INTERNATIONAL JOURNAL OF COMMUNICATIONS
Issue 3, Volume 4, 2010

Anomaly Detection in Wireless Sensor Networks using Self-Organizing Map and Wavelets

S. Siripanadorn, W. Hattagam, N. Teaumroong

***Abstract*—**This paper proposes an anomaly detection scheme which is able to detect anomalies accurately by employing only important features of data signals, instead of using all the sensor data traces. The contribution of this paper centers on anomaly detection by using Discrete Wavelet Transform (DWT) combined with a competitive learning neural network called self-organizing map (SOM) in order to accurately detect abnormal data readings while using just half of the data size. Experiment results from synthetic and real data injected with synthetic faults collected from a WSN show that the proposed algorithm outperforms the SOM algorithm by up to 18% and DWT algorithm by up to 35% in presence of bursty faults with marginal increase of false alarm rate. Furthermore, in the real-world datasets experiments show that our proposed algorithm can maintain acceptable anomaly detection accuracy as well as the SOM algorithm while using just half of the input data.

***Keywords*—**Anomaly Detection, Discrete Wavelet Transform, Self-Organizing Map, Wireless Sensor Networks, Agriculture Monitoring.

I. INTRODUCTION

Wireless sensor networks (WSNs) have been recently deployed in many areas of agriculture to increase yield and prevent outbreaks such as in hydroponics and paddy fields, fertilizer composting process, and livestock monitoring. However, these applications rely mainly on manually measuring and controlling the parameters such as moisture, temperature, pH, oxygen, soil nutrients, etc., which are both time consuming and laborious. Autonomous monitoring devices such as WSNs therefore warrant potential use in agriculture monitoring.

A WSN is a wireless network that consists of distributed autonomous devices using sensors to cooperatively monitor or collect environmental conditions at different locations. Several measurements can be collected from the WSN. The collected measurements from the WSN may be affected by anomalies in the sensor network. With the huge amount of data continually collected from the WSN, it becomes increasingly difficult to detect anomalies in the data measurements. Therefore, anomaly detection techniques are necessary to automatically detect faults and alert the system controller to take suitable action.

Research emphasizing on anomaly detection in communication networks has progressed in recent years,e.g., in network traffic [1], [2], [3], in IP networks [4], in cellular mobile networks [5]. In general anomaly detection refers to the problem of finding patterns in data that do not conform to expected behavior [1]. Abnormal data patterns can be caused by faulty sensors in the network or unusual phenomena in the monitored domain.

Anomalies caused by faulty sensor communications are presented in [6]. They proposed a distributed algorithm for detecting and isolating faulty sensor nodes in WSNs. Each sensor node identifies its own status based on local comparisons of sensed data with thresholds. Ref. [7] applied 4 different anomaly detection techniques for different types of faults obtained in the real–world datasets, namely, NAMOS [8], INTEL [9] and SensorScope [10]. They classified these faults into 3 types, i.e., noise faults, short faults and constant faults. This research suggested that there is presently no known anomaly detection method suitable for every type of faults.

Another application of anomaly detection is an unusual phenomenon in the monitored domain. Erroneous measurements may occur as a result of transducers drifting out of calibration, or from faults introduced by harsh environmental conditions. In a large network it is extremely difficult and time consuming to detect these erroneous measurements manually. In addition, energy is wasted in the network when forwarding the unwanted erroneous measurements to the base station for analysis. One solution to alleviate network energy consumption is to reduce the amount of data that needs to be communicated through the network. Energy is critical in WSNs, therefore anomaly detection methods in WSN must not only perform well but also demand low energy consumption. Distributed in-network processing can reduce transmission energy and eventually help prolong the overall network lifetime of the WSN [11]. Our work is motivated by this concept. In particular, we focus on reducing the amount of transmitted data by in-network processing for anomaly detection at the base station.

This paper considers anomalies caused by unusual phenomenon and faulty sensors. To detect these anomalies, a dynamic data classification scheme such as data mining method could be useful.

Data mining is an expanding area of research in artificial neural network and information management whose objective is to extract relevant information from large databases. One particular method, called the self-organizing map (SOM), has several beneficial features which make it a useful tool in data

mining. In particular, it follows the probability density function of the data and is, thus, an efficient clustering and quantization algorithm. The most important feature of the SOM in data mining is the visualization property [12].

SOM has been applied for anomaly detection in communication networks [13], [14], [15] as well as WSNs [16]. Ref. [16] focuses on evaluating the position of sensors in a WSN, or the localization problem. Their localization technique is based on a simple SOM, implemented on each sensor node. The main advantages of their solution are the limited storage and computing costs. However, SOM requires processing time which increases with the size of input data. To reduce the input data size, features of the data can be extracted without losing the significant data can be used for anomaly detection. This can be achieved by the Discrete Wavelet Transform (DWT). Wavelets have been extensively employed for anomaly [17] and fault detection [18]. DWT has also been integrated with SOM to detect system faults [19], [20]. In particular, feature vectors of the faults have been constructed using DWT, sliding windows and a statistical analysis. Classification of the feature vectors was obtained by using SOM.

To the best of our knowledge, DWT and SOM have not yet been applied for anomaly detection in WSNs. Therefore, the underlying aim of this paper is to propose an anomaly detection algorithm which determines the discrete wavelet transform, and detects the abnormality of the sensor readings by training the SOM using the wavelet coefficients. Our proposed algorithm, the integrated SOM and DWT algorithm, could help reduce wasted energy caused by transmitting all measurement data to the base station by applying DWT algorithm onto the sensor modes in order to reduce size of transmitted data without losing the significant feature of the data.

II. ANOMALY DETECTION

The first step of anomaly detection involves selecting the data parameters to be monitored and grouping them together in a pattern vector $\mathbf{x}^{\mu} \in \Re$, $\mu = 1, \ldots, N$,

$$x^{\mu} = \begin{pmatrix} x_1^{\mu} \\ x_2^{\mu} \\ \vdots \\ x_n^{\mu} \end{pmatrix} = \begin{pmatrix} \mathrm{KPI}_1^{\mu} \\ \mathrm{KPI}_2^{\mu} \\ \vdots \\ \mathrm{KPI}_n^{\mu} \end{pmatrix}, \quad (1)$$

where μ is the observation index, n is the number of parameter types or key performance indices (KPIs) chosen to monitor the environmental condition.

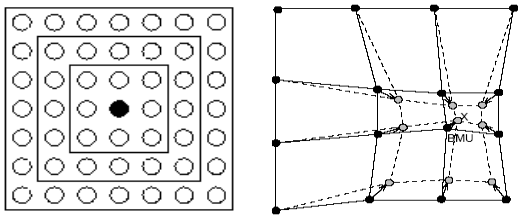
A. Self-Organizing Map

Competitive neural models such as the self-organizing map (SOM) [13], are able to extract statistical regularities from the input data vectors and encode them in the weights without supervision. It maps a high-dimensional data manifold onto a low-dimensional, usually two-dimensional, grid or display.

