


Automatic Positioning of Mobile Users via GSM Signal Measurements

Ercan Demir and Abdulkemir Öztekin


Abstract—Today the need for mobile communication systems and the high increase in the number of users have also made the development of new generation mobile applications indispensable. Obtaining location information has been one of the most interesting and significant areas of improvement. The purpose of the services used to determine the location is generally to obtain the information of the users such as approximate location, speed, and time. The GPS is the most preferred and globally accurate positioning system among global positioning systems. However, in addition high installation cost of the system; galactic and meteorological factors, high buildings, other physical obstacles, and especially indoor areas are some of the main constraints that can lead to serious signal degradation and losses which may cause the system to be out of service. In this context, there is an urgent need for positioning systems that will be alternative and complementary to global positioning systems. The cellular network is widely used by almost everyone and its coverage area is increasing day by day. The network has been trained and tested in the simulation environment using machine learning algorithms, namely, extreme learning machine (ELM), generalized regression neural network (GRNN), and k nearest neighborhood (kNN). When compared to other cellular localization methods in the literature, the proposed system performs positioning with much higher accuracies with distance error rates below a meter (m) at minimum, and between 76-216 m on average. The test results show that it can successfully localize the mobile users with a significant accuracy for indoor, where GPS signals are very weak or cannot be received at all; and it can also stand in the breach for outdoor, where GPS may be disabled for different reasons.

Index Terms—GSM positioning, GPS, localization, machine learning.

ERCAN DEMİR is with Department of Electrical Electronics and Engineering Batman University, Batman, Turkey, (e-mail: ercandemir23.ed@gmail.com).

 <https://orcid.org/0000-0002-3234-8728>

ABDULKERİM ÖZTEKİN is with Department of Electrical and Electronics Engineering Batman University, Batman, Turkey, (e-mail: abdulkemirimoztekin@gmail.com).

 <https://orcid.org/0000-0002-0698-3525>

Manuscript received January 04, 2021; accepted April 27, 2021.
DOI: [10.17694/bajece.852963](https://doi.org/10.17694/bajece.852963)

I. INTRODUCTION

THE HIGH interest in mobile communication systems triggers the development of new technologies and applications. Geolocation techniques are one of the most important developments in this field. Positioning techniques enable the position estimation of people, mobile devices or equipment. Global positioning systems are the most widely used positioning systems in many areas in today's technology. Global geolocation systems are mainly used in military fields (fighter jets, smart weapons and bombs, positioning vehicles and equipment) and scientific studies (geological studies, remote sensing research, geophysical measurements, cartography, etc.). On the other hand, such positioning systems are widely used in many transportation systems, mining activities, all kinds of security applications, especially in search and rescue operations, agricultural activities, and sports activities [1].

Among the satellite systems used for global positioning, the most widely used positioning system with the widest user mass is the global positioning system (GPS). This system developed by the USA was used for military purposes in the early days. The GPS global positioning system consists of two main parts: satellites positioned in the earth orbit and ground stations that control these satellites. Such global positioning systems are costly to install. However, these systems may not work because sufficient signal levels cannot be generated in closed areas such as tunnels, under bridges, and inside buildings, as well as blind spots created by obstacles. In addition to the high number of complex, multi-storey, and wide-spread buildings in today's modern settlements, this number is increasing day by day. Due to the fact that the buildings consist of tens of floors, hundreds of rooms, and corridors, they are both complex and the number of population they accommodate gives the appearance of small cities [2]. The inadequacy of global positioning systems such as the inability to work in such closed areas indicates the need for effective positioning systems that can also work in closed areas.

In this context, the Global System for Mobile Communications (GSM-Global System for Mobile Communications) emerges as a good alternative in determining the location of the mobile user. In addition to the increasing need for daily use of mobile devices used for communication, their popularity has also increased day by

day. In addition to the fact that cellular base stations are more and more frequent in metropolitan areas, the coverage area is rapidly increasing to include rural areas. However, the ability to receive signals from cellular base stations even in closed areas and in worse physical conditions brings the GSM-based positioning system one step further to determine the location of the mobile user.

There are different studies in the literature for cellular network-based location determination. Magro et al. have estimated the user's location with an average margin of error of 324 m using a genetic algorithm exploiting network parameters, such as cell ID and transmitted signal strength that can be accessed by mobile devices in 3G networks [3]. Türkyılmaz has proposed an environment aware location estimation method based on signal strength measurements for cellular networks, where it has been reported that the average error has been reduced from 642 m to 573 m (10.75% improvement), the standard deviation from 689 m to 481 m (30.19% improvement), and the maximum error from 4762 m to 2638 m (44.6% improvement) [4].

In the study conducted by Kurt, a positioning system using the fingerprint in cellular networks has been employed using received signal strengths (RSS) collected from mobile stations in two different regions, resulting in an average distance error of 435 m and 405 m, respectively [5]. Fritsche et al. have worked on obtaining approximate location information over the GSM network in case of interruption of GPS information [6]. A geolocation algorithm has been developed using Kalman filter exploiting the information with the help of both GPS satellites and measured information from the GSM network. Xuereb and Debono have proposed a mobile terminal location estimation using support vector machines with an error margin of 314m [7].

