Classification with missing data in a wireless sensor network

YuanYuan Li and Lynne E. Parker
Distributed Intelligence Laboratory, Department of Electrical Engineering and Computer Science
The University of Tennessee, 203 Claxton Complex, Knoxville, Tennessee 37996-3450
Email: {yli,parker}@eecs.utk.edu

Abstract

We have developed a novel method to estimate missing observations in wireless sensor networks. We use a hierarchical unsupervised fuzzy ART neural network to represent the data cluster prototypes and describe missing input patterns based on the network. We then estimate missing inputs by a spatial-temporal imputation technique. Our experimental results show that our proposed approach performs better than nine other missing data imputation techniques including moving average and Expectation-Maximization (EM) imputation.

1. Introduction

An important challenge in Wireless Sensor Networks (WSNs) research is to determine a systematic procedure to train these networks so that they are sensitive only to "unique" events that are of interest in specific applications – for example, the use of seismic sensors (i.e., "unattended ground sensors") for the purpose of detecting intruders. To address this challenge, we have incorporated a machine learning technique into the WSN so that the networks can learn automatically to recognize normal and abnormal modes of operation.

Our approach makes use of a fuzzy Adaptive Resonance Theory (ART) neural network, which was first implemented in a WSN by Kulakov and Davcev [8]. The motivation for choosing this fuzzy ART model lies in its simplicity of computation. The fuzzy ART neural network system is an unsupervised Artificial Neural Network (ANN) that can perform dimensionality reduction and pattern classification. Fuzzy ART considers sensor readings from all sensors simultaneously; change in the combination of sensor readings is what triggers an anomalous event. This is achieved through an iterative process of adjusting weights of interconnections between large numbers of simple artificial neurons. Additionally, the network can continually learn from new events without forgetting what has already been learned. No off-line training phase is required. The algorithm is simple enough to be implemented on the limited platform of the Crossbow motes [1], yet still achieve good performance.

One major problem that we discovered during the implementation of the fuzzy ART model on physical sensors is that the performance of the network is often degraded by missing sensor readings, due to synchronization, sensor faults, communication malfunctions, or malicious attacks. Thus, this problem is unavoidable in WSNs. However, this problem was not addressed by Kulakov and Davcev in their implementation. We have analyzed different ways to estimate the missing values that do not require significant changes to our base model and still improve the performance of the network. (Determining how to substitute for missing data is called *imputation* in statistics.) We propose to use a spatial-temporal replacement scheme that is shown to have better performance in the experiments compared to other techniques studied. This imputation technique works better than others due to the nature of WSNs – the data obtained by the sensor networks are often correlated over time, and over space.

In the remainder of paper, we first review related work in Section 2. Then, we define our problem in Section 3. Our approach is presented in Section 4. In Section 5, we present the hardware platform that we have used to test our system. Our experimental results from the physical experiments are presented in Section 6. Finally, we summarize our findings in Section 7.

2. Related work

Many machine learning algorithms have been proposed in the WSN area for anomaly detection. Examples of these algorithms are Multi-layer Perceptrons (MLPs) [11], Self-Organizing Maps (SOMs) [4], and Adaptive Resonance Theory (ART) [8]. We have chosen the fuzzy ART model proposed by Kulakov and Davcev in [8] for our WSN implementation, because of its unique abilities to learn in a short period of time and to continually learn from new events [2], [3], [8], [12].

After we have extended the original fuzzy ART system to learn a time-series [9], the performance for anomaly detection is still unsatisfactory due to missing data. Our experiments showed that a large percentage of the data are missing due to unstable wireless communication. Several solutions have been suggested to tolerate this error at the communication level such as link quality profiling [13]. Moreover, there are higher level algorithms developed to estimate missing sensor data such as Fletcher, et. al., [5] who estimated missing sensor data using a jump linear system and Kalman filtering. Granger, et. al., [6] suggested three different ways to modify the fuzzy ART neural network to compensate for missing input data. In this paper, we investigate an imputation scheme that utilizes time and space information. The advantages of this scheme are that it requires less memory, less computation and no offline-training. Our experimental results indicate that this replacement scheme outperforms the modifications suggested in [6].