The basic SOM consists of a regular grid of map units or neurons as shown in Fig 1(a). Each neuron, denoted by i (depicted by the black dot), has a set of layered neighboring neurons (depicted by the white dots) as shown in Fig 1(a).

Neuron i maintains a weight vector $\mathbf{m}_i$. In order to follow the properties of the input data, such vector is updated during the training process. For example, Fig.1(b) shows a SOM represented by a 2-dimensional grid of 4×4 neurons. The dimension of each vector is equal to the dimension of the input data. In the figure, a vector of input data (marked by x) is used to train the SOM weight vectors (the black dots). The winning neuron (marked by BMU) as well as its 1-neighborhood neurons, adjust their corresponding vectors to the new values (marked by the gray dots).

The SOM is trained iteratively. In each training step, one sample vector $\mathbf{x}$ from the input data set is chosen.



(a) (b)

Fig. 1 An illustration of the SOM (a) with rectangular lattice neighbors belonging to the innermost neuron (black dot) corresponding to 1, 2 and 3- neighborhoods, (b) SOM updates the BMU with 1-neighborhood.

The distances between the sample data and all of weight vectors in the SOM are calculated using some distance measure. Suppose that at iteration t, neuron i whose weight vector $\mathbf{m}_i(t)$ is the closest to the input vector $\mathbf{x}(t)$. We denote such weight vector by $\mathbf{m}_c(t)$ and refer to it as the Best-Matching Unit (BMU), that is

$$\|x(t) - m_c(t)\| = \arg\min_{\forall i} \|\mathbf{x}(t) - \mathbf{m}_i(t)\| \quad (2)$$

where $\|\cdot\|$ is the Euclidian distance.

Suppose neuron i is to be updated, the SOM updating rule for the weight vector of neuron i is given by

$$m_i(t+1) = m_i(t) + \eta_t h_c(i,t)[\mathbf{x}(t) - \mathbf{m}_i(t)] \quad (3)$$

where t is the iteration index, $\mathbf{x}(t)$ is an input vector, η_t is the learning rate, $h_c(i,t)$ is the neighborhood function of the algorithm. The Gaussian neighborhood function may be used, that is

$$h_c(i,t) = \exp\left(-\frac{\|r_c(t) - r_i(t)\|^2}{2\sigma^2(t)}\right) \quad (4)$$

where $r_i(t)$ and $r_c(t)$ are the positions of neurons i and the BMU, c respectively, and $\sigma(t)$ is the radius of the neighborhood function at time t. Note that $h_c(i,t)$ defines the width of the neighborhood. It is necessary that $\lim_{t\to\infty} h_c(i,t) = 0$ and $\lim_{t\to\infty} \eta_t = 0$ for the algorithm to converge [13].

B. Discrete Wavelet Transform

DWT is a mathematical transform that separates the data signal into fine-scale information known as detail coefficients,

and rough-scale information known as approximate coefficients. Its major advantage is the multi-resolution representation and time-frequency localization property for signals. Usually, the sketch of the original time series can be recovered using only the low-pass-cut off decomposition coefficients; the details can be modeled from the middle-level decomposition coefficients; the rest is usually regarded as noises or irregularities. The following equations describe the computation of the DWT decomposition process:

$$a_{j+1}^{DWT}(k) = \sum_n h_0(n-2k)a_j^{DWT}(k) \quad (5)$$

$$d_{j+1}^{DWT}(k) = \sum_n g_0(n-2k)a_j^{DWT}(k) \ , \quad (6)$$

where the rough-scale (or approximation) coefficients a_j^{DWT} are convolved separately with h_0 and g_0, the wavelet function and scaling function, respectively, *n* is the time scaling index, *k* is the frequency translation index for wavelet level *j*. The resulting coefficient is down-sampled by 2. This process splits a_j^{DWT} roughly in half, partitioning it into a set of fine-scale (or *detail)* coefficients d_{j+1}^{DWT} and a coarser set of approximation coefficients a_{j+1}^{DWT} [21].

DWT has the capability to encode the finer resolution of the original time series with its hierarchical coefficients. Furthermore, DWT can be computed efficiently in linear time, which is important while dealing with large datasets.

C. Integration of SOM and DWT

In the integration of SOM and DWT algorithm, the DWT algorithm is used as an input data preprocessor of the SOM algorithm in order to reduce the size of data without losing any significant feature of the data. This enables the implementation of in-network processing which helps to reduce the radio communication energy and eventually prolong the lifetime of the WSN [11]. The input data will be padded with zero if its length is odd data. After obtaining the wavelet coefficients, these coefficients will be fed to the SOM algorithm which can be divided into 2 sets. Each set contains both approximate and detail coefficients. The first set which is obtained from noiseless data, will be used to train the SOM algorithm. The second set which is obtained from the faulty data will be used to test the SOM algorithm. Then to reduce the false alarms the detected results will be double checked by using the univariate method [13], [14].

D. Anomaly Detection

A new observation data set can be considered abnormal if the distance between the weight vector of the winning neuron and the new state vector, given by

$$e^{\mu} = \left\| \mathbf{x}^{new} - \mathbf{m}_c^{\mu} \right\| \quad (7)$$

is greater than a certain percentage $p = 1-\alpha$ of the distances in the distance distribution profile. That is,

$$\begin{aligned} &\text{IF} \quad e^{\mu} \in \left[e_p^-, e_p^+ \right], \\ &\text{THEN } \mathbf{x}^{new} \text{ is NORMAL} \\ &\text{ELSE } \mathbf{x}^{new} \text{ is ABNORMAL.} \end{aligned} \quad (8)$$

Equation (8) is referred to as the global decision. In [6], an addition of local decisions of each KPIs is presented. Suppose that a data vector $\mathbf{x}^{new}$ is considered abnormal by the global decision. Then in the local anomaly detection, the absolute value of error in each component of the error vector is then computed by

$$\left| \mathbf{E}^{new} \right| = \begin{pmatrix} \left| x_1^{\mu} - m_{c,1}^{\mu} \right| \\ \left| x_2^{\mu} - m_{c,2}^{\mu} \right| \\ \vdots \\ \left| x_n^{\mu} - m_{c,n}^{\mu} \right| \end{pmatrix}. \quad (9)$$

The error in each KPI is then compared to the interval of normality component-by-component, and the anomaly decision is carried out as in (8).

III. Experiment Results

A. Evaluation on detecting synthetic faults

In this section, we evaluated the performance of the proposed integration of SOM and DWT algorithm by detecting anomalies in series of synthetic data and actual data collected from a wireless sensor network injected by various synthetic faults.

