

# Neural Network Based Approach for Malicious Node Detection in Wireless Sensor Networks

DANIEL-IOAN CURIAC, CONSTANTIN VOLOSENCU

Automatics and Applied Informatics Department

“Politehnica” University of Timisoara

Bd.V. Parvan Nr.2, 300223 Timisoara

ROMANIA

<http://www.aut.upt.ro/~curiac>

ALEX DOBOLI

Electrical and Computer Engineering Department

State University of New York

Stony Brook, NY 11794-2350

UNITED STATES OF AMERICA

<http://www.ee.sunysb.edu/~adoboli/>

OCTAVIAN DRANGA, TOMASZ BEDNARZ

School of Engineering

James Cook University

Townsville, QLD 4811

AUSTRALIA

<http://www.eng.jcu.edu.au/Staff/Profiles/octavian-dranga/>

*Abstract:* - Even if their resources in terms of energy, memory, computational power and bandwidth are strictly limited, sensor networks have proved their huge viability in the real world, being just a matter of time until this kind of networks will be standardized and used broadly in the field. One of the important problems that are related to the use of wireless sensor networks in harsh environments is the gap in their security. This paper provides a solution to discover malicious nodes in wireless sensor networks using an on-line neural network predictor based on past and present values obtained from neighboring nodes. This solution can be also a way to discover the malfunctioning nodes that were not a subject of an attack. Being localized on the base station level, our algorithm is suitable even for large-scale sensor networks.

*Keywords:* - wireless sensor network, prediction, neural network, analytical redundancy

## 1 Introduction

Wireless sensor networks are extremely distributed systems comprising a large number of independent and interconnected sensor nodes, with limited computational and communicative potential. They are deployed for data acquisition purposes on a wide range of locations, sometimes in resource-limited and hostile environments such as disaster areas, seismic zones, ecological contamination sites, military combat zones etc. Their deployment in such harsh environments can be dangerously disturbed by any kind of sensor malfunction or, more damaging, by malicious attacks from an adversary.

The design of wireless sensor networks faces two very important problems nowadays: restrictive power consumption and security. The nodes being

deployed in a certain environment without any previous network setup should be able to communicate secure and be able to avoid intruders. For this reason cryptographic algorithms should be helpful, but unfortunately not enough for the moment. The sensor network designer should decide on more security implying high power consumption and less security with low power consumption. For example, in wireless sensor networks used for detection (fire, flood, object movement, etc.) the information validity should be no more than few seconds, enough to be propagated to the sink (the sink of a wireless sensor network is a device that reports gathered information to monitoring groups or fire an alarm).

Although researchers have proposed more strategies, none can be used as a standard and none can guaranty very good security. In this research area, the development of new architectures/security systems that achieve protection by encrypting the transmissions is crucial, as well as protection against physical attacks on sensors (sensor capturing) or against attacks performed due to insecure routing protocols.

Sensor networks due to their restrictive constraints are vulnerable to some relevant types of attacks that cannot be avoided only by cryptography: eavesdropping, traffic analysis, spoofing, selective forwarding, sinkhole attack, wormhole attack, Sybil attack and Hello flood attack are the most important [1].

Probably the biggest threat for a wireless sensor network is node-capturing attack [2] where an adversary gains full control over sensor nodes through direct physical access. This type of attack is fundamentally different from the attacks already mentioned because it doesn't rely on security holes in protocols, broadcasting, operating systems, etc. It is based on the geographic deployment of the sensor nodes in the field. Realistically, we cannot expect to control access to hundreds of nodes spread over several kilometers and, by this, we make a node capturing attack very possible. In addition, sensors are rarely tamper resistant, so an attacker can damage or replace sensors and computation hardware or extract sensitive material such as cryptographic keys to gain unrestricted access to higher levels of communication. Moreover, all sensors are usually assumed to run the same software, in particular, the same operating system. Finding an appropriate bug in the sensor network, through reverse engineering techniques applied to the captured sensor, allows the adversary to control the entire sensor network.

Our proposed countermeasure relies on the fact that a corrupted sensor node, even if it may still send authentic messages (e.g., it can use the cryptographic keys already stored in its memory), it may not work according to its original specifications sending erroneous readings to the base station. We will identify these sensors by using a neural network based predictor and will eliminate their malicious effect.