3. Problem formulation

Let o denote observations made by sensors in the WSN; o_i denotes a vector of collective sensor observations made at time i and $o_{ij} = null$ means the observation of sensor j at time i is missing. Let c denote class/category, and p denote the prototype of c. The data is *missing at random* (MAR) – note that this assumption does not mean that the pattern itself is random, but rather that "missingness" does not depend on the data values. Verifying if the current observation o_i belongs to a pre-existing class/category c_n can be conducted by testing the null hypothesis. Note that c is discrete and we assume each class/category just has one prototype. For each n in p,

$$H_0 : o_i - p_n = 0 \quad \text{versus} \quad H_1 : o_i - p_n \neq 0 \qquad (1)$$

where H_0 hypothesizes that the current observation matches the prototype, meaning that it belongs to one of pre-existing classes. H_1 hypothesizes that the current observation is different from the pre-existing classes, and thus, a new class c_{n+1} is generated. Our problem is to estimate the value of o_{ij} when the observation of sensor j at time i is missing, and to classify o_{ij} into one of the classes in c.

4. Approach

In order to detect abnormal events in a previously unknown environment, we have to first learn an initial model of the environment online. We assume that this is the *normal* model of the environment. Representations of abnormal states of the environment will not be kept in the sensor nodes due to memory limitations. Therefore, any events that do not match the existing normal model are treated as abnormal events by the sensor motes. In this section, we first introduce our network architecture; then, we describe the basic fuzzy ART network.

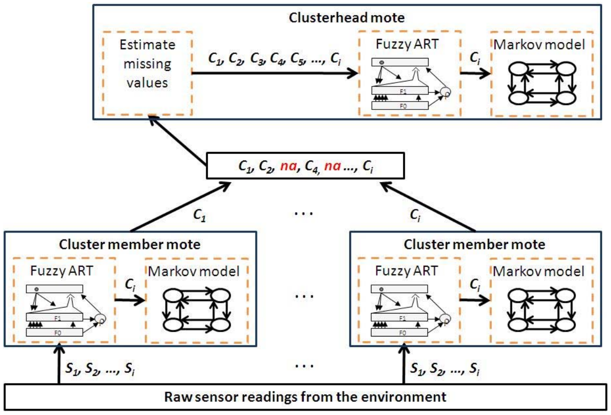


Fig. 1. Proposed fuzzy ART architecture, extended to estimate missing data and perform time series analysis.

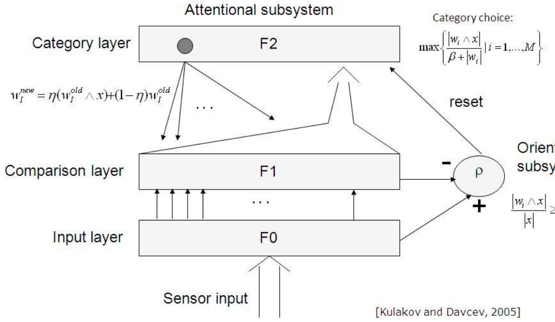
4.1. Architecture for the sensor networks

In our approach, sensors are arranged hierarchically, as shown in Figure 1. The hierarchical structure has been used in many sensor networks, such as [7]. In our WSN, sensor nodes are divided into clusters. Each cluster has a clusterhead and multiple cluster members. Each cluster covers a geometric region and is responsible for detecting the environmental changes in that region. Both cluster members and clusterheads run an identical detection system – a missing data estimator, a fuzzy ART network, and a Markov model. Cluster members read in raw sensor readings, s_i, (e.g., light and sound) from the environment as input, and then classify data into categories c_i. After the classification process, cluster members send their category labels to their clusterheads. The clusterheads first pre-process the collected category labels by identifying and estimating the missing values. Then, the processed categorizations are used as input to the fuzzy ART neural network and are fused together to reduce the dimensionality of the data. The output of the fuzzy ART network is a category label c_i. If the category label does not match one of the existing prototypes, a change is detected by the system. After the categorization process is finished, the system further checks if there are time-related changes.