In this paper, a study has been carried out to obtain approximate location information via cellular networks. A data set including the location information of these base stations together with a sufficient number of real signal measurements obtained from different GSM base stations from different points of the region, both indoor and outdoor, was created by means of a mobile application that have been developed in the Android Studio environment. In Matlab simulation environment, the training of the network has been provided with this obtained data set using machine learning methods: extreme learning machine (ELM), generalized regression neural network (GRNN) and k nearest neighbors (kNN) algorithm. Thus, a study was conducted to obtain approximate real location information by using signal measurements on the test data.

The study consists of four sections. After this introduction section, in the second section, a brief background on extreme learning machines, generalized regression neural network and k nearest neighbor algorithm are given. The third section includes the application findings. The last section concludes the paper by discussion and suggestion of further studies.

II. MATERIALS AND METHOD

A. Collection of the Data Set

The data used in this study consists of signal measurement records obtained from indoor and outdoor locations in a certain region in the city center of Siirt (a province in Turkey) via a convenient and easy-to-access mobile application that we have developed in the Android Studio environment. The developed application has been installed and runs on an android based mobile phone to collect and record data. The map of the scanned area where the data collection process has been taken is shown in Figure 1, using the Google Earth application.



Fig.1. Google Earth output illustrating the scanned area for data collection

The collected data consists of the actual signal strengths measured in dBm from different base stations of the serving cellular network and the geographic coordinates (i.e., latitude and longitude) of the currently serving cell location taken via internet and the actual position of the mobile user taken via GPS, which have been recorded instantly at certain time intervals that can be easily adopted according to the relative speed of the mobile user to ensure sufficient amount of data. The record has been arranged in a suitable form and has been purified from redundant data to acquire consistent information since it may inherently contain repetitive information from the same cells according to the availability of the cells of the serving network. Therefore, the data set has been ensured to contain measurements from at least three different cells, namely the camping cell and other neighboring cells, as the proposed algorithm relies on. The obtained parameters to be used in the simulation environment have been classified in Table 1.

TABLE I
THE CONTENT OF THE DATA SET

RSS1	RSS measured from serving cell
RSS2	RSS measured from neighboring cell
RSS3	RSS measured from neighboring cell
Lat1	Latitude information of the serving cell
Lon1	Longitude information of the serving cell
Lat2	Latitude information of the mobile user
Lon2	Longitude information of the mobile user

TABLE II
AN ARRANGED SAMPLE DATA SET

Record #	RSS1 (dBm)	RSS2 (dBm)	RSS3 (dBm)	Lat1 (°)	Lon1(°)	Lat2(°)	Lon2(°)
1	-94	-102	-102	37.9362619	41.9339838	37.93444416	41.94037064
2	-90	-103	-104	37.9362619	41.9339838	37.93438009	41.94037293
3	-92	-101	-102	37.9333024	41.9348472	37.93426857	41.94035169
4	-94	-99	-103	37.9333024	41.9348472	37.93428212	41.94035159
:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:
985	-102	-98	-105	37.9324745	41.9422683	37.93430771	41.94339912
986	-70	-83	-84	37.9351614	41.9419484	37.93448996	41.94061816
987	-72	-81	-83	37.9351614	41.9419484	37.93437705	41.94047073
:	:	:	:	:	:	:	:

In addition to the two basic data sets obtained by both indoor and outdoor measurements, a third data set has been formed by combining these two data sets to obtain a mixed type data to provide a more general and realistic model. Finally, the acquired data sets reflecting the three types of scenarios, namely indoor, outdoor and general, are then ready to be tested in the simulation environment. The arranged data set consists of a total of 2040 data, categorized as 1203 data for indoor and 837 data for outdoor, where a sample of the data set is given in Table 2.

B. Extreme Learning Machine

Extreme learning machine (ELM) is a method developed to be used in training feed-forward artificial neural networks (ANN) with only one hidden layer [8]. The scheme of the network is given in Figure 2.

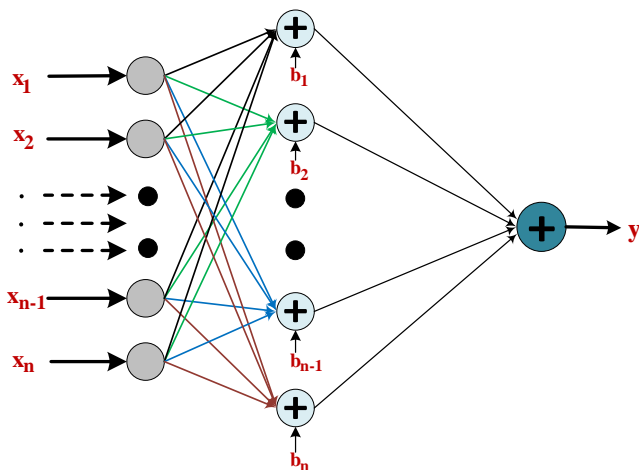


Fig.2. A feed-forward ANN model with one hidden layer

The output equation of the network is given as

$$Y(p) = \sum_{j=1}^m \beta_{j,k} g \left(\sum_{i=1}^n w_{i,j} x_i + b_j \right) \tag{1}$$

where $x_{1...n}$ are the input vectors, $y_{1...p}$ are the output vectors, $\beta_{1...m}$ are the output layer weights, $w_{1...n,1...m}$ are the connection weights between input layer and hidden layer, $b_{1...m}$ are the biases and $g(\cdot)$ is the activation function [9].