## 2 Malicious Node Detection using Neural Networks

Our goal is to design a node capture resilient scheme that eliminates the effect of corrupted data provided

by malicious sensors. For this, detecting anomalies and intruders in sensor networks is very important. After detection, the sensor network can take decisions to investigate, find and remove malicious nodes if possible.

### 2.1 Sensor Network Model

We make the following assumptions related to the sensor network:

a) The sensor network is static, i.e., sensor nodes are not mobile; each sensor node knows its own location [3] even if they were deployed via aerial scattering or by physical installation. If not, the nodes can obtain their own location through the location process described in [4]. Moreover, all the sensors passed a one-time authentication procedure done just after their deployment in the field.

b) The sensor nodes are similar in their computational and communication capabilities and power resources to the current generation sensor nodes, e.g. the Berkeley MICA motes. We assume that every node has space for storing up to hundreds of bytes of keying materials in order to secure the transfer of information through symmetric cryptography.

c) The base station, sometimes called access point, acting as a controller and as a key server, is assumed to be a laptop class device and supplied with long-lasting power. We also assume that the base station will not be compromised.

d) We rely on wireless cellular network (WCN) architecture (Fig. 1). In this architecture, a number of base stations are already deployed within the field. Each base station forms a cell around itself that covers part of the area. Mobile wireless nodes and other appliances can communicate wirelessly, as long as they are within the area covered by one cell [5].

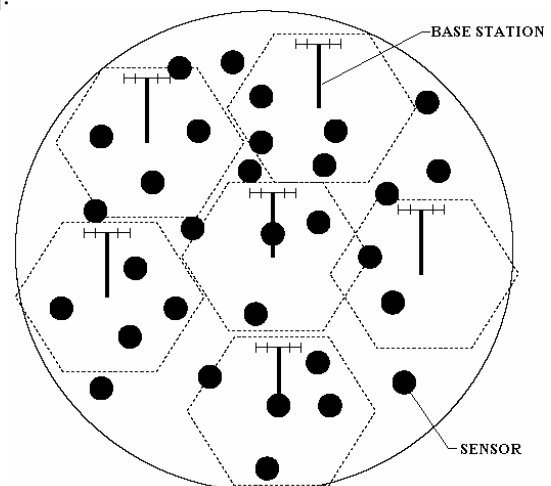


Fig.1: Cellular Network Architecture

Also, it is possible to extend our methodology to a SENMA (Sensor Network with Mobile Access) architecture that was proposed in [6] for large-scale sensor networks. The main difference related to the cellular network architecture is that base stations are considered to be mobile, so each cell has varying boundaries which implies that mobile wireless nodes and other appliances can communicate wirelessly, as long as they are at least within the area covered by the range of the mobile access point.

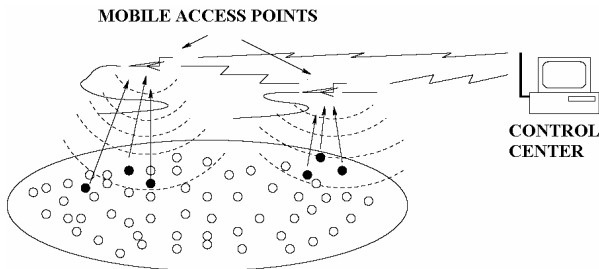


Fig. 2: SENMA Architecture

f) We are also assuming that multiple sensors nodes can detect an event situated in the surrounding area, so we rely on the redundancy of sensor networks.

The two types of architectures presented bellow (WCN and SENMA) have important properties that will be considered for developing a secure sensor network: nodes talk directly to base stations; no node-to-node communications; no multi-hop data transfer; sensor synchronism is not necessary; sensor do not listen, only transmit and only when polled for; complicated protocols avoided; reliability of individual sensors much less critical; system reconfiguration for mobile nodes not necessary.

## 2.2 Employing Redundancy in Sensor Networks

One important natural feature of sensor networks that will be employed by our strategy is inherent redundancy. New approaches for ensuring security and power savings in sensor networks are based on this characteristic. It is known that redundancy in sensor networks can provide higher monitoring quality [7][8] by employing the adjacent nodes to discern the rightness of local data. These highly localized results can be aggregated [9] to provide higher data reliability to requesting applications such as event/target detection [9][10]. Here, we will take this approach one step further: we will use redundancy as a feature that can bring a higher level of information security to sensor networks.

