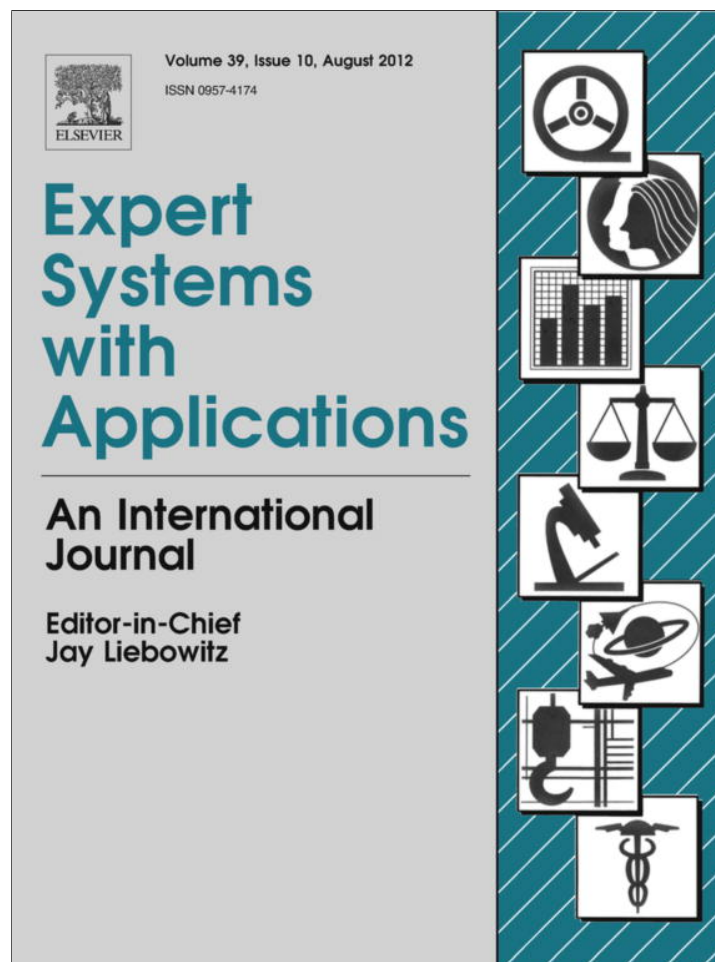


Provided for non-commercial research and education use.  
Not for reproduction, distribution or commercial use.



This article appeared in a journal published by Elsevier. The attached copy is furnished to the author for internal non-commercial research and education use, including for instruction at the authors institution and sharing with colleagues.

Other uses, including reproduction and distribution, or selling or licensing copies, or posting to personal, institutional or third party websites are prohibited.

In most cases authors are permitted to post their version of the article (e.g. in Word or Tex form) to their personal website or institutional repository. Authors requiring further information regarding Elsevier's archiving and manuscript policies are encouraged to visit:

<http://www.elsevier.com/copyright>



Contents lists available at SciVerse ScienceDirect

## Expert Systems with Applications

journal homepage: [www.elsevier.com/locate/eswa](http://www.elsevier.com/locate/eswa)

## Ensemble based sensing anomaly detection in wireless sensor networks

Daniel-Ioan Curiac\*, Constantin Volosencu

Automation and Applied Informatics Department, "Politehnica" University of Timisoara, Bd. V. Parvan nr. 2, 300223 Timisoara, Romania

## ARTICLE INFO

## Keywords:

Wireless sensor networks  
Binary classifier  
Ensemble based systems  
Sensors anomalies  
Data accuracy

## ABSTRACT

Wireless sensor networks are often used to monitor and measure physical characteristics from remote and sometimes hostile environments. In these circumstances the sensing data accuracy is a crucial attribute for the way these applications complete their objectives, requiring efficient solutions to discover sensor anomalies. Such solutions are hard to be found mainly because the intricate defining of the correct sensor behavior in a complex and dynamic environment. This paper tackles the sensing anomaly detection from a new perspective by modeling the correct operation of sensors not by one, but by five different dynamical models, acting synergically to provide a reliable solution. Our methodology relies on an ensemble based system composed of a set of diverse binary classifiers, adequately selected, to implement a complex decisional system on network base station. Moreover, every time a sensing anomaly is discovered, our ensemble offers a reliable estimation to replace the erroneous measurement provided by sensor.

© 2012 Elsevier Ltd. All rights reserved.

## 1. Introduction

A wireless sensor network (WSN) is a collection of tiny, inexpensive and low-power devices that can be deployed throughout a geographical space for fine-grained monitoring and event detection. Besides its computing and communication potential, a WSN is, first of all, an advanced distributed measurement system which is often prone to sensing anomalies that can cause erroneous data, compromising the objectives of the entire network.

There are three major sources of sensor anomalies within WSN: (i) software or/and hardware failures – breakdowns in any subsystem of a sensor node or even battery discharging can produce wrong sensing data; (ii) security attacks – when a malicious entity compels the sensors to report erroneous measurements or to drop measurement data packages (Walters, Liang, Shi, & Chaudhary, 2006); (iii) environment related sources – when sensor nodes cannot measure the physical value correctly due to unfavorable phenomena that can arise in the harsh environments in which they are often deployed.

Generally speaking, sensing anomaly detection refers to the problem of discovering patterns in measurement data that do not match with expected behavior (Chandola, Banerjee, & Kumar, 2009; Rajasegarar, Leckie, & Palansiwami, 2008; An, Heo, & Chang, 2011). This is not a simple task mainly because an estimated model of “correct behavior” is always hard to find. In the case of a WSN there is an inherent feature on which we can rely – sensing redun-

dancy (Curiac, Volosencu, Pescaru, Jurca, & Dobioli, 2009), which takes two basic forms: physical redundancy that implies the use of more than one sensor node for measuring the same localized value; and analytical redundancy that implies a mathematical model for evaluating the value provided by one sensor by taking into consideration the past and present values of neighboring sensors (spatial redundancy), the past values of the sensor itself (temporal redundancy) or both (spatiotemporal redundancy).

In the last decade a series of relevant approaches based on assortments employing different types of analytical redundancies and intelligent detection algorithms have been proposed for solving the issue of sensing anomaly discovery.

In (Siripanadorn, Hattagam, & Teaumroong, 2010) a mixture between a competitive learning method called the self-organizing map (SOM) and the discrete wavelet transform (DWT) is used to detect anomalies from synthetic and real-world datasets.

A two-step temporal modeling procedure, developed to analyze and extract semantic symbols from a sequence of observations, is presented in Li, Thomason, and Parker (2010) where an intelligent system detects time-related changes online by using a likelihood-ratio detection scheme. The algorithm is distributed, and supports a hierarchical learning structure that can scale to large number of sensors.

The use of Bayesian networks as means for unsupervised learning and anomaly detection in gas monitoring sensor networks for underground coal mines is described in Wang, Lizier, Obst, Prokopenko, and Wang (2008). The authors showed that the Bayesian network model can learn cyclical baselines for gas concentrations, thus reducing false alarms usually caused by flatline thresholds. Their solution was proved to be efficient in both distributed and centralized approach.

\* Corresponding author. Tel.: +40 256 403 227; fax: +40 256 403 214.

E-mail addresses: [daniel.curiac@aut.upt.ro](mailto:daniel.curiac@aut.upt.ro) (D.-I. Curiac), [constantin.volosencu@aut.upt.ro](mailto:constantin.volosencu@aut.upt.ro) (C. Volosencu).

An efficient method applying principal component analysis (PCA) simultaneously on multiple metrics received from various sensors is depicted in Chatzigiannakis and Papavassiliou (2007). One of the key features of this approach is that it provides an integrated methodology of taking into consideration and combining effectively correlated sensor data, in a distributed fashion. Furthermore, it allows the integration of results from neighboring network areas to detect correlated anomalies that involve multiple groups of nodes.