In order for ANN to be able to learn successfully, the transfer function, the biases and weights of the system to be modeled have to be selected properly.

In order to minimize the error that will occur during training in gradient-based approaches, the process of changing the given weights and biases continue until the most appropriate parameters are obtained. In the ELM method, the input weights and biases are given randomly, and the output weights are calculated accordingly [10]. Equation 1 can be rewritten in a more compact form as follows

$$y = H\beta \tag{2}$$

where H represents the hidden matrix of the network. H can be calculated as follows

$$H = \begin{bmatrix} g(w_{1,1}x_1 + b_1) & \dots & g(w_{1,m}x_m + b_m) \\ g(w_{n,1}x_n + b_1) & \dots & g(w_{n,m}x_m + b_m) \end{bmatrix} \tag{3}$$

Calculation of the inverse matrix of H matrix is performed with the generalized Moore-Penrose matrix. So the output weights (β) can be calculated as

$$\beta = H^\dagger y \tag{4}$$

where H^\dagger in Equation 4 indicates the Moore-Penrose generalized inverse matrix [11].

C. Generalized Regression Neural Network

Generalized regression neural network (GRNN) is a controlled type of feed-forward neural network (FFNN) and is one of the most popular neural networks. It was first introduced by Donald F. Specht in 1991. The training of GRNN networks is very fast, because unlike other networks where most of the data can be propagated back and forth many times until an acceptable error is found, in GRNN data only needs to be moved forward once [12]. The GRNN network works well on interpolation problems, a mathematical method developed to calculate missing data in a series. GRNN networks are used for estimating continuous variables as in standard regression techniques. By taking the function prediction directly from the training data, it approximates any function between the input and output vectors. In addition, as the size of the training set increases the prediction error approaches zero, but there are only slight restrictions on the function [13].

A GRNN consists of four layers: input layer, pattern layer, summation layer and output layer. The number of input units on the input layer depends on the total number of observation parameters. The first layer is connected to the pattern layer, and each neuron in this layer provides a training pattern and its output. The pattern layer is connected to the summation layer. In training of the network, radial-based and linear activation functions are used in the hidden and output layers [14]. Each pattern layer unit is connected to S - and D -summation neurons. The S -summation neuron computes the sum of the weighted responses of the pattern layer. On the other hand, the D -summation neuron is used to calculate the non-weighted output of the pattern neurons.

y_i is the target output value corresponding to the i th input pattern and represents the connection weight between the pattern layer and S -summation neuron. The connection weight for D -summation neuron is unity. The predicted value \hat{y}_i to an unknown input vector x , is obtained by the output layer which divides the output of each S -summation neuron by D -summation neuron

$$\hat{y}_i = \frac{\sum_{i=1}^n y_i \cdot \exp - D(x, x_i)}{\sum_{i=1}^n \exp - D(x, x_i)} \quad (5)$$

where n is the number of training patterns, and the Gaussian function D is defined as

$$D(x, x_i) = \sum_{k=1}^m \left(\frac{x_k - x_{ik}}{\sigma} \right)^2 \quad (6)$$

where m is the number of elements of the input vector, The x_k and x_{ik} are the k th element of x and x_i , respectively. The σ is generally called as spread and its optimal value is determined experimentally. It should be noted that in conventional GRNN applications all units in the pattern layer have the same single spread [15].

D. k Nearest Neighbor

The k nearest neighbor (k NN) is a sample based learning algorithm that performs the learning process according to the data in the training set. Classification of a new data is made according to the distance (similarity) between the data in the

training set [16]. In k NN regression, the algorithm is used for estimating continuous variables. The algorithm employs a weighted average of the nearest neighbors k , weighted by the inverse of their distance. In the k NN method, the data in the training set are recorded with numerical features. Each data represents a point in m -dimensional space and all of the data in the training set is included in an m -dimensional data area. In case a new data is encountered, the classification of the new data is performed by determining k pieces of data that are similar to the new data from the data in the training set [17].

One of the most important factors affecting the performance of the k NN is how to calculate the distance between data. The calculation process can be performed by Euclidean distance or other distance measurement parameters, such as Minkowsky, Hamming or Mahalanobis distance metrics or their variants, depending on the data. Since symmetrically non-propagated classes are often encountered while determining the class in the k NN, such classes are more dominant in determining the classes of new data. Therefore, methods that give weight values in different ways to the values that affect the distance measurement of the k NN are used [18].

The k NN is highly effective in training sets with a large number of data; it can achieve significant successful results. It can create a classification model even if there are data that are not similar to the data in the training set. The availability of such data increases the time required for training the data set [19]. Although the k NN is structurally simple, it has a high computational cost. Calculating the distance between the data to be classified and the data in the data set requires a very high calculation cost in training sets with a high number of data. In order to minimize such a high cost, methods that can be used with k NN can be preferred. For example, structurally powerful search trees or elementary component methods that can reduce data size can be used [20]. The k NN is not very successful in sets with high dimensions. In addition, it is very sensitive to factors such as neighborhood number and distance measurement and requires high memories [21].