Clusterheads may have higher level clusterheads which classify their output class labels. Finally, the root mote can obtain the final model of the environment. With this architecture, our WSN can be easily scaled to large numbers of sensors. At the same time, this hierarchical approach reduces communication, which in turn saves energy in the WSN.

The original fuzzy ART model cannot detect time-related abnormal events. We use a Markov model to learn a time series. The details of this approach are addressed in our companion paper [9].

4.2. The Fuzzy ART Network



Attentional subsystem
Category layer
F2
Category choice:
$\max\left\{\frac{|w_i \wedge x|}{\beta+|w_i|} \mid i=1,\ldots,M\right\}$
$w_I^{new} = \eta(w_I^{old} \wedge x)+(1-\eta)w_I^{old}$
reset
Comparison layer
F1
-
ρ
Orienting subsystem
+
$\frac{|w_i \wedge x|}{|x|} \geq \rho$
Input layer
F0
Sensor input

[Kulakov and Davcev, 2005]

Fig. 2. A typical fuzzy ART architecture. (see [8])

Adaptive Resonance Theory (ART) is a neural network architecture developed by Grossberg and Carpenter [3]. The basic fuzzy ART network is an unsupervised learning model, which is able to take analog input. Kulakov and Davcev proposed a fuzzy ART model for change detection in a WSN in [8]. Our basic fuzzy ART network is implemented in the same way. Figure 2 gives a representation of their fuzzy ART network. A typical fuzzy ART network has three layers: an input layer (F0), a comparison layer (F1) and a category layer (F2). For completeness, we present the learning process of the neural network in detail as follows.

Compare each input vector I ($I_j \in [0,1]$; and $j = \{1,2,...,N\}$) with each category/prototype in F2 to classify it with its best match. The choice function A_j is defined as, $A_j(I) = \frac{|I \wedge w_j|}{\epsilon + |w_j|}$, where parameter w_j is the binary weight vector of category j, and parameter $\epsilon \in [0,1]$. The operator $\wedge$ is defined by $x \wedge y \equiv min(x,y)$ and the operator $|\cdot|$ is defined by $|x| \equiv \sum_{i=1}^{M} x_i$. Then, the approach selects the F2 node J that has the highest match ($A_J = max\{A_j | j = 1,...,N\}$). The weight vector of the winning node (w_J) is compared to the current input at the comparison layer. The training process starts if the match function of the chosen category exceeds the vigilance; that is:

$$\frac{|I \wedge w_J|}{|I|} \geq \rho \tag{2}$$

where parameter $\rho \in [0,1]$ represents the vigilance level. The number of developed categories can be controlled by the vigilance level. The higher the value, the more sensitive the network is to changes in the environment, which can result in a larger number of finer categories. When $\rho = 1$, the network creates a new category for every unique input pattern.

The prototype node J in F2 captures the current input, and the network learns by modifying the weight vector w_J according to:

$$w_J^{new} = \gamma(I \wedge w_J^{(old)}) + (1-\gamma) w_J^{(old)} \tag{3}$$

where parameter $\gamma \in [0,1]$ is the learning rate. Fast learning occurs when $\gamma = 1$. If the stored prototype w_J does not match the input sufficiently (i.e., Equation (2) is not satisfied), the winning F2 node will be inhibited until a new input vector is applied. Then, another F2 node is selected with the highest A_j value, whose prototype will be matched against the input. This process is repeated until the network either finds a stored category whose prototype matches the input, or selects the uncommitted F2 node if all prototypes result in mismatches. In this case, learning a new category is initiated according to Equation (3).

The fuzzy ART system has a category proliferation problem. Complement coding can be used to overcome this problem [12]. The complement of input I can be achieved by preprocessing each incoming vector a by $a^c \equiv 1-a$. After the complement coding process, input I becomes a $2M$-dimensional vector, $I = (a, a^c) \equiv (a_1, ..., a_M, a_1^c, ...a_M^c)$. Note that normalization of input I can be achieved by preprocessing each incoming vector a, $I = \frac{a}{|a|}$. Inputs preprocessed into complement coding form are automatically normalized.