In the experiment, we generated the synthetic input data from a normal distribution N(0,1) and synthetic faults by additive white Gaussian noise (AWGN) with power 25 dBW generated from MATLAB. We used such fault because its statistical similarity to the synthetic input data thus, it is more difficult to be detected. Therefore, we can evaluate the performance of the algorithms under ambiguous faults. The amount of faults is represented by the notation n/s, where "n" is the amount of faults per series and "s" is the amount of series of faults, resulting in the total amount of n×s faults. The generated faults added to the input data ranged from bursty (20/10) to sparse (1/10). The exact positions of the faults injected in the input data were predetermined and was later used to detect true and false alarms. In the experiment using real data, we have chosen 2 parameters, namely temperature and moisture, as KPIs collected from samples of compost in a bioorganic fertilizer plant. In this paper, the data of the 2 KPIs at the WSNs were collected every 5 minutes for 3 days. We compared 3 anomaly detection methods: SOM algorithm, DWT algorithm, and integration of SOM and DWT algorithm.

We measured 2 performance metrics: 1) the *true alarm rate* which is defined by the number of detected true anomalies over the total number of true anomalies in the data set; and 2) the *false alarm rate* which is defined by the number of detected false alarms over the total number of detected anomalies.

In the DWT algorithm, we used the threshold in (11) in order to decide whether the data is normal or abnormal

$$\sigma_w = median(|d_1 - \overline{d_1}|) \quad (10)$$

$$T_w = \sigma_w \sqrt{2\log_e(N)}, \quad (11)$$

where N is the size of data and $\overline{d}_1$ is the sample mean of the level 1 detail coefficients [21].

This threshold was calculated from the low pass and high pass coefficients from the assumed normal data by using Haar and Daubechies4 mother wavelets. The Haar and Daubechies4 wavelets were used because they are relatively easy to cross-check by hand with computed coefficients from MATLAB program. Hence, we can compare the position of each coefficient with the actual fault position. After the threshold calculation, the set of coefficients which are obtained from the DWT of the noisy data will be compared with the threshold, coefficient by coefficient. For the real data scenario, the data was normalized by equation (12) before being processed by the DWT to eliminate potential outliers:

$$\text{Norm(Data)} = \frac{\text{(Data)} - \text{mean(Data)}}{\text{variance(Data)}} \quad (12)$$

If the absolute value of the coefficient is greater than the computed threshold, an anomaly is said to be detected.

In the SOM algorithm and the proposed integrated SOM and DWT algorithm, the initial value for learning rate in the SOM part was set to $\eta_0 = 0.9$, and gradually reduced to $\eta_T = 10^{-5}$, in order to guarantee convergence [13]. The number of training epochs was set to 50 because longer training epochs tend to over train the SOM [13]. The required percentage of distance in (8) was set to 99%. We used a Gaussian neighborhood function because the distribution of the collected data after the normalization fits well to the Gaussian distribution. The 30×30 size of neurons was used. Fig. 2 and 3 show that the anomaly detection in SOM algorithm and the integrated SOM and DWT algorithms improve as the number of neurons is increased. This suggests that the more neurons used, the "finer" SOM's classification becomes resulting in enhanced detection performance. However, at neuron size 50×50, the SOM requires much longer training time with a marginal improvement in the detection performance. Therefore, the 30×30 size of neurons was selected to train and test the SOM. We also improved the SOM algorithm by double checking with the univariate method in order to reduce the false alarm rate [13], [14]. To obtain accurate results, each algorithm was repeated for 70 runs.

To evaluate the performance of all algorithms, the results of each algorithm were compared to the (known) fault positions which were injected into the input data. In particular, when an anomaly was detected then its position was compared with the (known) fault position. If this position existed, then the anomaly detected was a true alarm; otherwise, it was a miss. On the other hand, if an anomaly was detected but the (known) fault position did not exist, then the anomaly was a false alarm.

Fig. 4 and Fig. 5 show the percentage of true alarm rate averaged over 70 runs, as a function of the amount of faults added into the input data. Note that the proposed integrated SOM and DWT algorithm which used Haar as a mother wavelet gives the best performance over other algorithms. This is because the DWT with Haar wavelet can detect changing points. In particular, the Haar wavelet uses 2 adjacent input data to compute a coefficient whereas the Daubechies4 uses 4 adjacent input data to compute a coefficient. However, Daubechies4 gave a lower performance than Haar because each coefficient was computed from an average over 4 input data. If a fault occurred in 1 of these 4 data, such fault will be averaged with the remaining 3 normal data resulting in a coefficient with an absolute value possibly lower than the decision threshold. Consequently, the true alarm rate is reduced. On the other hand, the Haar wavelet only uses 2 adjacent data to compute 1 coefficient. Thus, the true alarm rate is significantly higher than that of Daubechies4. The integrated SOM and DWT algorithm using Haar also outperforms the SOM algorithm. This is because in the Haar case, the coefficients obtained were transformed from two adjacent data. Therefore, if some data was faulty or differed greatly from the data nearby, this coefficient can detect such anomaly. On the other hand, the SOM algorithm directly checked the data one by one to detect an anomaly. If the data was faulty but had a small magnitude, then this fault may not be detected, and consequently the true alarm rate was reduced. Note that the DWT algorithm has the lowest performance because the decision threshold in (11) is rather conservative. Furthermore, the threshold is fixed throughout the detection and the algorithm does not have any double checking method.

Fig.4 and 5 show that the proposed algorithm can achieve up to 65% and 67% of true alarm rates in case of bursty faults for synthetic and real data, respectively. The proposed algorithm achieved a true alarm rate of up to 18% higher than the SOM algorithm alone in presence of bursty faults. Compared to the DWT alone, the proposed algorithm can attain a true alarm rate of up to 35% more in the bursty faults case.

As for sparse faults, the proposed algorithm can achieve up to 69% and 80% true alarm rates for synthetic and real data, respectively. The integrated SOM and DWT also gave true alarm rates of up to 10% higher than the SOM algorithm alone whereas DWT performed the weakest, in presence of sparse faults.

Fig.6 and Fig. 7 show the false alarm rate results in the synthetic and real data experiments, respectively. Note that most results have low false alarm rates, i.e., less than 1 % except in the case of sparse faults due to the increased detection difficulty.

The integration of SOM and Daubechies4 DWT also gave a weak performance due to the reasons previously explained. All these results show that the integration of SOM and DWT with Haar as a mother wavelet outperform the SOM algorithm and DWT method.

From these figures, the false alarm rate of the proposed algorithm is 0.11% and 0.13% in presence of bursty faults and 0.91% and 1% in presence of sparse faults with synthetic and real data, respectively. Note that the false alarm rate of the proposed algorithm is slightly higher than the other two algorithms. Since the gain in the true alarm rate is more significant, such tradeoff is therefore considered acceptable.

