Mobile Wireless Sensor Networks Localization Based on Support Vector Machines

Majid Sina[1],
Dr. Hamid Haj Seyed Javadi[2],
Dr. Mehdi Dehghan Takht Fooladi[3]

[1] Department of Computer, Science and Research Branch, Islamic Azad university, Tehran, Iran
sinamajid@yahoo.com
[2] Department of Basis Science, Shahed University, Tehran, Iran
h.s.javadi@shahed.ac.ir
[3] Department of Computer Engineering, Amirkabir University of Technology, Tehran, Iran
dehghan@ce.aut.ac.ir
Paper Reference Number:
Name of the Presenter: Majid Sina

Abstract

It is necessary for some of the applications of wireless sensor networks (WSNs) to estimate the geographical position of nodes. For the localization of wireless sensor nodes in wireless sensor networks, we took advantage of the learning machine approach based on binary support vector machine (SVM). During offline learning process, the received signal strength from reference nodes was chosen as learning machine input. By dividing the sensor field into several regions the geographical position of reference nodes was used as class information in SVM, and during online localization process the decision function of SVM was used for estimating the blindfolded nodes. In our suggested approach, the mobility of nodes in the sensor field was taken into consideration, and the mobility of nodes for reducing the reference node and increasing the localization accuracy was used. The expected efficiency of our approach was offered using simulation.

Key words: Localization, Mobile nodes, Received signal strength (RSS), Support vector machine (SVM), Wireless sensor network (WSN)

1. Introduction

Wireless sensor networks are a distributed collection of nodes with limited resources that have the functional ability with the least need to the user. Wireless sensor nodes perform in a distributed interactional manner. These nodes are usually embedded in a physical environment and report the sensed data to a central station. Knowing the geographical position of sensor nodes for many sensor networks tasks, such as network management, event detection, query processing based on geography, and routing is necessary. For localization, the nodes can be equipped with Global Positioning System (GPS); but this solution is expensive regarding expenses and it is not optimal regarding the energy consumption. Considering this fact that in some cases node positions change, an important issue is to provide an accurate and efficient solution for estimating the position of sensors whose real position information is unknown.

1.1. Related Works

Existing techniques for sensor networks localization can be categorized into various dimensions, such as centralized vs. decentralized, beacons vs. beacon-less, and ranging vs. ranging-free [23].

In the centralized localization techniques, information such as connectivity and measured distance among nodes, collected from the whole network in one place and the information is processed in that place in a centralized manner. In order to estimate sensor nodes position using mathematical techniques [5]. In [4], the convex optimization for position estimation has only been used based on connectivity constraints. The MPAMAP [14] technique has also improved the results of the suggested technique in [4] with the help of multi-dimensional scaling. The other technique such as semidefinite programming [4] is of this category.

Decentralized localization approaches do not need centralized calculations, and each node in terms of its relations with its neighboring nodes, determinates its position [23]. Techniques based on relaxation such as [18] and [16], coordinate system stitching techniques such as [2], [8], [9], [11], and techniques based on beacon such as [1], [6], [10], [12], [19], [20], [21], and [22] categorized in this class.

Ranging techniques use distance estimates or angle estimates in localization calculations, while ranging- free techniques are dependent on the content of received messages [23]. In the ranging process, the distance between two nodes is estimated using techniques such as Time Difference of Arrival (TDoA) [7], [15], Angle of Arrival (AoA) [12], [17], or Received Signal Strength Indication (RSSI) [22].

There are two ranging-free algorithms: techniques that relay on high density of reference nodes in such a way that every node can be placed in the sensor field of several reference nodes, like APIT [6] technique and hop count techniques like DV-HOP [13].

Duc A. Tran in [5] developed support vector classification solutions for sensor network localization. But this approach has not considered the node mobility problem and the mobile node localization. In this paper, the suggested idea in [5] was developed for mobile nodes localization with the help of an appropriate network model and SVM.

1.2. Our Contribution

The suggested approach in this paper has used the Support vector machine learning concept. The support vector machine is a classification method with two major parts: a kernel function and a support vector set.