Our paper tackles the sensing anomaly discovery from a new perspective: the modeling of the correct behavior for sensors is done not by one, but by five different models, acting synergically to provide a reliable solution. For this, we developed an ensemble based system (EBS) containing five different binary classifiers, each categorizing every network node as being accurate or erroneous, the final decision being taken by the entire ensemble using a voting procedure. It is broadly accepted that the overall efficiency of such committees of classifiers can occur only if there is diversity among its components (Polikar, 2006). In our proposal, the heterogeneity of classifiers is achieved by using various and carefully selected classifier architectures and different sets of input data. In order to completely solve the problem, our ensemble not only discovers sensing errors, but offers reliable estimations to replace the measurements affected by these anomalies.

The remainder of the paper is organized as follows. In Section 2, we introduce the philosophy of our ensemble based sensing anomaly detection. Section 3 describes the architecture of each individual classifier, pointing out the way in which the correct sensing behavior is modeled and predicted, while Section 4 depicts the decisional block of the ensemble based on weighted voting algorithm. In Section 5, we present the methodology used for training and testing of the ensemble. Section 6 covers a test case illustrating the entire methodology and, finally, conclusions are offered in Section 7.

## 2. Ensemble based sensing anomaly detection

Discovering sensing anomalies in the context of WSN is a challenging issue due to the complexity of the environment in which sensor nodes are deployed. Often, this subject is tackled using dedicated decisional systems. For acquiring node behavior related decisions, it makes sense to ask more than one decision making entity, because this practice assures indubitably a better, more informed, and trustable final decision. We label these decisional instances as classifiers or experts and their collections as ensemble based systems (Dietterich, 2000; Ho, Hull, & Srihari, 1994; Polikar, 2006; Wang, Hao, Ma, & Jiang, 2011).

In order to periodically detect and investigate each and every sensor anomaly, an ensemble based system has been designed.

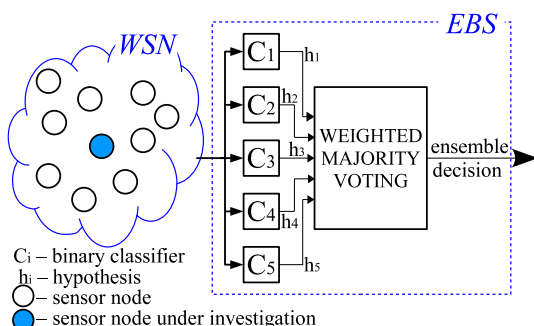


Fig. 1. Ensemble based sensing anomaly detection.

As presented in Fig. 1, this ensemble encloses several binary classifiers that independently categorize the state of each sensor as “reliable” or “unreliable”. All the classifier outputs will be aggregated using a weighted majority algorithm to obtain the concluding ensemble decision. This final ensemble decision will be further used by the base station to take all the required actions for the unreliable nodes (e.g. removal of the untrustworthy nodes from the WSN for a specified period of time).

Our ensemble based methodology reveals the following characteristics:

- The EBS input data are represented by past and present measurements gathered by the node under investigation (node A) and respectively, past and present measurements gathered by each of the node A neighboring nodes.
- There are five binary classifiers involving different prediction techniques and different types of inputs to assure the required diversity of classifiers. Each classifier offers its own hypothesis  $h_i$  (“reliable” or “unreliable”) about the sensing correctness of the node under investigation.
- The overall ensemble decision block is based on a weighted majority algorithm.
- If the investigated node is found as presenting sensor anomalies, the network base station acts in consequence and can exclude that specific sensor from the list of network functioning sensors for a limited period of time. As an example, this can be achieved based on the following rule: if the EBS indicated at least three times that the node A suffers from a sensor anomaly, the base station decides to inactivate the sensor. The base station could later reactivate the sensor after repeating the EBS investigation by testing if new readings became appropriate.
- Using the power of ensemble, a reliable estimation of the measurement affected by anomaly is automatically offered any time when needed.

In the following paragraphs the design of each EBS component, together with details regarding training and testing of the ensemble are offered.

## 3. Designing the classifiers

The design of individual classifiers to fulfill the EBS requirements is not a simple task. This process has to be governed by one magic word: diversity. As a result, any stratagem for generating the ensemble members must be focused on the ensemble’s heterogeneity improvement.

The diversity of classifiers may originate from three basic sources:

- a. classifier structure: the use of diverse classifier algorithms (Hsu & Srivastava, 2009; Tsoumakas, Katakis, & Vlahavas, 2004) can assure the required heterogeneity (diversity through structure);
- b. classifier internal parameters: by using different training datasets (García-Pedrajas & Ortiz-Boyer, 2009; Kim & Kang, 2010; Li & Sun, 2012; Shirai, Kudo, & Nakamura, 2009) or diverse initializations of the training algorithms, the classifiers having exactly the same structure may cover different regions of the classified workspace, assuring the heterogeneity (diversity through parameters); and
- c. classifier inputs: the diversity may be assured by applying different inputs to identical classifiers (diversity through inputs); usually this channel to obtain diversity is irrelevant, but when there is an overabundance of data sources it can be a viable alternative.

There are circumstances in which these three possibilities are combined resulting more complex ensembles, but applying metrics to evaluate diversity in these cases may become very difficult.

Due to the specificity of WSN applications we chose to assure the diversity of classifiers using a mixture between diversity through structure and diversity through inputs. From this perspective we developed five classifiers that combine the two kinds of input data (measurement time series provided by the sensor under investigation and the measurements gathered from neighboring sensors), with linear or complex nonlinear models:

- average based classifier;
- autoregressive linear predictor based classifier;
- neural network based classifier;
- neural network autoregressive predictor based classifier;
- Adaptive Neuro-Fuzzy Inference System (ANFIS) based classifier.

All these classifiers are built based on a preset configuration that includes:

- a prediction block – this block encapsulates a linear or nonlinear model of the “correct behavior” of the sensor and relies on the analytical redundancy feature of the WSNs. Manipulating measurement data often affected by errors gathered from a dynamic and complex environment, the predictor block parameters have to be very carefully chosen. By choosing different types of prediction algorithms, the required diversity can be assured.
- an error computing block that compares the sensor current measurement with its predicted value; Practically, this block compares the actual behavior of the sensor under investigation with a modeled “correct behavior”.
- an internal decision block that takes the individual resolution of the classifier – correct or incorrect behavior, using a carefully chosen threshold as the limit between good and bad behavior.

### 3.1. Average based classifier

The average based classifier (AVC) is computationally the simplest binary classifier used in our ensemble. Based on measurement gathered by the sensor under investigation at a specific moment in time  $\theta_A(t)$  and on the measurements provided by  $k$  adjacent sensors at the same moment in time aggregated in the input vector  $\theta_{AVC}(t)$ , it provides a binary output: “0” meaning normal functioning of sensor A and “1” meaning abnormal functioning.

$$\theta_{AVC}(t) = [\theta_1(t) \ \theta_2(t) \ \dots \ \theta_k(t)]. \quad (1)$$

This classifier exploits the analytical spatial redundancy feature of WSN that allows the estimation of the measurement gathered from sensor A using the measurement values provided by neighboring sensors. In this case, the estimation will be obtained using a simple averaging scheme. As can be seen in Fig. 2, the classifier's internal structure consists of one average computation block, one error computation block and one internal decisional block.

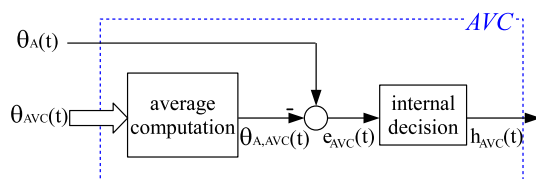


Fig. 2. Average based classifier.

