Neurocomputing 94 (2012) 111–120

Contents lists available at SciVerse ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Application of fuzzy inference systems to detection of faults in wireless sensor networks

Safdar Abbas Khan [a,c,*], Boubaker Daachi [a], Karim Djouani [a,b]

[a] LISSI Laboratory, EA-3956, University of Paris East, France
[b] F'SATI, Tshwane University of Technology, South Africa
[c] IIT, Quaid-i-Azam University, Islamabad, Pakistan

ARTICLE INFO

Article history:
Received 9 November 2011
Received in revised form
25 February 2012
Accepted 8 April 2012
Communicated by H.R. Karimi
Available online 23 May 2012

Keywords:
Wireless sensor networks
Fault detection
Fuzzy inference systems
Neural networks
Distributed scheme

ABSTRACT

In this paper we present a fault detection strategy for wireless sensor networks. The strategy is based on modeling a sensor node by Takagi–Sugeno–Kang (TSK) fuzzy inference system (FIS), where a sensor measurement of a node is approximated by a function of the sensor measurements of the neighboring nodes. We also model a node by recurrent TSK-FIS (RFIS), where the sensor measurement of the node is approximated as function of real measurements of the neighboring nodes and the previously approximated value of the node itself. Temporary errors in sensor measurements and/or communication are overcome by redundancy of data gathering. A node which has developed a faulty sensor is not completely discarded because it is useful for relaying the information among the other nodes. Each node has its own fuzzy model that is trained with input of neighboring sensors' measurements and an output of its actual measurement. A sensor is declared faulty if the difference between the outcome of the fuzzy model and the actual sensor measurement is greater than the prescribed amount depending on the physical quantity being measured. Simulations are performed using the fuzzy logic toolbox of Matlab. We also give a comparison of obtained results to those from a feed-forward artificial neural network, recurrent neural network and the median [1] of measured values of the neighboring nodes.

1. Introduction

Wireless sensor networks are emerging as computing platforms for monitoring various environments including remote geographical regions, office buildings and industrial plants [2]. They consist of the following: a set of nodes that can communicate with each other; sensors that measure a desired physical quantity; and the system base station for data collection, processing, and connection to the wide area network. Modern wireless sensor nodes have microprocessors for local data processing, networking, and control purposes [3]. WSNs have enabled numerous advanced monitoring and control applications in environmental, biomedical, and numerous other applications.

One of the motivations for WSN modeling stems from the need for intelligent fault detection in complex distributed sensory systems. Because sensor networks often operate in potentially hostile and harsh environments, most of the applications are mission critical. The sensors are often used to compute control actions [4–6], where sensor faults can cause catastrophic events. For instance, the National Aeronautics and Space Administration was forced to abort the launch of the space shuttle *Discovery* due to a failure in one of the sensors in the sensor network of the shuttle's external tank (the failure was discovered through human inspection) [7].

Sensors and actuators boarded on a WSN node are more prone to faults as compared to traditional integrated semiconductor chips. Feedback about the functionality status of nodes is mandatory for multisensor systems so that they could eventually recover and heal from possible faults. Components such as sensors and actuators have significantly higher fault rates than the traditional integrated semiconductor circuits-based systems. Multisensor systems need feedback information about the health status of their nodes in order to recover and heal from eventual faults. This would enhance the reliability on the system. Due to malfunctions or noise the sensor reading are more or less uncertain in the sense that no sensor will render an accurate reading at all times. Because low-cost sensor nodes are often deployed in an uncontrolled or even harsh environment, they are vulnerable to have faults. It is thus desirable to detect, locate the faulty sensor nodes, and exclude them from the network during normal operation unless they can be used as communication node. Consequently we need to design a WSN that is capable of fault detection [7–9]. Efficiency in converting data to features

* Corresponding author at: LISSI Laboratory, EA-3956, University of Paris East, France. Tel.: +923005193823.
E-mail addresses: safdar-abbas.khan@u-pec.fr (S.A. Khan), daachi@u-pec.fr (B. Daachi), djouani@u-pec.fr (K. Djouani).

0925-2312/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.neucom.2012.04.002

while consistently accommodating the uncertainty inherent in the measurements forms a key issue for diagnosing and dealing with sensor faults [8,9].

The ancient method for fault tolerance is to equip a node with multiple sensors but doing so would not only increase the cost of a node and hence that of the network but would also lead in more complexity and power consumption. So recent works are centered around analytical redundancy [10,11] in which the sensor measurements are processed analytically, and the mathematical models are compared with the physical measurements. Therefore, instead of using additional hardware we use analytical fault detection, and model each node of a WSN through Takagi–Sugeno–Kang (TSK) fuzzy inference system (FIS).

Fault detection and fault tolerance in wireless sensor networks have been investigated in many research works. In [12] diagnosis for sensor networks has been carried out with additional attention to the congestion avoidance at the central node. In [13] Koushanfar et al. have proposed an online detection technique for faulty sensors, where nonparametric statistical methods are used to identify the sensors that have the highest probability to be faulty. In [14] the problem of fault identification in ad hoc networks is addressed. The diagnostic model lies upon the comparison-based one-to-many communication paradigm. In [15] Ruiz et al. have developed a management architecture for detection of faults in event-driven WSNs. In [1] the identification of faulty sensors in reach of events is discussed. The proposed generic algorithms are localized and thus scalable for large networks, however those are limited due to uneven distribution of nodes. In [7] a node is identified as faulty depending upon the comparison of the output from a modified recurrent neural network to real measurement. In [16] a solution to the fault-feature disambiguation problem in sensor networks is proposed in the form of Bayesian fault-recognition algorithms exploiting the notion that measurement errors due to faulty equipment are likely to be uncorrelated, while environmental conditions are spatially correlated. In [17] the fault correction problem for distributed event detection in a WSN is studied. This distributed fault-tolerant detection scheme achieves optimal results when the neighborhood size is chosen based on the given detection error bound such that better balance between detection accuracy and energy usage is obtained.

In [18] the authors have presented a localized fault detection algorithm to identify the faulty sensors. It uses local comparisons with a modified majority voting, where each sensor node makes a decision based on comparisons between its own sensing data and neighbors' data, while considering the confidence level of its neighbors. The scheme, however, is a little complex in the sense that information exchange between neighboring nodes has to occur twice to reach a local decision based on a threshold. In addition, it does not allow transient faults in sensor reading and internode communication, which could occur for most normal sensor nodes [19]. Transient faults in sensing and communication have been investigated in [20], where a simple distributed algorithm has been proposed to tolerate transient faults in the fault detection process. In [21,22] the authors have proposed a filtering approach for the fault detection that is robust to false alarms. Our proposed approach does not require a false alarm. In [23], the authors have developed an adaptive intelligent technique based on artificial neural networks combined with advanced signal processing methods for systematic detection and diagnosis of faults in industrial systems based on classification method. In [24] the authors have presented a neuro-fuzzy networks based scheme for fault detection and isolation of a u-tube steam generator in a nuclear power plant. Some other fault management schemes can be found in the survey written by Yu et al. [25].

The rest of the paper is organized as follows: Section 2 presents the system model and the assumptions made. In Section 3 we discuss how we are treating the problem of fault detection. Sections 4 and 5 respectively represent the fuzzy inference modeling and the neural network modeling for the sensor fault detection. In Section 6 we discuss the implementation of the proposed approach. In Section 7 we present and discuss the simulation results, and finally in Section 8 we conclude this paper.

