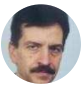
Application of Wavelet Neural-Networks in Wireless Sensor Networks.

Conference Paper · January 2005

DOI: 10.1109/SNPD-SAWN.2005.23 · Source: DBLP

3 authors:


Andrea Kulakov
Ss. Cyril and Methodius University


Danco Davcev
Ss. Cyril and Methodius University


Goran Trajkovski
Western Governors University

Application of wavelet neural-networks in wireless sensor networks

ABSTRACT

Most of the current in-network data processing algorithms are modified regression techniques like multidimensional data series analysis. In our opinion, some of the algorithms developed within the artificial neural-networks tradition can be easily adopted to wireless sensor network platforms and will meet the requirements for sensor networks like: simple parallel distributed computation, distributed storage, data robustness and auto-classification of sensor readings. As a result of the dimensionality reduction obtained simply from the outputs of the neural-networks clustering algorithms, lower communication costs and energy savings can also be obtained.

In this paper we will present two different data aggregation architectures with algorithms using artificial neural-networks which use unsupervised learning methods for categorization of the sensory inputs. They are tested on a data obtained from a set of several motes, equipped with several sensors each. Results from simulations of purposefully faulty sensors show the data robustness of these architectures.

These architectures are further developed adding one pre-processing level which will use wavelets for initial data-processing of the sensory inputs at different resolutions and later introduced into the artificial neural-networks. The effects of this additional wavelet pre-processing are given for the two above mentioned architectures.

Keywords: wavelet neural-networks, wireless sensor networks, dimensionality reduction, data robustness

INTRODUCTION

Sensor networks place several requirements on a distributed storage infrastructure. These systems are highly data-driven and are deployed to observe, analyze and understand the physical world. A fully centralized data collection strategy is infeasible given the energy constraints on sensor node communication, and is also inefficient given that sensor data has significant redundancy both in time and in space. In cases when the application demands compressed summaries of large spatio-temporal sensor data and similarity queries, such as detecting correlations and finding similar patterns, the use of a neural-network algorithm is a reasonable choice.

The development of the wireless sensor networks is accompanied by several algorithms for data processing which are modified regression techniques from the field of multidimensional data series analysis in other scientific fields, with examples like nearest neighbor search, principal component analysis and multidimensional scaling (e.g. [8,11]). We argue that some of the algorithms well developed within the neural-networks tradition for over 40 years, are well suited to fit into the requirements imposed to sensor networks for: simple parallel distributed computation, distributed storage, data robustness and auto-classification of sensor readings.

Auto-classification of the sensor readings is important in sensor networks since the data obtained with them is with high dimensionality and very immense, which could easily overwhelm the processing and storage capacity of a centralized database system. On the other hand, the data obtained by the sensor networks are often self-correlated over time, over space and over different sensor inputs, due to the nature of the phenomena being sensed which is often slowly changing, due to the redundant sensor nodes dispersed near each other, and due to the fact that often the sensor readings are correlated over different modalities sensed at one node (e.g. sound and light from cars in traffic control application).

Neural-networks algorithms, on the other hand, use simple computations and do not represent big burden to memory. The proposed adaptations of the ART neural networks models can be easily parameterized according to user needs for greater or lower level of details of the sensor data. The outputs of the ART neural networks can also be easily transformed into if-then decision rules understandable to humans (e.g. [4]).

Up to date, the only application of neural-networks algorithms for data processing in the field of sensor networks is the work of Catterall et al. [7] where they have slightly modified the Kohonen Self Organizing Maps model. Even this application was presented to a different kind of audience at a conference for Artificial Life. This has additionally motivated us to bring closer the work done in the field of Artificial Neural Networks to the community of researchers working in the field of Sensor Networks, since some of the problems for the processing of the sensory input data are similar.

Unsupervised learning Artificial Neural Networks typically perform dimensionality reduction or pattern clustering. They are able to discover both regularities and irregularities in the redundant input data by iterative process of adjusting weights of interconnections between a large numbers of simple computational units (called artificial neurons). As a result of the dimensionality reduction obtained easily from the outputs of these algorithms, lower communication costs and thus bigger energy savings can also be obtained.

A neural network algorithm can be implemented in the tiny platform of Smart-It units, which are kind of sensor nodes or motes. Thus instead of reporting the raw-data, each Smart-It unit can send only the cluster number where the current sensory input pattern has been classified. In that way a huge dimensionality reduction can be achieved depending on the number of sensor inputs in each unit. In the same time communication savings will benefit from the fact that the cluster number is a small binary number unlike sensory readings which can be several bytes long real numbers converted from the analog inputs.

Since the communication is the biggest consumer of the energy in the units, this leads to bigger energy savings as well.

In its simplest form, the wavelet transform is a series of filtering and subsampling operations, which repeatedly split a signal into low frequency and high frequency components in a reversible manner [13]. Several variants are possible, such as extending the number of channels and generalizing key results from uniformly sampled one-dimensional signals to irregular sampling grids over complex shapes [1,13,14]. Many of these algorithms are memory efficient and lend themselves to real-time implementations, which may be particularly suitable for interpreting streaming sensor data.