Choosing of the optimal k parameter is a crucial task in k NN, and it is mostly dependent on the type of the employed data. Large values of k generally can improve the noise effect in classification, but it may cause to a distinction of the boundaries of the classes; however, an optimal selection of k neighbors can be obtained using various heuristic algorithms [22]. In the special case of selecting $k = 1$ in prediction of the class to belong to the closest training sample is known as the nearest neighbor algorithm. In the two-class k NN (i.e. binary case classification), if k is chosen to be an odd number then it helps the algorithm to avoid tied votes. Bootstrap technique is one of the popular empirical method in selecting the optimal k value in the setting stage [23].

E. Proposed Method

The block diagram of the proposed positioning system is illustrated in Figure 3. The arranged data set obtained via our developed application consists of a total of 2040 data, categorized as 1203 data for indoor and 837 data for outdoor. In the first preprocessing stage, the data is retrieved from the application, recorded to a database and categorized to the

relevant classes. In the next stage, all data sets are put through the machine learning algorithms, namely ELM, GRNN and, kNN to predict the position of the current location of the mobile user. Eventually, the predicted location information is compared with the actual location information obtained via GPS to calculate the distance between the predicted and actual location coordinates, thus enabling to measure the performances of each algorithm in terms of minimum/mean distance error and RMSE.

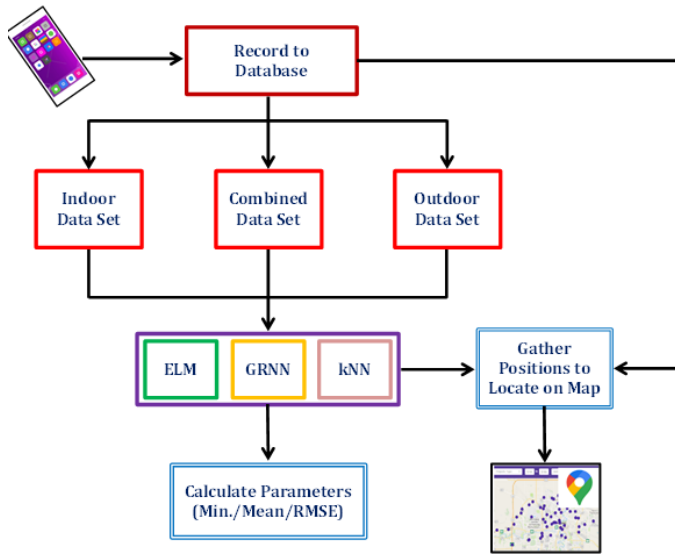


Fig.3. A general layout of the proposed algorithm

III. RESULTS AND DISCUSSION

For the selection of the activation function, which is one of the most important parameters affecting the performance of ELM, 14 different activation functions have been tested. Another important parameter affecting the performance is the number of neurons in the hidden layer, and tests are carried out for different numbers of neurons (5, 10, 25, 50, 75, 100, 125) and thus the network parameters (input and output weights and biases) that will work with best accuracy are determined separately for each data set. The optimum spread value which is the most important parameter for GRNN, was determined by conducting experiments for each data set using a total of 10 spread values (0.3, 0.5, 0.7, 0.9, 1, 1.5, 2, 2.5, 3, and 4). In the *k*NN method, *k* neighborhood parameter was tested for different numbers of

neighborhood values (1, 2, 3, 5, 10, 25, and 50) in the simulation environment and the minimum *k* value that would be optimum for each data set was determined.

Depending on the used methods and related parameters, the distance between the estimated position (i.e., latitude & longitude) and the actual position was calculated using the Haversine formula given below

$$d = 2r \arcsin \left(\sqrt{\sin^2 \left(\frac{lat2 - lat1}{2} \right) + \cos(lat1) \cos(lat2) \sin^2 \left(\frac{lon2 - lon1}{2} \right)} \right) \quad (7)$$

where *d* is the distance in kilometer (km), and the parameter *r* is the radius of the Earth taken as *r* ≈ 6371 km. The parameters *lat1*, *lon1*, *lat2*, and *lon2* represent the latitudes and longitudes of the actual and the estimated location in radians (rad), respectively. This distance expresses the amount of deviation from the actual position, i.e., the amount of error. The performances of all given methods were tested for each data set, namely: indoor, outdoor, and mixed data sets. The root mean square error (RMSE) metric, as well as the obtained distance errors, have been used as the performance parameters for evaluation.

A. Indoor Data Set

The RMSE values obtained in the tests performed with ELM method to determine the location with indoor data set are given in Table 3.

Among 14 activation functions, the best average RMSE values were provided by symmetric hard-limit, symmetric saturating linear, tangent sigmoid, pure linear and hyperbolic tangent activation functions. When all the activation functions and the number of neurons in the hidden layer are examined together, it can be seen that the best performance is achieved by the pure linear activation function with a value of 0.00145 RMSE and 10-125 neurons.