2. System model

The system under consideration accommodates n number of localized stationary homogeneous sensor nodes with unique identity number and same transmission range, which communicate via a packet radio network. The proposed algorithm assumes all nodes are fault free during deployment and during the training of the fuzzy inference system. For each node an FIS is computed and carried out at the base station and not on the node itself as is done in [7], where each node has a modified recurrent neural network installed on it. The communication algorithm ensures that each sensor knows the identity of its neighbor, MAC protocol solves contention problem over logical link, the link level protocol provides one hop broadcast.

2.1. Communication model

The communication graph of a WSN is represented as a graph $G(V,E)$, where V represents the set of sensor nodes in the network and E represents the set of edges connecting sensor nodes. The Cartesian coordinates of the node A_i are represented by $(A_{i,1},A_{i,2})$. Two nodes A_i and A_j are said to have an edge in the graph if the distance

$$d(i,j)=\sqrt{(A_{i,1}-A_{j,1})^2+(A_{i,2}-A_{j,2})^2} \tag{1}$$

between them is less than r (transmission range). That is

$$d(i,j)\le r \Leftrightarrow (A_i,A_j)\in E \tag{2}$$

For convenience we assume that G is undirected, which means that if $(A_i,A_j)\in E$ then $(A_j,A_i)\in E$. The communication graph can be a test graph in our fault detection if two nodes with an edge connecting them are compared. If some of the edges are not involved in the fault detection or ignored based on the previous test results, a test graph in our fault detection can be a subgraph of the communication graph. For simplicity, we assume that communication graph and test graph are the same. For the graph $G(V,E)$ and $A_i \in V$, the set of the neighbors of A_i, $N(A_i)$ is defined to be

$$N(A_i) := \{A_j \in V : (A_i,A_j)\in E\} \tag{3}$$

For two connected nodes $(A_i,A_j)\in E$ we define a set

$$D_{i,j} := N(A_j)-(N(A_i)\cup \{A_i\}) \tag{4}$$

2.2. Fault model

The value measured by node A_i at kth instant of time, t_k, is denoted by x_i^k. If the time instant is not explicitly required the sensor measurement shall simply be denoted by x_i. Nodes with permanent faulty sensors are to be identified but are not excluded from the network because they are useful in relaying data packets among the nodes. Nodes with transient errors in sensor reading are termed as fault-free.

For a homogeneous physical quantity the difference, between the measured value at a fault-free sensor with the measured values of its fault-free neighbors, is bounded. Thus, if A_i and A_j are neighbors then in case of possessing fault-free sensors the following condition is satisfied:

$$|x_i-x_j| \le \delta \tag{5}$$

where δ may vary depending on the application. If temperature is the physical quantity being measured, for example, then a sensor node and its neighbors are expected to have similar temperatures. Hence δ is expected to be a small number. If the local binary decision at each node, instead of the sensed data, is transmitted to its neighbors, δ is set to 0.

3. Fault detection

Two nodes A_i and A_j are compared only if $(A_i,A_j) \in E$. Thus at time instant t_k if two such nodes have fully functional sensors then

$$|x_i^k - x_j^k| \le \delta_{i,j}^k \tag{6}$$

Suppose A_i has m neighbors, i.e., $|N(A_i)| = m$. As shown in Fig. 1, let these neighbors be denoted by

$$N(A_i) = \{A_{i_1}, A_{i_2}, \ldots, A_{i_m}\}$$

So for this particular node we have

$$|x_i^k - x_{i_j}^k| \le \delta_{i,i_{j'}}^k \quad \text{for } 1 \le j \le m$$

Equivalently we can write it as

$$x_i^k = x_{i_j}^k + \epsilon_{i,i_j}^k \quad \text{for } 1 \le j \le m$$

where ϵ_{i,i_j}^k is the difference between the ith sensor measurement and that of its jth neighbor at the instant t_k. Whence

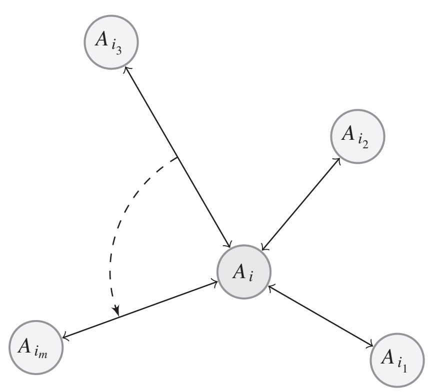

Fig. 1. Neighbors of node A_i.

we get

$$mx_i^k = \sum_{j=1}^{m} (x_{i_j}^k + \epsilon_{i,i_j}^k)$$

or

$$x_i^k = \frac{1}{m} \sum_{j=1}^{m} (x_{i_j}^k + \epsilon_{i,i_j}^k) \tag{7}$$

Eq. (7) represents a relation between the real sensor measurement of the node A_i and the sensor measurements of all of its neighbors. Which means the sensor measurement of A_i can be approximated by an m-variable function f of neighboring sensor measurements. That is

$$x_i^k \approx f(x_{i_1}^k, x_{i_2}^k, \ldots, x_{i_m}^k) \tag{8}$$

Hence for this node we create a TSK FIS which is trained with inputs as the sensor measurements of $N(A_i)$ nodes and output as the real sensor measurement of the node A_i.

4. TSK fuzzy treatment

The fuzzy logic system (FLS) [26,27] is an inference system which mimics the human thinking and its basic configuration consists of a fuzzifier, some fuzzy IF–THEN rules, a fuzzy inference engine and a defuzzifier. A fuzzy rule is written as the following statement:

$$R^l : \text{IF } x_1 \text{ is } B_1^l \text{ and } x_2 \text{ is } B_2^l \text{ and} \cdots x_n \text{ is } B_n^l \text{ THEN } y \text{ is } y^l$$

where R^l $(l = 1,2,\ldots,M)$ denotes the lth implication, x_j $(j = 1,2,\ldots,n)$ are input variables of the FLS, y^l is a singleton, B_j^l is the fuzzy membership function which can represent the uncertainty in the reasoning. When we use the product inference, center-average and singleton fuzzifier, the output of the fuzzy system for an input $\mathbf{x} = (x_1,x_2,\ldots,x_n)^T$ can be expressed as

$$y = \frac{\sum_{i=1}^{n} \alpha_i y^i}{\sum_{i=1}^{n} \alpha_i} \tag{9}$$

where α_i implies the overall truth value of the premise of the ith implication, and is computed as

$$\alpha_i = \prod_{l=1}^{M} A_l^i(x_i) \tag{10}$$

We are also using a recurrent FIS in which the added input is the previously approximated value, as shown in Fig. 2. The structure of the fuzzy controller is depicted in Fig. 3.