4.3. The Missing Data Estimator

Incomplete data is common in WSNs and may arise due to hardware failures, synchronization, packet collisions, signal strength fading, or environmental interference (e.g., a microwave, walls, or human blockage). Our experimental findings showed that, on average, approximately 40% of data from our sensor nodes were missing when they were presented to the clusterheads' classifier. The existence of missing data degrades the performance of the classifier.

4.3.1 Existing techniques. Several missing data replacement strategies, such as those listed in Table I, have been used to improve the performance of the classifiers.

The first strategy is called "doing nothing". When new sensor data comes in, the new sensor data replaces the old data. If the sensor values are missing, the system keeps the last available data. Specifically, for received sensor observations a, if $a_i^t = null$, let $a_i^t = b_i^{t-1}$, where b is the processed input used for the fuzzy ART algorithm, t is the current time instance and $t-1$ is the previous time instance. We have used this strategy as the base strategy in comparison to the other missing data strategies. The computational and space complexities of this imputation strategy for each mote are $O(1)$.

Strategies 2 and 3 use a fixed constant to replace the missing data. Specifically, strategy 2 uses the minimum non-existing value and strategy 3 uses the maximum non-existing value. "Non-existing values" means we do not have sensor readings or classes numbered as 0 or 1. The fuzzy ART algorithm remains unchanged.

Strategies 4 and 5 use a moving average to replace the missing data. Specifically, strategy 4 uses the mean of the past 5 sensor values used, including the estimated missing values. Strategy 5 uses the mean of the past 5 seen observations of the current sensor i, excluding the processed missing values. Note that strategy 4 is different from strategy 5 when the past 5 sensor values contain data that were estimated. The computational and space complexity of both strategies for each mote are a fixed constant, therefore, $O(1)$.

Strategies 6, 7 and 8 are proposed by Granger, et. al., in [6]. We implemented these 3 strategies exactly as proposed. Strategy 6 replaces the missing data with a fixed minimum value (like strategy 2). In addition, strategy 6 modifies the original fuzzy ART system by setting the complement coding of the missing data to the minimum value as well. Strategy 7 replaces the missing data with a fixed maximum value (like strategy 3). The complement coding of strategy 7 is also set to the fixed maximum value. However, as the number of prototypes grows, it becomes harder and harder to pass the vigilance testing. Thus, strategy 7 changed the vigilance testing rule to $\frac{|a_i \wedge A'|}{M} \geq \rho$. Instead of using fixed constant values for missing data, strategy 8 uses an indicator vector. When a sensor value is missing, the indicator sets the missing data to "0" and the complement coding to "1". The new prototype choice becomes, $T_j(A, \delta) = \frac{|w_j \wedge A' \wedge \delta|}{\alpha + |w_j \wedge \delta|}$. The new vigilance level

TABLE I

Comparison of missing value imputation strategies. The time and space complexities are calculated based on a single mote, where n is the number of observations made from time 1 and k is the number of the mote's neighbors.