In the paper we will review first the ART1 [2] and FuzzyART [3] models. Later some related work will be considered and after that our proposal of three different kinds of architectures for incorporating the Artificial Neural Networks into the small Smart-It units' network will be given. Shortly we will present the hardware platform that has been originally used to obtain the data that we later used to test our proposal and we will give some results of the classifications of the data within different architectures. We will also give results from the simulations where we have purposefully made one of the input sensors malfunctioning in order to show the data robustness of our approach. Later we show how wavelet transform of the signals can be incorporated in the proposed architectures and give one example. Finally we will give some discussions and directions for future work.

ART AND FUZZYART ALGORITHMS

Several models of unsupervised Artificial Neural Networks have been proposed like Multi-layer Perceptron (MLP), Self-Organizing Maps (SOMs), and Adaptive Resonance Theory (ART) ([9] and [10]). Out of these we have chosen the ART models for implementation in the field of sensor networks because they do not constrain the number of different categories in which the input data will be clustered. Although the later extensions of MLP and SOMs involve the principle of incrementally growing structure, their topological self-organization is possible only with so called off-line learning cycle separate from the classification cycle. Having two separate cycles is inconvenient in the presence of potentially unlimited stream of input data with no reliable method of choosing the suitably representative subset for a learning cycle. ART algorithms offer another example of topological self-organization of data but they can adapt structure quickly in the so called fast-learning mode explained later.

Adaptive Resonance Theory (ART) has been developed by Grossberg and Carpenter for pattern recognition primarily. Models of unsupervised learning include ART1 [2] for binary input patterns and FuzzyART [3] for analog input patterns.

ART networks develop stable recognition codes by self-organization in response to arbitrary sequences of input patterns. They were designed to solve the so called stability-plasticity dilemma: how to continue to learn from new events without forgetting previously learned information. ART networks model several features such as robustness to variations in intensity), detection of signals mixed with noise, and both short- and long-term memory to accommodate variable rates of change in the environment. There are several variations of ART-based networks: ART1 (three-layer network with binary inputs), Fuzzy ART (with analog inputs, representing neuro-fuzzy hybrids which inherit all key features of ART), their supervised versions ARTMAP and FuzzyARTMAP and many others. ARTMAP models [4], for example, combine two unsupervised modules to carry out supervised learning.

In Figure 1 typical representation of an ART Artificial Neural Network is given. Winning F2 category nodes are selected by the attentional subsystem. Category search is controlled by the orienting subsystem. If the degree of category match at the F1 layer is lower than the so called vigilance level ρ, a reset signal will be triggered, which will deactivate the current winning F2 node for the period of presentation of the current input.

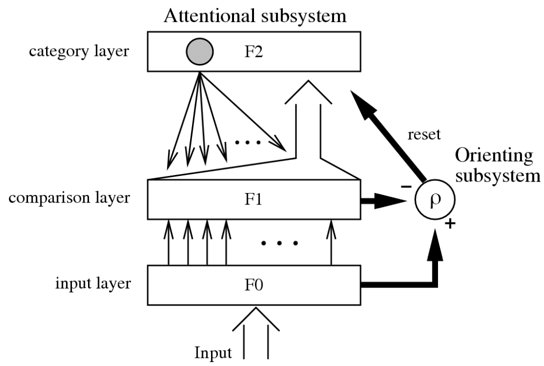


Figure 1 Architecture of the ART network.

An ART network is built up of three layers: the input layer (F0), the comparison layer (F1) and the recognition layer (F2) with N, N and M neurons, respectively. The input layer stores the input pattern, and each neuron in the input layer is connected to its corresponding node in the comparison layer via one-to-one, non-modifiable links. Nodes in the F2 layer represent input categories. The F1 and F2 layers interact with each other through weighted bottom-up and top-down connections that are modified when the network learns. There are additional gain control signals in the network (not shown in Figure 1) that regulate its operation, but those will not be detailed here.

The learning process of the network can be described as follows: At each presentation of a non-zero binary input pattern x ($x_j \in \{0, 1\}$; $j = 1, 2, \ldots, N$), the network attempts to classify it into one of its existing categories based on its similarity to the stored prototype of each category node. More precisely, for each node i in the F2 layer, the bottom-up activation T_i is calculated, which can be expressed as

$$T_i = \frac{|\mathrm{w}_i \cap \mathrm{x}|}{\beta + |\mathrm{w}_i|} \quad i = 1, \ldots, M \qquad (1)$$

where $|\cdot|$ is the norm operator (for a vector u it is: $|\mathrm{u}| \equiv \sum_{j=1}^{N} u_j$), w_i is the (binary) weight vector or prototype of category i, and $\beta > 0$ is a parameter. Then the F2 node I that has the highest bottom-up activation, i.e. $T_I = \max\{T_i \mid i = 1, \ldots, M\}$, is selected (realizing so called winner-takes-all competition).