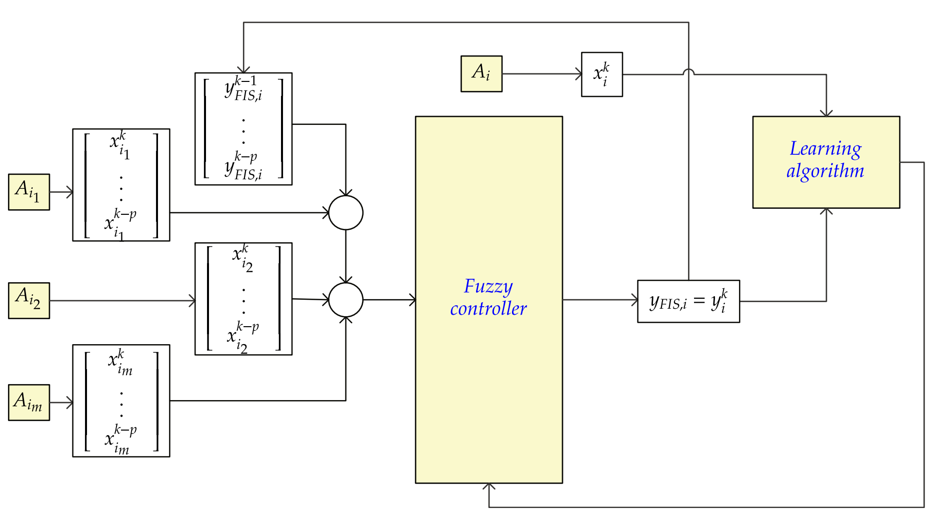

Fig. 2. Recurrent fuzzy inference system for fault detection of a sensor.

The first block in the fuzzy controller is *fuzzifier*, which converts each piece of input data degrees of membership by a lookup in one or several membership functions. The fuzzification block thus matches the input data with the conditions of the rules to determine how well the condition of each rule matches that particular input instance. The *fuzzy rule base* uses several variables both in the condition and the conclusion of the rules. The rules are presented in the *if–then* format. In the *fuzzy inference engine* the corresponding rules are activated and all the activations are accumulated using max–min operations. In the *defuzzifier* the crisp output value (9) is calculated which is the abscissa under the center of gravity of the fuzzy set.

5. Neural network treatment

Neural networks (NNs) imitate the human brain to perform intelligent tasks. They can represent complicated relationships between input and output variables, and acquire knowledge about these relationships directly from the data [28,29]. In this work we have used a multilayer perceptron (MLP) NN that consists of an input layer, a nonlinear hidden layer and a linear output layer, as shown in Fig. 4. The input vector $\boldsymbol{\alpha}=[\alpha_1,\alpha_2,\alpha_3,\alpha_4,\alpha_5]$ has components $[x_1^k,x_2^k,x_7^k,x_{11}^k,x_{12}^k]$, which are the sensed values at the neighbors of node A_6, as would be discussed latter in this paper. The 5×6 matrix $\mathbf{U}=[\beta_{i,j}]$ represents the input-to-hidden layer weights. The activation function of each of the hidden layer neuron is denoted by σ_i for $i=1,2,\ldots,6$. Each of the σ_i is the logsigmoid function. These activation functions are represented by a vector $\boldsymbol{\sigma}^T=(\sigma_1,\sigma_2,\ldots,\sigma_6)$. The vector $\mathbf{w}^T=(w_1,w_2,\ldots,w_6)$ represents the hidden-to-output layer weights. The activation function of the output layer is denoted by η and is the linear identity function. The single scalar output x_{NN} is the sensor measurement approximated by the neural network

$$x_{NN}\approx\eta\left\{\sum_{j=1}^{6}w_j\sigma_j\left(\sum_{i=1}^{5}\alpha_i\beta_{i,j}\right)\right\} \tag{11}$$

In matrix form which is written as

$$x_{NN}(\boldsymbol{\alpha})=\eta\{\mathbf{w}^T\boldsymbol{\sigma}(\mathbf{U}^T\boldsymbol{\alpha})\} \tag{12}$$

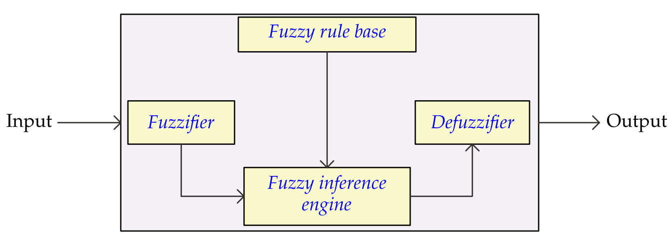

Fig. 3. The fuzzy controller as shown in Fig. 2.

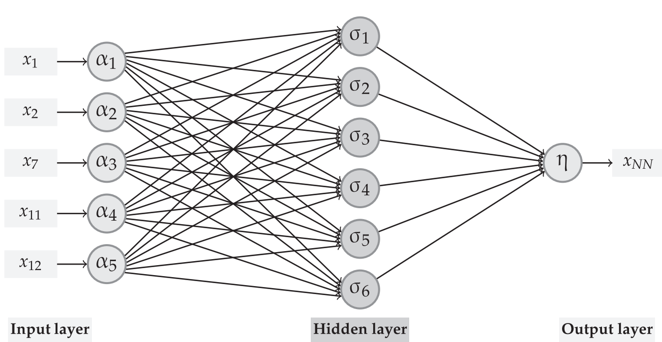

Fig. 4. Three layered neural network for node A_6 with five input variables and one output variable.

6. Implementation of the proposed approach

Suppose we want to measure the health status of the node A_i. So for this node we train an initial TSK FIS with input $\mathbf{x}_{FIS}=(x_{i_1},x_{i_2},\ldots,x_{i_m})^T$ and output $y_{FIS}=x_i$. The type of membership function is Gaussian. The number of membership functions for each component of input vector depends upon the range of temperature being measured. Here we are using five membership functions for each neighboring sensed value x_{i_j} for $j=1,2,\ldots,m$. So the fuzzy rules for node A_i are given by

$$R^l: \text{IF } x_{i_1}^k \text{ is } F_1^l \text{ and } x_{i_2}^k \text{ is } F_2^l \cdots \text{and } x_{i_m}^k \text{ is } F_m^l \text{ THEN } y_{FIS}^l=x_i^k$$

for $l=1,2,\ldots,M$, where M is the total number of rules (in present case $M=5^m$). Here F_j^l are the membership functions. The plot of membership functions of the variable x_{i_3} (where $i=6$) obtained through fuzzy tool box of Matlab is shown in Fig. 5. After training FIS we apply it through simulation on a WSN scenario. So at an instant t_k the output of a fuzzy controller is $y_{FIS}=y_i^k$ as shown in Fig. 6. Then we compare this output value with the actual sensed measurement at node A_i and if

$$|y_i^k-x_i^k|\geq TOLERENCE \tag{13}$$

holds true then the sensor of the node A_i is identified as faulty. Now we talk about the members of $N(A_i)$ that can participate in finding health status of the node A_i.