	Missing data imputation strategies	**Time**	**Space**
1. Doing nothing	If $a_i^t = null$, set $a_i^t = b_i^{t-1}$, where b is the processed input used for the fuzzy ART algorithm. t is the current time instance and $t-1$ is the previous time instance.	$O(1)$	$O(1)$
2. Replace by minimum value	If $a_i = null$, set $a_i = 0$ and complement coding $a_i^c = 1 - a_i$. Note: Neither 0 or 1 are used as normal category numbers.	$O(1)$	$O(1)$
3. Replace by maximum value	If $a_i = null$, set $a_i = 1$ and $a_i^c = 1 - a_i$.	$O(1)$	$O(1)$
4. Replace by 5 average used	If $a_i = null$, set $a_i^t = \frac{(b_i^{t-1} + \ldots + b_i^{t-5})}{5}$.	$O(1)$	$O(1)$
5. Replace by 5 average seen	If $a_i = null$, set $a_i^t = \frac{(a_i^{t-1} + \ldots + a_i^{t-5})}{5}$.	$O(1)$	$O(1)$
6. Replace by "0" [6]	If $a_i = null$, set $a_i = a_i^c = 0$.	$O(1)$	$O(1)$
7. Replace by "1" [6]	If $a_i = null$, set $a_i = a_i^c = 1$. The new vigilance testing becomes $\frac{\|a_i \wedge A'\|}{M} \geq \rho$.	$O(1)$	$O(1)$
8. Replace by indicator vector [6]	If $a_i = null$, $\delta_i = 0$; otherwise $\delta_i = 1$. ($\delta = (\delta_1, \delta_2, ..., \delta_{2M})$). The new prototype choice becomes $T_j(A, \delta) = \frac{\|w_j \wedge A' \wedge \delta\|}{\alpha + \|w_j \wedge \delta\|}$. The new vigilance testing is $\frac{\|w_j \wedge A' \wedge \delta\|}{\alpha + \|w_j \wedge \delta\|} \geq \rho$. The new prototype learning is $w_J^{'} = \beta((A \vee \delta^c) \wedge w_J) + (1 - \beta) w_J)$.	$O(1)$	$O(1)$
9. EM imputation	See [10].	$O(k * n)$	$O(n)$
10. Replace by spatial-temporal data	See Algorithm 1.	$O(k)$	$O(1)$

testing becomes, $\frac{|w_j \wedge A' \wedge \delta|}{\alpha + |w_j \wedge \delta|} \geq \rho$, and the new prototype learning is $w_J^{'} = \beta((A \vee \delta^c) \wedge w_J) + (1 - \beta) w_J$. The computational and space complexities of these strategies for each mote are $O(1)$. Since these strategies only modify the vigilance testing rules and/or learning rules from the original fuzzy ART algorithm, there are no additional time and space requirements for these strategies.

To further test the imputation strategy, the standard missing data imputation strategy from the literature of statistics, i.e., the Expectation Maximization (EM) imputation strategy was also applied to the same data. Little and Rubin provide a detailed explanation of the EM imputation algorithm in [10]. It is important to notice that this EM imputation strategy is computationally intensive and, thus, is not practical in a WSN. However, we consider it here for comparison purposes.

4.3.2 Spatial-temporal algorithm. The existing missing data imputation strategies do not take into account spatial-temporal information. We hypothesized that if an environment is highly correlated in time and space, using spatial and temporal information to estimate missing observations should have high accuracy. To confirm our hypothesis, we have to show that the environment is highly correlated both in time and space. We utilized techniques from statistics, i.e. the space and time correlation testing.

We made use of two time-series tests to determine if the data are correlated in time, namely, the Durbin-Watson test and the partial AutoCorrelation function (PACF). The Durbin-Watson test determines whether or not the data set is time correlated. and the PACF gives us information on how the sensor data are correlated in time with each other.

For space correlation testing, the Pearson correlation coefficients and R-squared testing are used. The Pearson correlation coefficient is a common measure of the correlation between two random variables X and Y. Pearson's correlation reflects the degree of association between two variables.

In order to deal with the missing data, we studied ways of modifying the fuzzy ART algorithm which require only minor changes to the current fuzzy ART network. We need a simple replacement algorithm which runs online and still has a satisfactory performance.

Algorithm 1 Spatial-temporal imputation approach

```
for each missing input sensor $s_i$ do
    for all sensors $s_j$ within one hop of communication range of $s_i$ do
        if If the sensing distance between $s_i$ and $s_j < \alpha$ then
            $a_i^t$ = the most common reading of $a_j^t$
        else
            $a_i^t = a_i^{t-1}$
        end if
    end for
end for
```

Thus, we developed Algorithm 1 to estimate the missing data. We first check if a neighbor is within the missing sensor's minimum sensing range. If there are neighbors within the sensor range, we can use the neighbor's reading observation, resulting in a spatially correlated replacement. If there are multiple neighbors within the sensor range, and they do not have the same readings, we choose the majority reading. Otherwise, use the last seen data of the sensor-temporal reading. We use this strategy because in WSNs, sensor data in the environment tends to be highly correlated for sensors that are geographically close to each other (spatially correlated), and also highly correlated for a period of time (temporally correlated).