Support vectors were achieved during the training phase and with the help of the learning data. For the localization problem, the sensor field was divided into several geographical regions and every sensor node was classified in there regions. Then, each node position was achieved through the intersection of regions to which that node is classified. Our contribution included defining the kernel function and classes for sensor nodes classification, a solution for the classifier imposition, nodes mobility detection, with the Radio Frequency (RF) Technique, and using the nodes mobility for increasing the learning data over time.

1.3. Paper Organization

The rest of this paper has been organized as follows. A brief background on SVM has been provided in section 2. In section 3, our approach has been explained. The evaluation results based on a simulation study are presented in Section 4. Finally, in section 5, the paper has been ended with the conclusion.

2. Support Vector Machine

Consider the data classification problem in a X data space to one of the two classes of $\neg G$ or G. Suppose each data point x has a feature vector $\vec{x}$ in $\vec{X} \in \Re^n$ feature space. We suppose k data point $x_1, x_2, \ldots, x_k$ as training points with label $y_1, y_2, \ldots, y_k$ that if $x_i \in G$, then $y_i = l$, and else $y_i = -l$. It is required that estimating a new data point x be in class G or not. Support vector machine is an effective solution for this problem. For the nodes localization in the sensor network, the following steps have been achieved.

- A kernel function must be defined as $K : X \times X \to \Re$ that is symmetric and $k \times k$ matrix $\left[K\left(x_i, x_j\right)\right]_{i,j=1}^{k}$ must be positive semidefinite (that is, it should have non-negative eigenvalves).
- Maximize

$$W(\alpha) = \sum_{i=1}^{k} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{k} y_i y_j \alpha_i \alpha_j K\left(x_i, x_j\right) \tag{1}$$

- Subject to

$$\sum_{i=1}^{k} y_i \alpha_i = 0, \tag{2}$$

$$0 \le \alpha_i \le C, i \in [1, k]. \tag{3}$$

Suppose that $\left\{\alpha_1^*, \alpha_2^*, \ldots, \alpha_k^*\right\}$ is a solution to this optimization problem. $b = b^*$ is chosen in such a way that $y_i h_K\left(x_i\right) = 1$ for all i with $0 < \alpha_i^* < C$.

Training points that are corresponding to such $\left(i, \alpha_i^*\right)$ are labeled as support vectors.

The decision rule for classifying a data point x is in this way that if and only if $sign\left(h_K(x)\right) = 1$, then $x \in G$, where

$$h_K(x) = \sum_{i=1 \to k,\ \mathrm{x_i\ is\ a\ support\ vector}} \alpha_i^*, y_i K\left(x, x_i\right) + b^*. \tag{4}$$

The function $h_K(.)$ in the equation 4 shows a hyper plane in $\vec{X}$ that separates training points in X with a maximum margin, where G is the points in the positive side and $\neg G$ is the points in the negative side.

If the classes are separated linearly, the hyperplane shows a line that separates the classes with a maximum margin, and if the classes are not separated linearly, the data are mapped to a feature space with a higher dimension so that they can be separated linearly. This map is like $\mathrm{x} \to \phi(\mathrm{x})$. Generally, feature space dimensions are infinite. The calculations in the feature space can be expensive because this space has higher dimension. To solve this problem, this kernel technique is used in which kernel functions are used. The kernel function is considered as polynomial kernel function because of its efficiency in experience that comes, in equation 5.

$$K(x,x')=(1+(x,x'))^{m} \tag{5}$$

3. Localization Based On Support Vector Machines

3.1. Network Model

A large-scale wireless sensor network of N node $\{S_1,S_2,\ldots,S_N\}$ is deployed in a 2D geographical field $[0, D]\times[0, D]\ (D > 0)$. This field is divided to $M_1 \times M_2$ region where M1 and M_2 definition is dependent on the localization accuracy. Generally, it is supposed that $M_1 = M_2$. Communication range of each S_i node is indicated with $r(S_i)$ and in this paper it is supposed all nodes have an equal $r\,(r>0)$. Two nodes form each other are reachable, if there is a communication route between them. There are a number of signal transmitter nodes $\{FN_1, FN_2,..., FN_n\}$ that, in this paper, supposed $n=5$. These nodes are deployed in 4 corners and the center of sensor field, and they are fixed in their positions. k nodes $\{MB_1, MB_2,..., MB_k\}$ are beacon nodes that are scattered in the field uniformly or randomly that know their position with the help of Global Positioning System (GPS). These nodes can be mobile and directly listen to all transmitters. There are p nodes $\{MN_1, MN_2,..., MN_p\}$ without GPS that are scattered in the field uniformly or randomly, and their positions must be estimated with an appropriate localization algorithm. In this paper, an approach based on SVM is used for localization. Figure 1 shows the locations of 50 beacon nodes, 500 blindfolded nodes and 5 radio signal transmitters.