A node $A_{i_j}\in N(A_i)$ shall participate in the fault identification of the node A_i if the condition (18) is satisfied, in which the node A_{i_j} shall tally its own status with that of the elements of D_{i,i_j}. So there is a possibility that one or more elements of $N(A_i)$ shall not be

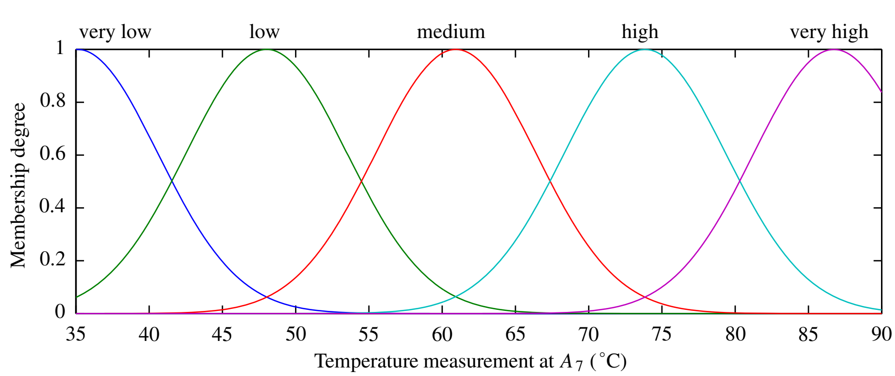

Fig. 5. Plot of membership functions for the variable x_{i_3}, where $i=6$.

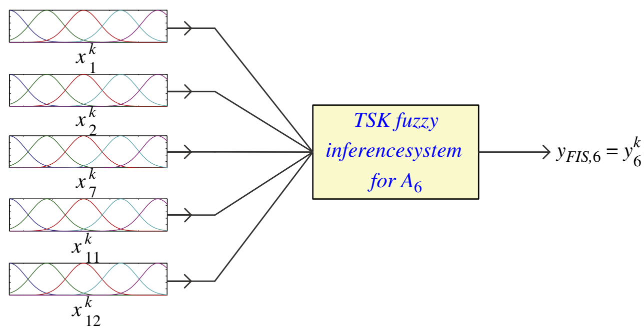

Fig. 6. Approximated value for A_6 by the fuzzy controller.

involved in A_i's fault identification. If $|N(A_i)| = m$ and l of these nodes are not participating then there are

$$\binom{m}{l} \equiv \frac{m!}{l!(m-l)!}$$

combinations for the participating neighboring nodes with l varying from 1 to $m-1$. The total number of possible combinations is

$$\sum_{l=1}^{m-1} \binom{m}{l} = 2^m - 2$$

where each combination corresponds to an FIS. Now we describe the condition (18). For the node A_{i_j}, we have

$$D_{i,i_j} = N(A_{i_j}) - (N(A_i) \cup \{A_i\}) \qquad (14)$$

Let $|D_{i,i_j}| = r$ and these nodes be denoted by $u_1, u_2, \ldots, u_r$. The sensor measurement of the node A_{i_j} is compared with the sensor measurements of the nodes $u_1, u_2, \ldots, u_r$. To tackle the transient faults we shall have this comparison for multiple times $(t_1, t_2, \ldots, t_k)$. Let us denote $x_{i_j}^q$ by $T(A_{i_j}, t_q)$ where $q = 1,2,\ldots,k$. So on the same pattern we shall have sensor measurements of these r nodes as $T(u_\gamma, t_q)$ for $\gamma = 1,2,\ldots,r$ and $q = 1,2,\ldots,k$. Let us define a function

$$g(x_{i_j}^q, T(u_\gamma, t_q)) = \begin{cases} 1 & \text{if } A_{i_j} \text{ and } u_\gamma \text{ satisfy condition (6)} \\ 0 & \text{otherwise} \end{cases} \qquad (15)$$

So the results from function (15) are stored in an $r \times k$ matrix $H = [h_{\gamma,q}]$ where

$$h_{\gamma,q} = g(x_{i_j}^q, T(u_\gamma, t_q)) \qquad (16)$$

A label C_{i_j,u_γ} is attached to A_{i_j} with

$$C_{i_j,u_\gamma} = \begin{cases} 1 & \text{if } \sum_{q=1}^{k} h_{\gamma,q} \geq (k-\mu) \\ 0 & \text{otherwise} \end{cases} \qquad (17)$$

where μ depends upon the number of instances the data is gathered. Now, if

$$\sum_{\gamma=1}^{r} C_{i_j,u_\gamma} \geq \lambda \qquad (18)$$

where λ is selected as a threshold for this condition, on whose fulfilment the node A_{i_j} participates in the fuzzy fault identification of the node A_i.

7. Simulation results

We have simulated a sensor network with 15 sensor nodes as shown in Fig. 7 and one sensor per node. Each node has at least three 1-hop neighbors. The quantity being measured is the temperature. The temperature of all nodes is gathered for a period of 80 h equally divided into 100 instances. For the simulation purpose the temperature T at a point (x,y) and at time t is given by

$$T(x,y,t) = \sqrt{x^2 + y^2} + L\cos(\phi + 2\pi f t) + \sin(\tfrac{5}{2}t) + 60 \qquad (19)$$

where $L=25$, $f=0.025$ and $\phi = \pi$. The reason for choosing this particular heuristic function is that with this expression the temperature varies from 34.15 °C to 88.88 °C. The temperature changes smoothly and there are no sudden jumps or discontinuities. The differences in the data output are small enough to guarantee and justify the theoretical approach described in Section 3. Each sensor is modeled using an FIS as described in previous sections. An FIS has inputs consisting of the sensor measurements of the neighboring nodes. Each input variable to FIS has five membership functions of type Gaussian. An FIS is generated by using the grid partition and is trained by using hybrid method. We have used Matlab as a simulation software. Here we consider and discuss the status of the node A_6 with

$$N(A_6) = \{A_1, A_2, A_7, A_{11}, A_{12}\} \qquad (20)$$

The initial FIS is trained with input of sensor measurements of all the five neighboring (in order) nodes. The kth sample input vector to FIS has the components

$x_1^k \; x_2^k \; x_7^k \; x_{11}^k \; x_{12}^k$

where k is varied from 1 to 100, that is, the FIS is trained with the temperature values of neighborhood nodes for the entire period of 80 h. Similarly the neural network is also trained from these data spanned over 80 h.

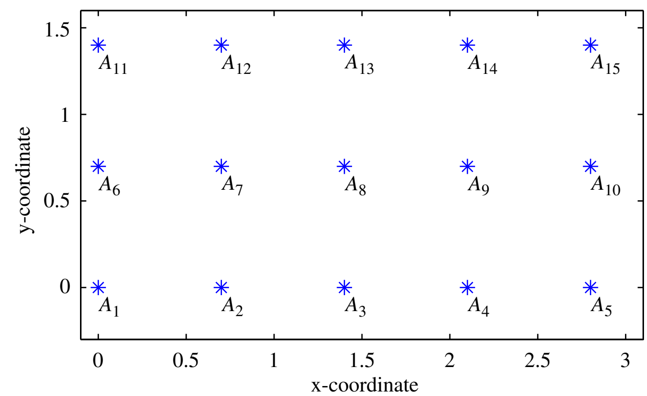

Fig. 7. A WSN scenario with 15 nodes.

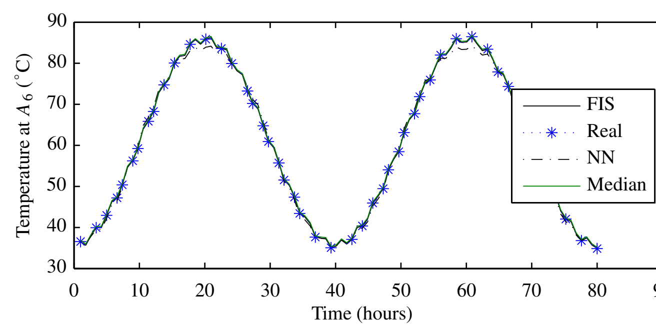

Fig. 8. Real sensor measurement of node A_6 and its models using TSK FIS, NN, and median.