Classifier's average computation block receives all present measurement values of each of the  $k$  neighbors and computes the averaged value as presented in (2):

$$\theta_{A,AVC}(t) = \frac{1}{k} \sum_{i=1}^k \theta_i(t). \quad (2)$$

The average value  $\theta_{A,AVC}(t)$  is subtracted from node's current measurement value, resulting the error  $e_{AVC}(t)$ :

$$e_{AVC}(t) = \theta_A(t) - \theta_{A,AVC}(t). \quad (3)$$

The obtained error value is then evaluated by classifier's decisional block for producing  $h_{AVC}(t)$  hypothesis. This evaluation is performed by comparing the error value with a given threshold  $\varepsilon_{AVC}$ . If the absolute value of the error exceeds  $\varepsilon_{AVC}$ , the classifier outputs a hypothesis equal to “1”, meaning that the measurement provided by the node A was classified as being erroneous:

$$h_{AVC}(t) = \begin{cases} 0 & \text{if } |e_{AVC}(t)| < \varepsilon_{AVC} \\ 1 & \text{if } |e_{AVC}(t)| \geq \varepsilon_{AVC} \end{cases}. \quad (4)$$

In the training process, there is only one AVC parameter that has to be established: the threshold  $\varepsilon_{AVC}$ . This is done based on trials, experience and intuition of the human expert which plays a central role.

### 3.2. Autoregressive linear predictor based classifier

The autoregressive linear predictor based classifier (ALC) exploits the analytical temporal redundancy feature of WSN that allows the estimation of the measurement gathered from a specific sensor A using past measurements of the same sensor to supply a binary output: “0” meaning normal functioning of the sensor and “1” meaning abnormal functioning.

In this case, the estimation will be obtained using an autoregressive predictor based on a numerically robust variant of recursive least square (RLS) method Ljung & Soderstrom, 1987; Yang & Bohme, 1992. As can be seen in Fig. 3, the classifier's internal structure consists of one autoregressive prediction block, one error calculation block and one internal decisional block.

Classifier's autoregressive prediction block receives past measurement values provided by the sensor under investigation:

$$\theta_{A,ALC}(t) = [\theta_A(t-1) \ \dots \ \theta_A(t-n)] \quad (5)$$

and predicts its current value as shown in (6), where  $n$  is the autoregression order, the parameters  $a_1(t)$ ,  $a_2(t)$ ,  $\dots$ ,  $a_n(t)$  are estimated using a robust recursive least square algorithm, and  $\xi(t)$  is considered to be a white-noise perturbation

$$\theta_{A,ALC}(t) = a_1(t) \cdot \theta_A(t-1) + \dots + a_n(t) \cdot \theta_A(t-n) + \xi(t). \quad (6)$$

The predicted value  $\theta_{A,ALC}(t)$  is subtracted from node's current measurement value, obtaining the error  $e_{ALC}(t)$  as shown in (7):

$$e_{ALC}(t) = \theta_A(t) - \theta_{A,ALC}(t). \quad (7)$$

The obtained error value is then evaluated by classifier's internal decisional block for producing  $h_{ALC}(t)$  hypothesis in a similar manner as the one described by (4). If the error value exceeds the value

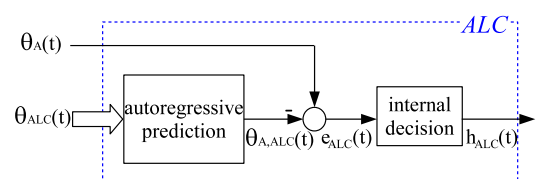


Fig. 3. Autoregressive linear predictor based classifier.



of a given threshold  $\varepsilon_{ALC}$ , the classifier outputs a hypothesis equal to “1”, meaning that the measurement provided by the node A was classified as being erroneous.

The classifier has some internal parameters that have to be thoroughly established: the threshold  $\varepsilon_{ALC}$ , the autoregression order  $n$ , and the initialization parameters of the RLS algorithm. The threshold  $\varepsilon_{ALC}$  is set based on trials, experience and intuition of the human expert, the autoregression order is a positive integer with values among 3 and 6 to make a reasonable balance between precision and computational time, and initialization parameters of RLS are set according to prescriptions when no a priori information are available (Ljung, 2007).

An important remark upon the use of ALC: the classifier can be used as a part of EBS only after some time steps (e.g.  $2n$  time samples) because the prediction offered by RLS needs some time until the parameter convergence (Ljung, 1999).

### 3.3. Neural network based classifier

The neural network based classifier (NNC) processes the current value provided by the sensor under investigation and the current/past values gathered from adjacent sensors to provide a binary output: “0” meaning normal functioning of the sensor and “1” meaning abnormal functioning. This classifier exploits the analytical spatial redundancy feature of WSN that allows the estimation of the measurement gathered from a specific sensor A using only measurements provided by neighboring nodes. In this case, the estimation will be obtained using a neural network predictor. By using a strong nonlinear model (neural network) to approximate the “normal” operation state of a sensor we can model more sophisticated situations than the linear ones (ALC or AVC).

As can be seen in Fig. 4, the classifier's internal structure consists of one feed-forward neural network, one error calculation block and one internal decision block. The neural network, based on its input vector that contains current/past measurements provided by its  $k$  neighbors:

$$\theta_{NNC}(t) = [\theta_1(t) \cdots \theta_1(t-n) \cdots \theta_k(t) \cdots \theta_k(t-n)], \quad (8)$$

predicts the actual value gathered from sensor A –  $\theta_{A,NNC}(t)$ . This prediction is subtracted from the current measurement of the sensor under investigation  $\theta_A(t)$  to obtain the error:

$$e_{NNC}(t) = \theta_A(t) - \theta_{A,NNC}(t), \quad (9)$$

which will be used by the internal decision block to obtain the classifier's hypothesis  $h_{NNC}(t)$ . Classifier's decisional block functions the same as the one included in the average based classifier but uses another threshold:  $\varepsilon_{NNC}$ .

The classifier has some internal parameters that have to be carefully established: the threshold  $\varepsilon_{NNC}$ , the neural network structure (number of layers, number of neurons on each layer, etc.) and the neural network's weights. The threshold  $\varepsilon_{NNC}$  and the structure of the neural network are set based on trials, experience and intuition of the human expert. The neural network training can be done in an automated fashion using a backpropagation algorithm (e.g. Levenberg–Marquardt). To make a reasonable compromise

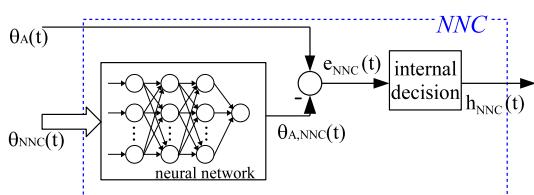


Fig. 4. Neural network based classifier.

between complexity and computing time, we chose to use only measurements provided at last three moments in time:  $t$ ,  $t - 1$  and  $t - 2$ .

This classifier can be used as a component in our EBS only after  $n$  time samples, this period being necessary to populate all components of the input vector  $\theta_{NNC}(t)$  with real values.

### 3.4. Neural network autoregressive predictor based classifier

The neural network autoregressive predictor based classifier (NNAC) processes the values provided only by the sensor under investigation to provide a binary output: “0” meaning normal functioning of the sensor and “1” meaning abnormal functioning. In fact this is an alternative to the autoregressive predictor based classifier that uses a strong nonlinear model (neural network) instead of a linear model. This classifier exploits the analytical temporal redundancy feature of WSN that allows the estimation of the measurement gathered from a specific sensor A using past measurements of the same sensor. In this case, the autoregressive prediction process is done by a neural network predictor.

As can be seen in Fig. 5, the classifier's internal structure is almost identical with the one presented for NNC, with the difference that the input vector includes only past measurements obtained from sensor A:

$$\theta_{NNAC}(t) = [\theta_A(t-1) \theta_A(t-2) \cdots \theta_A(t-n)]. \quad (10)$$

This will impose some changes in the neural network structure (e.g. number of neurons on input layer).

