J. Basic. Appl. Sci. Res., 2(8)7589-7596, 2012
© 2012, TextRoad Publication

ISSN 2090-4304
Journal of Basic and Applied Scientific Research
www.textroad.com

Localization Algorithm for Large Scale Mobile Wireless Sensor Networks

Samira Afzal

Department of Computer Engineering, Sharif University of Technology, Tehran, Iran

ABSTRACT

Localization is one of the most crucial problems in Wireless Sensor Networks. In this paper, we present a new scheme to solve this issue in the mobile and large scale networks based on learning based algorithms. The proposed algorithm is cost efficient and applicable to deploy in 3 dimension area. We propose two crucial methods to obtain the best results: position check method and distance check method. The simulation results show the performance of the proposed algorithm.

KEY WORDS: wireless sensor networ, support vector machine, localization problem, learning algorithms.

1. INTRODUCTION

Localization is an important challenge in wireless sensor network (WSN) technology. Localization problem refers to the process of computing positions of nodes [1]. Location estimation is essential in the most wireless sensor network applications, including target tracking, vehicle tracking, event detection, people monitoring, routing, forest fire detecting [2] and, etc. In all of these applications collected data is not usable without knowing about the location of an event which is the location of the sensor. The importance of this fact led researchers to seek a solution for localization problem. Broadly, localization algorithms can categorize in two groups, Non-learning and learning-based algorithms.

In the non-learning based approaches the distance of an unknown sensor node from neighbor beacons (sensors aware of their position) are measured in the first step. Then the probable coordinate of sensor position will be estimated. In this category, most of the available algorithms for mobile nodes are based on Monte-Carlo methods.

Learning based approaches use function in two phases: offline training phase and online localization phase. A model data is build based on some data gathered from the network and each sensor can use it to localize itself without knowledge about other sensors.

Therefore, learning approaches can be used in a distributed manner, and each sensor can be localized independently from other sensor nodes. So, it's not required that all the sensors run the localization algorithm and only the target localizes. Learning based approaches has been considered because of simple implementation, fast result and Low computation required for each node.

For more information about localization taxonomy and evaluation of localization algorithm refers to [3] and [4].

This paper proposes a technique to solve localization problem in mobile WSNs based on SVM as a learning machine method. SVM is mathematically powerful to estimate the location of nodes. In this algorithm SVM localizes a node based on connectivity information. Therefore, it doesn't require any special hardware and makes the algorithm suitable for large scale networks.

Many of learning based systems become inaccurate in dynamic networks when new data comes sequentially. In this method, model data builds online depends on the new data comes. In time intervals each beacon node will estimate its location according to its information and predicted model by SVM. If its location was out of threshold or the predicted class region was wrong, it will calculate the new connectivity information and sends it to the sink or head beacon in order to make a new predicted SVM model. The simulation results show that it has a good result in a dynamic network with high density and just with using a small set of beacon nodes like 5%. It also works properly in 3D area.

The rest of the paper is organized as follows. Section 2 contains the brief background on SVM. Section 3 overviewed the localization algorithms. In section 4, our technique is proposed. The result of experiments has been described in section 5. Section 6 discusses benefits of the proposed algorithm. Finally, section 7 concludes the paper with pointers to our future research.

2. Support vector machine classification

Support Vector Machine (SVM) [5] is a method used for classification and regression. SVM as classification establishes a hyperplane or set of hyperplanes in a high-dimensional space. It has a good functional margin and lower generalization error.

***Corresponding Author**: Samira Afzal, Department of Computer Engineering, Sharif University of Technology, Tehran, Iran. E-mail:Afzal.samira@gmail.com

It can works as linear and nonlinear classifier. Suppose that we have input feature vectors $X_1, X_2, \ldots, X_k$ with label $y_i \in \{1, -1\}$, where $X_i \in R^d$ is training data. The aim is predict y_i for a new X_i.
SVM uses the following function for linear data:

$$L_D = \sum \alpha_i - 1/2 \sum_{i,j} \alpha_i \alpha_j y_i y_j x_i x_j, \quad (1)$$

with subject to

$$\sum_i \alpha_i y_i = 0, \quad (2)$$

$$0 \leq \alpha_i \leq c.$$

(3)