Fig. 8 shows a comparison of the actual measurement of node A_6 with the FIS model, NN model, and the median of the real sensor measurements from $N(A_6)$. The advantage of FIS model over the median method is that it always takes into account the individual measurement from each of the neighbor nodes. An individual erroneous sensor measurement extends its error to the combined input when we take the median. The estimation for the sensed measurement of A_6 by FIS outperforms the approximated values both from NN and median models. Since the temperature data in Fig. 8 is condensed and the approximated values from different models are not clearly distinguishable so a portion has been zoomed in and is shown in Fig. 9.

The absolute value of the difference between the approximations by different models and the real measurement is shown in Fig. 10. Since the FIS model closely approximates the real value and the difference between the two is very small therefore, we are using a logarithmic scale on the temperature measurement axis.

In order to detect a fault in the sensor of node A_6 we introduced an increasing deviation, as a function of time, in its temperature measurement

$$\epsilon(t) = \left[\sin\frac{t}{4} + \frac{t-10}{5}\right] H(t-10) \tag{21}$$

where t is in hours and $H : \mathbb{R} \rightarrow \{0,1\}$ is the unit step function

$$H(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases}$$

Then we plotted the gradually deviating real measurement of node A_6 and the approximated measurements by the FIS for the entire period of 80 h. The results are shown in Fig. 11. The temperature measurement for the first 10 h behaves normally

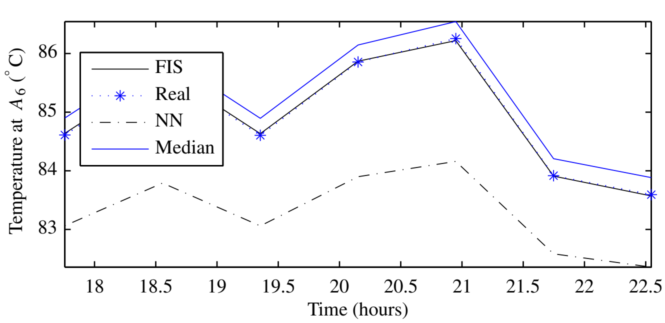

Fig. 9. A magnified portion from Fig. 8.

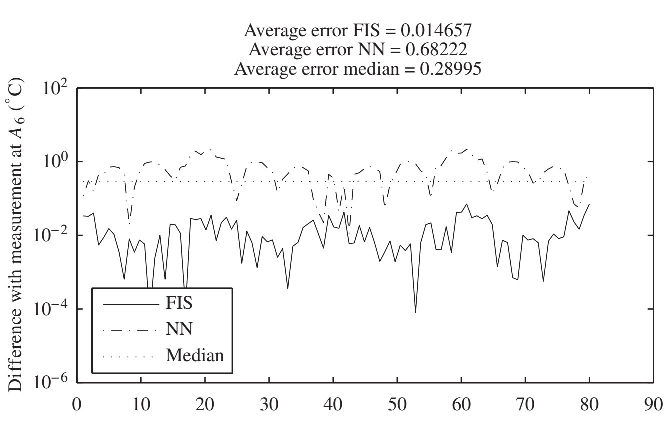

Fig. 10. Absolute difference of FLS, NN, and median model value with real sensor measurement.

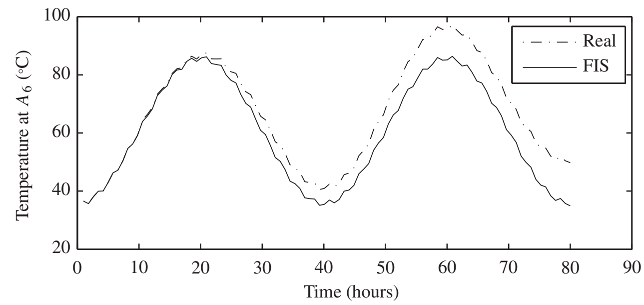

Fig. 11. Sensor measurement and FIS values for the entire period of 80 h.

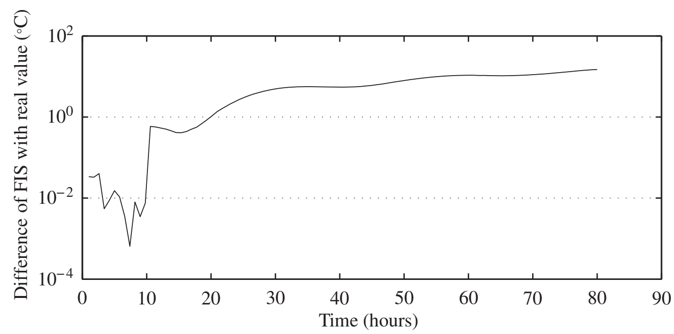

Fig. 12. Difference between the actual and FIS estimated values.

but after that there arise a gradually increasing difference between the real value at A_6 and the value estimated by the FIS. The absolute value of the difference between the two measurements is shown in Fig. 12. Once again the difference between the two measurements for the first 18 h is so small that it is better to scale the temperature measurement axis logarithmically. From $t=20$ onwards the real measurement starts differing from the FIS estimated value by more than 1 °C. Also from Fig. 12 one can decide when to identify the node as faulty depending upon the tolerance allowed by the application.

7.1. *Transient fault tolerance*

Now we discuss the fault tolerance of the proposed approach. By fault tolerance we mean an intermittent perturbation in the sensor measurement of a node that shall be ignored by our scheme. The results for the estimated value for node A_6 are discussed here to elucidate the fault tolerance aspect in the presented method. On its turn every member of $N(A_6)$, as mentioned in (20), is made to show an irregular behavior. The transient error and hence the disturbed sensor reading, $\tilde{x}_j^k$, of neighboring nodes at an instance t_k is as follows:

$$\tilde{x}_j^k = x_j^k + EB \sin\left(\frac{t_k}{4}\right) \tag{22}$$

for $j=1,2,7,11,12$, where EB is the bound on the introduced perturbation. The number of neighboring nodes with transient fault is varied from 1 to m, for the present example $m=5$. Then these perturbed values are used as an input to FIS and obtained output value is compared with the real observed value of the sensor measurement, x_6^k in this case. The results for different values for EB are shown in Tables 1 and 2, and in Figs. 13 and 14. From Table 1 we can infer that even if 50% of the neighbors are manifesting a disturbed behavior than usual, the difference between the real sensed measurement and the FIS estimated value is acceptably small.

As shown in Fig. 15 the measurement of node A_1 is perturbed and the rest of the $N(A_6)$ sensor measurements show the usual behavior. Still the difference between the sensor measurement of node A_6 and its estimated value is very less as is shown in Fig. 16 which is a magnified portion of Fig. 15.

Table 1
Transient fault with absolute value less than 1.