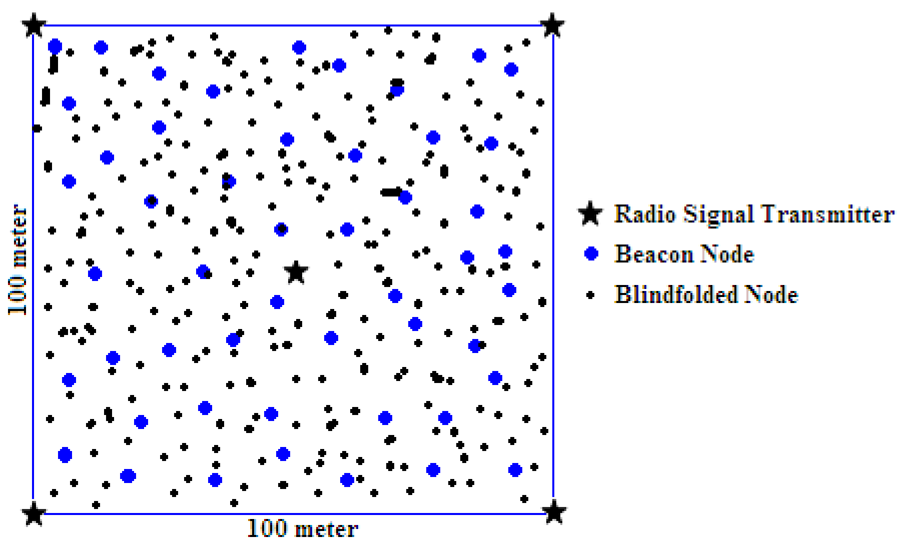


Fig 1: Network model

3.2. Suppositions and Notifications

Suppose that $\left(x\left(S_i\right), y\left(S_i\right)\right)$ determines the real coordinates of S_i node. If MB_i node is from our reference nodes set ($MB_i \in \{MB_1, MB_2, ..., MB_k\}$), then $RSS\left(MB_i, FN_j\right)$ show MB_i node's received signal strength from FN_j transmitter radio signal node. Every MB_i node is represented with the following feature vector:

$$MB_i = < RSS(MB_i, FN_1), RSS(MB_i, FN_2), ..., RSS(MB_i, FN_n) >$$

The beacon nodes set $\{MB_i\}$ (i=1:n) is considered as the training date for SVM. Every blindfolded node is represented with the following feature vector:

$$MN_i = < RSS(MN_i, FN_1), RSS(MN_i, FN_2), ..., RSS(MN_i, FN_n) >$$

After training SVM, the node position can be estimated using vectors.

Class definition: the sensor field from X and Y dimensions are divided to several geographical regions $\{C_1, C_2, ..., C_{M-1}\}$ in where M can be a power of 2.

- M-1 classes for the X-dimension $\{cx_1, cx_2, ..., cx_{M-1}\}$, each class cx_i containing nodes with the x-coordinate $x < iD/M$
- M-1 classes for the Y-dimension $\{cy_1, cy_2, ..., cy_{M-1}\}$, each class cy_i containing nodes with the y-coordinate $y < iD/M$

Therefore, $(2M - 2)$ binary classification must be performed. Obviously, class cx_i includes nodes that are placed at the right side of vertical line $x = i\ D/M$ and every class such as cy_i includes nodes that is placed above the horizontal line $y = i\ D/M$. Therefore, if SVM predicts that each S node is placed in a cx_i class not in a cx_{i+1} class, and is placed in a cy_i class not in a cy_{i+1} class, it is concluded that S is placed in square cell $[iD/M, (i+1)D/M] \times [jD/M, (j+1)D/M]$, and the center of this cell can be estimated for S node position.

