International Journal on
"Technical and Physical Problems of Engineering"
(IJTPE)
Published by International Organization of IOTPE

ISSN 2077-3528
IJTPE Journal
www.iotpe.com
ijtpe@iotpe.com

September 2012 | Issue 12 | Volume 4 | Number 3 | Pages 41-46

AN EFFICIENT APPROACH TO DETECT FAULTY READINGS USING FUZZY LOGIC AND RISK NUMBER: APPLICABILITY IN LONG-THIN WIRELESS SENSOR NETWORKS

M. Sedighi [1] S.J. Dastgheib [1] A. Nikfarjam [2]

1. Department of Computer Engineering, Dezful Branch, Islamic Azad University, Dezful, Iran
sedighi.mohsen@yahoo.com, dastqeib@gmail.com
2. Young Researchers Club, Bushehr Branch, Islamic Azad University, Bushehr, Iran, nikfarjam.alireza@gmail.com

Abstract- Wireless sensor networks (WSNs) consist of hundreds or thousands of tiny sensor nodes that work together to do some special tasks. Detecting node with faulty readings is one of the important issues in WSNs. The weighted voting is usually used to detect faulty readings. Most of the existing algorithms use correlation of two nodes read vectors as weight to detect faulty readings, which result in high computational complexity. In addition, algorithms that use inverse of distance as weight have weaknesses in accuracy of calculations. This paper proposes a new fuzzy-based algorithm to detect faulty readings. Using an effective fuzzy inference system can improve the decision-making algorithm, which is used for detecting faulty readings in WSNs. We use normal and weighted averages of read vectors sensed values instead of single value or the entire of these vectors. The computation complexity of our algorithm is low and the accuracy is at a high level.

Keywords: Wireless Sensor Network, Faulty Readings, Fuzzy Logic, Weighted Averaging, Read Vector.

I. INTRODUCTION

Wireless sensor networks (WSNs) consist of hundreds or thousands tiny nodes that work together to do some special tasks. Each node consists of sensor, processor, antenna, a tiny memory and one energy supply. Because of the limited energy resources in WSN, it is required to use algorithms with less energy consumption. Detecting faulty nodes is one of the fundamental challenges in WSNs. The nodes are prone to produce incorrect data in noisy and unreliable environments. Faulty nodes produce arbitrary readings, which do not reflect the true state of environmental phenomenon or events under monitoring. Furthermore, nodes may report incorrect readings, which come from wrong inference and decision-making. Both arbitrary and noisy readings are caused by faulty readings. These failures may even prevent the benefit of WSNs. Thus, it is very important to identify and filter the faulty readings [1].

The Long-Thin Network (LTN) is a type of network topologies, which is widely used in wireless sensor network applications. Some applications of LTN are in surveillance application, including leakage detection of fuel pipes, monitoring tunnels, stage measurements in sewer, street lights monitoring in highway systems, flood protection of rivers, vibration detection of bridges, roadside networks, pedestrian detection systems and etc. In LTN, nodes may form several long backbones and these backbones extend the network to intend coverage areas. A backbone is a linear path, which may contain tens or hundreds of routers [2].

The fact that readings data from neighboring nodes are similar can be expressed by the spatial correlation. The set of neighboring nodes is called set of witnesses. Therefore, an idea to detect faulty readings is to use this correlation space. In other words, if a node (s_j) receives an unusual reading, this node asks its neighbors if the reading is faulty or not by sending suspicious reading to them (referring to the set of witnesses). Based on the classic voting of majority:

1. Each node within the set of witnesses (e.g. node s_i) makes a judgment by comparing its reading with an abnormal reading submitted by the suspect node (s_j).
2. If the difference between these two readings exceeds a predefined threshold, s_i considers the reading posted by s_j as faulty and gives a negative vote to s_j. Otherwise, s_i claims that s_j is normal and a positive vote returns to s_j.
3. After collecting votes from the neighbors, s_j decides if the reading is faulty or not. If the number of negative votes is more than positive votes, reading is considered as faulty; otherwise, it is considered as usual reading.

Nevertheless, this simple voting method does not work well when the number of faulty nodes is large. To solve this problem weighted voting algorithm is introduced [3, 4]. Assuming that closer nodes have more readings similarity, weighted voting algorithms allocate more weight to closer neighbors (e.g., weights of reverse of distance of a node with its neighbors).