Data with $\alpha_i > 0$ are Support vectors and $\alpha_i = 0$ are non-support vectors.
In nonlinear data function, SVM will be used with using kernel trick. It will map the data to some other dimensional space H (Euclidean space), with a mapping which called $\emptyset$.

$$\emptyset: R^d \rightarrow H. \quad (4)$$

And it is obvious that the algorithm depends on the training data through dot products in H.

$$X_i, X_j \rightarrow \emptyset(X_i).\emptyset(X_j) = K(X_i, X_j). \quad (5)$$

Where k must be symmetric, and the $k \times k$ matrix $[k(X_i, X_j) \quad]_{i,j=1}^{k}$ must be positive semidefinite.
Suppose that the solution will be $\{\alpha_1^*, \alpha_2^*, \ldots, \alpha_k^*\}$ and $b = b^*$. Then in the test phase there is:

$$H_k(x) = \sum \alpha_i^* y_i K(x, x_i) + b^* \quad (6)$$

3. Localization algorithms

Localization importance in WSN applications led researchers to seek solutions for this problem. In this way, several methods introduce in the literature. One easy way is a manual configuration but this is impractical in large scale or when sensors are mobile or deployed in inaccessible areas such as volcanoes. Another way is to add global positioning system (GPS) to each sensor, which requires direct line of sight. It causes low accuracy due to poor signal reception, and it is unusable for indoor applications. Alternatively, it is costly in terms of additional hardware and energy requirements. Therefore, several localization algorithms introduced to solve localization problem. Mobile localization algorithms can broadly classify in two groups based on the learning criteria: Non-learning and learning based algorithms.

In the past, researchers have more concentration on non-learning based algorithms based on RSS (received signal strength) values. These methods assume that there are several access points with known location [6]. The location of mobile nodes will be determined relative to some fixed access points based on RSS. These algorithms are not affordable for large sensor networks because of using RSS. Typically, the most important work to determine the location of a mobile node in non-learning algorithms developed based on Monte-Carlo methods. Recently, researchers used range free measurement methods to make more flexible algorithms [7].

Alternatively, machine learning approaches use RSS data which gathers by a mobile node and labeled with location [8]. This data known as training data, and they are used to build predicted model. Later, this predicted model will be used by nodes to estimate their location. In general, data gathering is done in the offline manner and localizing is done in the online manner. most of the proposed method in this category are for indoor mobile tracking [8], [9], and[10].

The authors of [11] used an ensemble based support vector regression (SVR) to determine the location of nodes. At the first, training data is divided into subsets. Sub predictor builds for each subset. Then they all are combined with proper weights.
Pan et al. used a mobile device walking through the network and gathering labeled data. Then a semi-supervised approach applied on labeled and unlabeled data to solve the localization problem in the regression manner. The mapping function can work online based on RSS [8].

The approach given in [12] combines two methods to estimate the location of nodes for indoors. In the first step, they assumed there is unlabeled RSS data available in the network. They used Latent Semantic Indexing (LSI) or Singular Value Decomposition (SVD) to estimate the relative location of access points and mobile devices. In the second step, they assumed there is a small amount of labeled data. They proposed a learning approach based on a manifold to obtain the absolute locations. Both labeled and unlabeled data are from access points and mobile devices. The above mentioned algorithms are based on RSS information and assume that direct signal from beacons is possible for nodes. So these algorithms are suitable for small networks.

The work [13] gave a localization algorithm based on learning approach and range free estimation. Using connectivity information caused the algorithm could be applied for large scale. In this algorithm, beacon nodes used as training data and information of network gathers as range free measurement. The model broadcast to the

network and each node can estimate its location individually. This approach has very good accuracy for fixed network.

4. THE PROPOSED METHOD

In this section, the proposed localization method will be presented in more details.

We consider N nodes $\{S_1, S_2, ..., S_n\}$ in the 2D or 3D area with d_l, d_w, d_h length for each dimension and $d_l = d_w = d_h = D$. The area covered by the network is partitioned into $M \times M$ square cells for 2D area and $M \times M \times M$ cube for 3D area. Each cell or cube constitutes the label for out training data (Fig.1). Intuitively, x-coordinate has M classes cx_i and each node believes that exists in class $[cx_i, cy_i]$ in 2D area or in class $[cx_i, cy_i, cz_i]$ in 3D area. In other words respectively exists in $[\frac{(i-1)D}{M}, \frac{iD}{M}] \times [\frac{(j-1)D}{M}, \frac{jD}{M}]$ unit or $[\frac{(i-1)D}{M}, \frac{iD}{M}] \times [\frac{(j-1)D}{M}, \frac{jD}{M}] \times [\frac{(k-1)D}{M}, \frac{kD}{M}]$ unit. We assign the center of estimated unit as location for each S_i and denotes by (x_i^c, y_i^c) in 2D and (x_i^c, y_i^c, z_i^c) in 3D area.