Training data: For every classification problem (for one class $c \in \{cx_1, cx_2, ..., cx_{M-1}, cy_1, cy_2, ..., cy_{M-1}\}$), training data are beacon nodes set with corresponding labels $\{l_1, l_2, ..., l_k\}$, that if beacon node MB_i is belonged to class c, then $l_i = 1$, and else, $l_i = -1$.

After defining the kernel function and training data for each class, we can solve the above mentioned optimization problem of the support vector machine to gain coefficients $\{a_1^*, a_2^*, ..., a_k^*\}$ and b^*. Then, the decision function $h_K(.)$ in equation 6 is used to investigate if the given MN_i node is placed in class c or not.

$$h_K(S) = \sum_{i=1}^{k} \alpha_i^* l_i K(S, S_i) + b^* \qquad (6)$$

The learning procedure is implemented as follows: the sink node gets the received signal strength vectors and positions of all beacon nodes and transmits them to the base station. The base station execute the learning procedure of the support vector machine on all $(2M - 2)$

classes $cx_1, cx_2, ..., cx_{M-1}, cy_1, cy_2, ..., cy_{M-1}$, and then calculates b^* and $(i, l_i\alpha_i^*)$ information for each class. This information forms the support vector machine prediction model.

3.3. Position Estimation

We focus on the X-dimension classification and the Y-dimension has also a similar state. In this paper, we organize cx classes in a binary decision tree. As it is seen in figure 2, each true node is a cx class, and the output with value l is indicative of belonging and output 0 is indicative of no belonging to this class.

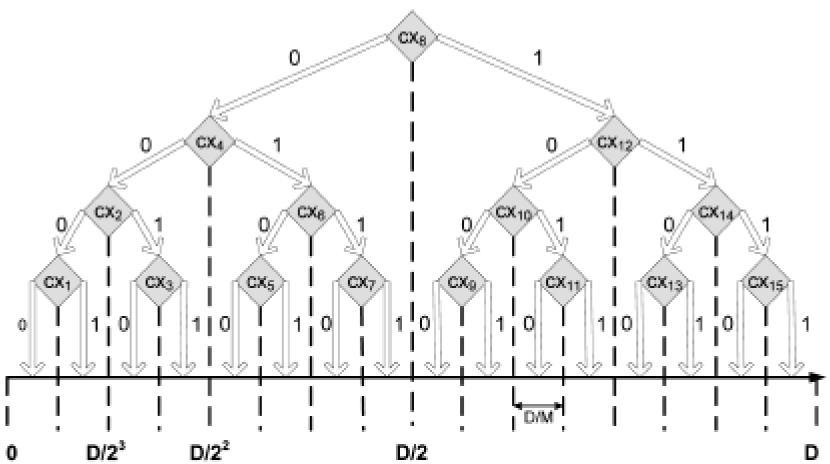


Fig 2: Decision tree. m = 4.

Classes are assigned to tree nodes in such a way that the ordered list $cx_1 \rightarrow cx_2 \rightarrow ... \rightarrow cx_{M-1}$ is achieved with tree inorder traversal.
Considering this BST, each sensor node can estimate its coordinates on X with the help of the following algorithm.

1. $Initially,\ i = M/2\ (start\ at\ root\ of\ the\ tree\ cx_{M/2})$
2. $IF\ (SVM\ predicts\ S\ in\ class\ cx_i)$
   - $IF\ (cx_i\ is\ a\ leaf\ node\ -\ i.e.,\ having\ no\ child\ decision\ node)$
     $RETURN\ x'(S) = (i - 1/2)D/M$
   - $ELSE\ Move\ to\ left\text{-}child\ cx_j\ and\ set\ i = j$
3. $ELSE$
   - $IF\ (cx_i\ is\ a\ leaf\ node)\ RETURN\ x'(S) = (i + 1/2)D/M$
   - $ELSE\ Move\ to\ right\text{-}child\ cx_t\ and\ set\ i = t$
4. $GOTO\ Step\ 2$
5. END

A similar BST is used for Y-dimension. The estimated position for node S is $(x'(S), y'(S))$. The localization of each node requires visiting $\log_2^M$ decision tree nodes using this algorithm. M or m parameter controls the localization accuracy. The localization algorithm is faced with errors because of support vector machine errors. For every class c, when support vector machine predict that one sensor node is in class c but in fact it is not in this class or vice versa, a miss classification has been formed. In addition to these errors, even with true region selection in high algorithm, the maximum amount of error will be $\frac{D}{M\sqrt{2}}$.