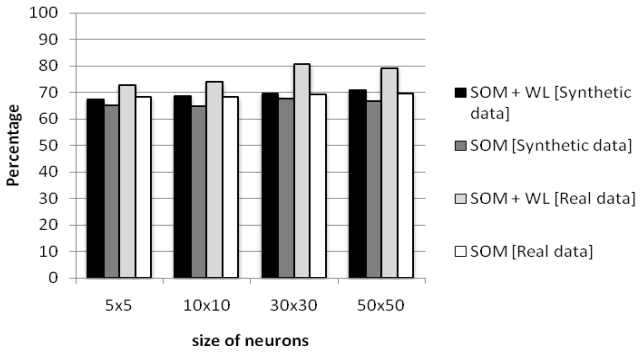


Fig. 2 True alarm rates with different size of neurons in the sparse faults case

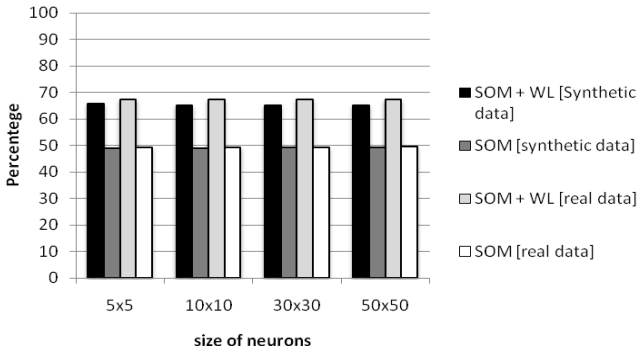


Fig. 3 True alarm rates with different size of neurons in the bursty faults case

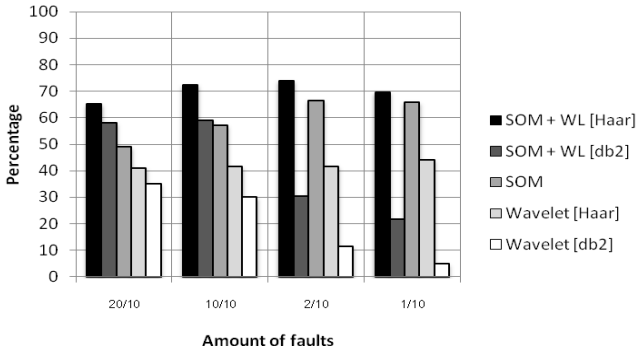


Fig. 4 True alarm rates with synthetic data

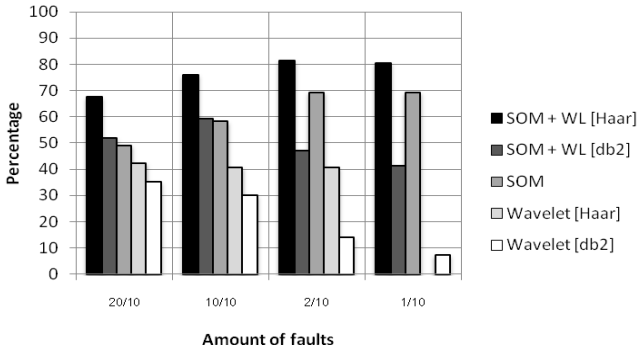


Fig. 5 True alarm rates with real data

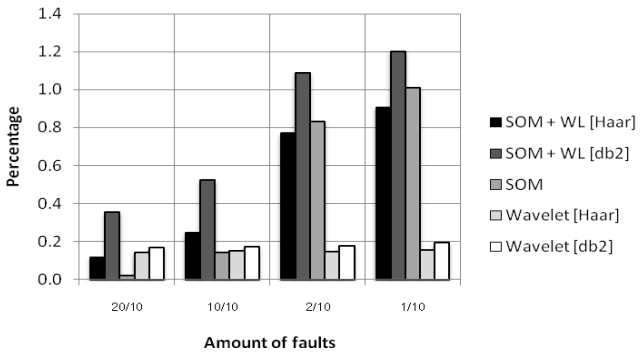


Fig. 6 False alarm rates with synthetic data

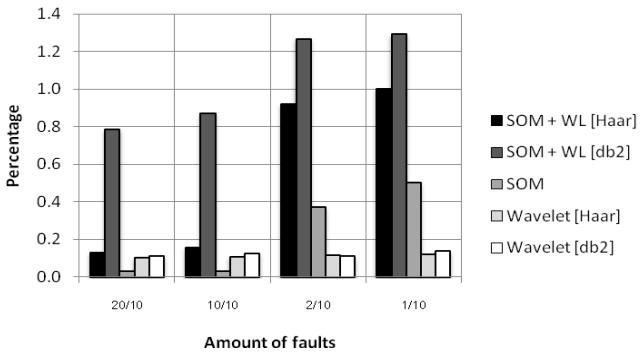


Fig. 7 False alarm rates with real data

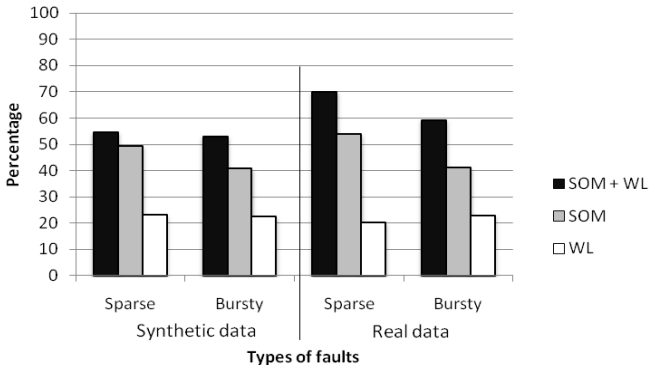


Fig. 8 True alarm rate with 10 dBW AWGN faults

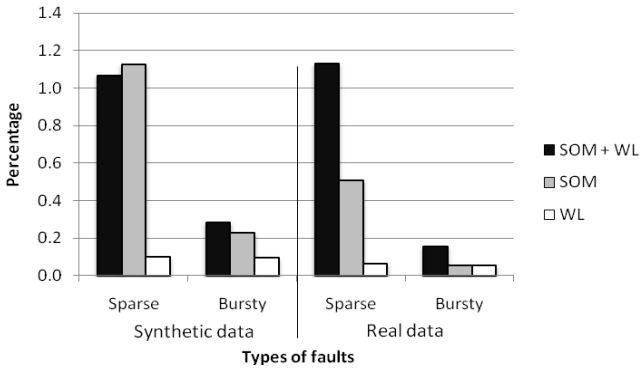


Fig. 9 False alarm rate with 10 dBW AWGN faults

Fig. 8 and 9 show the effect of the decreasing of AWGN noise power from 25dBW to 10dBW in both synthetic and real data scenarios. Only the Haar wavelet was used in the proposed algorithm and the DWT algorithm. The Daubechies4 was not included due to its weak performance. Though the anomaly detection is more difficult, the proposed integrated SOM and DWT still consistently outperforms the other two methods in terms of true alarm rate but with marginal increase in the false alarm rate as tradeoff.