There are two categories of weighted voting method:

- The first category is based on inverse of distance as weight.
- The second category is based on correlation between nodes readings as weight.

The second batch methods that will be explained later are very complex and expensive as energy. The first category has some weaknesses such as the fact that faulty node near the voting node can have a devastating impact on voting. Debraj de proposed a weighed voting algorithm in [1] and tried to cover these weaknesses but this problem has not been fully resolved.

In the second category, weight is considered as correlation of two nodes read vectors. These algorithms have high accuracy but are very costly. In paper [5] we try to reduce the energy consumption of these methods; however, this problem still exists. In this paper, we propose a novel approach to detect faulty reading using fuzzy logic, weighted averaging, normal averaging and risk number. Fuzzy logic, which will be explained briefly in section three, is used in the proposed method because it can reduce computational complexity, delay, and energy consumption and improve accuracy and performance [6-8].

Since sending entire read vector is very costly the suspected node sends normal and weighted averaging of read vector to its neighbors in the proposed method. We propose a new method to obtain the average of read vector in section five. Though sending entire read vector is costly, it provides high accuracy. In the proposed method, the averages are used instead of entire read vector which may reduce accuracy. To improve accuracy we use risk number, which will be introduced in section four, along with readings.

Based on above observations in this paper an innovative in-network voting scheme is proposed to detect faulty reading of a node by calculating the correlation of two nodes read vectors averages and their risk number. Since several sequences of numbers may have the same average, weighted averaging is calculated in addition to normal averaging. It will be proved that our voting algorithm is extremely accurate and imposes negligible cost to the network.

The rest of this paper is organized as follows. In section II, fault tolerant deployment techniques for long-thin wireless sensor networks are discussed. The fuzzy logic concept and its characteristics and applications are described in section III. Section IV introduces risk number and section V proposes a novel approach to averaging. Section VI proposes an effective fuzzy-based algorithm to detect faulty readings in WSN. Experimental results, analysis, and comparison the obtained results with the other models are discussed in section VII. Finally, section VIII concludes this paper.

II. FAULT TOLERANT DEPLOYMENT TECHNIQUE FOR LONG-THIN WIRELESS SENSOR NETWORKS

The Long-Thin Network (LTN) is a type of network topologies, which is widely used in wireless sensor network applications [9-11]. The form of nodes distribution in the Long-Thin Network (LTN) causes each node to have fewer neighbors. Few neighbors will cause fault in network. The number of neighbors should not be so little that compromise the health of network.

Long-Thin structures are usually used in environments that are included in the restrictions. These restrictions limit the number of neighbors. In this structure, failure of some close together nodes may pull some parts of network into isolation, or in a worse case the entire network may stop working. A structure for LTNs is shown below (Figure 1). This structure is an optimal deployment for the sensor nodes within the LTN, and is useful in most practical applications [1].

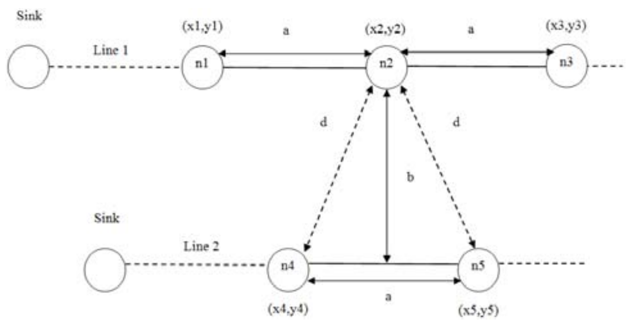

Figure 1. Proposed fault tolerant sub-structure for long-thin network

According to the definition of rectangular triangle:

$$d^2 = b^2 + \left(\frac{a}{2}\right)^2 \quad (1)$$

Therefore, the number of neighbors of a node can also increase by increasing the number of parallel lines and changing the value of parameters *a*, *b*.

III. FUZZY LOGIC

Before discussing the proposed method to detect faulty readings it is necessary to present an overview of fuzzy logic. Fuzzy Logic (FL) is defined as the logic of human thought, which is much less rigid than the calculations computers generally perform. Fuzzy Logic offers several unique features that make it a particularly good alternative for many control problems. It is inherently robust since it does not require precise, noise-free inputs and can be programmed to fail safely [12-15].