3.4. Nodes Mobility Detection

Each mobile node preserves the received signal strength value that comes for transmitter nodes. By calculating Received signal strength average value, the mobile node determines if it has been moved or not. Because received signal strength is not stable, and changes at any time, the average of 3 recent received signal strength from each transmitter node ($\mathrm{RSS}_{\mathrm{avg-i}}$) is calculated. Regarding the system requirements, two lower and upper thresholds are defined. Threshold N shows the least change of received signal strength value. When the position of a node is changed, received signal strength changes over time, too.
Also if the changes of received signal strength are very high, the result will be wrong. To solve this problem, a simple but efficient strategy is suggested, and that is to define the maximum threshold M. the recent average value of received signal strength that has been stored in the node, is compared to the last value of received signal strength ($\mathrm{RSS}_{\mathrm{lastavg-i}}$). If the absolute of their difference is greater than threshold N and less than threshold M, it shows that the node has been moved. Otherwise, it is thought that the node do not moved. Then, the mobile node sends a message to the sink node for coordination and the base station realizes that the nodes have been moved. Figure 3 shows this mechanism.
Suppose that the environment is like a square that the length of every side is L meter, and then the threshold approximation is shown with equation 7.

$$thresholdN = 10n\log(L/d_0) \tag{7}$$

Where

n : path loss exponent , d_0: the close-in reference distance

We also can use other manner to evaluate thresholdN such as statistical methods [23].

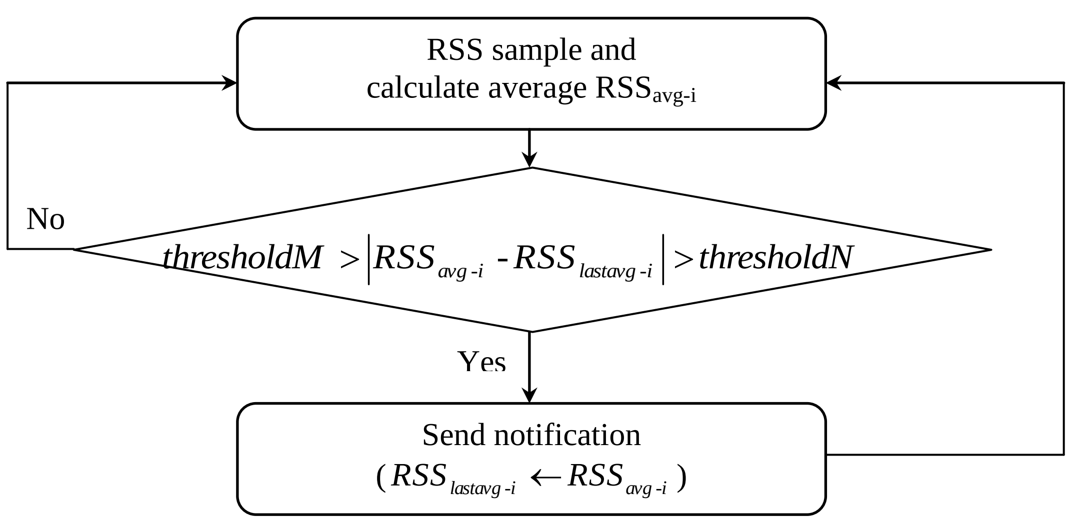


Fig 3: Mobile nodes determine flowchart.

3.5. Taking Advantage of Nodes Mobility

In previously suggested approaches such as LSVM [5] the nodes in the network have been considered fixed, and for increasing the localization accuracy the number at beacon nodes must considerably be high. In this paper, every beacon node forms its received signal strength vector through the transmitted signal strength from radio transmitters. If the previous position of mobile node is supposed as valid node, and the new position of mobile node as a new node, adding this supposedly new node to the network has no effect on the number of received signal strength vectors.