The weight vector of the winning node (w_I) will then be compared to the current input at the comparison layer. If they are similar enough, i.e. if they satisfy the matching condition:

$$\frac{|w_I \cap x|}{|x|} \geq \rho \qquad (2)$$

where ρ is a system parameter called vigilance ($0 < \rho \leq 1$), then the F2 node I will capture the current input and the network learns by modifying w_I:

$$w_I^{new} = \eta(w_I^{old} \cap x) + (1-\eta)w_I^{old} \qquad (3)$$

where η is the learning rate ($0 < \eta \leq 1$) (the case when $\eta = 1$ is called "fast learning"). All other weights in the network remain unchanged.

If, however, the stored prototype w_I does not match the input sufficiently, i.e. if the condition (2) is not met, the winning F2 node will be reset (by activating the reset signal in Figure 1) for the period of presentation of the current input. Then another F2 node (or category) is selected with the highest T_i, whose prototype will be matched against the input, and so on. This "hypothesis-testing" cycle is repeated until the network either finds a stored category whose prototype matches the input well enough, or allocates a new F2 node in which case learning takes place according to (3).

As a consequence of its stability-plasticity property, the network is capable of learning "on-line", i.e. refining its learned categories in response to a stream of new input patterns, as opposed to being trained "off-line" on a finite training set.

The number of developed categories can be controlled by setting the vigilance ρ: the higher the vigilance level, the larger number of more specific categories will be created. At its extreme, if $\rho = 1$, the network will create a new category for every unique input pattern.

FuzzyART is an analog version of the ART1 algorithm which takes analog inputs and classifies them in a similar way as ART1. The main ART1 operations of category choice (1), match (2), and learning (3) translate into Fuzzy ART operations by replacing the ordinary set theory intersection operator ∩ of ART1 with the fuzzy set theory conjunction MIN operator ∧.

In FuzzyART (but as well in ART1), complement coding of the input vector prevents a type of category proliferation that could otherwise occur when weights erode. Complement coding doubles the dimensionality of an input vector b ≡ (b_1, …, b_N) by concatenating the vector b with its complement b^c. The input to the FuzzyART network (F0 in Figure 1) is then a 2N-dimensional vector: I=B≡(b, b^c), where $(b^c)_i \equiv (1- b_i)$. If b represents input features, then complement coding allows a learned category representation to encode the degree to which each feature is consistently absent from the input vector, as well as the degree to which it is consistently present in the input vector, when that category is active. Because of its computational advantages, complement coding is used in nearly all ART applications, and we have used it in our models as well.

The strengths of the ART models include its unique ability to solve a stability-plasticity dilemma, extremely short training times in the fast-learning mode, n incrementally growing number of clusters based on the variations in the input data and the fact that there is a need to set up any initial categories. The network runs entirely autonomously; it does not need any outside control, it can learn and classify at the same time, provides fast access to match results, and is designed to work with infinite stream of data. ART networks can be controlled not to become too sensitive to the variations in the input. All these features make it an excellent choice for application in wireless sensor networks.

RELATED WORK

As we mentioned in the introduction, Catterall et al. [7] have slightly modified the Kohonen Self Organizing Maps (SOMs) model. Kohonen SOMs and ART models are similar in a way that they are both prototype-based networks where they both create a set of prototypes and then compare an unknown input vector with the stored prototypes in order to implement the mapping or clustering.

The advantages of SOMs over other Artificial Neural Network models include the ability to provide real-time nearest-neighbor response as well as topology-preserving mapping of the input data. Still, the limitations are extensive off-line learning and most importantly, the need of a predefined map size, i.e. a fixed number of output clusters or categories. In [7] a rather simple and straightforward implementation of the Kohonen neural-network architecture is used, where one cluster unit corresponds to one Smart-It hardware unit.

In many real-world situations, there is no a priori information on variability present in the data stream, so we can not determine in advance the required number of output clusters in which the input patterns will fit. Thus this straightforward implementation of the Kohonen neural network seems rather rudimentary and the only justification for it can be the mere possibility to apply some principles of Artificial Neural Networks for data processing in wireless sensor networks.

DIMENSIONS [8] is another model where they treat the problems of data storage and handling in sensor systems. DIMENSIONS incorporates spatio-temporal data reduction to distributed storage architectures, introduces local cost functions to data compression techniques, and adds distributed decision making and communication cost to data mining paradigms. It provides unified view of data handling in sensor networks incorporating long-term and short-term storage with increasing lossy compression over time, multi-resolution data access using different wavelet parameters at different hierarchical levels, and spatio-temporal pattern mining.

Several aspects of our approach are common to DIMENSIONS such as spatio-temporal data reduction, limited communication costs, long-term and short-term storage and hierarchical access with different level of details, although these aspects are achieved by completely different algorithms.