The output control is a smooth control function despite a wide range of input variations. Since, the FL controller processes user defined rules governing the target control system; it can be modified and tweaked easily to improve system performance. Fuzzy Logic deals with the analysis of information by using fuzzy sets, each of which may represent a linguistic term like "Warm", "High", etc. The range of real values over which the set is mapped is called domain and the membership function describes fuzzy sets. A membership function assigns a true (crisp) value between 0 and 1 to each point in the fuzzy set's domain. Depending upon the shape of the membership function, various types of fuzzy sets can be used such as triangular, trapezoidal, beta, PI, Gaussian, sigmoid, etc. We use triangular and trapezoidal membership functions because they are suitable for real-time operation, do not increase complexity of computations and have enough accuracy [16, 17].

The fuzzified values are processed by the inference engine, which consists of a rule base and various methods for inferring the rules. One of the fuzzy systems that is used in the inference engine of the expert system is the Mamdani fuzzy system. The Mamdani fuzzy system is a simple rule-base method that does not require complicated calculations which can employ the IF…THEN… rules to control systems [18]. All the rules in the rule-base are processed in a parallel manner by the fuzzy inference engine. The defuzzifier performs defuzzification on the fuzzy solution space. That is, it finds a single crisp output value from the solution fuzzy space. Some techniques are introduced for deffuzification like Center of Area (COA), mean of maximum and etc. COA is the most suitable technique for WSN so we use this technique for defuzzification [19]. In this study, the crisp value adopting the COA defuzzification method was obtained by the following formula.

$$Crisp\ Output\ (\alpha) = \frac{\int_z \mu_A(x) z dz}{\int_z \mu_A(x) dz} \quad (2)$$

where α is the crisp value for the "z" output and $\mu_A(z)$ is the aggregated output membership function [20].

IV. RISK NUMBER

In this section, we define risk number. It means that a faulty node may destroy voting results. Therefore, we should control risky nodes before voting is done. We know that readings of a node are in a range. If these readings are scattered, it means that something is wrong. Each node can calculate the amount of its scattering readings. Suppose that all readings of node s_i include a sequence of readings inside the sliding window Δt, This sequence readings is called Read Vector. The s_i readings can be expressed as follows:

$$b_i(t) = \{x_i(t - \Delta t + 1),\ x_i(t - \Delta t + 2), ..., x_i(t)\} \quad (3)$$

Each node calculates the risk number when the suspected node is required to vote. The node can obtains average, minimum and maximum of read vector. If the differences between minimum and maximum, between minimum and average, or between maximum and average exceed a predefined threshold it means that the node is risky. In this case, the risk number is set one, otherwise is set zero. The node decides about faulty readings of voting node. To lessen the effect of risky node on voting, the weight of this node is reduced to half.

V. AVERAGING ALGORITHM

We explained before in part I that weighted voting methods to detect faulty readings based on correlation between read vectors of two nodes as weight are accurate but costly. To reduce the cost of these methods one approach is to use average of read vectors instead of whole vectors in computations. This is a good approach, which has an acceptable level of accuracy and very low computational complexity. However, there is a major problem. Average is not a good representative of a sequence (vector) of numbers. Note the following example. It shows that several different data vectors have the same average.

Average (148, 1, 1) = Average (51, 50, 49) = Average (99, 50, 1) = ... = 50

To solve this problem a weighted averaging approach is suggested. Each number is multiplied by a coefficient, which is a number between 0 and 1. Weighted averaging is done as follows.

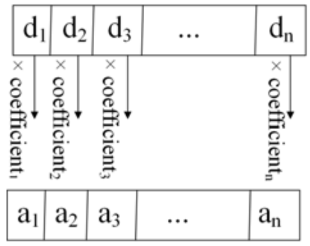

Figure 2. Changing the numbers sequence

$$Wighted\ Averaging = \frac{\sum_{i=1}^{n} a_i}{\sum_{i=1}^{n} coefficeint_i} \quad (4)$$

where d is normal number and a is changed number. However, in some cases weighted averaging of two vectors may be the same. To prevent this problem we use these two averages. For simplicity, we consider three coefficients repeated respectively in the vector. These coefficients can be 0.2, 0.5 and 0.8.