All nodes in the network have the same communication range $r > 0$. Each node can communicate with other nodes if exists in its range and no obstacle exists between them. Nodes $\{S_1, S_2, ..., S_k\}$ are the beacon nodes which know their positions. The aim of this thesis is to propose a localization algorithm to estimate the location of all other nodes $\{S_{k+1}, ..., S_n\}$ in distributed manner.

All nodes and beacons are mobile with the same mobility model. For each beacon node S_i, a vector $\langle h(S_i, S_1), h(S_i, S_2), ..., h(S_i, S_k) \rangle$ will be made. $h(S_i, S_j)$ denotes the shortest hop count between beacon sensors S_i, S_j .

The first step of our approach is training phase. Among this phase, the shortest paths between beacons will be calculated and send to the head beacon. Central unit uses received information as the input of SVM algorithm and make a model for all sensor nodes. Model is broadcasted to the network and each node uses the model to predict its location individually, without knowledge of other sensor positions. Since beacons and other nodes are mobile, the algorithm needs additional methods to calculate the results efficiently.

Movement in the environment will be discovered by two crucial methods in beacons, position check method and distance check method.

In check position method, each beacon checks its position periodically and if its predicted position was in wrong class, then calculates a new hop count distance vector and sends it to the head beacon. Another helpful method which used in our approach is threshold checking. According to this step, each beacon finds and sends a new hop count distance vector if it shifts its location more than the threshold T. These two methods are applied periodically to discover movement in the environment. Therefore, new information is gathered and sends to the head beacon if it is necessary. So, these methods avoid extra communication. In additional, the information is always updated for the head beacon. It runs SVM on the updated information and makes new predicted model. Recent made model is broadcasted to the network. Hence, always the best model is available for each sensor. The rest of this section explains proposed algorithm in details:

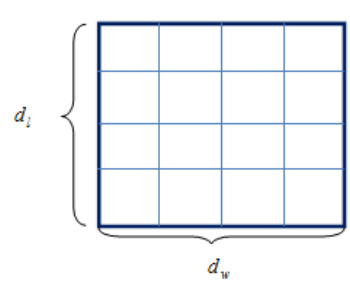

Fig.1. An area with M=4 and 16 cell.

A. Training phase:

In this step, all beacon nodes broadcast a HELLO message through the network to calculate the shortest path to other beacons. In this time, other nodes use beacon's HELLO message to calculate their shortest path to the beacons. Nodes use these distance vectors in the localization phase. One of the beacons which should be

more resourceful will be chosen as a head beacon and all other beacons send an INFO message contains their location and hop count distances to the head beacon $\{(x_i, y_i), \langle h(S_i, S_1), h(S_i, S_2), ..., h(S_i, S_k)\rangle\}$ in 2D area and $\{(x_i, y_i, z_i), \langle h(S_i, S_1), h(S_i, S_2), ..., h(S_i, S_k)\rangle\}$ in 3D area. Head beacon runs SVM on the received information and makes groups of $(x_{ix}, \alpha^*_{ix}, b^*_x), (x_{iy}, \alpha^*_{iy}, b^*_y)$ and also $(x_{iz}, \alpha^*_{iz}, b^*_{iz})$ if it is 3D area. The calculated information called predicted model, which is sent to all other nodes.

B. Localization phase:

Predicted model will be broadcasted through the network via ADVERTISMENT message. Each node S can compute its location by itself with using received model and stored distance vector" $\langle h(S_i, S_1), h(S_i, S_2), ..., h(S_i, S_k)\rangle$. These hop count distances achieved through the HELLO message broadcasting of beacon nodes. In fact, the result of localization for each node is a class which exists in. And the centroid of the class is used as estimated position of the node. (x'_i, y'_i) in 2D and (x'_i, y'_i, z'_i) in 3D.

C. Check phase:

Each beacon node applies periodically two crucial methods: position check method and distance check method to discover mobility in the network.