As a result of the tests obtained by cross validation using different activation functions and different number of neurons, the min. and mean distances between the estimated and actual locations are given in Table 4. The results show that each activation function has performed a min. error of approximately a few meters, and the best result was obtained as 0.9 m, using multiquadratic activation function with 75 neurons. It is seen that almost all of the activation functions used has performed an average error of 200 to 250 m, where the best mean error distance was achieved as 199 m using the pure linear activation function with 10-100 neurons. Also, the

TABLE III
AVERAGE RMSE VALUES USING ELM / INDOOR

Activation Functions	Number of Hidden Neurons						
	5	10	25	50	75	100	125
Sym. Hard-limit	0.00167	0.00166	0.00166	0.00163	0.00163	0.00160	0.00159
Sym. Sat. Linear	0.00167	0.00165	0.00165	0.00162	0.00161	0.00160	0.00159
Tangent Sigmoid	0.00167	0.00167	0.00165	0.00163	0.00160	0.00159	0.00159
Pure Linear	0.00216	0.00145	0.00145	0.00145	0.00145	0.00145	0.00145
Hyp. Tangent	0.00167	0.00166	0.00163	0.00165	0.00161	0.00160	0.00160
Multiquadratic	0.00167	0.00166	0.00162	0.00160	0.00158	0.00157	0.00157

TABLE IV
MIN. AND MEAN DISTANCE ERRORS (M) USING ELM/ INDOOR

Activation Functions	Number of Hidden Neurons													
	5		10		25		50		75		100		125	
	Min.	Mean	Min.	Mean	Min.	Mean	Min.	Mean	Min.	Mean	Min.	Mean	Min.	Mean
Sym. Hard-limit	22	5234	10	223	15	221	15	438	4	220	8	320	5	382
Sym. Sat. Linear	21	224	1	221	4	221	12	320	12	218	7	216	7	244
Tangent Sigmoid	11	228	5	224	10	229	5	220	4	238	8	227	7	239
Pure Linear	5	295	2	199	2	199	3	199	3	199	2	199	2	200
Hyp. Tangent	6	224	8	224	8	221	4	222	10	343	7	242	8	419
Multiquadratic	7	224	8	223	5	219	2	217	0.9	226	5	220	5	258

best performances were mostly obtained with the pure linear activation function.

The obtained min. and mean distance errors for each activation function depending on the number of neurons in the hidden layer is shown graphically in Figure 4. It can be seen from the figure that the best results are mostly obtained with the pure linear activation function.

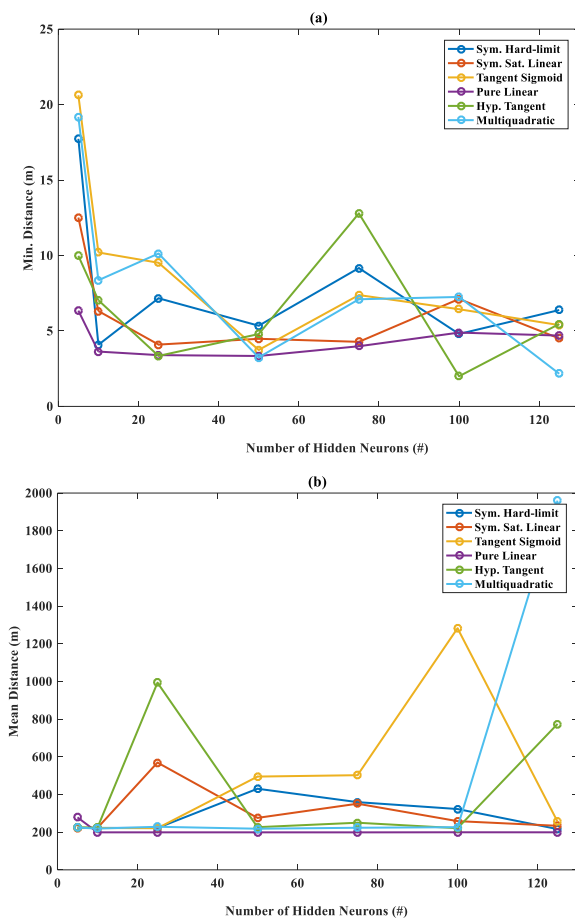


Fig.4 Distance errors vs number of hidden neurons, (a) Min. (b) Mean

The obtained test results using GRNN method for indoor data set depending on the spread values (0.3-4.0) is given in Table 5.

TABLE V
MIN. AND MEAN DISTANCE ERRORS (M), AND RMSE VALUES USING GRNN / INDOOR

Spread Values	Min.	Mean	RMSE
0.3	0.001	228	0.0019
0.5	0.001	224	0.0018
0.7	0.001	217	0.0017
0.9	0.001	213	0.0017
1.0	0.001	212	0.0016
1.5	0.014	210	0.0016
2.0	0.021	210	0.0016
2.5	0.189	210	0.0016
3.0	2.153	211	0.0016
4.0	10.953	212	0.0016

When the results given in Table 5 are evaluated in general, the best performance in terms of RMSE values was obtained as 0.0016. The best min. distance error was obtained as 0.001 m with spread values 0.3-1.0, and the best mean distance error was found to be 210 m with spread values 1.5-2.5.