In the case of NNAC, the internal parameters that have to be cautiously established are: the threshold  $\varepsilon_{NNAC}$ , the neural network structure and the neural network's weights. The threshold  $\varepsilon_{NNAC}$  and the neural network structure (number of layers, number of neurons on each hidden layer) are set based on trials, knowledge and intuition of the human expert. The neural network weights can be obtained using an automated procedure based on a backpropagation algorithm. To make a rational compromise between complexity and computing time, we decided to use only measurements gathered at last four moments in time:  $t$ ,  $t - 1$ ,  $t - 2$  and  $t - 3$ .

This classifier can be used as a component in our EBS only after  $n$  time samples, this period being necessary to populate all components of the input vector  $\theta_{NNAC}(t)$  with real values.

### 3.5. ANFIS based classifier

The ANFIS based classifier (ANFISC) processes the past values provided by the sensor under investigation and current/past values provided by adjacent sensors to provide a binary output: “0” meaning normal functioning of the sensor and “1” meaning abnormal functioning. This classifier exploits the analytical spatiotemporal redundancy feature of WSN that allows the estimation of the measurement gathered from a specific sensor A using past measurements of the same sensor and measurements provided by neighbor nodes. By using a strong nonlinear model (ANFIS) to

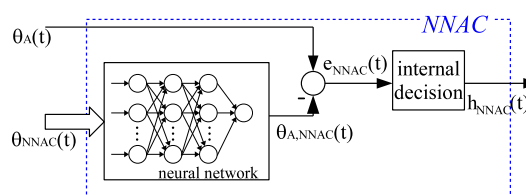


Fig. 5. Neural network autoregressive predictor based classifier.

approximate the “normal” operation state of a sensor we can model very sophisticated situations.

As can be seen in Fig. 6, the classifier’s internal structure consists of one ANFIS prediction block, one error calculation block and one internal decisional block.

Adaptive Neuro-Fuzzy Inference System (ANFIS) was developed by Jang (1991, 1993) and is a hybrid neuro-fuzzy system that uses the Takagi–Sugeno-type fuzzy inference. Basically, ANFIS is a fuzzy inference system, framed in a feed-forward neural network. Hence, the advantages of a fuzzy system can be combined with a learning algorithm. ANFIS architecture consists of five layers of nodes: the first and the fourth layers consist of adaptive nodes implementing fuzzification and defuzzification (represented by squares in the ANFIS structure from Fig. 6), while the second, third and fifth layers consist of fixed nodes that are implementing the fuzzy rule ( $\Pi$ ), the normalization ( $N$ ) and summation ( $\Sigma$ ). The adaptive nodes are associated with their respective parameters, get duly updated with each subsequent iteration while the fixed nodes are devoid of any parameters.

By including two intelligent approaches (neural networks and fuzzy reasoning), the ANFIS block can achieve good reasoning in quality and quantity. In other words we have fuzzy reasoning and network calculation.

The ANFIS predictor, based on its input vector  $\theta_{ANFISC}(t)$  that contains past measurements obtained from sensor A and current/past measurements provided by its  $k$  neighbors:

$$\theta_{ANFISC}(t) = [\theta_A(t-1) \cdots \theta_A(t-n) | \theta_1(t) \cdots \theta_1(t-n) | \cdots | \theta_k(t) \cdots \theta_k(t-n)] \quad (11)$$

predicts the actual value gathered from sensor A –  $\theta_{A,ANFISC}(t)$ . This prediction is subtracted from the current measurement of the sensor under investigation  $\theta_A(t)$  to obtain the error:

$$e_{ANFISC}(t) = \theta_A(t) - \theta_{A,ANFISC}(t) \quad (12)$$

which will be exploited by the internal decision block to achieve the classifier’s hypothesis  $h_{ANFISC}(t)$ . Classifier’s decisional block operates the same as the ones included in above mentioned classifiers (AVC, ALC, NNC and NNAC), but utilizes a different threshold:  $e_{ANFISC}$ .

The classifier has some internal parameters that had to be carefully determined: the threshold  $e_{ANFISC}$  and the ANFIS structure and parameters. The threshold  $e_{ANFISC}$  is set based on trials, knowledge and intuition of the human expert, while the ANFIS network parameters are obtained using a combination of the least-squares method and the backpropagation gradient descent method (MathWorks, 2002).

The training process of this classifier is very computationally intensive. To make a reasonable compromise between model complexity and computing time, we chose to use only two measurements from each sensor:

$$\theta_{ANFISC}(t) = [\theta_A(t-1) \theta_A(t-2) \theta_1(t) \theta_1(t-1) \cdots \theta_k(t) \theta_k(t-1)]. \quad (13)$$

The ANFIS based classifier can be used as a part of our EBS only after  $n$  time samples, this period being necessary to populate all components of the input vector  $\theta_{NNC}(t)$  with real measurements.

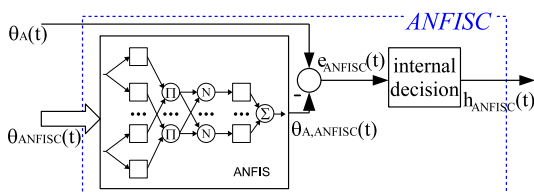


Fig. 6. ANFIS based classifier.

#### 4. Decision block

The ensemble decision block is in charge of two tasks: (a) to provide a confident decision of “reliable” or “unreliable” regarding each and every sensor measurement; and, (b) to offer a trustworthy estimation of the measurement value when a value provided by the sensor is considered to be affected by an anomaly.

Taking the ensemble final decision can be efficiently implemented using the weighted majority voting algorithm (Littlestone & Warmuth, 1994). The five classifiers provide a set of hypothesis –  $h_{AVC}(t), h_{ALC}(t), h_{NNC}(t), h_{NNAC}(t), h_{ANFISC}(t) \in \{0,1\}$  – that are later aggregated using a weighted majority voting scheme formularized by the following equations:

$$WMV(t) = \sum_i w_i \cdot h_i(t) \quad (14)$$

with  $h_i(t) \in \{h_{AVC}(t), h_{ALC}(t), h_{NNC}(t), h_{NNAC}(t), h_{ANFISC}(t)\}$  and  $\sum_i w_i = 1$ .

The overall ensemble decision is computed using the following relation:

$$h_{ensemble} = round(WMV(t)) = \begin{cases} 0 & \text{if } WMV(t) < 0.5 \\ 1 & \text{if } WMV(t) \geq 0.5 \end{cases} \quad (15)$$

where the function  $round(X)$  rounds  $X$  to the nearest integer.

The weights  $w_i$  related to each hypothesis are chosen based on simulations and experiments. In order to provide equal rights for components based on temporal redundancy and spatial redundancy we included the following restriction:

$$W_{ALC} + W_{NNAC} + 0.5 \cdot W_{ANFISC} = W_{AVC} + W_{NNC} + 0.5 \cdot W_{ANFISC} = 0.5. \quad (16)$$

In case a measurement provided by the sensor under investigation is considered abnormal, the ensemble decision block offers a reliable estimation that can replace it. This value is obtained by averaging the estimated values of the measurement provided by individual classifiers:

$$\theta_{A,trust}(t) = \frac{1}{5} \cdot [\theta_{A,AVC}(t) + \theta_{A,ALC}(t) + \theta_{A,NNC}(t) + \theta_{A,NNAC}(t) + \theta_{A,ANFISC}(t)]. \quad (17)$$

In order to prevent incorrect classifying of the ensemble components based on autoregressive forecasts (ALC, NNAC and ANFISC), their input values  $\theta_A(t-i)$ , when affected by anomalies, have to be replaced by these trustworthy estimations  $\theta_{A,trust}(t-i)$ . The reason is simple: wrong data in inputs conduct to misclassification.