If a beacon finds movement in the network then sends a message to find new distance vector. The physical position and distance vector will be sent to the head beacon.

Position check method: Each beacon estimates its region class periodically. If it finds a difference in its predicted class and real class, which stands on, it will be sent a REPOSITION message to other beacons in order to find the new hop count distance from other beacons.

The difference happens because of some reasons. The first reason is when beacon moves from one class to the other class. The second reason arises from the other nodes, movement, whether beacons or nodes. In this case, a beacon has no change in its position but because of other nodes' mobility, hop counts have been changed. In other words, beacon's location is the same as before but the distance vector is changed. There are both beacon movement and changed distance vector. So, new information should be gathered and inform to the head beacon.

Check threshold (T) method*:* To make our work more accurate we use threshold T for the beacons. If the proposed method discovers that a beacon changes its location more than T, then it will send REPOSITION message to other beacons in order to find the new hop count distances from other beacons, and it sends INFO message to head beacon to find new SVM model.

D. Revise phase:

The new gathered information by beacons informs to head beacon sequentially. Head beacon has the last information of the whole network and updates it with new received information from beacons. The new predicted model makes by running SVM on the updated information. These processes occurred in the online manner. The pseudo code is available below.

Training phase:

1. Each beacon broadcasts a HELLO message to the network.
2. Distance vector $\langle h(S_i,S_1), h(S_i,S_2), \ldots, h(S_i,S_k)\rangle$ is calculated for each beacon.
3. Distance vector $\langle h(S,S_1), h(S,S_2), \ldots, h(S,S_k)\rangle$ is calculated for each non beacon node.
4. Each beacon sends its physical position and its distance vector to the head beacon in the INFO message.

Localization phase:

1. Predicted model is broadcasted through the network.
2. Each node uses received model and its distance vector to estimate its class region.
3. Centroid of each estimated class is assumed as node's position.

Check position method:

1. Beacon class label is predicted by using SVM model in each beacon
2. The real position of beacon is given via GPS and the real class label which sensor exists in is calculated.
3. **If** (predicted class label is equal to the real class label)

   No change in the beacon position found

Else

Change in the beacon position is found.

The beacon node sends a broadcast REPOSITION message to find the new distance vector.

1. An INFO message contains its new location and new hop count distances to all other beacons, sends to head beacon in order to find the new model via SVM.

Check threshold (T) method:

1. Calculating distance of movement by beacon itself.
2. **If** (distance is more than T)

   The beacon broadcasts REPOSITION message to find the new distance vector.

4. An INFO message contains its new location and new distance vector to all other beacons, sends to head beacon in order to find new the model via SVM.

Revise phase:

1. New information receives to the head beacon from beacons.
2. New predicted model makes via SVM algorithm.
3. Head beacon broadcasts new model data to the whole network.

In the next section we study the implementation of the fallowing study: the performance of LSVM under the effect of different number of classes, beacon population, speed of nodes and some other basic keys.

5. Simulation

In this paper, we deploy the algorithm on a network with 1000 sensors in an 100×100 area. And 2 dimension. We assume a uniform distribution for the location of sensors and beacons. Beacons know their position priori, for example, by GPS device. Both nodes and beacons have the same r. We assume that the node deployed in the obstacle free area. All nodes are mobile and all use the same mobility model.

We use the libsvm [14] software and Radial Basis Function $K(S_i, S_j) = e^{-\gamma\|s_i - s_j\|^2}$ as a kernel function because of its empirical effectiveness where $\gamma > 0$ is a constant. Fig. 2 illustrates the error decreases by increasing the number of M. when M is 69, there is proximity 7.8% error in average but when M increases to 1027 average error decreases to 0.60%. Fig. 3 suggests increasing the beacon population decreases localization error. Even with 5% beacon it has very good accuracy. As shown in Fig. 4, the accuracy of our proposed algorithm is roughly closed to the static approach. Fig. 5 shows the effect of speed on the accuracy of the proposed algorithm. It has good results for different speed we experimented. Fig. 6 shows the effect of check threshold *T* for varies values of *T*.

5. DISCUSSIONS

The main important point in a sensor network is resource constraint. Therefore, an efficient localization algorithm is required for sensors with limited memory, computation and communication capacity.