Finally the indoor data set was tested with k NN method, where Table 6 shows the min. and mean distances, as well as the corresponding RMSE values depending on the number of neighbors k .

TABLE VI
MIN. AND MEAN DISTANCE ERRORS (M), AND RMSE VALUES USING k NN / INDOOR

k	Min.	Mean	RMSE
1	0.0000	75	0.0009
2	0.0000	85	0.0009
3	0.0000	103	0.0011
5	0.0024	130	0.0013
10	0.0024	170	0.0016
25	0.0024	203	0.0018
50	0.0000	254	0.0022

TABLE VII
MIN. AND MEAN DISTANCE ERRORS (M) USING ELM/ OUTDOOR

Activation Functions	Number of Hidden Neurons													
	5		10		25		50		75		100		125	
	Min.	Mean	Min.	Mean	Min.	Mean	Min.	Mean	Min.	Mean	Min.	Mean	Min.	Mean
Sym. Hard-limit	28	236	24	237	12	233	8	231	1	422	9	517	6	348
Sym. Sat. Linear	21	237	15	939	28	511	15	577	10	224	9	236	19	452
Tangent Sigmoid	9	7097	15	238	4	269	11	244	8	9032	21	471	10	578
Pure Linear	8	316	5	207	6	207	5	207	4	207	5	207	9	207
Hyp. Tangent	20	244	27	245	17	299	22	1921	12	417	12	1052	9	8340
Multiquadratic	13	233	28	237	23	240	8	246	15	805	10	333	9	289

When the results in Table 6 are evaluated, the best performance in terms of RMSE values was obtained as 0.0009. The best min. distance error was obtained as 0 m with $k = 1, 2$ and 3, and the best mean distance error as 75 m with $k = 1$.

B. Outdoor Data Set

Using the same analogy used for indoor data set, the outdoor data set has been tested in the same manner using ELM, GRNN and kNN methods, and the results are given in Table 7, Table 8 and Table 9, respectively.

The results given in Table 7 shows that almost all activation functions have performed well in terms of min. distance and the best result was obtained as 1 m using symmetric hard-limit activation function with 75 neurons. Similarly, the best mean distances best distance error was obtained as 207 m using pure linear activation function with 10-125 neurons. When considering all the results together, it is seen that most of the activation functions reach an average error of between 200 to 300 m, and the best results (about 222 m on average) are mostly obtained with the pure linear activation function.

TABLE VIII
MIN. AND MEAN DISTANCE ERRORS (M), AND RMSE VALUES USING GRNN / OUTDOOR

Spread Values	Min.	Mean	RMSE
0.3	0.905	237	0.0021
0.5	1.087	232	0.0020
0.7	1.087	220	0.0019
0.9	0.638	209	0.0018
1.0	1.087	207	0.0018
1.5	1.086	196	0.0016
2.0	1.087	193	0.0016
2.5	1.087	191	0.0016
3.0	1.087	193	0.0016
4.0	1.323	195	0.0016

The obtained test results using GRNN method for outdoor data set depending on the spread values (0.3-4.0) is given in Table 8, where the best min. distance error was obtained as 0.638 m with 0.9 spread value and the best mean distance error as 191 m with 2.5 spread value. Also, the best result in

terms of RMSE value was obtained with 1.5-4.0 spread values as 0.0016.

TABLE IX
MIN. AND MEAN DISTANCE ERRORS (M), AND RMSE VALUES USING kNN / OUTDOOR

k	Min.	Mean	RMSE
1	0.61	93	0.0011
2	1.08	98	0.0011
3	1.38	108	0.0012
5	1.45	135	0.0014
10	1.45	195	0.0018
25	3.80	261	0.0022
50	7.31	353	0.0028

Finally, the test results of outdoor data set using kNN method are shown in Table 9, where the best min. and mean distance error was found as 0.61 m and 93 m with $k = 1$, respectively. And the best result in terms of RMSE value was obtained with $k = 1$ and 2 as 0.0011.

C. Combined Data Set

In order to generalize the results obtained from indoor and outdoor data sets, an integrated data set was created by combining these data sets. Similarly, the combined type data set was tested using ELM, GRNN, and kNN methods, where the obtained results are shown in Table 10, Table 11, and Table 12, respectively.

The results given in Table 10 shows that the best min. and mean distance errors was obtained as 1 m using the hyperbolic tangent activation function with 25 neurons and as 204 m using pure linear activation function with 10 neurons, respectively.

When Table 11 is examined, it is seen that the best performance in terms of RMSE values was obtained with 2.0 spread value as 0.0016, and the best min. distance error as 0.001 m with 0.3-1.0 spread values. The best mean distance error is seen to be 216 m with 1.5 spread value.

Finally, the test results of mixed data set using kNN method are shown in Table 12, where the best RMSE value was obtained as 0.0013 with $k = 1$. The best min. and mean distance errors were found as 0 m with $k = 50$ and 120 m with $k = 1$, respectively.