PROPOSED ARCHITECTURES OF SENSOR NETWORKS

Three types of network architectures are proposed. The results of the classifications of a real-world data will be given later for each of the architectures.

1) One Clusterhead collecting all sensor data

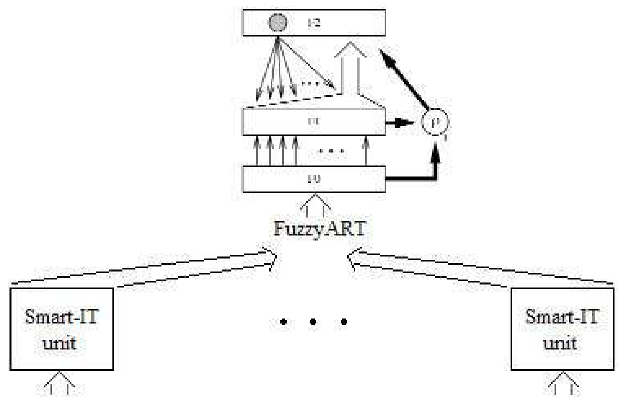


Figure 2 Clusterhead collecting all sensor data from its cluster of units

First we have made this architecture to compare the work of Catterall et al. [7] (which used Kohonen SOMs), in order to show that ART model can be used straightforwardly instead of SOMs. This model brings advantages in that we do not have to fix in advance the number of clusters (categories) that the network should learn to recognize. Here the Smart-It units send the sensory readings to one of them chosen to be a Clusterhead, where a FuzzyART network is implemented.

The sensor data can be classified with different vigilance parameter ρ, thus providing a general overall view on the network, (smaller ρ) or more and more detailed views of the network (greater ρ). Depending on the level of details needed at the moment, some of the Clusterheads can be adjusted to different vigilance parameters and later can be queried giving coarser or more detailed results.

2) Clusterhead collecting only clustering outputs from the other units.

Each Smart-It unit has FuzzyART implementations classifying only its sensor readings. One of the Smart-It units can be chosen to be a Clusterhead collecting and classifying only the classifications obtained at other units. Since the clusters at each unit can be represented with integer values, the neural-network implementation at the Clusterhead is ART1 with binary inputs.

With this architecture a great dimensionality reduction can be achieved depending on the number of sensor inputs in each unit (in our case it's a 6-to-1 reduction). In the same time communication savings benefit from the fact that the cluster number is a small binary number unlike raw sensory readings which can be several bytes long real numbers converted from the analog inputs.

If the number of sensors in each unit is n, the Clusterhead collects data from k units, and the number of different categories in each unit can be represented by c-byte integer, while the sensor readings are real numbers represented with r bytes, then the communication saving can be calculated as:

$$\frac{n \cdot k \cdot r}{k \cdot c} = \frac{n \cdot r}{c}$$

Since the communication is the biggest consumer of the energy in the units, this leads to bigger energy savings as well.

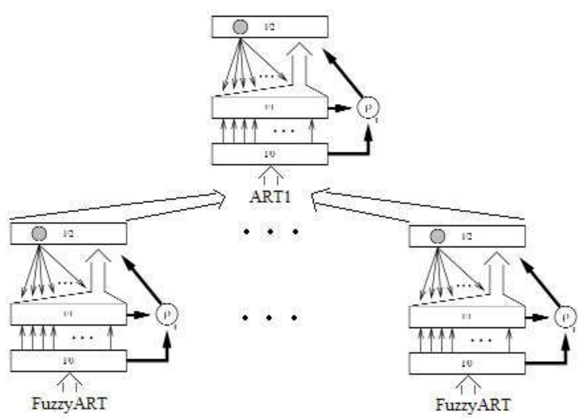


Figure 3 One Clusterhead collecting and classifying the data after they are once classified at the lower level

Another benefit from this architecture is the fact that we can view the classifications at the Clusterhead as an indication of the spatio-temporal correlations of the input data.

3) Wavelet preprocessors of the sensor input.

Each of these two architectures can use a wavelet transformation on each sensor signal, treated as a one dimensional time series. Different wavelet transformation methods have been developed, more or less complex. We have chosen one of the simplest methods Haar 1D wavelet transform, which can be explained as:

if $c_1(k)=½\ [c_0(k) + c_0(k-1)]$ is the first smoothing of the signal, then $w_1(k)= c_1(k) - c_0(k)$ is the first wavelet coefficient. This can be generalized as

$w_i(k)= c_i(k) - c_{i-1}(k)$.

The first smoothing $c_0(t)=x(t)$ is the signal itself and it is the finest scale.

The combination of wavelet coefficients and the residual vector c_p $\{w_1(t), w_2(t), \ldots, w_n(t), c_p(t)\}$ can be taken as a multi-resolution wavelet feature at time *t* of the signal and it can be processed further as a representation of the signal itself.

In the Figure 4 we give the first architecture enhanced with Discrete Wavelet Transform (DWT) preprocessor. We will also give results of the simulations with the real-world data later.