There are two possible approaches: using hardware redundancy and using analytical redundancy. Hardware redundancy implies the use

of supplementary sensors (in normal circumstances they are already deployed in the field due to the necessity of covering the area in case of malfunctioning of some sensor nodes) and selection of data that appears similarly on the majority of sensors. Analytical redundancy is done through a process of comparison between the actual sensor value and the expected/estimated sensor value. This approach is based on a mathematical model that can predict the value of one sensor by taking into consideration the past and present values of neighboring sensors. The computational cost of this approach can become prohibitive as the number of sensors and model complexity is increased, but it can be done in our methodology at base station level (laptop class device) where all requirements are satisfied. Furthermore, our approach is suitable even when hardware redundancy conditions are not met, for example when, due to malfunctions, some sensors had to be ejected from the network.

## 2.3 Proposed Strategy

Our strategy uses the analytical redundancy and relies on a knowledge-based system (KBS) placed in base stations (Fig.3). The principle is the following: a malicious sensor node that will try to enter false information into the sensor network will be identified by comparing its output value  $x$  with the value  $\hat{x}$  predicted using past/present values provided by contiguous sensors. Taken into consideration a specific node denoted by  $A$  (Fig.3), this process is done in the following steps:

- a) Associate a trust factor  $b$  with every sensor node. Initially all this factors have the same value. The specified sensor node  $A$  will have a trust factor denoted by  $b_A$ .
- b) Estimate the future value  $\hat{x}_A(t)$  provided by sensor node  $A$ , using the past/present values of adjacent sensors and the trust factor of each sensor; For sensor  $A$ , we can write the following equation:

$$\hat{x}_A(t) = f(X_{A,adj}(t-1), \dots, X_{A,adj}(t-n), B_A) \quad (1)$$

where

$$X_{A,adj}(t-i) = (x_{A,adj1}(t-i), \dots, x_{A,adjm}(t-i))^T \quad (2)$$

is a vector that contains the values provided by all  $m$  adjacent sensors of sensor  $A$  at instant  $(t-i)$ ,

$$B_A = (b_{A,adj1}, \dots, b_{A,adjm})^T \quad (3)$$

is a vector that contains the trust factor of each of the  $m$  adjacent nodes of  $A$ , and  $n$  is the estimator's order. In our approach, an on-line neural network predictor performs this step.

c) Compare the present value  $x_A(t)$  of the sensor node with its estimated value  $\hat{x}_A(t)$  by computing the error  $e_A(t) = x_A(t) - \hat{x}_A(t)$  (4);

d) Increase/decrease the trust factor  $b_A$  by using a function  $g$  that can be either linear or non-linear:  $b_A(t) = b_A(t-1) - g(e_A(t))$  (5)

The structure of such a KBS is depicted in Fig. 3 and contains two important blocks:

- Neural Network Prediction Block: this block provides the estimate  $\hat{x}_A$  following equation (1) and is able to memorize the past values provided by adjacent sensors and the related trust factors.
- Decision Block: here, based on a priori information (statistics, attack's model), the trust factor  $b_A$  is modified using (5), and, in particular circumstances, alarm signals can be transmitted to a higher hierarchical level.