Nodes with transient faults	Min. diff. (°C)	Max. diff. (°C)	Average diff. (°C)
0	6.8143×10^{-5}	0.070821	0.014657
1	0.00054166	0.323710	0.125320
2	0.00472610	0.578390	0.249150
3	0.00956560	0.774180	0.374370
4	0.00914440	0.911180	0.501320
5	0.00872320	1.013200	0.630710

Table 2
Transient fault with absolute value less than 2.

Nodes with transient faults	Min. diff. (°C)	Max. diff. (°C)	Average diff. (°C)
1	0.0040496	0.90014	0.24740
2	0.0091450	1.53390	0.48816
3	0.0083028	1.91960	0.73420
4	0.0074602	2.06490	0.98838
5	0.0066175	2.01350	1.25680

7.2. Recurrent FIS treatment

We have also conducted our approach with recurrent fuzzy inference system (RFIS). Also we have done a comparison with recurrent neural network (RNN) and median of the neighboring node sensor measurements. The RFIS is demonstrated in Fig. 2. The RFIS is trained with input $\mathbf{x}^k = (x^k_{i_1}, x^k_{i_2}, \ldots, x^k_{i_m}, y^{k-1}_i)^T$ and output $y^k_i = x^k_i$. We use three membership functions for each neighboring sensed value x_{i_j} for $j = 1,2,\ldots,m$. So the fuzzy rules for node A_i are given by

$$R^l: \quad \text{IF } x^k_{i_1} \text{ is } F^l_1 \cdots \text{and } x^k_{i_m} \text{ is } F^l_m \text{ and } y^{k-1}_{FIS} \text{ is } F^l_{m+1}$$
$$\text{THEN } y^l_{FIS} = x^k_i$$

for $l = 1,2,\ldots,M$, where M is the total number of rules (in present case $M = 3^{m+1}$). The plot of membership functions of the variable x_{i_3} (where $i = 6$) obtained through fuzzy tool box of Matlab is shown in Fig. 17. Since, for the sake of example we have chosen node A_6, therefore, the k input to RFIS sample vector has the components

$x^k_1 \; x^k_2 \; x^k_7 \; x^k_{11} \; x^k_{12} \; y^{k-1}_6$

Fig. 18 shows a comparison of the real measurement of node A_6 with the RFIS model, RNN model, and the median of the real sensor measurements from $N(A_6)$. Since the temperature data in Fig. 18 is condensed and the approximated values from different models are not clearly distinguishable so a portion has been zoomed-in and is shown in Fig. 19. The absolute value of the difference between approximations by different models and the real measurement is shown in Fig. 20. Fig. 21 shows the RFIS approximated values and the real sensor measurement of node A_6

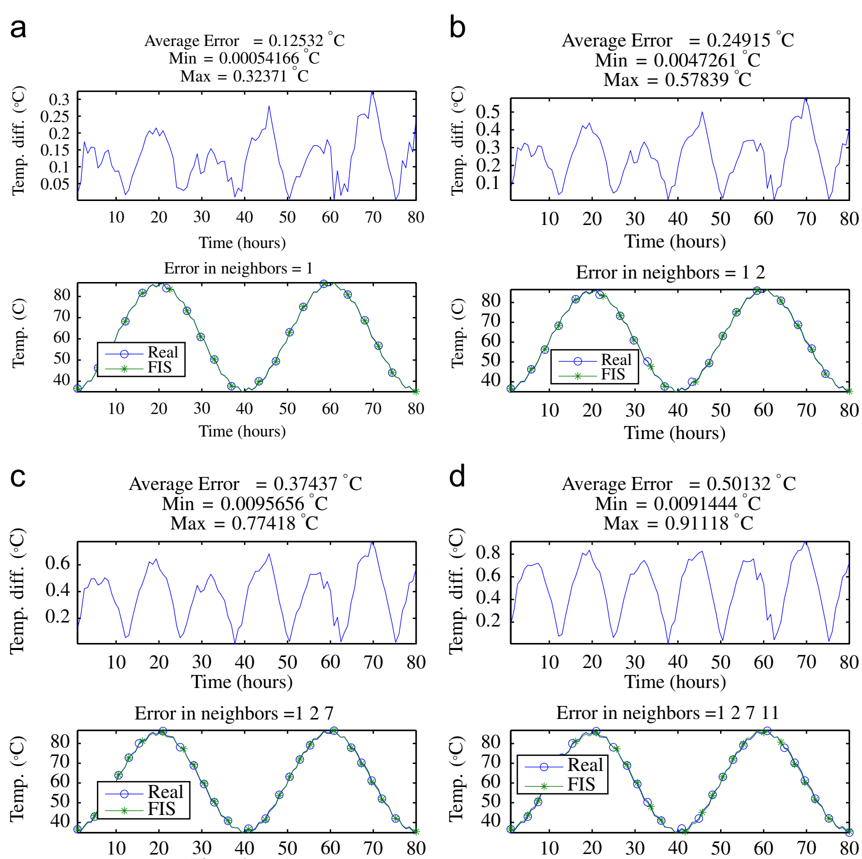

Fig. 13. Transient faults in neighboring nodes with absolute value less than 1.

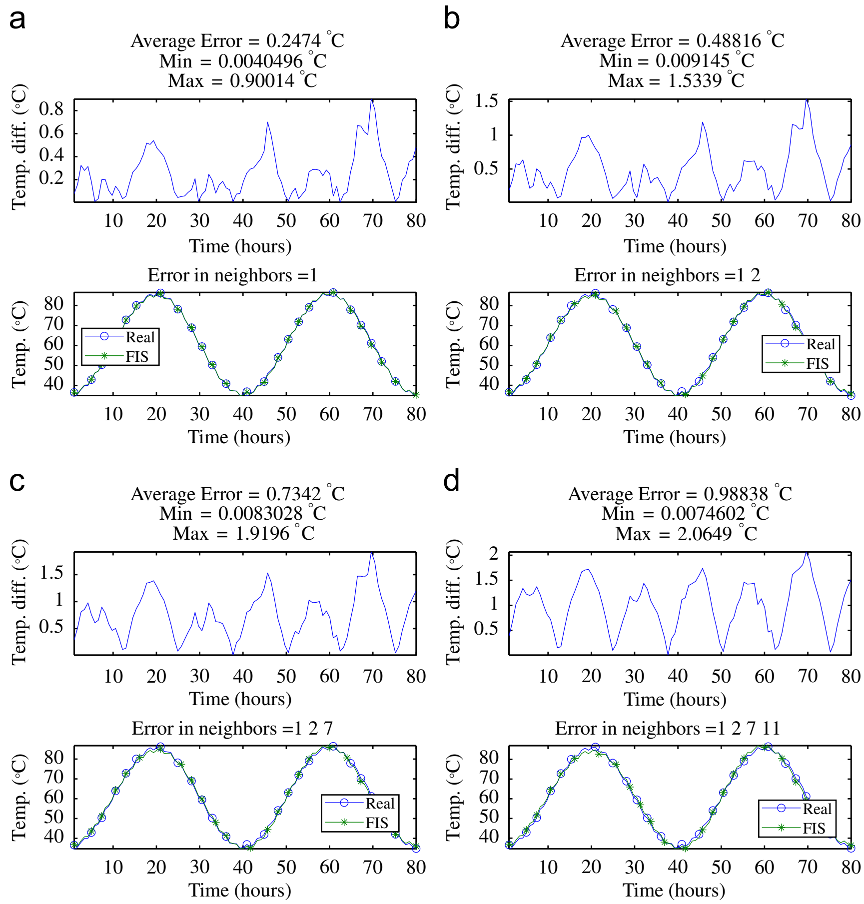

Fig. 14. Transient faults in neighboring nodes with absolute value less than 2.