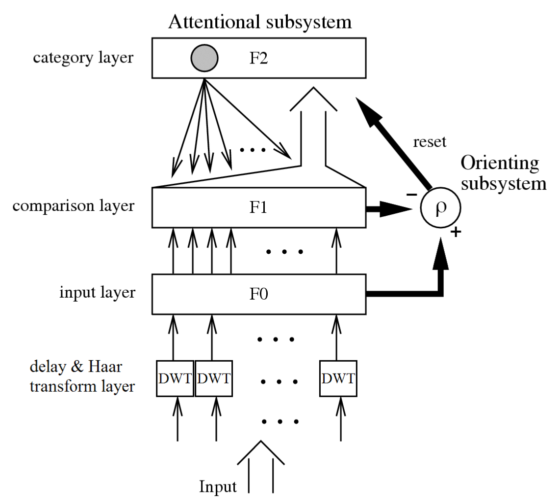


Figure 4 Discrete Wavelet Transform is applied to each signal before analyzing it with an ART neural-network

HARDWARE PLATFORM

The platform for the experiments, from which the data analyzed in this paper were obtained, is a collection of 'Smart-Its' (see [7], for more details). One Smart-It unit embodies a *sensor module*, and a *communication module*, which are interconnected. The core of sensor module is a PIC 16F877 microcontroller clocked at 20 MHz, which offers 384 bytes of data memory and 8Kx14 bits of memory. The sensor module consists of a light sensor, a microphone, 2 accelerometers, a thermometer and a pressure sensor. An RF stack provides wireless communication, at a maximum rate of 125 kbps, which only supports broadcast.

EXPERIMENTAL RESULTS

The data used in these experiments were provided courtesy of Catterall et al[1]. All results presented here were produced using datasets containing real-world data.

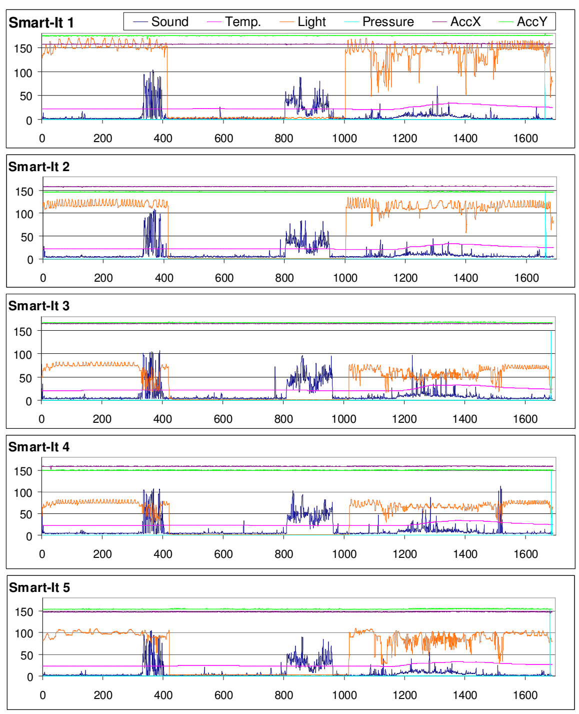


Figure 5 Datasets with sensor values from each of the Smart-It units (taken over 1700 smaples) during several states of the environment ('contexts'): Lights on (1-330), talking people nearby while lights remain on (331-400), lights turned off (401-800), talking people nearby while light remain turned off (801-1000), and heating on (1090-1400).

---

[1] The data files are available for download at this website:

www.comp.lancs.ac.uk/~catterae/alife2002/

The five datasets (one for each Smart-It) are visualized by time series plots in Figure 5. Note that, although the sensor data are very similar (as the units are physically close to each other), they are not *exactly* the same. In all subsequent charts, the y-axis shows the recognized category of the sample input, while the x-axis shows the sample sensory input.

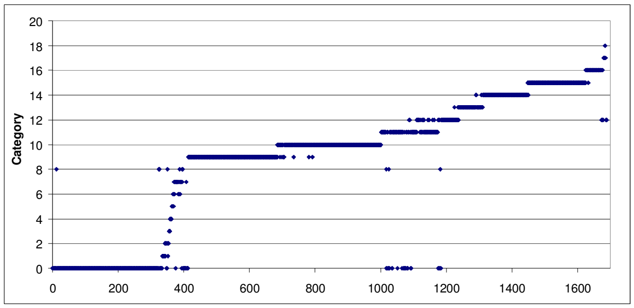


Figure 6 One possible classification of the input data

All the experiments were conducted with complement coding of the input vector and fast learning mode. Figure 6 shows one possible classification of these input data in an architecture presented in Figure 2 with vigilance level ρ=0.93.