In normal algorithm, averaging task is performed on an array (vector) of data. Using array is usually costly and causes increasing delay. The array is omitted in the proposed method. Therefore, this approach is useful for environments that require real-time processing. Averaging proposed algorithm for node n_i is shown below:

Averaging Algorithm:

```
While (ni senses)
    {
        sum1 = sum1 + sensed_value
        normal_average = sum1/counter
        if (counter%3 == 0)
        {
            coefficient = 0.2
            divided = divided + 0.2
        }
        if (counter%3 == 1)
        {
            coefficient = 0.5
            divided = divided + 0.5
        }
        if (counter%3 == 2)
        {
            coefficient = 0.8
            divided = divided + 0.8
        }
        changed_data = sensed_value * coefficient
        sum2 = sum2 + changed_data
        weighted_averaging = sum2/divided
        counter = counter + 1
    }
```

VI. PROPOSING AN EFFECTIVE FUZZY-BASED ALGORITHM TO DETECT FAULTY READINGS

A node with suspected readings starts the process of voting to detect faulty reading. Consider node n_2 in the substructure long-thin wireless sensor network (Figure 1) starts voting to understand whether its readings are faulty or not.

According to averaging algorithm, each node has averages of its readings. The node n_2 sends averages of its readings to the neighboring nodes (i.e. 1, 3, 4, 5). Each of them runs fuzzy system in Figure 3 to obtain its weight on voting and decides about the suspected node readings.

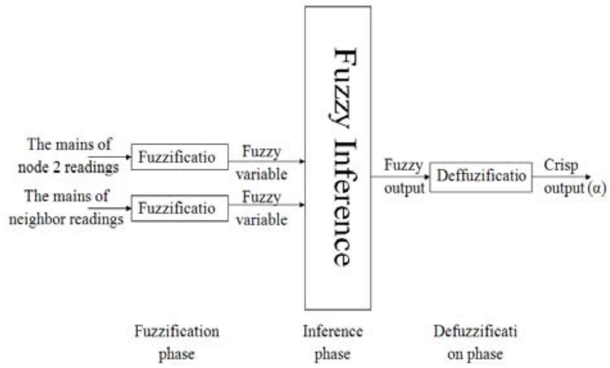

Figure 3. The components of the fuzzy system

The crisp output is the weight of the neighbor node in voting. This value is obtained by using Mamdani inference system on the fuzzy inputs. The rules, which are used in Mamdani inference system are listed in the Table 1.

Table 1. Existing rules

Node 2 averages	Neighbor averages	Crisp output
H	H	VH
H	L	M
H	M	H
M	H	H
M	M	M
M	L	L
L	L	VL
L	H	M
L	M	L

As maintained in section three, the fuzzy membership function is used for converting crisp values to fuzzy variables. The input and output fuzzy membership functions are shown in Figure 4.

According to Figure 4, Table 1 and Equation (2), the COA method can obtain the crisp output of the fuzzy system that is the amount of weight of neighbor node in voting. After performing the fuzzy system each neighboring node turns back its vote and weight to node 2. The node 2 does the voting by Equation (5).

$$Voting_2 = \sum_j weight_j \times vote_j \tag{5}$$

The superiority of our method over other approaches to detect faulty readings will be shown in the next section.

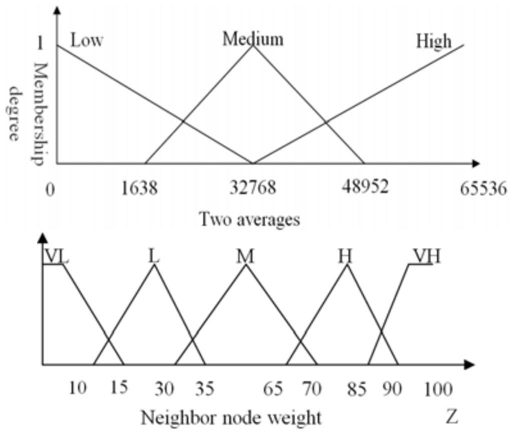

Figure 4. The fuzzy membership functions

VII. ANALYZING AND COMPARING

In This section, we review previous methods to detect faulty readings and compare them with the proposed method and show that the new method works interestingly better than the existing methods. As mentioned, the weighted voting methods are divided in two categories. The first category is weighted voting algorithms that use distance inverse as weight. They have lower complexity but are very vulnerable. One of the weaknesses of these algorithms is the high effect of faulty nodes that are near the voting node on the voting result. One of these algorithms is Debraj de that decreases the vulnerable down to the acceptable level, but this problem still exists. Debraj de uses the voting Equation (6).