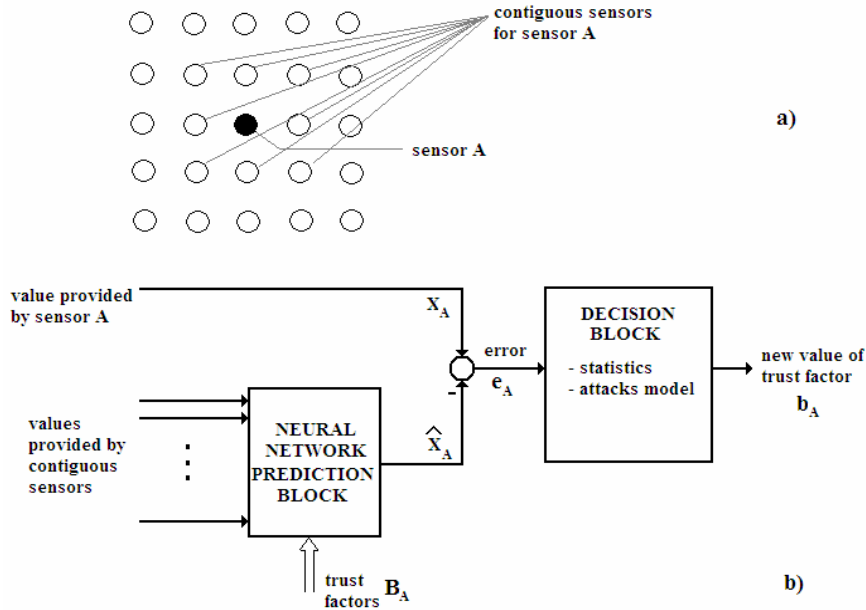


Fig. 3: Knowledge Based System Structure

### 3 Case Study

In this paragraph we will present how our strategy works in the case of a temperature sensor network. The following presumptions are made according to the considerations mentioned above. Let us consider the propagation of a temperature wave in a homogenous planar field where several temperature sensors, part of a sensor network, have been deployed. For malicious sensors detection we developed and trained a neural network that estimates the value provided by the sensor by taking into consideration the present values of neighboring sensors.

We presume the time  $t$  distribution of the temperature  $\theta$  through the homogenous medium in space to be:

$$\theta = \theta(z, t) \quad (6)$$

where  $\theta(z, t)$  is the temperature at the moment  $t$ , at distance  $z$  from the heat source.

The heat conduction, when neglecting the heat losses in the environment, is described by the heat equation [11]:

$$c_\theta \frac{\partial^2}{\partial z^2} \theta(z, t) = \frac{\partial}{\partial t} \theta(z, t) \quad (7)$$

where  $c_\theta$  is the heat conductivity coefficient of the medium.

The input in the system is the power of the heat source  $P$ , which at a certain point of the field is:

$$P(t) = \alpha \frac{\partial}{\partial z} \theta(z, t) \Big|_{z=0} \quad (8)$$

where  $\alpha$  is a constant depending on the heat transfer from the source to the medium.

The above description requires the function  $\theta(z, t)$  in order to determine the temperature variation in every point of the space. The function is the medium state at time  $t$ , so we have to measure and store many temperature values, one for each value of  $x$ , to know the state. The medium is an infinite dimensional system and it is described by partial differential equations.

To get a more approximate model that is more manageable for practical purposes we can use discretization of the medium. Let us make a nine-order model for the heat distribution in a two dimensions  $xOy$  plane, presented in Fig. 4.

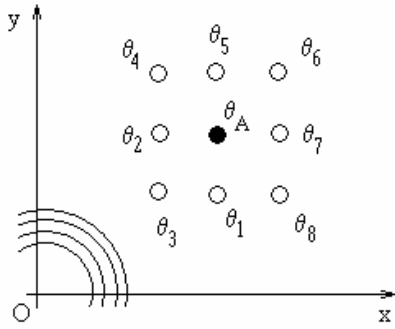


Fig. 4. Sensor Distribution

We make a discretization of the homogenous medium into 9 parts, where the temperature values at time  $t$  are denoted by  $\theta_i, i=A, 1, 8$ , respectively. This means that we work with 9 state variables  $\theta_i$  for each point  $i$  at a distance  $z_i$  from the origin of the heat source. The detected sensor's value is  $\theta_A$ .

For this model we may write the conservation of energy relationship for each point:

$$\frac{d}{dt}W_i = P_{in,i} - P_{out,j} \quad (9)$$

where  $W_i$  is the energy stored in point  $i$ ,  $P_{in,i}$  is the input power to point  $i$  and  $P_{out,j}$  is the output power of point  $j$ .

Let the heat capacity of each point be denoted by  $C$  and the heat transfer coefficient between the points by  $K_{i,j}$ . We can write the following equation:

$$\begin{aligned} \frac{d}{dt}C\theta_i(t) = & K_{k,i}[\theta_k(t) - \theta_i(t)] + K_{j,i}[\theta_j(t) - \theta_i(t)] + \\ & + K_{l,i}[\theta_l(t) - \theta_i(t)] - K_{i,m}[\theta_i(t) - \theta_m(t)] - \\ & - K_{i,n}[\theta_i(t) - \theta_n(t)] - K_{i,p}[\theta_i(t) - \theta_p(t)], \end{aligned} \quad (10)$$