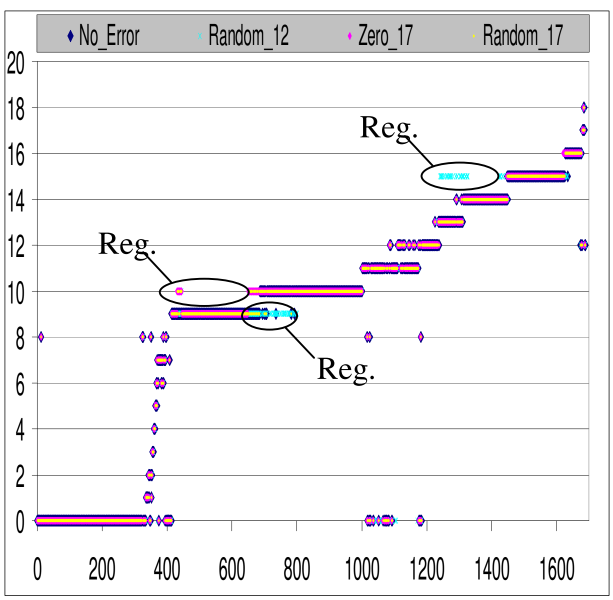


Figure 7 Different classifications when some of the sensors are defective giving zero or random values.

For testing the data robustness of the models, we have synthetically made one of the sensors at a time defective in way that it gives either a zero constant signal or a random measurement signal. Training was done with vigilance set to 0.93, while testing was done with vigilance set to 0.90. In Figure 7 the effects of the representative sensor errors are shown (sensor numbered 12 and sensor numbered 17, out of 30), where with the ovals are highlighted the regions where the classifications differ from the case when all sensors are functioning correctly. In Regions 1 and 2, the classification of the case when the sensor number 12 gives random values differs from the regular case, while in Region 3, the defective sensor number 17 results in different classification than the regular case. In Region 2, the cases when the 17th sensor gives random or zero values also results in different classifications. All this can be concluded according to the different colors and shapes of the dots in the chart in Figure 7.

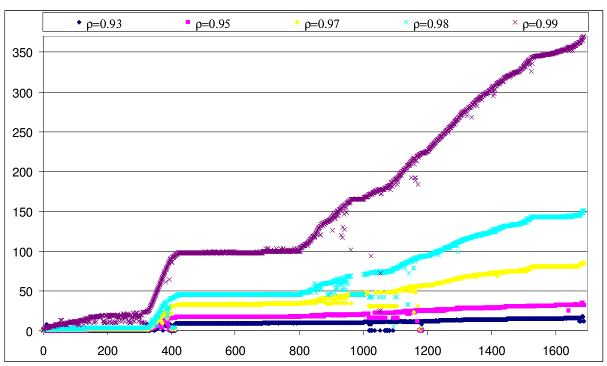


Figure 8 **Different numbers of categories appear when the data is classified at different levels of details depending on the vigilance parameter ρ**

If in this architecture different values of the vigilance parameter ρ are used (ranging from 0.93 up to 0.99 in our experiments) we get different number of output categories (from 20 up to 370) for the 1700 samples taken as a learning dataset, which is shown in Figure 8.

For the second architecture (Figure 3) we have also conducted experiments with the original data and with the synthetically made erroneous data. In Figure 9 we give the results of the classifications of the Clusterhead collecting only the classifications from the other Smart-It units. The training was done with vigilance level of 0.88, while the testing with 0.70. The results show no significant difference among the classifications when all sensors are functioning correctly or when some of the sensors give only zero or random signal (in our case sensors number 12 and 17).

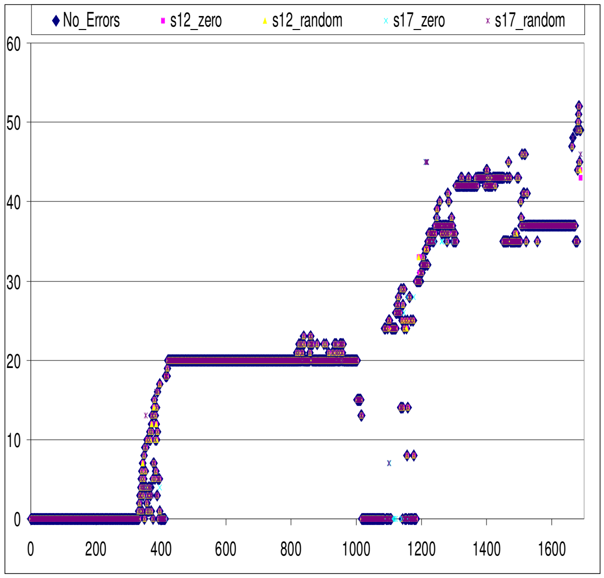


Figure 9 Results of the classifications show significant data robustness of the second architecture with one Clusterhead collecting only clustering outputs from the other units.

In order to test the possibility to preprocess the signals before analyzing them with a neural-network, we have implemented only the calculation of one-scale wavelets $w_1(k)$ and taken them as an input for the neural-network clustering.

