–392.

[9] J. Khalife, K. Shamaei, and Z. Kassas, “Navigation with cellular CDMA signals – part I: Signal modeling and software-defined receiver design,” *IEEE Transactions on Signal Processing*, 2017, accepted.

[10] J. Khalife and Z. Kassas, “Navigation with cellular CDMA signals – part II: Performance analysis and experimental results,” *IEEE Transactions on Signal Processing*, 2017, accepted.

[11] K. Shamaei, J. Khalife, and Z. Kassas, “Exploiting LTE signals for navigation: Theory to implementation,” *IEEE Transactions on Wireless Communications*, vol. PP, no. 99, pp. 1–1, 2018.

[12] V. Seshadri, G. Zaruba, and M. Huber, “A bayesian sampling approach to in-door localization of wireless devices using received signal strength indication,” in *Proceedings of IEEE International Conference on Pervasive Computing and Communications*, March 2005, pp. 75–84.

[13] C. Feng, W. Au, S. Valaee, and Z. Tan, “Review on determining number of cluster in k-means clustering,” *IEEE Transactions on Mobile Computing*, vol. 11, no. 12, pp. 1983–1993, December 2012.

[14] C. Yan and H. Fan, “Asynchronous differential TDOA for non-gps navigation using signals of opportunity,” in *Proceedings of IEEE Symposium on Computers and Communications*, June 2006, pp. 1530–1346.

[15] S. Ergen, H. Terikol, M. Kontik, R. Sevlian, R. Rajagopal, and P. Varaiya, “RSSI-fingerprinting-based mobile phone localization with route constraints,” *IEEE Transactions on Vehicular Technology*, vol. 63, no. 1, pp. 423–428, January 2014.

[16] C. Chitraranjan, A. Denton, and A. Perera, “A complete observation model for tracking vehicles from mobile phone signal strengths and its potential in travel-time estimation,” in *Proceedings of IEEE Vehicular Technology Conference*, September 2016, pp. 1–7.

[17] K. Saab and S. Saab, “Application of an optimal stochastic newton-raphson technique to triangulation-based localization systems,” in *Proceedings of IEEE/ION Position, Location, And Navigation Symposium*, April 2016, pp. 981–986.

[18] V. Moghtadaiee, A. G. Dempster, and B. Li, “Accuracy indicator for fingerprinting localization systems,” in *Proceedings of IEEE/ION Position, Location, And Navigation Symposium*, April 2012, pp. 1204–1208.

[19] Y. Xu and Y. Sun, “Neural network-based accuracy enhancement method for WLAN indoor positioning,” in *Proceedings of IEEE Vehicular Technology Conference*, September 2012, pp. 1–5.

[20] G. Chen, Y. Zhang, L. Xiao, J. Li, L. Zhou, and S. Zhou, “Measurement-based RSS-multipath neural network indoor positioning technique,” in *Proceedings of IEEE Canadian Conference on Electrical and Computer Engineering*, May 2014, pp. 1–7.

[21] T. Kodinariya and P. Makwana, “Review on determining number of cluster in k-means clustering,” *International Journal of Advance Research in Computer Science and Management Studies*, vol. 1, no. 6, pp. 90–95, November 2013.

[22] K. Hornik, M. Stinchcombe, and H. White, “Multilayer feedforward networks are universal approximators,” *Neural Networks*, vol. 2, no. 5, pp. 359–366, 1989.

[23] D. Rumelhart, G. Hinton, and R. Williams, “Learning representations by back-propagating errors,” *Nature*, vol. 323, pp. 533–536, October 1986.

[24] M. Hagan and M. Menhaj, “Training feedforward networks with the marquardt algorithm,” *IEEE Transactions on Neural Networks*, vol. 5, no. 6, pp. 989–993, November 1994.

[25] Neural designer software. [Online]. Available: https://www.neuraldesigner.com/