The proposed integration of SOM and DWT algorithm with Haar wavelet outperformed the SOM algorithm and the DWT algorithm alone. Our results suggest that the proposed integrated SOM and DWT anomaly detection scheme can be deployed in a resource-constrained network such as a WSN. In particular, the DWT using Haar wavelet can be implemented at the sensor nodes as a data preprocessor to reduce the amount of data to be transmitted by at least half (for one-level DWT). Since energy consumption is critical in WSNs, such distributed in-network processing can reduce transmission energy and eventually help prolong the overall network lifetime of the WSN [11] while still maintaining acceptable anomaly detection accuracy.

B. *Evaluation on detecting faults in real-world datasets*

In this section, we apply the anomaly detection methods to three real-world datasets, i.e., NAMOS [8], INTEL Berkeley Lab [9], and SensorScope [10], to detect anomalies in sensor traces. However, since we did not have ground truth information about faults for these datasets, visual inspection and the histogram method are used to decide whether the data is normal or abnormal. The histogram method was used because it displays the data distribution which allows us to determine a suitable threshold according to that data series.

The histogram method divides the time series of sensor readings into groups of *N* samples. We then plot the histogram of the samples and select a threshold according to outliers of the histogram. However, this approach is sensitive to the choice of *N*. Fig. 10 [10] shows the effect of *N* on the histogram computed for sensor measurements taken from a real-world deployment [7]. Therefore, selecting the correct value for the parameter *N* requires a good understanding of the *normal* sensor readings. In practice, one should also try a range of values for *N* to ensure that the samples flagged as faulty are not just an artifact of the value selected for *N* [7]. With heuristic adjustments on the parameter value of N and some domain knowledge of the normal data profile, the histogram method was used as reference to identify abnormal data samples.

In the real-world datasets experiment, we evaluated the performance of 3 anomaly detection methods: the SOM, DWT using the Haar wavelet methods, and the integration of SOM and DWT using the Haar wavelet. For the SOM and the integration of SOM and DWT using Haar wavelet algorithms, we also considered the effects of changing the number of training samples, the number of training epochs which were 10 and 50 iterations, and the size of neurons which were 10x10 and 30x30. We also compared the performance of the low and high pass Haar wavelet coefficients (LP and HP, respectively) in the DWT algorithm and the integration of SOM and DWT algorithm.

1) *NAMOS*

In the NAMOS dataset, 9 buoys with temperature and chlorophyll concentration sensors (fluorimeters) were deployed in Lake Fulmor, James Reserve for over 24 hours in August, 2006 [8]. We analyzed the measurements from chlorophyll sensors on buoys no. 103 for 10^4 samples as shown in Fig.11. In the experiment, the histogram method was used to identify anomalies in the NAMOS dataset from which we selected the threshold of 0 and 500 as lower and upper bounds of the normal region, respectively. The size of training samples of 1500 and 3000 samples were used to train both the SOM and the integration of SOM and DWT algorithms.

Fig. 12 shows the percentage of detection alarm rates for true, miss and false alarms which were obtained from changing the size of training samples. Note that both the SOM algorithm and the proposed integrated SOM and DWT algorithm with low pass wavelet coefficients gave the best true alarm detection performance of nearly 100% while their false alarm rates is negligible. The integrated SOM and DWT algorithm and DWT algorithm with high pass coefficients gave the lowest performance. This is because the high pass coefficients are more suitable for detecting the changing points of the data whereas most of faults appear constant as seen from $9x10^3$ samples onwards in Fig. 11. In addition, reducing the size of training samples did not have any effect on the anomaly detection in the SOM algorithm and the proposed integrated SOM and DWT algorithm. This is because both training samples are obtained from a normal period of data which differ only in sample sizes.

Fig. 13 shows the percentage of detection alarm rates for true, miss and false alarms which were obtained by reducing the number of training epoch from 50 to 10 iterations. In this case, the SOM algorithm gave the best performance with nearly 100% of true alarm detection rate and no false alarm rate. DWT algorithm which used low pass coefficient gave high performance while the proposed integrated SOM and DWT algorithm with either coefficient failed on detecting any anomaly. The reason could be caused by the constant features of the faults in NAMOS which may be difficult to decide whether samples are normal or abnormal, in particular, if the wavelet coefficients are under trained. Hence, care must be taken when selecting the suitable number of training epochs. In addition, we also investigated the effect of reducing the size of neurons. Results in Fig.14 show that there is no significant effect from reducing size of neurons from 30x30 to 10x10.

2) *INTEL*

In the INTEL dataset, 54 Mica2Dot motes with temperature, humidity and light sensors were deployed in the Intel Berkeley Research Lab between February 28th and April 5th, 2004 [9]. In this paper, we present the results on the anomaly detection in the temperature readings.

In the experiment, we selected the threshold value of 16 and 30 as the upper and lower bounds of the normal data regions. These values were obtained from the histogram method. The size of training samples used was 1000 and 2000 samples as shown in Fig. 15.

Fig. 16, shows the percentage of detection alarm rates for true, miss and false alarms which were obtained from changing the size of training samples. According to the results as shown, the SOM and the proposed algorithm can achieve a true alarm rate of up to 100% with very small false alarm rate. Their true alarm rate is 67% higher than the DWT method using high pass coefficients. Note that the high pass coefficients can detect spike faults better than low pass coefficient since the high pass coefficients reflect the rate of change between two successive samples. Note that the DWT using low pass coefficient gave the lowest performance. The results of changing number of training epochs are shown in Fig. 17 and the size of neurons are shown in Fig. 18. From both figures there are no significant effects on the detection rate because the fault in this dataset has a high amplitude and can be easily detected.

3) SensorScope

The SensorScope project is an ongoing outdoor sensor network deployment consisting of weather-stations with sensors for sensing several environmental quantities such as temperature, humidity, solar radiation, soil moisture, and so on [10]. We did not have the ground truth regarding faulty samples for this dataset. We used a combination of visual inspection and the histogram method to identify anomaly samples [7].

In the experiment, we present the results on the anomaly detection in two types (KPIs) of data in the pdg2008-metro-1 dataset, i.e., the surface and ambient temperature readings. Using visual inspection and the histogram method, the lower and upper threshold values used for anomaly detection in SensorScope were -14 and 4 for the surface temperature and -12 and 4 for the ambient temperature. The sizes of training samples were 700 and 2000 samples for both KPIs as shown in Fig. 19.