$$w_{ij} = \frac{confidence_{ij}}{d_{ij}}, \; Vote_i = \sum_j w_{ij} s_j \tag{6}$$

In this formula, d_{ij} is the distance between two nodes, w_{ij} and $confidence_{ij}$ are respectively weight of node j in node i voting and the confidence number between i,j, which is obtained from Debraj delocalization error detection algorithm. The s_i is the current sensed value by node i.

The second category is weighted voting algorithms that use correlation between two nodes read vector as weight. The complexity of these algorithms is high and about $O(n^3)$. We can reduce this complexity by proposing a new method for voting in our previous method that is proposed in [5]. However, the complexity, energy consumption and delay still are high. The order of computational complexity in the previous proposed method is about $O(n^2)$ which is high.

It is easily proved that our approach is much more accurate than the first category algorithms. Note the following figure.

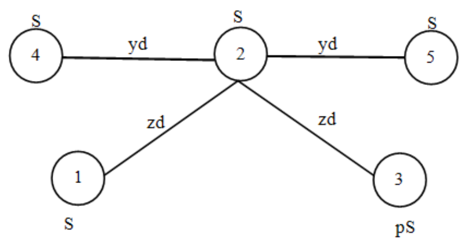

Figure 5. Evaluation of faulty node effect on the voting

In Figure 5, S is sensed value by the nodes and d indicates the distance. The z, y, p are fix numbers. According to Debraj de voting method that is in the first category, the following relations are obtained. These relationships are used to detect the faulty readings of node 2. Node with Number 3 is a faulty node that is going to destroy the voting result.

$$w_{12}=\frac{3}{zd},\ w_{23}=\frac{1}{zd},\ w_{24}=w_{25}=\frac{3}{yd}$$

$$Vote_i=\frac{3}{zd}\times S-\frac{1}{zd}\times pS+\frac{6}{yd}\times S=\frac{S}{d}\left(\frac{3}{z}-\frac{p}{z}+\frac{6}{y}\right)$$

To node 3 can destroy voting must:

$$\frac{3}{z}-\frac{p}{z}+\frac{6}{y}<0 \xrightarrow{\text{multiply } xy \text{ in both side}} 3y+py+6z<0$$

We can write wz instead of y, and then we have:

$$3wz-wpz+6z<0 \ \rightarrow\ \text{discard } z \ \rightarrow\ 3w+6<pw$$

As was seen, in this case if unequal is established, voting is wrong! For example, if $w=2$ then:

$$3\times 2z-2pz+6z<0 \ \rightarrow\ 6<p$$

According to this example, if p is greater than six voting failed. Obviously, it shows a very low accuracy and very high risk of inaccuracy. With the decreased distance between faulty node and voting node (increasing w) the effect of the faulty node on voting brutally increases.

To resolve this problem we can use algorithms that use the amount of their readings similarity or correlation instead of the distance between two nodes. These algorithms are the second category of voting methods. They are often time consuming and expensive.

Here we deal with the proposed algorithm (Equation (5)). According to figure 5, node with number 3 tries to destroy voting result. We assume that node 3 is risky node. Consider the following relations.

$\alpha_{1,4,5}=100$; α_3 is maximum 100; Since 3 is risky node its α changed to 50.

$$Voting_2=+100-50+100+100=+250>\text{zero}$$

$$\rightarrow \text{voting is correct!}$$

We prove that our algorithm solved the problem of the first category of voting algorithms. The previous proposed method can solve this problem but is very costly and since it does not consider the entire read vector may have some problems. The proposed method is an efficient approach by using entire read vector, fuzzy logic and risk number.