The computational requirement for an individual mote to estimate missing data using this algorithm is a linear function of the number of motes, k, within the communication range. In our application, it is the number of cluster members within one cluster. Therefore, the computational complexity for the proposed imputation technique is $O(k)$. Let m denote the number of motes in a WSN, R denote the radius of the WSN, and S denote the communication range or sensing range of each sensor. In our problem, we assume that the sensing range is the same as communication range and all sensors have the same communication range/sensing range. If R is much larger than the mote i's communication range S, then k is much smaller than m. In the worst case, k=m. In general, sensor motes are deployed such that each of them covers a local region, and together they cover the entire environment. Thus, searching within a local cluster of k motes is typically not computationally intensive. The space requirement for the proposed approach is $O(1)$, since we just need to store one previous observation.

5. Hardware platform

To compare these strategies, we conducted experiments using Crossbow [1] sensor motes. A Crossbow mote includes a processing unit, a sensor module, and a communication module. The processing board contains an 8-bit processor at $8MHz$, a $128KB$ programming memory and $512KB$ additional data flash memory. The wireless transmission range is around 10 meters inside a building. The sensor board has a buzzer, a light sensor, a microphone, 2 magnetometers and 2 accelerometers. For the experiments reported in this paper, we used the light and sound sensing components. Note that every mote runs the same fuzzy ART program image. Motes are assigned role, and can either be a cluster member or a clusterhead.

6. Experimental results

Accuracy is used as our performance measure. The mismatch between the algorithm's categorization and the true category is considered an error. The accuracy A is defined as, $A = C/T$, where C denotes the number of correct categorizations, and T denotes the total number of observations. To ensure fair comparison, the vigilance parameter of the fuzzy ART algorithm has been readjusted for each replacement strategy until the best performance is obtained.

In this experiment, we used five cluster members and one clusterhead. The five cluster members were uniformly deployed around the clusterhead. In addition, all cluster members are within communication range of the clusterhead. Experimental results were obtained from 3 sets of trials. In each trial, each sensor mote made 6500 observations. For testing time and space correlations, we only used the first trial of collected data, since the other two trials repeat the first trial and our environment settings did not change. For the purposes of correlation testing, we first removed the samples with missing values. All testing results were made from a complete data set of approximately 1500 samples. Our sensory data passed the Durbin-Watson test and has a value of 0.0059 with 95% confidence level. A DW value of less than 2 indicates there is a high correlation in time. Our number is near 0, which is an evidence that our sensory data have time correlation.

TABLE II

THE PARTIAL AUTOCORRELATIONS

Lag	1	2	3	4	5
Correlation	0.997	-0.002	-0.002	-0.002	-0.002

Our next test is to determine the sensory data's time correlations by using partial autocorrelations. The partial autocorrelations are shown in Table II. The result shows that our sensor data has high correlation with one previous data point, i.e., lag 1 value is close to 1; however, there is little association with 2 or more sensory observations made in the past, i.e., low lag 2-5 values.

TABLE III

PEARSON CORRELATION COEFFICIENTS

	sensor1	**sensor2**	**sensor3**	**sensor4**	**sensor5**
sensor1	1.0	0.998	0.997	0.995	0.995
sensor2	0.998	1.0	0.998	0.996	0.997
sensor3	0.997	0.998	1.0	0.996	0.998
sensor4	0.995	0.996	0.996	1.0	0.996
sensor5	0.997	0.997	0.998	0.996	1.0

To show space correlation, we calculated the correlation coefficients between the sensors at each observation as well as R^2. The Pearson correlation coefficients between nodes are shown in Table III. The correlation coefficients between sensor nodes are close to 1. Therefore, there are high positive associations between sensor nodes.

We further tested the goodness of fit of the model of one sensor's observations replaced by other sensor's observations. As an example, we took entire observations made by sensor 1 and used it against observations made by sensor 2 to obtain the R^2 value. The R^2 value is almost perfect (close to 1), which means if we use sensor 1's reading to replace sensor 2's reading when sensor 2's observation is missing, it should result in high accuracy, due to the model fitness being high. Our sensory data passed both the time and space correlation tests, therefore, our sensory data is highly correlated in time and space.