We have calculated the wavelet coefficients over different time windows. The results of the clustering of the wavelet coefficients calculated over 512 samples are given in Figure 10. Generally the results have shown that the number of clusters into which the signals are classified is smaller for the same parameters of the ART neural-networks.

And again, in order to show the data-robustness of this newly proposed data-processing method, we have analyzed the signals which were purposefully made faulty (sensor 17 giving zero or random signals), the results of which are shown in Figure 11.

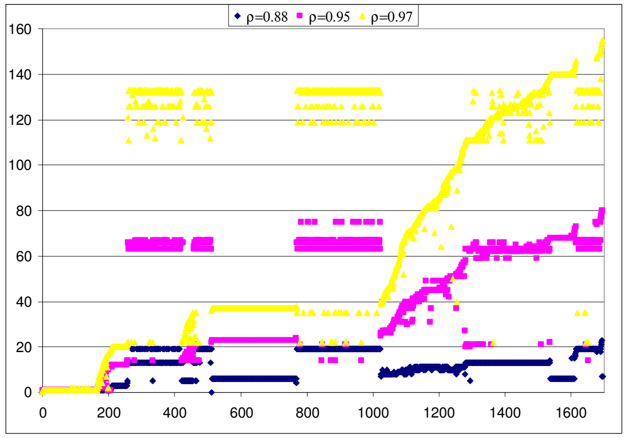


Figure 10 **Clustering of the 1D Haar wavelet coefficients calculated over 512 samples, in ART neural-network with different vigilance ρ**

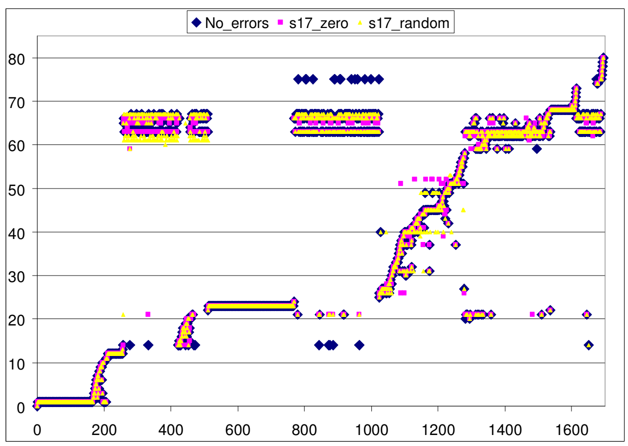


Figure 11 **Clustering of the 1D Haar wavelet coefficients calculated over 512 samples, in an ART neural-network with ρ=0.95, where the 17th sensor is faulty giving zero or random signals**

In Figure 12 we show the data robustness of the second architecture when using the wavelet preprocessing units and again we have analyzed the signals which were purposefully made faulty (sensor 17 giving zero or random signals).

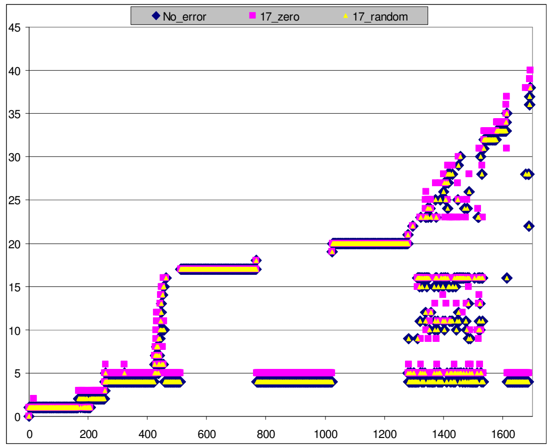


Figure 12 **Clustering of the 1D Haar wavelet coefficients calculated over 512 samples, with the second architecture ρ=0.80, where the 17th sensor is faulty giving zero or random signals**

DISCUSSION

The second proposed architecture (Figure 3) with one Clusterhead collecting only clustering outputs from the other units can be generalized to a hierarchical cascade classification scheme where small Smart-It units at the lowest level will be grouped in small groups having one Clusterhead. Then several Clusterheads can be grouped and their outputs can be classified using a binary input ART1 classifier at a Clusterhead one-level higher and so on, up to a level where the classification will be read by a human user or stored in a database, after achieving a huge dimensionality reduction (see Figure 13).