$i, j, k, m, n, p = A, 1, \dots, 8$

The heat source of power  $P$  is positioned in the origin of the coordinate systems. We can apply a coordinate transformation and move the source to point  $\theta_3$ . In this case, for unity coefficients, we may consider the following state space equations for the model:

$$\begin{bmatrix} \dot{\theta}_A \\ \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \\ \dot{\theta}_4 \\ \dot{\theta}_5 \\ \dot{\theta}_6 \\ \dot{\theta}_7 \\ \dot{\theta}_8 \end{bmatrix} = \begin{bmatrix} -5,41 & 1 & 0,7 & 1 & 0 & 1 & 0,7 & 1 & 0 \\ 1 & -3,7 & 1 & 0 & 0 & 0 & 0 & 0 & 0,7 \\ 0,7 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & -2,7 & -3,7 & 1 & 0,7 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & -2 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0,7 & 1 & -3,7 & 1 & 0 & 0 \\ 0,7 & 0 & 0 & 0 & 0 & 1 & -2 & 1 & 0 \\ 1 & 0,7 & 0 & 0 & 0 & 0 & 1 & -5,4 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & -2 \end{bmatrix} \begin{bmatrix} \theta_A \\ \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \\ \theta_5 \\ \theta_6 \\ \theta_7 \\ \theta_8 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} P \quad (11)$$

The unit step response ( $P=1$ ) of the above system is presented in Fig. 5.

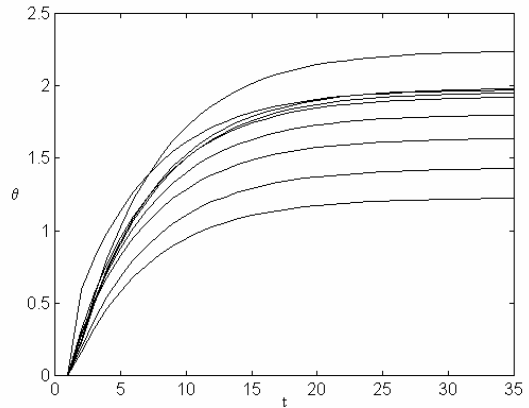


Fig. 5. Time propagation of temperature for all nine sensors

Using the above presumptions we can develop a feedforward neural network with continuous values

to obtain an estimate  $\hat{\theta}_A$  of the state  $\theta_A$  based on the adjacent measured states  $\theta_i, i=1, \dots, 8$ . The neural network used for this purpose has the structure presented in Fig. 6.

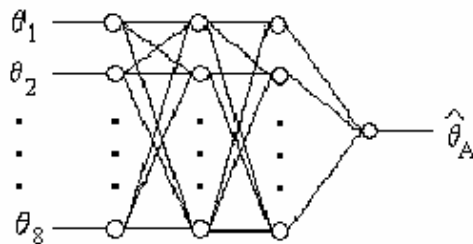


Fig. 6. Neural Network Structure

According to Kolmogorov's theorem [12] we are using 2 hidden layers of neurons with biases to obtain a reduced error of approximation of the estimate. The input layer has 8 neurons each for the measured temperatures. The output layer has one neuron for the estimated temperature. The first and the second hidden layers have 64 and 36 neurons, respectively. These numbers resulted after an iterative training of the neural network using Levenberg-Marquardt method with a training set  $(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6, \theta_7, \theta_8; \theta_A)$ , obtained from the step response of the above system.

The activation functions of the neural network are represented by a hyperbolic tangent function for the hidden layers and a first-order linear function for the output layer. The sum square error after 9 training epochs is presented in Fig. 7.

The neural network was tested with other different test set. The output error was less than 0.1%.

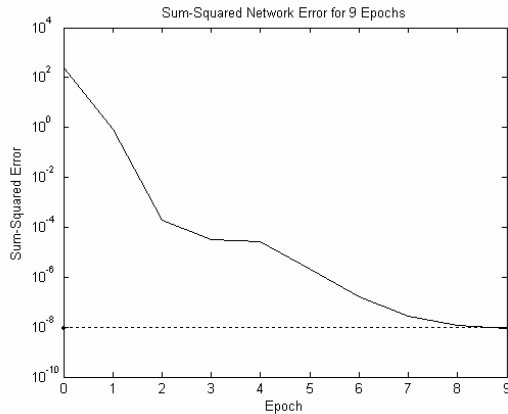


Fig. 7. Training sum squared error

We assumed that sensor node  $A$  was attacked at  $t=18$  and as a result its output value  $\theta_{Ac}$  has the shape depicted in Fig.8. On the other hand, the sensor's estimated output value  $\hat{\theta}_A$ , predicted by the above-mentioned neural network differs from the actual value of the malicious sensor  $A$  showing that something wrong happened to sensor  $A$  (Fig. 8). In these circumstances, the decision block will decrease the corresponding trust factor  $b_A$ .