#### 5. Training and testing the ensemble

After setting the structure for each individual classifier and for the decision block, a complex training and testing process must be carried out to assure optimal sets of parameters (Fig. 7). This is done using a variety of automated/nonautomated procedures individualized for each ensemble component and relies on training datasets obtained from two sources: computer simulation and experimental deployments of the same type of WSNs.

The training process is undoubtedly one of the most challenging tasks when designing an ensemble system. There are three specificities that must be carefully taken into account in the training process of the proposed ensemble:

- (a) *The impact of the sensor deployment:* Three of the classifiers included in our ensemble use measurements provided by neighboring sensors. This characteristic implies that the sensor deployment (shape, distances between adjacent nodes, etc.) is a key aspect, affecting decisively the training datasets and, by this, the entire training process.

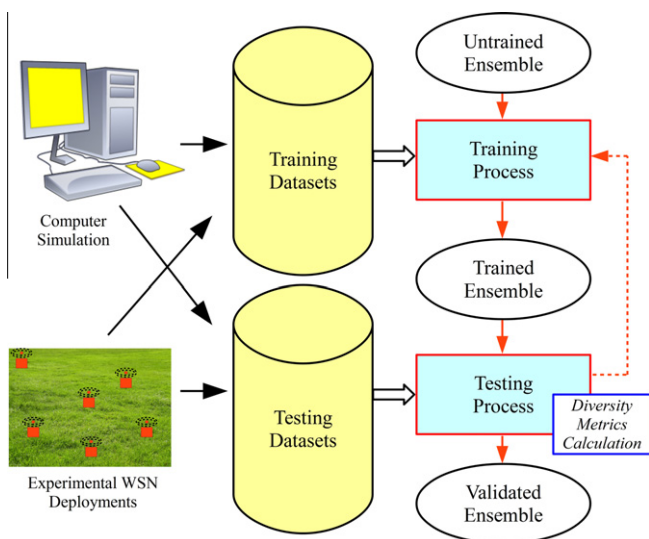


Fig. 7. Training and testing processes of the ensemble.

- (b) *The way training datasets can be obtained:* The environment in which the WSN will be deployed is extremely complex and, as a result, hard to simulate and predict. The training datasets will have to approximate this intricate environment which cannot be done using only computer simulation or only experimental deployments. A mixture between the two possibilities is more feasible.
- (c) *Assuring the classifiers diversity is a must:* The training process must guarantee the diversity of classifiers as mentioned in Section 3. The diversity is estimated using proper metrics that are computed during validation of the training process. If the diversity requirements are not met the training must be redone.

The training and testing methodology applied in the case of our ensemble is governed by the following steps:

- i. *Choosing the sensor deployment shapes that will be considered:* Due to the specificity of WSN deployment procedures (aerial scattering, manual deployment, etc.) and due to possible existence of non-reporting sensors, the geographical placement of the sensors in the field can conduct to a plethora of configurations. From this huge number, we must select a reasonable set that can approximately cover all the situations. For each selected sensor deployment configuration, a set of parameters for ensemble components will be obtained through training.
- ii. *Forming the training and testing datasets for each sensor deployment shape:* In order to obtain consistent training and testing datasets, we must concentrate on two directions: development of computer simulation programs that can model fairly accurate the environment; and, deployment of experimental WSNs to collect relevant data from real environments.
- iii. *Training the ensemble components:* all the parameters of each and every ensemble component will be obtained using automated (Levenberg–Marquardt algorithm in case of neural networks, a combination of the least-squares method with the backpropagation gradient descent method in case of ANFIS) and nonautomated techniques. Obtaining the thresholds for each classifier or the weights for the decision block is based mainly on experience and intuition.

- iv. *Testing the trained ensemble using testing datasets and ensemble evaluation based on proper diversity metrics:* Each trained ensemble is validated using independent testing datasets (data not used in ensemble training) obtained through computer simulations or using experimental sensor deployments in a real environment. In addition, the diversity of the classifiers inside each of the ensembles is computed using proper metrics.
- v. *If the testing results do not match with the desired results or if the diversity of the classifiers is not appropriate, the training process will be redone.*

Internal parameters of the individual classifiers can be obtained using a variety of techniques as follows: the thresholds  $\epsilon_{AVC}$ ,  $\epsilon_{ALC}$ ,  $\epsilon_{NNG}$ ,  $\epsilon_{NNAC}$  and  $\epsilon_{ANFISC}$  are tuned based on trials, experience and intuition, the weights of the two neural networks implied in NNC and NNAC classifiers are computed using Levenberg–Marquardt algorithm and the ANFIS parameters can be calculated using a combination of the least-squares method with the backpropagation gradient descent method. A good choice for implementing the training process is proved to be Matlab/Symulink package that offers powerful functions for simulation, neural networks and ANFIS training, etc.

After training, each ensemble is subject to a testing and validation procedure, where new data are classified and diversity metrics (Q-statistic) are computed. If the results of this validation are not acceptable the training process must be performed once again.

### 5.1. The link between the training process and the sensor deployment in the field

The training process depends decisively on the real sensor deployment. Because the ensemble contains classifiers that rely on the measurements provided by adjacent sensors, the in-field position of the sensors has a major impact in the way the ensemble has to be trained.

In order to standardize the training procedure we chose to train the ensemble for a set of nine different deployment shapes, presented in Fig. 8. This set covers a larger variety of deployments because it envelops the shapes obtained through rotation or/and symmetrization.

Basically, we have to obtain the parameters for each ensemble component corresponding to this set of deployments and to store them in a small Access-type database or in a text file that will be parsed any time when needed, on the base station.

When the deployment of WSN in the field is finished, for each sensor node, a procedure to find the most appropriate shape will be started. This can be done using shape matching procedures similar to the ones presented in Cao, Lisani, Morel, Musé, and Sur (2008), Belongie, Malik, and Puzicha (2002), Huttenlocher, Klanderman, and Rucklidge (1993). When the closest shape of the deployment is obtained, the parameters for the related ensemble are loaded from the database. This way, the ensemble can work even when some sensors are not reporting or even when sensors are deployed nonuniformly (randomly) in the field.

### 5.2. Obtaining the training and testing datasets

Sensor anomalies within WSNs are hard to be discovered mainly because the complexity of environment in which they are deployed. Our ensemble must envelope this complex environment in mathematical models contained in each classifier. These models have to reflect different facets of an intricate reality and, for this, they have to be trained using proper datasets obtained from two independent sources (Fig. 7):

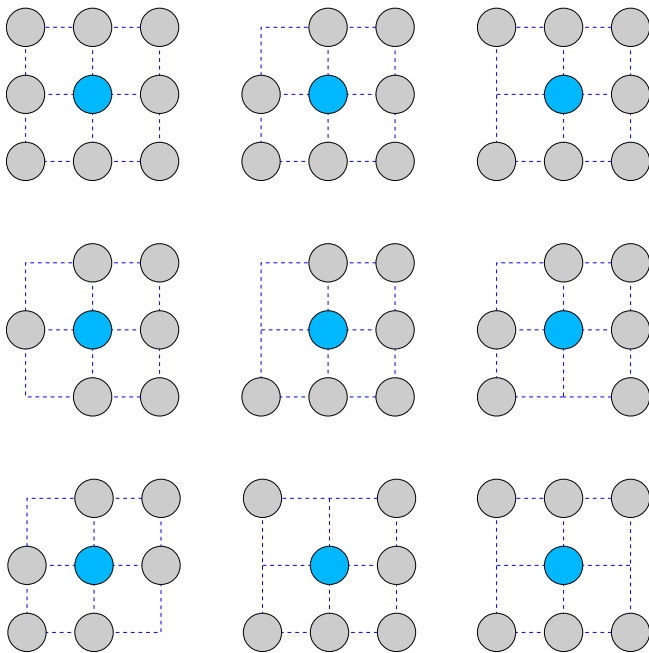


Fig. 8. Training deployment shapes.