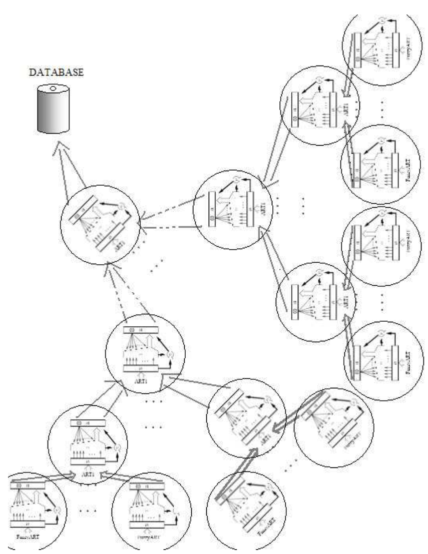


Figure 13 Hierarchical cascades of ART neural-network classifiers implemented in units of a sensor network

If at each level classifications from k units are clustered into one Clusterhead (represented with the same number of bytes), the dimensionality reduction after l levels will be:

$$\frac{n \cdot r}{c} \cdot k^{l}$$

For future work we are also considering to apply the supervised learning versions of the ART algorithms, namely ARTMAP, FuzzyARTMAP and dARTMAP [6] where along with the sensor input vector, a vector of corresponding "right-answers" is obtained by the user (so called teacher), or possibly automatically from another system. Also improvements of the ARTMAP algorithms such as ARTC [12] may be taken into account.

CONCLUSION

In this paper we have demonstrated a possible adaptation of one popular model of Artificial Neural Networks algorithm (ART model) in the field of wireless sensor networks. The positive features of the ART class algorithms such as simple parallel distributed computation, distributed storage, data robustness and auto-classification of sensor readings are demonstrated within two different proposed architectures, and then it is also demonstrated when combining them with initial wavelet preprocessing of the input signals.

One of the proposed architectures with one Clusterhead collecting only clustering outputs from the other units provides a big dimensionality reduction and in the same time additional communication saving, since only classification IDs (small binaries) are passed to the Clusterhead instead of all input samples.

Results from the simulated deliberately erroneous sensors, where we imitate defective sensors giving only zero or random output, show that the model is robust to small variations in the input.

The Discrete Wavelet Transform of the signals can be incorporated as a preprocessing unit of the neural-networks giving the ability to extract important features in the data like abrupt changes at various scales. This architecture also shows significant data-robustness to errors.

REFERENCES

[1] Amaratunga, K. and J. E. Castrillón-Candás, "Surface Wavelets: A Multiresolution Signal Processing Tool for 3D Computational Modeling," *International Journal for Numerical Methods in Engineering*, **52**, 3, 239-271, 2001.

[2] Carpenter, G.A., and Grossberg, S., A massively parallel architecture for a self-organizing neural pattern recognition machine, Computer Vision, Graphics, and Image Processing, vol. 37, pp. 54-115, 1987.

[3] Carpenter, G.A., Grossberg, S., and Rosen, D.B., Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system, Neural Networks, vol. 4, pp. 759-771, 1991.

[4] Carpenter, G.A., Grossberg, S., Markuzon, N., Reynolds, J.H., and Rosen, D.B., Fuzzy ARTMAP: A neural network architecture for incremental supervised learning of analog multidimensional maps, IEEE Transactions on Neural Networks, vol. 3, pp. 698-713, 1992.

[5] Carpenter, G.A. and Grossberg, S., Integrating symbolic and neural processing in a self-organizing architecture for pattern recognition and prediction. In V. Honavar and L. Uhr (Eds.), Artificial intelligence and neural networks: Steps towards principled prediction. San Diego: Academic Press, pp. 387-421, 1994.

[6] Carpenter, G.A., Milenova, B.L., Noeske, B.W., Distributed ARTMAP: a neural network for fast distributed supervised learning, Neural Networks vol. 11, pp. 793–813, 1998.

[7] Catterall, E., Van Laerhoven, K., and Strohbach, M., Self-Organization in Ad Hoc Sensor Networks: An Empirical Study, in Proc. of Artificial Life VIII, The 8th Int. Conf. on the Simulation and Synthesis of Living Systems, Sydney, NSW, Australia, 2002.

[8] Ganesan, D., Estrin, D., Heidemann, J., DIMENSIONS: Why do we need a new Data Handling architecture for Sensor Networks?, in Proc. Information Processing in Sensor Networks, Berkeley, California, USA, 2004.

[9] Grossberg, S. Adaptive pattern classification and universal recoding: I. parallel development and coding of neural feature detectors, Biological Cybernetics, vol. 23, pp. 121-134, 1976.

[10] Grossberg, S. Adaptive Resonance Theory in Encyclopedia Of Cognitive Science, Macmillan Reference Ltd, 2000.

[11] Guestrin, C., Bodik, P., Thibaux, R., Paskin M., and Madden, S., Distributed Regression: an Efficient Framework for Modeling Sensor Network Data, in Proceedings of IPSN'04, April 26–27, 2004, Berkeley, California, USA, 2004.

[12] Sapojnikova, E., ART-based Fuzzy Classifiers: ART Fuzzy Networks for Automatic Classification, PhD Thesis, Department of Computer and Cognitive Science at the University of Tübingen, Germany, 2003.

[13] Strang, G. and Nguyen, T. Q., Wavelets and Filter Banks, Wellesley-Cambridge Press, 2nd Ed., 1997.

[14] Wu, J. and Amaratunga, K., "Wavelet Triangulated Irregular Networks", in International Journal of Geographic Information Science, 2003.