Fig. 20 shows the percentage of detection alarm rates for true, miss and false alarms obtained from changing the size of training samples. Note that the proposed algorithm using low pass coefficients achieved a true alarm rate 2% higher than the SOM algorithm while false alarm rate remained less than 0.5%. The proposed algorithm using low pass coefficients can attain a true alarm rate of up to 17% more than the DWT algorithm alone. The integrated SOM and DWT algorithm and DWT algorithm which used high pass coefficients gave the lowest performance. This is because high pass coefficients are more suitable for short duration faults such as, spike or sparse faults while the data in Fig. 19 contains noise faults which affect a larger number of successive samples with an increase in their variance.

The effect of reducing the number of training epochs is shown in Fig. 21. According to the results, there is no significant effect on the performance of SOM and the integrated SOM and DWT.

Fig. 22 shows the percentage of detection alarm rates for true, miss and false alarms which were obtained from reducing the size of neurons. Note that the proposed algorithm using low pass coefficients achieved a true alarm rate 2% lower than the SOM algorithm, whereas the false alarm rate remains lower than 0.5%. On the other hand, the proposed algorithm using low pass coefficients can attain a true alarm rate of up to 13% more than the DWT algorithm alone.

The results from the real-world dataset show that our proposed algorithm, the integrated SOM and DWT algorithm performs as equally well as the SOM algorithm while using just half of the input data (using level 1 of DWT). This is because DWT is able to extract relevant data features without any significant loss in information, thereby reducing wasted energy from transmitting all measurements to the base station. Hence, by applying DWT onto the sensor modes, to achieve in-network data processing, the size of transmitted data can be reduced while still maintain good anomaly detection abilities.

However, a variety of data characteristics can affect the anomaly detection in the integrated SOM and DWT algorithm as can be seen from the NAMOS dataset. Hence, a suitable setting of the algorithm, such as the size of training epochs, has to be considered carefully. In terms of the number of neurons, the more neurons used, the finer SOM's classification becomes, generally resulting in enhanced detection performance. However, the results in the real-world datasets show that there is no significant change in detection performance. In terms of the selection of wavelet coefficients, high pass coefficients are more suitable for detecting the changing points of the data, whereas low pass coefficients are more suitable for detecting the changing of trend of the data. These settings can be determined by considering the nature of the sensors deployed. For example, calibration errors in sensors can cause offset faults (whereby the measured value can differ from the true value by a constant), low battery voltage causes a combination of noise and constant faults, while short faults are caused by software error during communication and data logging [7].

IV. CONCLUSION

This paper proposed an integration of a competitive learning method called the self-organizing map (SOM) and the discrete wavelet transform (DWT), to detect anomalies from synthetic faults and faults obtained from real-world datasets.

In the synthetic faults experiment, the results show that the integration of SOM and DWT with Haar as a mother wavelet can attain 65% and 67% of true alarm rates in the case of bursty faults, and 69% and 80% of true alarm rates in case of sparse faults for synthetic and real data, respectively. In terms of the true alarm rate, the proposed algorithm outperforms the SOM algorithm by up to 18% and DWT algorithm by up to 35% in presence of bursty faults. With sparse faults, the proposed algorithm can gain a true alarm rate up to 10% above the SOM algorithm alone and entirely outperforms the DWT algorithm alone. Such gain in true alarm rates come with a marginal increase of false alarm rate.

In the real-world datasets, the integration of SOM and DWT with Haar as a mother wavelet can attain up to 99%, 100% and 83% of true alarm rates in the NAMOS, INTEL and SensorScope dataset, respectively. Our proposed algorithm also performs as equally well as the SOM algorithm and outperforms the DWT algorithm by up to 15%, 100%, and 17% in the NAMOS, INTEL and SensorScope dataset, respectively.

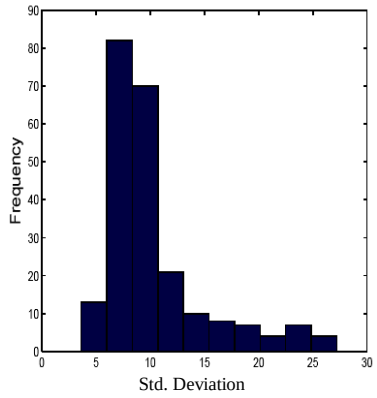


Fig. 10a Histogram shape with N = 100

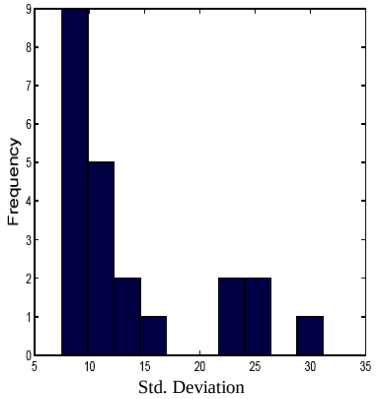


Fig. 10b Histogram shape with N = 1000

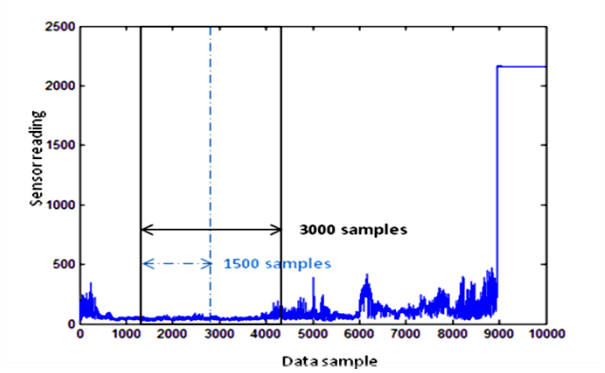


Fig. 11 NAMOS dataset of 10^4 samples

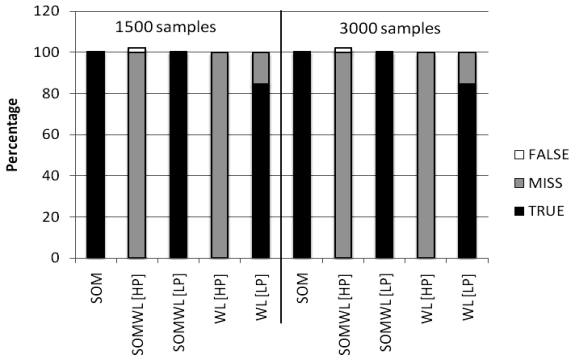


Fig. 12 Detection rate in the NAMOS dataset using training epoch of 50 iterations