Memory required maintaining information: Each node should keep the hop count from k beacons. Each one needs 1 byte. There is M class and SVM makes $u_i^* y_i$, b^* for each class. Both $u_i^* y_i$, b^* required 4 bytes. In a formula each node needs D$\times$ m $\times$(4k+4) +k bytes. D denotes dimensions in the formula. Example: each node in a network with 50 beacons and m=128 in 2D area required 254 $\times(4\times 50 + 4) + 50$ bytes. In terms of computation cost each node needs to compute $H_x(k)$ to estimate its position and beacons compute $H_x(k)$ periodically to check their position by employing position methods.

The SVM algorithm runs in the head beacon which assumed to be resourceful. Computation cost for SVM is $o(k_{sv}^2 k)$[13], where k_{sv} denotes the number of support vectors. This is less than the number of beacons. $o(M(k_{sv}^2 k))$ is the total runtime for M class.

In the point of communication cost, at the first time each beacon sends a broadcast message through the network, therefore, each sensor should broadcast k message. It means for each sensor less than 1 Kbyte to its neighbor nodes [13]. After that in the next steps if a beacon finds movement in the network then sends another broadcast. So it is not necessary for all beacons to send broadcast repeatedly. It avoids an extra communication overhead.

In terms of coverage, experiment shows fully coverage in the network. So this algorithm is feasible for applications with mobility required in large scale and dense nodes. It has good results by using only 5%

beacons. It is proved that error in x coordinate is equal to error in y coordinate if M is the same for both axes [13]. It is provable that error in z coordinate is also equal to x coordinate with equal M.

6. Conclusions

This paper presented a distributed localization algorithm for mobile wireless sensor networks based on SVM. Beacons produced training data to the learning algorithm. Connectivity information helps the algorithm be applicable for large scale area. Also the algorithm is suitable for three-dimensional regions. The experimental results demonstrate good accuracy. For future work, a network with obstacle will be considered.

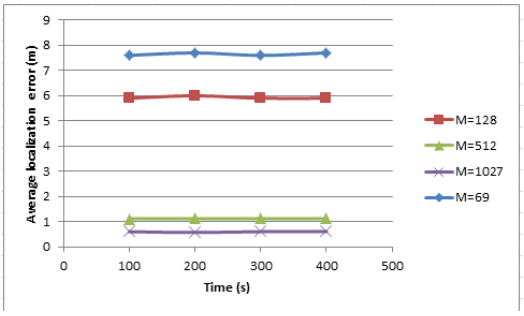

Fig.2. M comparison in the network with r =10 , B=5%, v=1 m/s and T=0.5 m.

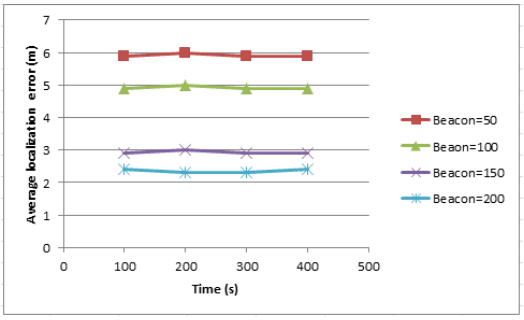

Fig.3. The effect of difference beacon population in a network with B=5%, communication range=10m and M=128 and T=0.5 m.

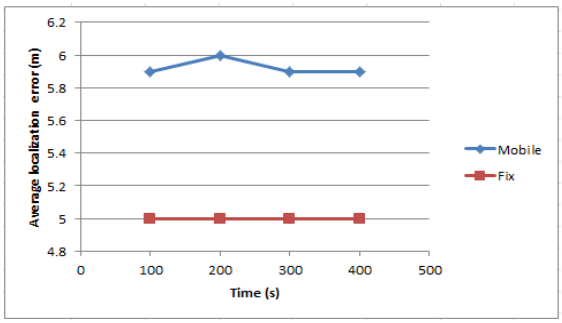

Fig.4. The proposed algorithm for mobile is very similar to the static algorithm. This plot is for B=5%, r =10 m and T=0.5 m.