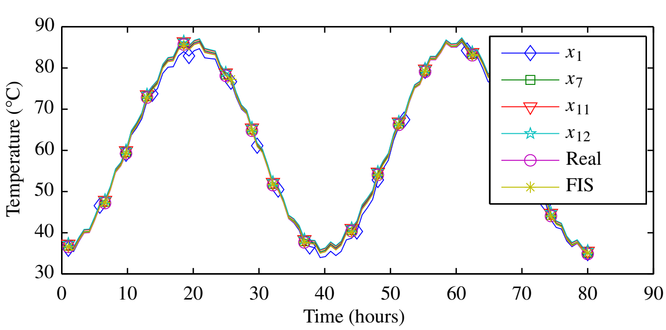

Fig. 15. Sensor measurements of neighbors and estimated value of the node itself.

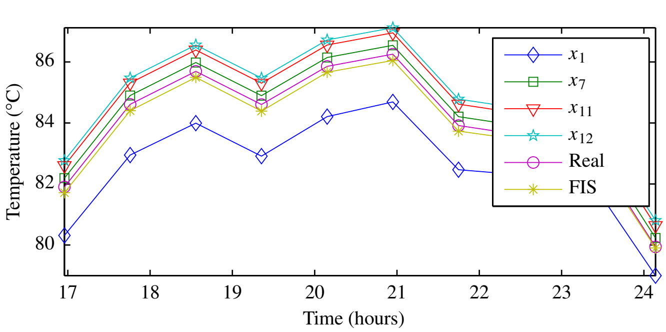

Fig. 16. A zoomed in portion of Fig. 15.

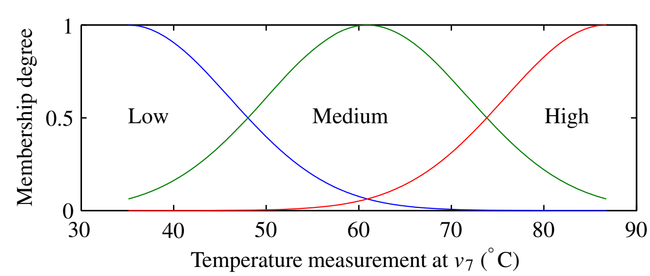

Fig. 17. Plot of membership functions for the variable x_{i_3}, where $i=6$.

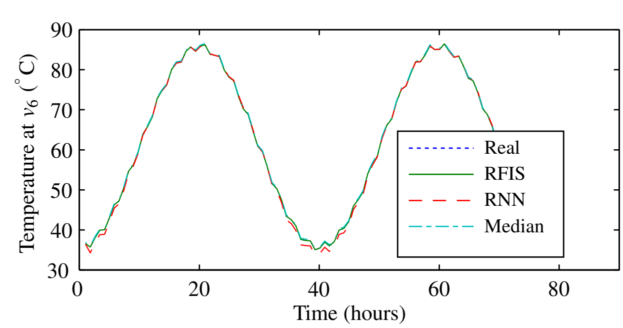

Fig. 18. Real sensor measurement of node v_6 and its models using recurrent TSK FIS, RNN, and median method.

with increasing deviation introduced. For the recurrent technique, the results for different values for *EB*, in (22), are shown in Tables 3 and 4. From the tables we can see that RFIS is performing better than FIS. Once again, like earlier, from Fig. 21 we can decide when to declare the node as faulty depending on the desired difference between the real and RFIS approximated value.

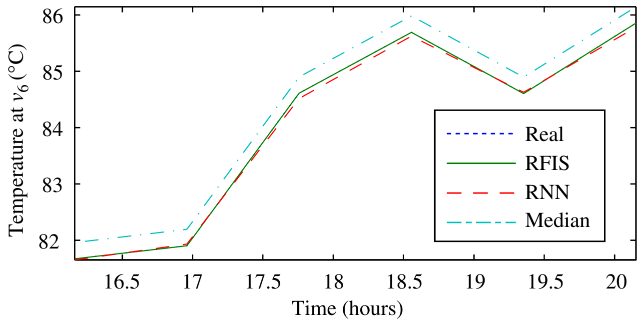

Fig. 19. A portion magnified from Fig. 18.

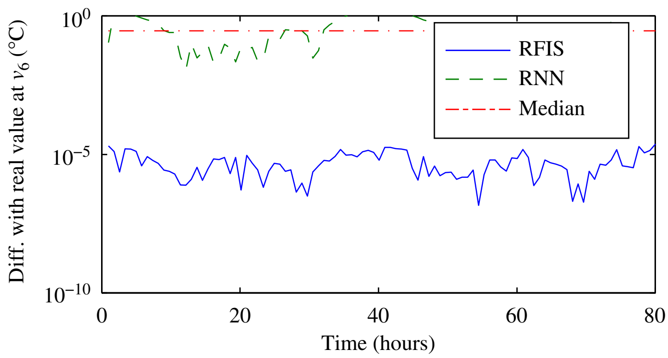

Fig. 20. Absolute difference of RFIS, RNN, and average model value with real sensor measurement.

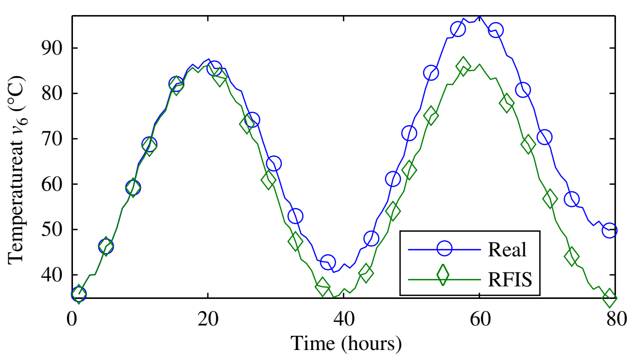

Fig. 21. Sensor measurement and the RFIS values for the entire period of 80 h.

Table 3
Transient fault with absolute value less than 1.

Nodes with transient faults	Min. diff. (°C)	Max. diff. (°C)	Average diff. (°C)
0	3.996×10^{-7}	3.473×10^{-4}	6.522×10^{-5}
1	0.00049349	0.25241218	0.13505085
2	0.00090606	0.47156798	0.26456387
3	0.00130369	0.66836709	0.39195162
4	0.00168033	0.83957573	0.51623019
5	0.00204862	0.99968624	0.63941108

8. Conclusion

In this paper we have described a sensor fault identification scheme for a wireless sensor network. Each node of the sensor network is modeled by a fuzzy inference system which approximates the measurement of that node as a function of the real measurements of the neighboring nodes. In an online environment the difference between the real value detected at a node and the estimated value given by its corresponding FIS model is used to decide whether or not to declare the node as faulty. Since the scheme is distributed and that the computations are performed at the base station the suggested method is less energy consuming. Simulation results show the efficiency of proposed scheme and that the fuzzy inference model outperforms the results given by artificial neural network and median of the one-hop neighbor measurements.