Figure 6 shows the number of bits which should be transferred among the nodes to detect faulty readings in the proposed method and previous proposed method. Since sending and receiving data consume very high energy of nodes in WSN than the other operations, the following diagram clearly shows that the proposed method is more energy efficient that the previous proposed method.

The proposed method has less complexity and the energy consumption to calculate faulty reading is modest. Look at Table 2.

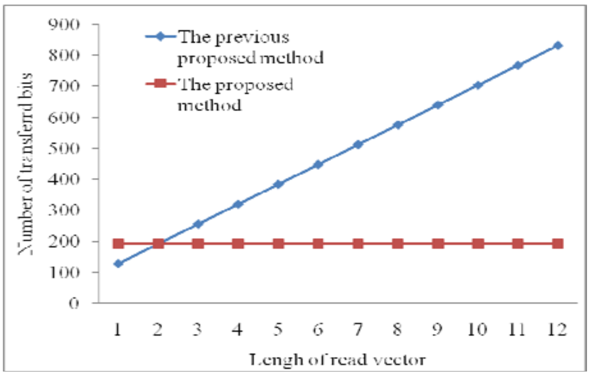

Figure 6. The number of transferred bits

Table 2. Comparing between algorithms with some parameters

Algorithm / Parameter	Pervious proposed method	Debraj de's Algorithm	weighted voting with correlation	proposed algorithm
Precision	high	low	very high	very high
Complexity	high	low	very high	low
Energy Consumption	high	low	very high	low
Delay	high	low	very high	low

VIII. CONCLUSIONS

Our motivation of doing this research was to achieve an efficient technique to detect nodes with faulty readings. The voting technique is used for detecting faulty readings in most of the existing methods. Voting mainly is done in two ways: weight based on correlation and weight based on distance. Use of voting based on calculating the correlation of two nodes is often costly. Voting based on distance have less complexity but are very vulnerable. Debraj de can reduce vulnerability of these voting. Debraj de voting in accordance with what was shown in comparing chapter, (because of using distance parameter in voting) does not always operate properly. The complexity of our algorithm is low and near the algorithms that use inverse of distance as weight. In addition, our proposed algorithm solves the problems of these algorithms.

REFERENCES

[1] D. De, "A Distributed Algorithm for Localization Error Detection Correction, Use in In-Network Faulty Reading Detection: Applicability in Long-Thin Wireless Sensor Networks", IEEE Wireless. Communications and Networking Conference, pp. 1-6, April 2009.

[2] F. Caballero, L. Merino, P. Gil, I. Maza, A. Ollero, "A Probabilistic Framework for Entire WSN Localization Using a Mobile Robot", Robotics and Autonomous Systems, Vol. 56, Issue 10, pp. 798-806, 31 October 2008.

[3] B. Krishnamachari, S. Iyengar, "Distributed Bayesian Algorithms for Fault-Tolerant Event Region Detection in Wireless Sensor Networks", IEEE Transactions on Computers, Vol. 53, No. 3, pp. 241-250, 2004.

[4] T. Sun, L.J. Chen, C.C. Han, M. Gerla, "Reliable Sensor Networks for Planet Exploration", IEEE