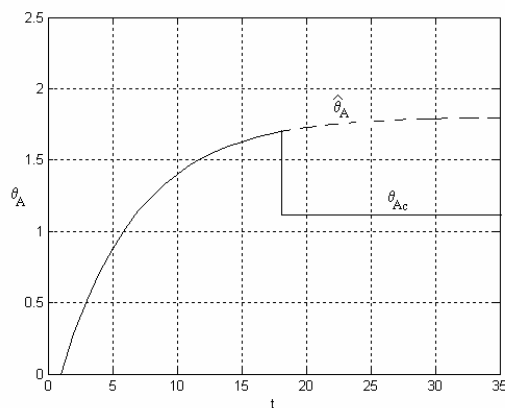


Fig. 8. Estimated  $\hat{\theta}_A$  and corrupted  $\theta_{Ac}$  values for sensor  $A$

### 4 Conclusion

In this paper, we proposed a strategy for detecting malicious sensor nodes and eliminating their effects by using a neural network based predictor. This strategy employs the analytical redundancy to estimate the value provided by a sensor by considering the past/present values given by adjacent sensors. This estimate is compared with the actual value of the sensor to increase/decrease its trust factor.

### References:

- [1] Karlof C., Wagner D., "Secure routing in wireless sensor networks: attacks and countermeasures", *Proceedings of the 1st IEEE International Workshop SNPA2003*, Anchorage, USA, May 2003, pp. 113-127.
- [2] Becher A., Benenson Z., Dornseif M., "Tampering with notes: Real-world physical attacks on wireless sensor networks" *Proceedings of the 3rd International Conference on Security in Pervasive Computing (SPC)*, York, UK, April 2006, pp.104-118.
- [3] He T., Huang C., Blum B.M., Stankovic J.A., Abdelzaher T., "Range-Free Localization Schemes in Large-Scale Sensor Networks", *Proceedings of the Intl. Conference on Mobile Computing and Networking (MOBICOM)*, San Diego, USA, September 2003, pp.81-95.
- [4] Savvides A., Han C.-C., Srivastava M.B., "Dynamic fine-grained localization in ad-hoc networks of sensors", *Proceedings of the 7th ACM MobiCom*, Rome, Italy, 2001, pp.166-179.
- [5] Feng J., Koushanfar F., Potkonjak M., "System-Architectures for Sensor Networks Issues, Alternatives, and Directions", *Proceedings of the 2002 IEEE International Conference on Computer Design (ICCD'02)*, Freiburg, Germany, September 2002, pp.226-231.
- [6] Tong L., Zhao Q., Adireddy S., "Sensor Networks with Mobile Agents", *Proceedings IEEE 2003 MILCOM*, Boston, USA, October 2003, pp.688-694.
- [7] Gao Y., Wu K., Li F., "Analysis on the Redundancy of Wireless Sensor Networks", *ACM WSNA Proceedings*, San Diego, USA, September 2003, pp.108-114.
- [8] Sun T., Chen L.J., Han C.C., Gerla M., "Improving Data Reliability via Exploiting Redundancy in Sensor Networks", *UCLA Technical Report CSD-TR No. 040037*.
- [9] Clouqueur T., Ramanathan P., Saluja K.K., Wang K.C., "Value fusion versus decision-fusion for fault tolerance in collaborative target detection in sensor networks", *Proceedings of Fusion 2001*, Montreal, August 2001, pp.25-30.
- [10] Li D., Wong K., Hu Y., Sayeed A., "Detection, Classification and Tracking of Targets in Distributed Sensor Networks", *IEEE Signal Processing Magazine*, March 2002, pp.17-19.
- [11] Ljung L., Glad T., "Modeling of Dynamical Systems", *Prentice Hall*, Englewood Cliffs, USA, 1994.
- [12] Hertz J., Krogh A., Palmer R.G., "Introduction to the Theory of Neural Computation", *Addison-Wesley Publishing*, Redwood Cliffs, USA, 1991.