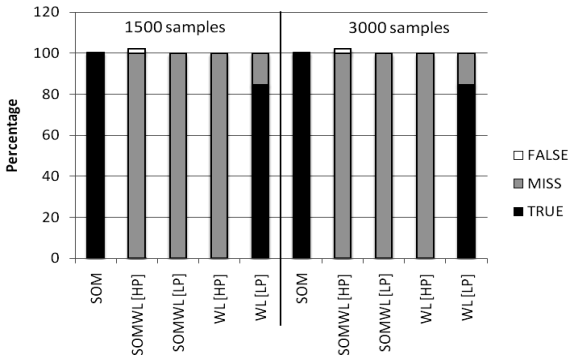


Fig. 13 Detection rate in the NAMOS data set using training epoch of 10 iterations

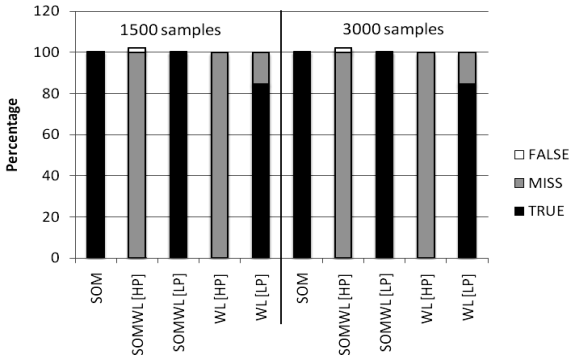


Fig. 14 Detection rate in the NAMOS dataset using the size of neurons of 10x10

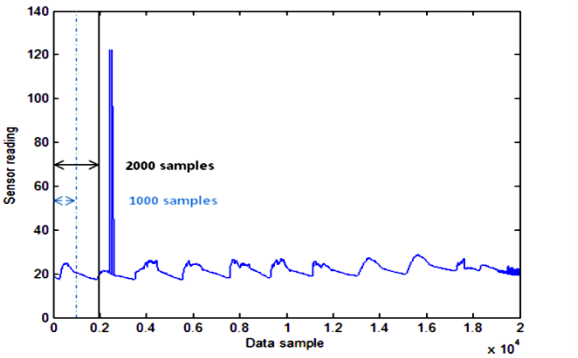


Fig. 15 INTEL dataset of $2x10^4$ samples

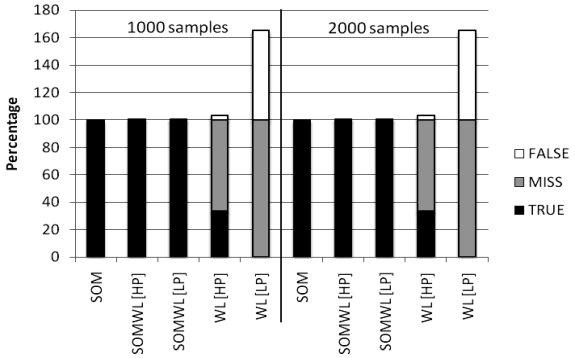


Fig. 16 Detection rate in the INTEL dataset using training epoch of 50 iterations

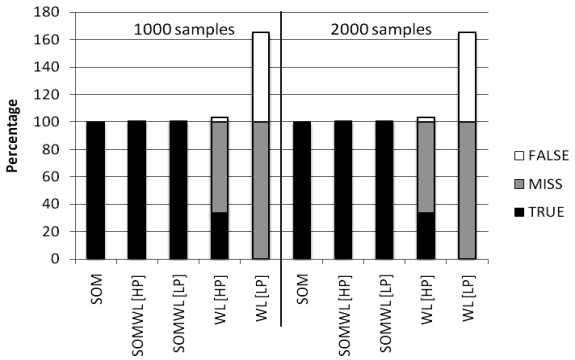


Fig. 17 Detection rate in the INTEL dataset using training epoch of 10 iterations

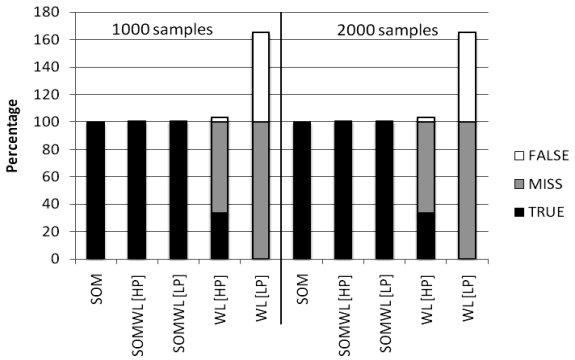


Fig. 18 Detection rate in the INTEL dataset using the size of neurons of 10x10

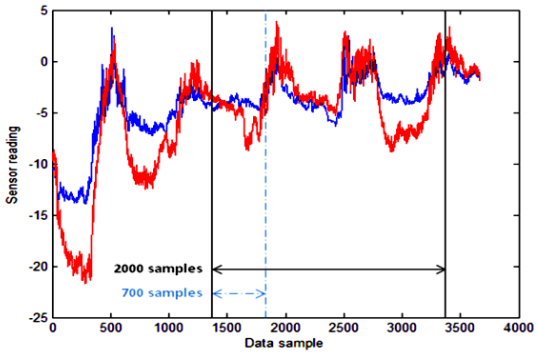


Fig. 19 SensorScope dataset of 4000 samples

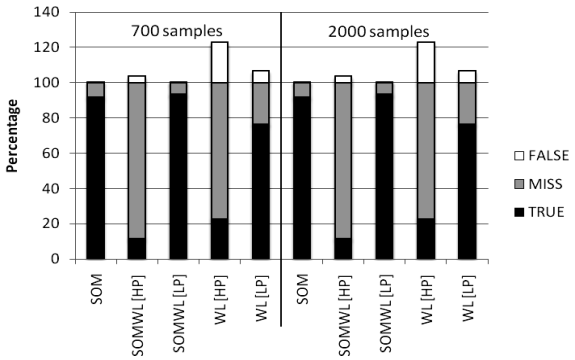


Fig. 20 Detection rate in the SensorScope dataset using training epoch of 50 iterations

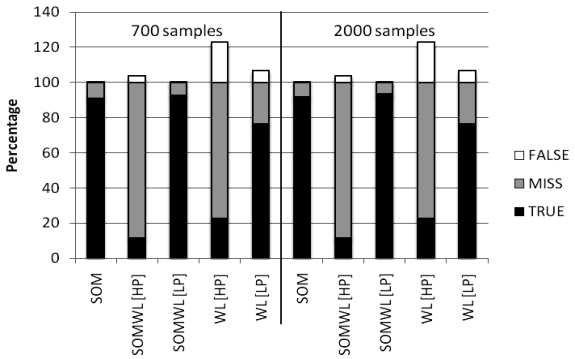


Fig. 21 Detection rate in the SensorScope dataset using training epoch of 10 iterations

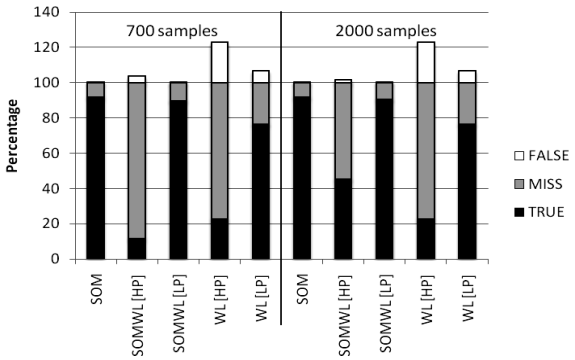


Fig. 22 Detection rate in the SensorScope dataset using the size of neurons of 10x10

In terms of the true alarm rate when reducing the number of training epochs, the proposed algorithm has a poor performance due to the detection ability of wavelet coefficients is unsuitable for the anomaly in the NAMOS dataset. In the INTEL dataset, the proposed algorithm outperforms the DWT algorithm and performs equally well when compared to the SOM algorithm while using just half of the input data. In the SensorScope dataset, the proposed algorithm outperforms the DWT algorithm but is slightly lower than the SOM algorithm.