- *Computer simulation:* In principle, the environment in which the WSN will be deployed can be simulated to obtain relevant data for training and testing the classifiers, but this action is highly problematical due to the complexity of involved phenomena. For this reason, the computer simulations are error prone and cannot be the only source for training and testing datasets.
- *Experimental deployments:* A relevant source of training and testing datasets is represented by experimental deployments of WSNs in similar environments. Although, the real environment is hard to be replicated, data obtained from experimental deployments are relevant, but sometimes statistically inefficient.

By combining information provided by these two sources, consistent training and testing datasets can be formed.

### 5.3. Metrics to estimate the diversity of classifiers

In order to quantitatively evaluate the diversity of classifiers, several metrics have been defined (Kuncheva & Whitaker, 2003; Polikar, 2006). Among them, the pairwise metrics defined between two classifiers are the simplest. In our case, the ensemble incorporates  $T = 5$  classifiers, so there is a need to compute  $T(T - 1)/2 = 10$  pairwise heterogeneity metrics, and then to obtain the overall diversity of the ensemble by averaging these pairwise metrics.

Given the hypotheses  $h_i$  and  $h_j$  provided by classifiers  $i$  and  $j$ , we can define four probabilities ( $a$ ,  $b$ ,  $c$  and  $d$ ) as presented in Table 1:  $a$  – the probability of the occurrence of a case in which both  $h_i$  and  $h_j$  hypotheses are correct,  $b$  – the probability of the occurrence of a case in which  $h_i$  hypothesis is correct and  $h_j$  hypothesis is incorrect, and so on. Evidently,  $a + b + c + d = 1$ .

Table 1  
Relationship between a pair of classifiers.

	$h_j$ is correct	$h_j$ is incorrect
$h_i$ is correct	a	b
$h_i$ is incorrect	c	d

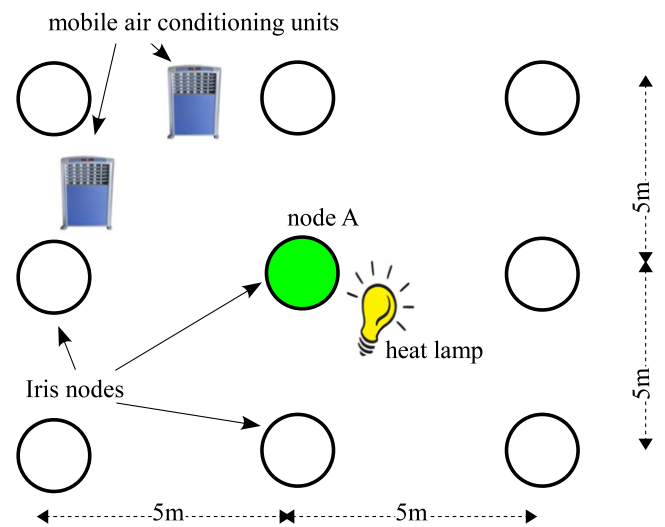


Fig. 9. Experimental indoor WSN deployment.

Based on these probabilities we can compute the  $Q$ -statistic, an indicator that is intuitive and simple to implement (Kuncheva & Whitaker, 2003):

$$Q_{i,j} = \frac{ad - bc}{ad + bc}, \quad (18)$$

$Q$  has positive values if the same instances are correctly classified by both classifiers and negative values, otherwise. Maximum diversity is obtained for  $Q = 0$ .

The overall  $Q$ -statistic indicator of the ensemble will be computed using the following formula:

$$Q_{ensemble} = \frac{1}{n_{pairs}} \sum |Q_{i,j}| = \frac{2}{T(T - 1)} \sum |Q_{i,j}|, \quad (19)$$

where  $T$  represents the number of classifiers included in the ensemble, and  $|Q_{i,j}|$  represents the absolute value of  $Q_{i,j}$ .  $Q_{ensemble} \in [0, 1]$  and has to take values as close as possible to zero to meet the requirement of classifiers diversity. Otherwise, the trained ensemble cannot pass the testing process and must be retrained.

## 6. Implementation and case study

For validating the above concept and related methodologies we performed a series of studies using a WSN designed for measuring the temperature in an indoor environment. Our experimental sensor network is composed of nine Crossbow-Iris nodes equipped with MTS310 sensors boards which report the measured values through a gateway (MIB520CB) to a laptop-class device where our software modules can efficiently operate.

All the programs were developed in Matlab/Simulink mainly because of its strong capabilities within the fields of recursive parameter estimation, neural networks and ANFIS related techniques. Basically we developed two different programs, one related to training and testing process where the parameters of the EBS are tuned for fulfilling the expectations and one for on-line sensing anomalies discovery that implements the trained ensemble of classifiers.

The datasets for training and testing were obtained from two different sources, experimental WSN deployment and computer simulation, to cover a wider variety of environmental heat-transfer related phenomena.

Our indoor WSN deployment has the configuration depicted in Fig. 9, where the temperature can be intentionally perturbed using



**Table 2**  
Configuration of the training and testing datasets.

Package number	Time (min)	Node NW (°C)	Node N (°C)	Node NE (°C)	Node E (°C)	Node SE (°C)	Node S (°C)	Node SW (°C)	Node W (°C)	Node A (°C)
...										
3	86	23.32	23.28	23.12	23.31	23.31	23.07	23.26	23.21	23.18
4	0	18.32	18.28	18.27	18.17	18.30	18.27	18.29	18.27	18.19
4	1	18.74	18.42	18.41	18.35	18.29	18.32	18.39	18.52	18.30
4	2	19.32	19.03	18.70	18.61	18.32	18.48	19.01	19.23	18.69
...										

**Table 3**  
Structure of the training and testing datasets.

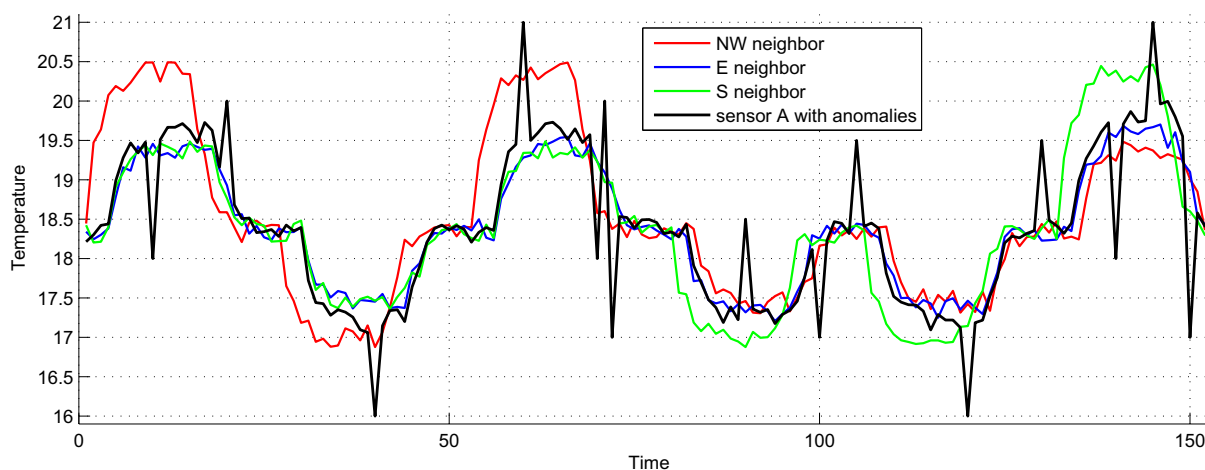
Package number	Destination	Number of time samples	Number of abnormal functioning samples	Source
1	Training	127	15	Real WSN
2	Training	226	32	Real WSN
3	Training	87	12	Real WSN
4	Training	320	63	Computer simulation
5	Training	342	45	Computer simulation
6	Training	287	42	Computer simulation
7	Testing	79	14	Real WSN
8	Testing	162	23	Real WSN
9	Testing	154	15	Computer simulation
10	Testing	303	41	Computer simulation

**Table 4**  
Pairwise Q-statistics between the five classifiers.

Q-statistics	AVC	ALC	NNC	NNAC	ANFISC
AVC	1.00	0.82	0.86	0.84	0.86
ALC	0.82	1.00	0.79	0.91	0.89
NNC	0.86	0.79	1.00	0.82	0.90
NNAC	0.84	0.91	0.82	1.00	0.86
ANFISC	0.86	0.89	0.90	0.86	1.00

either mobile air conditioning units (affect groups of sensors with heating/cooling waves), either a heat lamp (affects a single sensor by increasing the temperature in its immediate vicinity).