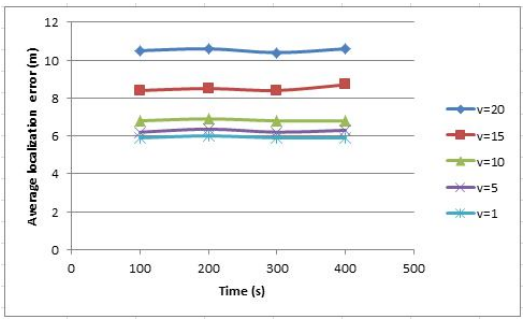

Fig.5. Comparison among different speed in a sensor network with B=5%, M=128 and r=10 m and T=0.5 m.

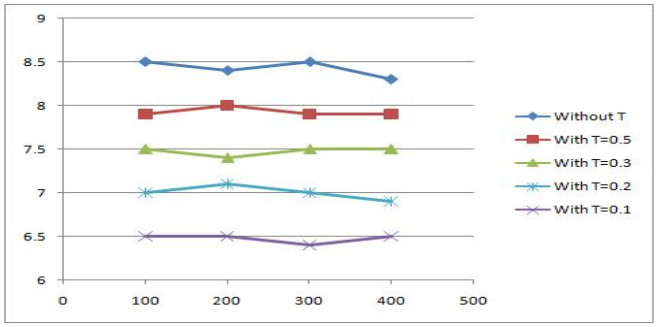

Fig.6. Efficiency of *T* when M=69 and r=10 m.

REFERENCES

[1] Aspnes,J., T. Eren, D.K. Goldenberg, A.S. Morse, W. Whiteley, Y.R. Yang, B.D. Anderson, and P.N. Belhumeur,2006. A theory of network localization:IEEE Transactions on Mobile Computing, 1663–1678.

[2] Le, T.N., Chong, P.H.J., Xue Jun Li, Wai Yie Leong, 2010. A simple Grid-based Localization Technique in Wireless Sensor Networks for Forest Fire Detection: Second International Conference on Communication Software and Networks, ICCSN'10: 93–98.

[3] Pal, A. 2010. Localization algorithms in wireless sensor networks: Current approaches and future challenges: Network Protocols and Algorithms, (2): 45–73.

[4] Afzal,S. and H. Beigy,2010. Techniqual report Sharif : University of Technology, International Campus-Kish Island.

[5] Burges,C.J., 1998. A tutorial on support vector machines for pattern recognition:Data mining and knowledge discovery, (2): 121–167.

[6] Letchner,J., D. Fox, and A. LaMarca,2005.Large-scale localization from wireless signal strength.Proceedings of the national conference on artificial intelligence: 15.

[7] Baggio, A. and K. Langendoen, 2008. Monte Carlo localization for mobile wireless sensor networks:Ad Hoc Networks, (6): 718–733.

[8] Pan,J.J., Q. Yang, H. Chang, and D.Y. Yeung,2006. A manifold regularization approach to calibration reduction for sensor-network based tracking:PROCEEDINGS OF THE NATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE, 988.

[9] Lorincz,K. and M. Welsh, 2007.MoteTrack: a robust, decentralized approach to RF-based location tracking:Personal and Ubiquitous Computing,(11): 489–503.

[10] Xuanlong,N.L., M.I. Jordan, and B. Sinopoli, A Kernel-Based Learning Approach to Ad Hoc Sensor.

[11] Kim,W., J. Park, and H.J. Kim, 2010.Target Localization Using Ensemble Support Vector Regression in Wireless Sensor Networks. Wireless Communications and Networking Conference (WCNC), 2010 IEEE, pp: 1–5.

[12] Pan,J.J. and Q. Yang, Co-localization from labeled and unlabeled data using graph laplacian:tC, (40): 70.

[13] Tran,D.A. and T. Nguyen, 2007. Localization in wireless sensor networks based on support vector machines:IEEE Transactions on Parallel and Distributed Systems, 981–994.

[14] Chang,C.C. and C.J. Lin, 2008. LIBSVM—A Library for Support Vector Machines, 2008:Last updated, (13).

Samira Afzal received the MS degree from Sharif University of Technology in Iran on January 2011. Since 2009, conducting research in the areas of machine learning, data mining and computer networks specially on ad-hoc and wireless sensor networks. She has been a professor and researcher of information technology at the Islamic Azad University, Marvdasht branch. From March 2011, she has been a professor in Sama technical and vocational training college in Islamic Azad University, Shiraz and Marvdasht branch. During 2010-2011 She was member of the Intelligence Systems laboratory in Sharif University of Technology in Iran.