As a result, a few beacon nodes can be used at the time of network deployment, and over time the mobility of nodes are used to increase the number of support vector machine training data. This process is performed in this way that the new position of a moved beacon node is supposed as a new beacon node in the network, and in a suitable interval this new training data are added to SVM training phase.

4. Results and Analysis

Simulation with fixed nodes: In this model 1000 sensor nodes have been scattered in the sensor field randomly. In every simulation execution, some percent of all nodes (5%, 10%, 15%, 20%, and 25%) are considered as beacon nodes, and the rest of them as blindfolded nodes. In every execution, transmitters send radio signals with 30db signal strength to the entire sensor field that receives all nodes. Then, the nodes prepare a received signal strength vector that includes the received signals from the transmitters. After that, the beacon nodes that know their position send their position information and their received signal strength vector to the sink node. The blindfolded nodes also send their received signal strength vector to the sink node.

The sink node sends all information to the base station on which the localization algorithm is performed. Finally, the estimated positions in the bas station are reported to all nodes by the sink node. We used the algorithms in the libsvm [3] software for SVM classification. The localization error has been shown in the simulation performed in figure 4.

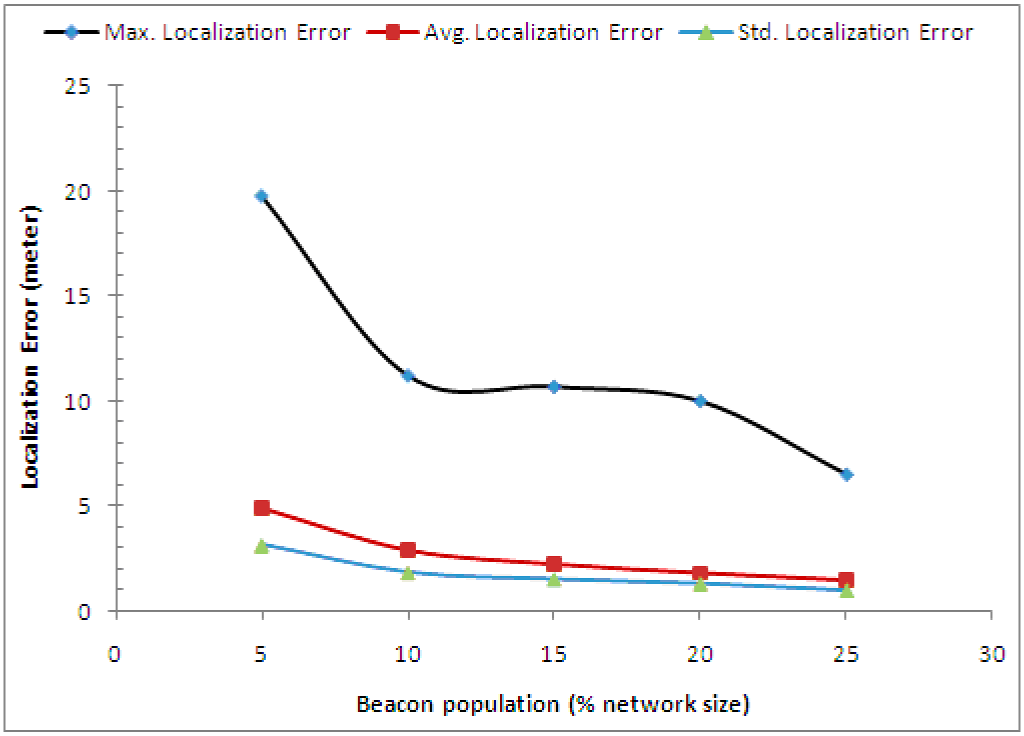
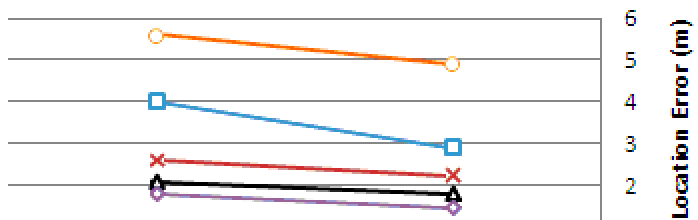


Fig 4: Statistics on localization error under different beacon populations.