Table 4
Transient fault with absolute value less than 2.

Nodes with transient faults	Min. diff. (°C)	Max. diff. (°C)	Average diff. (°C)
1	0.00094208	0.56351787	0.26891727
2	0.00176691	1.02979571	0.52720860
3	0.00256196	1.42193454	0.78187899
4	0.00331539	1.73497709	1.03106361
5	0.00405246	1.99931670	1.27880158

References

[1] M. Ding, D. Chen, K. Xing, X. Cheng, Localized fault-tolerant event boundary detection in sensor networks, in: Proceedings of 24th IEEE INFOCOM Conference, vol. 2, 2005, pp. 902–913.
[2] I. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cyirci, Wireless sensor networks: a survey, Comput. Networks 38 (4) (2002) 393–422.
[3] Crossbow, Technical Report, available online at 〈http://www.xbow.com〉.
[4] X. Di, B.K. Gosh, T. Tarn, Sensor-based hybrid position/force control of a robot manipulator in an uncalibrated environment, IEEE Trans. Control Syst. Technol. 8 (4) (2000) 635–645.
[5] S. Katsura, Y. Matsumoto, K. Ohnishi, Analysis and experimental validation of force bandwidth for force control, in: IEEE International Conference on Control Systems and Technology, 2003, pp. 796–8001.
[6] S. Lysheyski, Smart flight control surfaces with microelectromechanical systems, IEEE Trans. Aerosp. Electron. Syst. 38 (2) (2002) 543–552.
[7] A.I. Moustapha, R.R. Selmic, Wireless sensor network modeling using modified recurrent neural networks: application to fault detection, IEEE Trans. Instrum. Meas. 57 (5) (2008) 981–988.
[8] A. Zhirabok, O. Preobragenskaya, Instrument fault detection in nonlinear dynamic systems, in: IEEE Conference on Systems, Man, and Cybernetics, 1993, pp. 114–119.
[9] A. Pouliezos, G. Stavrakankis, Real Time Fault Monitoring of Industrial Processes, Kluwer Academic Publishers, Norwell, MA, 1994.
[10] M. Leushen, J. Cavallaro, I. Walker, Robotic fault detection using analytical redundancy, in: IEEE Conference on Robotics and Automation, 2002, pp. 456–463.
[11] S.C. Lee, Sensor value validation based on systematic exploration of the sensor redundancy for fault diagnosis, IEEE Trans. Syst. Man Cybern. 24 (4) (1994) 594–605.
[12] C. Jaikaeo, C. Srisathapornphat, C. Shen, Diagnosis of sensor networks, in: International Conference on Communications, 2001, pp. 1627–1632.
[13] F. Koushanfar, M. Potkonjak, A. Sangiovanni-Vincentelli, On-line fault detection of sensor measurements, IEEE Sensors 2 (2003) 974–980.
[14] S. Chessa, P. Santi, Comparison-based system-level fault diagnosis in ad hoc networks, in: 20th IEEE Symposium on Reliable Distributed Systems, 2001, pp. 257–266.
[15] L. Ruiz, I. Siqueira, L. Oliveira, H. Wong, J. Nogueira, A. Liureiro, Fault management in event-driven wireless sensor networks, in: MSWIM'04, 2004.
[16] B. Krishnamachari, S. Iyengar, Distributed Bayesian algorithms for fault-tolerant event region detection in wireless sensor networks, IEEE Trans. Comput. 53 (3) (2004).
[17] X. Luo, M. Dong, Y. Huang, On distributed fault-tolerant detection in wireless sensor networks, IEEE Trans. Comput. 55 (1) (2006) 58–70.
[18] J. Chen, S. Kher, A. Somani, Distributed fault detection of wireless sensor networks, in: Proceedings of DIWANS Workshop, 2006, pp. 65–72.
[19] M. Lee, Y. Choi, Fault detection of wireless sensor networks, Comput. Commun. 31 (14) (2008) 3469–3475.
[20] M. Lee, Y. Choi, Localized detection of faults in wireless sensor networks, in: ICACT, 2008, pp. 637–641.
[21] D. Wang, P. Shi, W. Wang, Robust fault detection for continuous-time switched delay systems: an linear matrix inequality approach, IET Control Theory Appl. 4 (1) (2010) 100–108.
[22] D. Wang, W. Wang, P. Shi, Robust fault detection for switched linear systems with state delays, IEEE Trans. Syst. Man Cybern., Part B: Cybern. 39 (3) (2009) 800–805.
[23] M. Barakat, F. Druaux, D. Lefebvre, M. Khalil, O. Mustapha, Self adaptive growing neural network classifier for faults detection and diagnosis, Neurocomputing 74 (18) (2011) 3865–3876.

[24] R. Razavi-Far, H. Davilu, V. Palade, C. Lucas, Model-based fault detection and isolation of a steam generator using neuro-fuzzy networks, Neurocomputing 72 (13–15) (2009) 2939–2951.
[25] M. Yu, H. Mokhtar, M. Merabti, Fault management in wireless sensor networks, IEEE Wireless Commun. (2007) 13–19.
[26] T. Takagi, M. Sugeno, Fuzzy identification of systems and its applications to modeling and control, IEEE Trans. Syst. Man Cybern. 15 (1) (1985) 116–132.
[27] M. Sugeno, G.T. Kang, Fuzzy modelling and control of multilayer incinerator, Fuzzy Sets Syst. 18 (3) (1986) 329–345.
[28] B. Daachi, T. Madani, A. Benallegue, Adaptive neural controller for redundant robot manipulators and collision avoidance with mobile obstacles, Neurocomputing 79 (2012) 50–60.
[29] A. Benallegue, D. Meddah, B. Daachi, Neural network identification and control of a class of non linear systems, in: Proceedings of the 16th IMACS World Congress, Lausanne, Switzerland, 2000.

Safdar Abbas Khan completed his PhD from the Laboratory of Images, Signals, and Intelligent Systems (LISSI) of University of Paris East, France. His PhD specialization concerns wireless sensor networks. Currently he is an assistant professor at the Institute of Information Technology, Quaid-i-Azam University Islamabad, Pakistan. He is also a research fellow at the LISSI lab. He research interest include localization techniques in wireless sensor networks, soft computing techniques, and optimization theory.

Boubaker Daachi is Associate Professor of robotics and computer science since 2003. He received his PhD in Robotics from the University of Versailles in 2000, his MS in robotics from the University of Paris 6 in 1998 and his BS in computer science from the University of Sétif (Algeria) in 1995. His research interests include adaptive control of complex systems, soft computing and ambient intelligence. Application fields are robotics, pervasive and distributed systems. He has published more than 40 papers in scientific journals, books and conference proceedings. He has been involved in the organizing committees of some national and international events.

Karim Djouani is a full professor at University of Paris East Créteil and received his PhD degree from the same institute in 1994. He is also a full professor at French South African Institute of Technology at Tshwane University of Technology, Pretoria, South Africa. His current works focus on the development of novel and highly efficient algorithms for reasoning systems with uncertainty as well as optimization, for distributed systems, networked control systems, wireless ad hoc network, wireless and mobile communication, and wireless sensor networks as well as Robotics. He has authored/co-authored over 100 papers in archival journals and conference proceedings as well as five chapters in edited books.