TABLE X
MIN. AND MEAN DISTANCE ERRORS (M) USING ELM/ COMBINED

Activation Functions	Number of Hidden Neurons													
	5		10		25		50		75		100		125	
	Min.	Mean	Min.	Mean	Min.	Mean	Min.	Mean	Min.	Mean	Min.	Mean	Min.	Mean
Sym. Hard-limit	4	232	5	231	11	230	11	479	11	227	4	226	8	608
Sym. Sat. Linear	6	231	6	231	4	234	9	228	5	229	6	225	8	227
Tangent Sigmoid	4	231	4	235	6	236	4	703	6	243	5	248	5	449
Pure Linear	27	3009	9	204	9	205	9	205	10	205	9	205	9	205
Hyp. Tangent	9	231	8	231	1	230	7	230	4	290	4	9858	8	655
Multiquadratic	11	535	9	242	13	232	4	261	12	234	5	980	6	583

TABLE XI
MIN. AND MEAN DISTANCE ERRORS (M), AND RMSE VALUES USING GRNN / COMBINED

Spread Values	Min.	Mean	RMSE
0.3	0.001	241	0.0020
0.5	0.001	234	0.0019
0.7	0.001	224	0.0018
0.9	0.001	220	0.0017
1.0	0.001	219	0.0017
1.5	0.012	216	0.0017
2.0	0.026	217	0.0016
2.5	0.266	217	0.0017
3.0	2.187	218	0.0017
4.0	3.503	219	0.0017

TABLE XII
MIN. AND MEAN DISTANCE ERRORS (M), AND RMSE VALUES USING kNN / COMBINED

k	Min.	Mean	RMSE
1	0.0009	120	0.0013
2	0.0009	135	0.0014
3	0.0011	160	0.0016
5	0.0045	198	0.0018
10	0.0125	253	0.0021
25	0.0138	326	0.0026
50	0.0000	380	0.0030

TABLE XIII
A SUMMARY OF MIN. AND MEAN DISTANCE ERRORS (M) DEPENDING ON THE DATA SET

Location	ELM		GRNN		kNN	
	Min.	Mean	Min.	Mean	Min.	Mean
Indoor	0.89	199	0.001	210	0	75
Outdoor	1	207	0.638	191	0.61	93
Combined	1	204	0.001	216	0	120

All the results are given in Table 13 in order to summarize and compare the performances of the given methods depending on the used type of data sets. When the given results are compared, it is clearly seen that the best results yielding minimum distance errors have been achieved with indoor data set using *k*NN method, with a min. of 0 m and an average of 75 m. Although the performances of both ELM and GRNN are nearly the same in all data sets, the performance of *k*NN considerably drops for outdoor and combined data sets.

IV. CONCLUSION

The need for positioning of people and/or devices, either being for personal, scientific, judicial or commercial purposes, is in a big demand, especially in indoors and obstructed areas. This requirement can be met with the help of different systems such as global, cellular or wi-fi technologies. Positioning systems show a great improvement through auxiliary and complementary systems and applications developed in parallel with technology. It is expected that geolocation systems will continue to develop by taking the advantage of the properties of signals that can be obtained from different generation cellular communication networks, such as GSM, CDMA, WCDMA, and LTE via mobile devices.

In this study, indoor and outdoor locations in a certain region were chosen as the application area. The data used in the tests consist of real signal measurements obtained from cellular base stations via a convenient and easy-to-access mobile application we have developed in the Android Studio environment. Since accessing, saving and organizing the limited number of network data offered by GSM operators requires an important and time consuming process, the software of the developed mobile application constitutes one of the most critical and important parts of this study. The data we obtained in this study was trained in the simulation environment with ELM, GRNN, and *k*-NN, enabling the mobile user to estimate the real position using only instant GSM signal measurements.

The proposed study shows that better results were obtained with indoor data which is actually more critical in positioning and the *k*NN works with the highest accuracy in all data sets when compared to other algorithms. It is thought that the reason of high difference in the obtained minimum and mean error is due to the incompatible data in the data sets which causes large deviations in predictions. In some cases, namely according to the current occupancy rate, connection speed and other optimization priorities of the operator; the mobile user

may be camped to another base station at a longer distance; which then causes to a higher positioning error as the proposed algorithm uses the camped base station's location information as its input. In fact, obtaining mostly higher accuracies with indoor data supports this idea. Because, possible remote base stations that can be directed by the operator due to the aforementioned priorities, will be prevented due to weak signal levels indoor. Thus, with the proposed algorithm, the mobile user will be able to camp to nearby stations with higher signal levels leading to positioning with higher accuracy.

When our study is compared to similar localization methods using cellular network data in the literature, it is seen that it performs positioning with much higher accuracies with all three methods. It is considered that the proposed method can successfully position the mobile users with a good accuracy for indoor environments, where GPS signals are very weak or cannot be received at all; and that it can be also used as a good alternative for outdoor environments, in cases where GPS may be disabled for different reasons. Designing a hybrid positioning system by integrating wi-fi based technologies that provide successful results in indoor positioning with cellular and/or global positioning systems is considered as a future work.