In figure 5, there is a comparison between the suggested approach in the paper and LSVM approach. Figure 5 shows the average localization error achieved in the performed simulations for three mention approaches when 5%, 10%, 15%, 20%, and 25% of all nodes are beacon nodes.



	LSVM	My Approach
5%	5.6	4.89
10%	4	2.89
15%	2.6	2.23
20%	2.1	1.81
25%	1.8	1.46

Fig 5: Statistics on the average localization error under different beacon populations.

Simulation with mobile nodes: In this phase of simulation, all nodes are mobile. The number of beacon nodes in the primitive deployment of the network has been limited to 5% of all nodes (50 nodes). The beacon mobility is used as an advantage for extending support vector machine training data. After the first localization, with knowing the movement of every beacon node its new position along with its received signal strength vector is sent to the sink node and the sink node sent this information to the base station.

As it was mentioned, this new position is used as a supposed new node. Over time and with the movement of beacon nodes, the input date to support vector machine training phase are increased. Thus, using the suggested network model in this paper and the beacon nodes mobility, the localization accuracy can be increased with the help of a few beacon nodes.

The simulation results are shown in figure 6. This figure shows that the suggested approach in this paper can increase the localization accuracy with a few beacon nodes over time.

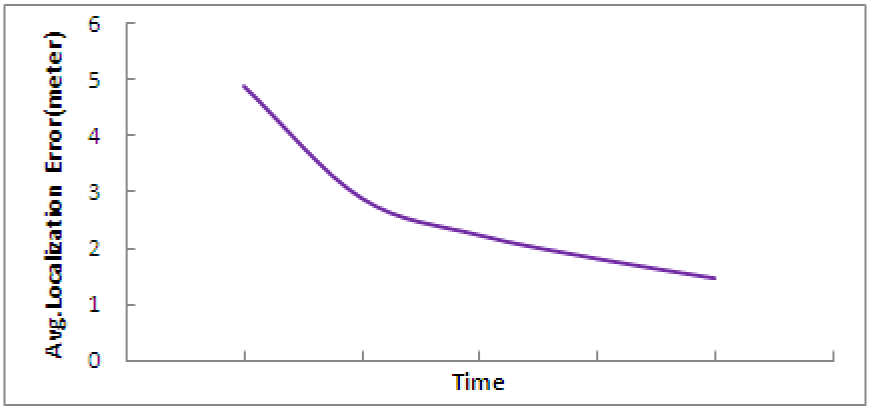


Fig 6: Error reduction over time in our approach

5. Conclusions

The wireless sensor network localization is one of the major challenges in network designs that has been taken into consideration in this paper. The learning machine technique was used as before for localization. With some changes in the network model and with the help of nodes mobility, we could achieve less average localization error compared with the previous approaches such as LSVM, while less beacon nodes were used. In this paper, the received signal strength was used for forming vectors that required received signal strength tools and handling noise problem. Instead of the received signal strength, hop count can be used for forming support vector machine vectors that will be investigated in our future study.

References

[1] Bulusu, N. , & Bychkovskiy, V. , & Estrin, D. , & Heidemann, J. (2002). Scalable Ad Hoc Deployable RF-Based Localization, *Proc. Grace Hopper Celebration of Women in Computing Conf*.

[2] Capkun, S. , & Hamdi, M. , & Hubauz, J. P. (2001). GPS-Free Positioning in Mobile Ad Hoc Networks, *Proc. 34th Ann. Hawaii Int'l Conf. System Sciences (HICSS)*.

[3] Chang, C.-C. , & Lin, C.-J. (2007). *LIBSVM—A Library for Support Vector Machines*. Nat'l Taiwan Univ., http://www.csie.ntu.edu.tw/cjlin/libsvm.

[4] Doherty, L. , & Ghaoui, L.E. , & Pister, K.S.J. (2001). Convex Position Estimation in wireless Sensor Networks, *Proc. IEEE INFOCOM, pp. 1655–1663. IEEE,* USA.