Computer simulation is another source of training/testing datasets. We started from the heat equation which describes the distribution of heat (or variation in temperature) in a given region over time. For a function  $u(x,y,t)$  of two spatial variables  $(x,y)$  and the time variable  $t$ , the heat equation with peak  $p(x,y)$  as the source is as follows:



**Fig. 10.** Temperature variations for node A and its NW, E and S neighbors.

**Table 5**  
Correct, anomalous and estimated values for the sensor A.

Time (min)	10	20	40	60	70	71	72	90	100	105	120	130	140	145	150
Measurements for sensor A without anomalies (°C)	19.43	18.98	17.22	19.57	19.58	19.35	19.06	17.15	18.33	18.34	17.33	18.39	19.75	19.89	19.11
Measurements for sensor A with anomalies (°C)	18	20	16	21	18	20	17	18.5	17	19.5	16	19.5	18	21	17
Estimated values (°C)	19.51	18.80	17.29	19.47	19.40	19.17	18.87	17.34	18.23	18.39	17.22	18.39	19.71	19.74	18.96

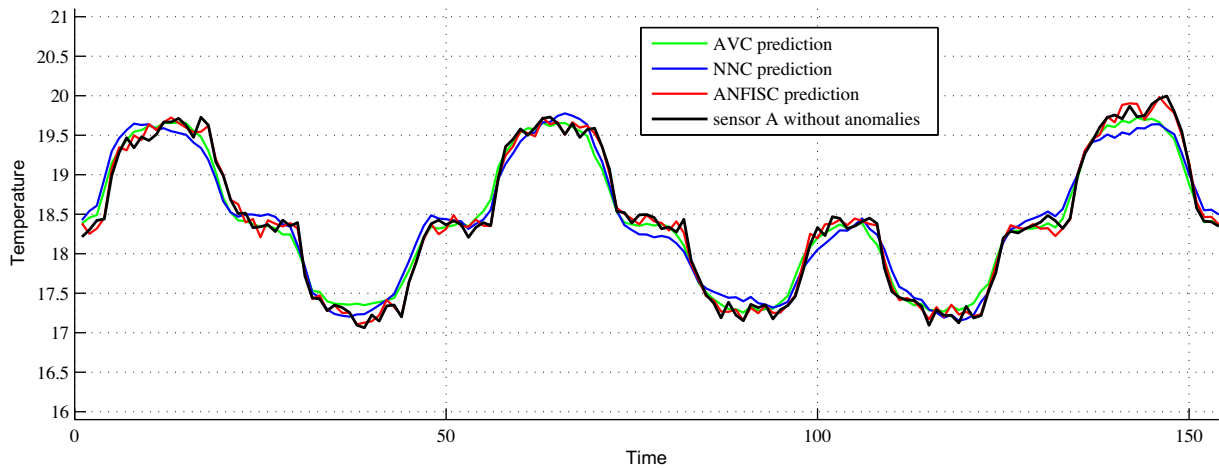


Fig. 11. Estimations of the sensor A temperature done by the AVC, NNC and ANFISC classifiers.

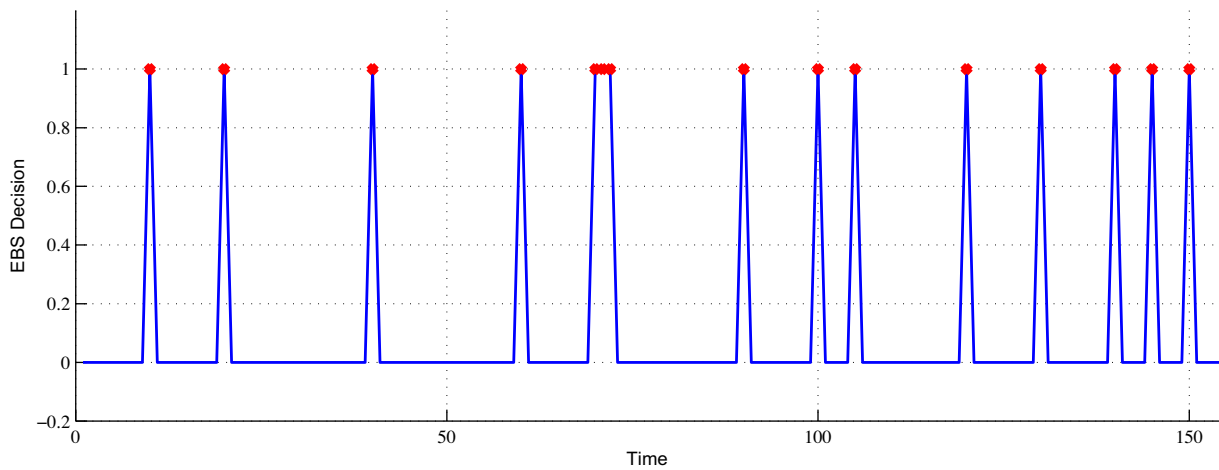


Fig. 12. The overall decision provided by the EBS.

$$\frac{\partial u}{\partial t} = \alpha \cdot \left( \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) - p(x, y). \quad (20)$$

We simulated this equation in Matlab (PDEToolbox) with different parameters to model diverse dynamics of the heat in the 2D environment where the sensor nodes are placed. In order to introduce erroneous values, we changed deliberately some of the temperature values near the node under investigation, at random moments in time.

For our case study, we considered all the weights of the EBS decision block to be 0.2 and the thresholds of the individual classifiers to be 0.35 °C.

Due to the way abnormal activity is discovered in our methodology, the datasets are not composed of independent values, but of packages of values obtained for different consecutive moments in time to catch the dynamics of the environment. For this reason the datasets for training and testing are structured as in Table 2 (neighboring nodes of the sensor A are individualized using their relative position described by cardinal and intercardinal directions: e.g. NorthWest – NW, East – E, etc.), where a package is represented by a number of measurements reported at successive moments in time.

The training datasets included six packages, half of them obtained through computer simulation and half obtained using the

experimental WSN deployment. The testing datasets consisted of only four packages, two obtained from the real deployment and two through computer simulation (Table 3).

In the training process we imposed that the  $Q$ -statistics between every two classifiers to be less than 0.91 and the overall  $Q$ -statistics obtained using Eq. (19) to be under 0.87. These goals have been achieved: the trained and validated ensemble reported an overall  $Q$ -statistics of 0.855, while the pairwise  $Q$ -statistics for the five classifiers are presented in Table 4.

The obtained results prove the efficiency of our ensemble based methodology, in the sense that the ensemble categorized accurately 99.71 of the situations and the best individual classifier in terms of results (ANFISC) had 97.99%, while the worst individual classifier (AVC) had 91.26% accuracy.