REFERENCES

- [1] Sevindi, C. (2005). (Global Positioning System (GPS) and Its Usage in Geographical Researches. *Turkish Journal Geographical Sciences*, 3(1), 101-112.
- [2] Teunissen, P., & Montenbruck, O. (Eds.). (2017). *Springer handbook of global navigation satellite systems*. Springer.
- [3] Magro, M. J., & Debono, C. J. (2007, September). A genetic algorithm approach to user location estimation in umts networks. In *EUROCON 2007-The International Conference on "Computer as a Tool"* (pp. 1136-1139). IEEE.
- [4] Türkyılmaz, O. (2007). Environment aware location estimation in cellular networks. Master Thesis. Boğaziçi University, İstanbul.
- [5] Kurt, Ö. F. (2009) Location estimation by fingerprinting in cellular networks. Master Thesis. Boğaziçi University, İstanbul.
- [6] Fritsche, C., Klein, A., & Wurtz, D. (2009, March). Hybrid GPS/GSM localization of mobile terminals using the extended Kalman filter. In *2009 6th Workshop on Positioning, Navigation and Communication* (pp. 189-194). IEEE.
- [7] Xuereb, D., & Debono, C. J. (2010, March). Mobile terminal location estimation using Support Vector Machines. In *2010 4th International Symposium on Communications, Control and Signal Processing (ISCCSP)* (pp. 1-4). IEEE.
- [8] Huang, G. B., Zhu, Q. Y., & Siew, C. K. (2006). Extreme learning machine: theory and applications. *Neurocomputing*, 70(1-3), 489-501.
- [9] Ertuğrul, Ö. F., & Kaya, Y. (2014). A detailed analysis on extreme learning machine and novel approaches based on ELM. *American Journal of computer science and engineering*, 1(5), 43-50.
- [10] Öztekin, A., & Erçelebi, E. (2019). An efficient soft demapper for APSK signals using extreme learning machine. *Neural Computing and Applications*, 31(10), 5715-5727.
- [11] Huang, G. B., Wang, D. H., & Lan, Y. (2011). Extreme learning machines: a survey. *International journal of machine learning and cybernetics*, 2(2), 107-122.
- [12] Celikoglu, H. B., & Cigizoglu, H. K. (2007). Public transportation trip flow modeling with generalized regression neural networks. *Advances in Engineering Software*, 38(2), 71-79.
- [13] Cigizoglu, H. K., & Alp, M. (2006). Generalized regression neural network in modelling river sediment yield. *Advances in Engineering Software*, 37(2), 63-68.

- [14] Kim, B., Lee, D. W., Park, K. Y., Choi, S. R., & Choi, S. (2004). Prediction of plasma etching using a randomized generalized regression neural network. *Vacuum*, 76(1), 37-43.
- [15] Jang, J. S. R., Sun, C. T., & Mizutani, E. (1997). Neuro-fuzzy and soft computing-a computational approach to learning and machine intelligence [Book Review]. *IEEE Transactions on automatic control*, 42(10), 1482-1484.
- [16] Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill Higher Education. New York.
- [17] Han, J., Pei, J., & Kamber, M. (2011). *Data mining: concepts and techniques*. Elsevier.
- [18] Coomans, D., & Massart, D. L. (1982). Alternative k-nearest neighbour rules in supervised pattern recognition: Part 1. k-Nearest neighbour classification by using alternative voting rules. *Analytica Chimica Acta*, 136, 15-27.
- [19] Aha, D. W., Kibler, D., & Albert, M. K. (1991). Instance-based learning algorithms. *Machine learning*, 6(1), 37-66.
- [20] Shmueli, G., Patel, N. R., & Bruce, P. C. (2011). *Data mining for business intelligence: Concepts, techniques, and applications in Microsoft Office Excel with XLMiner*. John Wiley and Sons.
- [21] Duda, R. O., & Hart, P. E. (2006). *Pattern classification*. John Wiley & Sons.
- [22] Everitt, B. S., Landau, S., Leese, M., & Stahl, D. (2011). Miscellaneous clustering methods. *Cluster analysis*, 215-255.
- [23] Hall, P., Park, B. U., & Samworth, R. J. (2008). Choice of neighbor order in nearest-neighbor classification. *the Annals of Statistics*, 36(5), 2135-2152.

BIOGRAPHIES

ERCAN DEMİR Siirt, Turkey, in 1996. He received the B.S. degree in electrical and electronics engineering from Siirt University, Siirt, in 2018. He received the M.S. degree in electrical and electronics engineering from Batman University, Batman, in 2020.

He worked as an electrical and electronics teacher with the Private Siirt OSB Science Vocational and Technical Anatolian High School in 2019, and worked as an electrical and electronics engineer with the SAGTEK Biomedical Informatics in 2020. His research interests include machine learning, android programming, and signal processing.



ABDULKERİM ÖZTEKİN Berlin, Germany, in 1978. He received the B.S. degree in electrical and electronics engineering from Hacettepe University, Ankara, in 2001. He received the M.S. and the Ph.D. degrees in electrical and electronics engineering from Gaziantep University, Gaziantep, in 2018.

From 2001 to 2011, he worked as a Senior Engineer in several companies in the industry, and from 2012 to 2018; he was a Lecturer with the Electronic Communication Department, Batman University. Since 2018, he has been an Assistant Professor with the Electrical and Electronics Engineering Department, Batman University. His research interests include signal processing, video coding, communication systems, and machine learning.