International Conference on Networking, Sensing and Control (ICNSC), pp. 816-821, 19-22 March 2005.
[5] A. Barati, S.J. Dastgheib, A. Movaghar, I. Attarzadeh, "An Optimized Algorithm to Detect Faulty Readings along the Substrate Access Wireless Long-Thin Sensor Networks", UKSim 5th European Symposium on Computer Modeling and Simulation, IEEE Computer Society, pp. 372-377, 16-18 Nov. 2011.
[6] H.Y. Lee, T.H. Cho, "Fuzzy Logic Based Key Disseminating in Ubiquitous Sensor Networks", 10th International Conference on Advanced Communication Technology (ICACT), pp. 958-962, Feb. 2008.
[7] B. Lazzerini, F. Marcelloni, M. Vecchio, S. Croce, E. Monaldi, "A Fuzzy Approach to Data Aggregation to Reduce Power Consumption in Wireless Sensor Networks", Fuzzy Information Processing Society, NAFIPS, Annual Meeting of the North American, pp. 436-441, June 2006.
[8] S.Y. Chiang, J.L. Wang, "Routing Analysis Using Fuzzy Logic Systems in Wireless Sensor Networks", 12th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems, pp. 966-973, September 2008.
[9] M.S. Pan, H.W. Fang, Y.C. Liu, Y.C. Tseng, "Address Assignment and Routing Schemes for ZigBee-Based Long-Thin Wireless Sensor Networks", IEEE Vehicular Technology Conference, pp. 173-177, 2008.
[10] F.J. Wu, Y.F. Kao, Y.Ch. Tseng, "From Wireless Sensor Networks Towards Cyber Physical Systems", Pervasive and Mobile Computing, Vol. 7, Issue 4, pp. 397-413, August 2011.
[11] G. Montenegro, S. Dawkins, M. Kojo, V. Magret, N. N. Vaidya, "Long Thin Networks", RFC 2757, January 2000.
[12] T.J. Ross, "Fuzzy Logic with Engineering Applications", Third Edition, John Wiley and Sons Ltd. Publication, ISBN: 978-0-470-74376-8, 2010.
[13] L.A. Zadeh, "Fuzzy Logic", IEEE Computing & Processing (Hardware/Software), Vol. 21, Issue 4, pp. 83-93, 1988.
[14] M. Alivirdizadeh, N.M. Tabatabaei, A. Demiroren, N.S. Boushehri, "Transient Stability Improving of Power System Including DFIG Based Wind Farm by Using Fuzzy Logic Controller", International Journal on. Technical and Physical Problems of Engineering. (IJTPE), Issue 10, Vol. 4, No. 1, pp. 66-73, March 2012.
[15] A.A. Allahverdiyev, "Application of Fuzzy-Genetic Algorithm for Solving an Open Transportation", International Journal on Technical and Physical Problems of Engineering (IJTPE), Issue 7, Vol. 3, No. 2, pp. 119-123, June 2011.
[16] Zh.J. Bose, "BK: Evaluation of Membership Functions for Fuzzy Logic Controlled Induction Motor Drive", IECON Proceedings (Industrial Electronics Conference), Vol. 1, pp. 229-234, 2002.
[17] W. Pedrycz, "Why Triangular Membership functions?", Fuzzy Sets and Systems, Elsevier, Vol. 64, Issue 1, pp. 21-30, 1994.
[18] Sh. Shamshirband, S. Kalantari, Z.S. Daliri, L.Sh. Ng, "Expert Security System in Wireless Sensor Networks Based on Fuzzy Discussion Multi-Agent Systems", Scientific Research and Essays, Vol. 5, No. 24, pp. 3840-3849, 18 December, 2010.
[19] A. Patel, B. Mohan, "Some Numerical Aspects of Center of Area Defuzzification Method", Fuzzy Sets and Systems, Vol. 132, pp. 401-409, 2002.
[20] A. Barati, S.J. Dastgheib, A. Movaghar, I. Attarzadeh, "An Effective Fuzzy Based Algorithm to Detect Faulty Readings in Long Thin Wireless Sensor Networks", International Journal on Technical and Physical Problems of Engineering. (IJTPE), Issue 10, Vol. 4, No. 1, pp. 52-58, March 2012.

BIOGRAPHIES

Mohsen Sedighi was born in Shiraz, Iran, in 1987. He received the B.Sc. degree in Computer Software Engineering from Sepahan University, Isfahan, Iran, in 2009. He is currently M.Sc. student in Computer Hardware Engineering from Dezful Branch, Islamic Azad University, Dezful, Iran. His research interests include computer arithmetic, wireless sensor network, intelligent and fault tolerant systems.

Seyyed Jalaleddin Dastgheib was born in Shiraz, Iran, in 1986. He received the B.Sc. degree in Computer Software Engineering from University of Tabriz, Tabriz, Iran, in 2008. He also received the M.Sc. degree in Computer Hardware Engineering from Dezful Branch, Islamic Azad University, Dezful, Iran, in 2011. His research interests include computer arithmetic, wireless sensor network, fuzzy systems, and fault tolerant systems.

Alireza Nikfarjam was born in Bushehr, Iran in January 1990. He received his B.S. degree in Information Technology from Bushehr Branch, Islamic Azad University, Bushehr, Iran in 2012. He is also a member of Young Researchers Club of the same university. His research interests are evolutionary computation, optimization, wireless sensor networks and artificial neural networks.