By reducing the size of neurons, the proposed algorithm still obtained a true alarm rate up to 16%, 100% and 84% higher than the DWT algorithm in NAMOS, INTEL and SensorScope dataset, respectively. The proposed algorithm performed equally well as the SOM algorithm in the NAMOS and INTEL dataset but only 2% lower than the SOM algorithm in the SensorScope dataset. The reduction of the size of neurons did not show any significant change in detection performance.

Our results suggest that the integration of SOM and DWT with Haar wavelet can lead to more effective anomaly detection. In particular, our results confirm that the proposed algorithm can maintain acceptable anomaly detection accuracy while using just half of the input data (using level 1 DWT).

In the future, we plan to extend our work to investigate anomaly detection with actual faults obtained from the bioorganic fertilizer plant environment, and study its performance by increasing the DWT level and considering other different types of wavelets. Furthermore, we also plan to investigate ways to identify and eliminate erroneous sensor readings at the sensor nodes, which could help further reduce wasted energy from transmitting unwanted erroneous measurements to the base station.

V. Acknowledgment

The authors would like to thank Assoc. Prof. Dr. Boon Hee Soong and Mr. Edwin Chan of Intelligent Systems Laboratory, Nanyang Technological University, Singapore for their assistance and suggestions on the sensor development and implementation.

References

[1] G. Kaur, V. Saxena and J.P. Gupta, "Anomaly Detection in Network Traffic and Role of Wavelets," *IEEE Trans. on Instrumentation and Measurement*, vol. 7, no. 5, pp. 46-51, April. 2010.

[2] D.E. Dening, "An Intrusion-Detection Model," *IEEE Trans. on Software Engineering*, vol. SE-13, vo. 2, pp. 222-232, 1987.

[3] S. Pervez, I. Ahmad, A. Akram and S.U. Swati, "A Comparative Analysis of Artificial Neural Network Technologies in Intrusion Detection Systems," *WSEAS Int. Conf. on Multimedia, Internet & Video Technol*ogies, pp. 84-89, 2006.

[4] M. Thottan and J. Chuanyi, "Anomaly detection in IP network," *IEEE Trans. on Signal Processing*, vol.51, no.8, 2003.

[5] J. Laiho, K. Raivio, P. Lehtimaki, K. Hatonen, and O. Simula, "Advanced Analysis Methods for 3G Cellular Networks," *IEEE Trans. on Neural Networks*, vol.4, no.3, pp. 930-942, 2005.

[6] M.H. Lee and Y.H. Choi, "Fault detection of wireless sensor networks," *Computer Communications*, vol. 31, pp. 3469-3475, 2008.

[7] A.B. Sharma, L. Golubchik, and R. Govindan, "Sensor Faults: Detection Methods and Prevalence in Real-World Datasets," *Trans. on Sensor Networks*, vol.5, pp. 1-34, 2010.

[8] NAMOS. 2006. Networked Aquatic Microbial Observing System. Data set available at: http://robotics.usc.edu/~namos/data/jr aug 06/.

[9] INTEL. 2004. The Intel Lab Data. Data set available at: http://berkeley.intel-research.net/labdata/.

[10] SENSORSCOPE. 2006. The SensorScope Lausanne Urban Canopy Experiment (LUCE) Project. Data set available at: http://sensorscope.epfl.ch/index.php/LUCE.

[11] S. Rajasegarar, C. Leckie, and Palaniswami, "Anomaly Detection in Wireless Sensor Networks," *IEEE Wireless Communications*, vol.15, no.4, pp. 34-40, 2008.

[12] J. Laiho, M. Kylvaja, and A. Hoglund, "Utilization of advanced analysis methods in UMTS networks," *IEEE Vehicular Technology Conf.*, vol. 2, pp. 726-730, May. 2002.

[13] G.A. Barreto, J.C. Mota, L.G. Souza, R.A. Frota, and L. Aguaya, "Condition monitoring of 3G cellular network through," *IEEE Trans. Neural Networks*, vol. 16, no. 5, pp. 1064-1075, Sep. 2006.

[14] P. Sukkhawatchani and W. Usaha, "Performance Evaluation of Anomaly Detection in Cellular Core Networks using Self-Organizing Map," *Proc. of ECTI-CON 2008*, vol.1, pp. 361-364, May. 2008.

[15] J. Zheng and M. Hu, "Detection of TCP Attacks Using SOM with Fast Nearest-Neighbor Search," *WSEAS Int. Conf. on Neural Networks*, pp.176-182, 2005.

[16] L. Paladina, M. Paone, G. Jellamo, and A. Puliafito, "Self organizing maps for distributed localization in wireless sensor networks," *Computers and Communications, 2007*, 12th IEEE Symposiumpp, pp.1113-1118, July. 2007.

[17] V.A. Aquino and J.A. Barria, "Anomaly detection in communication Networks using wavelets," *IEEE Proc. in Commun*ications, vol.148, no.6, pp. 355-362, Dec. 2001.

[18] N. Yadaiah, and Nagireddy Ravi, "Fault detection techniques for power transformers," *Industrial & Commercial Power Systems Technical Conf*., pp. 1- 9, 2007.

[19] Z. Xu and Q. Zhao, "A novel approach to fault detection and isolation based on wavelet analysis and neural network," *Electrical and Computer Engineering*, vol. 1, pp. 572–577, May. 2002.

[20] S. Postalcıoglu, K. Erkan and ED. Bolat, "Implementation of Intelligent Active Fault Tolerant Control System," *Springer-Verlag Berlin Heidelberg 2007*, pp. 804–812.

[21] R. J. Brychta, S. Tuntrakool, M. Appalsamy and D. Robertson, "Wavelet Methods for Spike Detection in Mouse Renal Synpathetic Nerve Activity," *IEEE Trans. Biomedical Engineering*, vol.54, no.1, pp. 82-93, Jan. 2007.