[5] Duc, A. , & Nguyen, T. (2007). Localization in Wireless Sensor Networks based on Support Vector Machines. *Submitted to IEEE Transactions on Parallel and Distributed Systems*.

[6] He, T. , & Huang, C. , & Blum, B. , & Stankovic, J. , & Abdelzaher, T. (2003). Range-Free Localization Schemes in Large Scale Sensor Networks, *Proc. ACM MobiCom*.

[7] Kwon, Y. , & Mechitov, K. , & Sundresh, S. , & Kim, W. , & Agha, G. (2004). Resilient Localization for Sensor Networks in Outdoor Environments, technical report, Univ. of Illinois, Urbana-Champaign.

[8] Meertens, L. , & Fitzpatrick, S. (2004). *The Distributed Construction of a Global Coordinate System in a Network of Static Computational Nodes from Inter-Node Distances* technical report, Kestrel Inst.,.

[9] Moore, D. , & Leonard, J. , & Rus, D. , & Teller, S. (2004). Robust Distributed Network Localization with Noisy Range Measurements, *Proc. Second ACM Int'l Conf. Embedded Networked Sensor Systems (SenSys '04)*.

[10] Nagpal, R. , & Shrobe, H. , & Bachrach, J. (2003). Organizing a Global Coordinate System from Local Information on an Ad Hoc Sensor Network, *Proc. Second Int'l Workshop Information Processing in Sensor Networks (IPSN)*.

[11] Niculescu, D. , & Nath, B. (2001). Ad Hoc Positioning System (APS), *Proc. IEEE Global Telecommunications Conf. (Globecom)*.

[12] Niculescu, D. , & Nath, B. (2003). Ad Hoc Positioning System (APS) Using AOA, *Proc. IEEE INFOCOM*.

[13] Niculescu, D. , & Nath, B. (2003). Dv based positioning in ad hoc networks. *J. Journal of Telecommunication Systems* 22, 267–280.

[14] Paramvir, B. , & Venkata, N. (2000). RADAR: An In-Building RF-Based User Location and Tracking System, *IEEE InfoCom 2000, vol. 2*, pp. 775–784.

[15] Priyantha, N.B. (2005). *The Cricket Indoor Location System*. PhD dissertation, Massachusetts Inst. Technology.

[16] Priyantha, N.B. , & Balakrishnan, H. , & Demaine, E. , & and Teller, S. (2003). Anchor-Free Distributed Localization in Sensor Networks, *Proc. First ACM Int'l Conf. Embedded Networked Sensor Systems (SenSys)*.

[17] Priyantha, N. , & Miu, A. , & Balakrishnan, H. , & Teller, S. (2001). The Cricket Compass for Context-Aware Mobile Applications, *Proc. ACM MobiCom*.

[18] Savarese, C. , & Rabaey, J. , & Beutel, J. (2001). Locationing in Distributed Ad-Hoc Wireless Sensor Networks, *Proc. IEEE Int'l Conf. Acoustics, Speech, and Signal Processing (ICASSP)*.

[19] Savvides, A. , & Han, C.-C. , & Strivastava, M.B. (2001). Dynamic Fine-Grained Localization in Ad Hoc Networks of Sensors, *Proc. ACM MobiCom '01*, pp. 166-179.

[20] Savvides, A. , & Park, H. , & Srivastava, M.(2002). The Bits and Flops of the N-Hop Multilateration Primitive for Node Localization Problems, *Proc. Workshop Wireless Networks and Applications*.

[21] Simic, S. , & Sastry, S.S. (2002). Distributed Localization in Wireless Ad Hoc Networks, technical report, Univ. of California, Berkeley.

[22] Whitehouse, C. (2002). The Design of Calamari: An Ad Hoc Localization System for Sensor Networks, master's thesis, Univ. of California, Berkeley.

[23] Yang, B., & Yang, & J., Xu, & J., & Yang, D. (2007). Area Localization Algorithm for Mobile Nodes in Wireless Sensor Networks Based on Support Vector Machines, H. Zhang et al. (Eds.): MSN 2007, LNCS 4864, pp. 561–571