To show how our EBS works, in Fig. 10 we presented the testing dataset package number 9, where the values related to sensor A have been intentionally modified to simulate anomalies at 15 instances in time ( $t = 10, 20, 40, 60, 70, 71, 72, 90, 100, 105, 120, 130, 140, 145, 150$ ), according to Table 5.

The estimated temperatures for the sensor A done by three of our five classifiers ( $\theta_{A,AVC}(t), \theta_{A,NNC}(t), \theta_{A,ANFISC}(t)$ ) in comparison to the original sensor A time series (without anomalies), are presented in Fig. 11, while the overall decisions taken by the ensemble are depicted in Fig. 12, with the remark that for the first six instances in time the decisions were taken only by the average

based classifier due to the reasons mentioned in Section 3 (the necessity to populate input vectors for ALC, NNC, NNAC, ANFISC classifiers with real measurements and the initial convergence of ALC).

The estimated values proposed by our ensemble to replace the abnormal measurements are presented in the last line of Table 5. By comparing the sensor A unperturbed measurements with the estimated values, it results a mean absolute error of only 0.118 °C and a maximum absolute error of 0.19 °C obtained at  $t = 72$  and  $t = 90$ , confirming once again the preciseness of our methodology.

## 7. Conclusions

It is in the human nature to ask for two, three or maybe more different authorized opinions before taking a significant decision. This human characteristic was transferred to the domain of artificial intelligence through the concept of ensemble based system, producing relevant results in a large variety of domains. Being exposed to numerous risks, WSN often use complex decisional systems for controlling their lifecycle, for processing measurement data or even for dealing with malicious security attacks. Our paper presents a novel perspective upon the sensing anomaly detection in wireless sensor networks by involving an ensemble of five carefully selected binary classifiers. Each of these five classifiers comprises a dynamical model of “correct behavior” of the sensor, estimated based on past measurements provided by the sensor itself or by neighboring sensors. The algorithms involved vary from a simple average computing to complex neural or ANFIS networks assuring the desired diversity of the classifiers to implement an efficient decision making system. Furthermore, using the power of the same ensemble we can precisely estimate the correct value of any measurement affected by sensing anomalies.

## Acknowledgment

This work was developed in the frame of PNII-IDEI-PCE-ID923-2009 CNCIS – UEFISCSU.

## References

- An, S. H., Heo, G., & Chang, S. H. (2011). Detection of process anomalies using an improved statistical learning framework. *Expert Systems with Applications*, 38(3), 1356–1363.
- Belongie, S., Malik, J., & Puzicha, J. (2002). Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4), 509–522.
- Cao, F., Lisani, J.-L., Morel, J.-M., Musé, P., & Sur, F. (2008). A theory of shape identification. *Lecture Notes in Mathematics*, 1948, 264.
- Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. *ACM Computing Surveys*, 41(3), 1–58.
- Chatzigiannakis, V., & Papavassiliou, S. (2007). Diagnosing anomalies and identifying faulty nodes in sensor networks. *IEEE Sensors Journal*, 7(5), 637–645.
- Curia, D., Volosencu, C., Pescaru, D., Jurca, L., & Doboli, A. (2009). Redundancy and its applications in wireless sensor networks: A survey. *WSEAS Transaction on Computer*, 8(4), 705–714.
- Dietterich, T. G. (2000). An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. *Machine Learning*, 40(2), 139–157.
- García-Pedrajas, N., & Ortiz-Boyer, D. (2009). Boosting k-nearest neighbor classifier by means of input space projection. *Expert Systems with Applications*, 36(7), 10570–10582.
- Ho, T. K., Hull, J. J., & Srihari, S. N. (1994). Decision combination in multiple classifier systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(1), 66–75.
- Hsu, K.-W., & Srivastava, J. (2009). Diversity in combinations of heterogeneous classifiers. In *Proceedings of the 13th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining (PAKDD'09)*, Bangkok, Thailand, LNCS 5476 (pp. 923–932).
- Huttenlocher, D., Klanderman, G., & Rucklidge, W. (1993). Comparing images using the hausdorff distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(9), 850–863.
- Jang, J.-S. R. (1991). Fuzzy modeling using generalized neural networks and Kalman filter algorithm. In *Proceedings of the ninth national conf on artificial intelligence (AAAI-91)* (pp. 762–767).
- Jang, J.-S. R. (1993). ANFIS: Adaptive-network-based fuzzy inference systems. *IEEE Transactions on Systems, Man, and Cybernetics*, 23(3), 665–685.
- Kim, M., & Kang, D. (2010). Ensemble with neural networks for bankruptcy prediction. *Expert Systems with Applications*, 37(4), 3373–3379.
- Kuncheva, L., & Whitaker, C. (2003). Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine Learning*, 51(2), 181–207.
- Li, H., & Sun, J. (2012). Case-based reasoning ensemble and business application: A computational approach from multiple case representations driven by randomness. *Expert Systems with Applications*, 39(3), 3298–3310.
- Li, Y. Y., Thomason, M., Parker, L. E. (2010). Detecting time-related changes in wireless sensor networks using symbol compression and probabilistic suffix trees. In *Proceedings of the IEEE/RSJ international conference on intelligent robots and systems (IROS2010)*, Taipei (pp. 2946–2951).
- Littlestone, N., & Warmuth, M. (1994). The weighted majority algorithm. *Information and Computation*, 108(2), 212–261.
- Ljung, L. (1999). *System identification – Theory for the user* (2nd ed.). Upper Saddle River, NJ: Prentice Hall.
- Ljung, L. (2007). *System identification toolbox for use with matlab* (7th ed.). Natick, MA: The MathWorks Inc.
- Ljung, L., & Soderstrom, T. (1987). *Theory and practice of recursive identification*. Cambridge, MA: MIT Press.
- MathWorks (2002). *Matlab fuzzy logic toolbox. User's guide version 2*. Natick, MA: The Math Works Inc.
- Polikar, R. (2006). Ensemble based systems in decision making. *IEEE Circuits and Systems Magazine*, 6(3), 21–45.
- Rajasegarar, S., Leckie, C., & Palansiwami, M. (2008). Anomaly detection in wireless sensor networks. *IEEE Wireless Communications*, 15(4), 34–40.
- Shirai, S., Kudo, M., & Nakamura, A. (2009). Comparison of bagging and boosting algorithms on sample and feature weighting. In *Proceedings of the 8th international workshop multiple classifier systems (MCS2009)*, Reykjavik, Iceland, LNCS 5519 (pp. 22–31).
- Siripanadorn, S., Hattagam, W., & Teaumroong, N. (2010). Anomaly detection in wireless sensor networks using self-organizing map and wavelets. *International Journal of Communications*, 4(3), 74–83.
- Tsoumakas, G., Katakis, I., & Vlahavas, I. (2004). Effective voting of heterogeneous classifiers. In *Proceedings of the 15th European conference on machine learning (ECML2004)*, Pisa, Italy (pp. 465–476).
- Wang, G., Hao, J. X., Ma, J., & Jiang, H. B. (2011). A comparative assessment of ensemble learning for credit scoring. *Expert Systems with Applications*, 38(1), 223–230.
- Walters, J. P., Liang, Z., Shi, W., & Chaudhary, V. (2006). Wireless sensor network security: A survey. In Xiao Yang (Ed.), *Security in distributed, grid, and pervasive computing, computing* (pp. 367–410). CRC Press.
- Wang, R. X., Lizier, J. T., Obst, O., Prokopenko, M., & Wang, P. (2008). Spatiotemporal anomaly detection in gas monitoring sensor networks. In *Proceedings of the European conference on sensor networks (EWSN2008)*, Bologna, Italy, LNCS 4913 (pp. 90–105).
- Yang, B., & Bohme, J. F. (1992). Rotation-base RLS algorithms: Unified derivations, numerical properties and parallel implementations. *IEEE Transactions on Signal Processing*, 40(5), 1151–1166.