The averaged accuracy and standard deviation of different imputation techniques have been calculated and the results are shown in Figure 3. We used "do nothing" (strategy 1) as the baseline for comparison. To determine the significance of these results, we applied the Student's t-test to the accuracy results for the spatial-temporal scheme compared against other imputation strategies. This test confirms that the differences in these results are statistically significant, with a confidence level of 99.5%. The spatial-temporal imputation strategy (strategy 10) performed better than the other strategies. The results suggest that the spatial-temporal imputation strategy (strategy 10) and EM (strategy 9) imputation work better than the "do nothing" as well

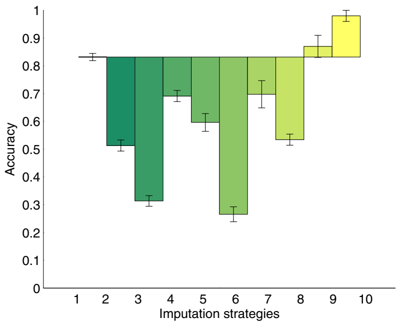


Fig. 3. Accuracies of different imputation techniques, with the x axis indicating the strategy number. (Refer to Table I for the definitions of the strategies.)

as other replacement strategies. This is due to the fact that our application involves continuous data gathering for large scale and distributed physical phenomena using a dense wireless sensor network which results in high correlation both in time and space. If the nodes are densely deployed, the readings from nearby nodes are likely to be highly correlated and hence contain redundancies. By using this combination of spatial and temporal replacement for the missing input we can achieve good performance with relatively low computational cost. The “do nothing” strategy has a better performance than moving average 5 (strategies 4 and 5). This is because the partial autocorrelation results indicate that our sensory data is highly correlated with 1 past history data point, not 5 data points. If the partial autocorrelation indicates the sensory data is highly correlated with 5 past history data points, i.e., a high lag 5 value, the performance of strategies 4 and 5 may outperform “do nothing”. This shows that the correct time model directly affects the performance. Strategies 2 and 3 just use a fixed value for missing data. As the missing pattern changes, the fuzzy ART treats them as different classes as the number of classes/categories increase, which results in more false alarms. Strategies 6, 7, and 8 tried to learn the missing patterns with a fixed value or an indicator. These strategies do better than strategies 2 and 3, but still have high misclassifications due to changing missing patterns. The EM imputation has a relatively high accuracy since it finds the best distribution for the observations in terms of likelihood. However, it is computationally intensive and requires offline learning. In our application, the EM algorithm is too large to be implemented on sensor motes. In a WSN application, if the sensory data is highly correlated in time and/or space (i.e., passes the proposed time and space statistical tests), our proposed spatial-temporal strategy works the best. Note that the time and space models in our algorithm are subject to change based on the environment. It is important to use the appropriate time and/or space model to estimate the missing values. In most environments, if there is no time and memory constraint, the EM imputation strategy is recommended.

7. Conclusion

In this paper, we presented a new technique to improve the accuracy of the sensor network: we estimate the missing data by using a spatial-temporal replacement scheme. Experimental results show our algorithm performs better than the proposed techniques in the literature. The proposed spatial-temporal replacement scheme requires less computation and space than the EM imputation scheme, and is affordably implemented in wireless sensor networks. We have supported our findings using statistical tests, which showed our proposed scheme works the best because the environment is highly correlated in time and space. In ongoing work, we plan to investigate the impact of different imputation strategies in an anomaly detection system.

8. Acknowledgments

This research was sponsored in part by the Oak Ridge National Laboratory SensorNet program. We also thank Michael Bailey for help with implementing the fuzzy ART algorithm, the operator control program and integrating the Deluge system.

References

[1] http://www.xbow.com/.

[2] G. Carpenter and S. Grossberg. Fuzzy art: Fast stable learning and categorization of analog patterns by an adaptive resonance system. *Neural Networks*, 4:759–771, 1991.

[3] G. Carpenter and S. Grossberg. *Adaptive Resonance Theory*, pages 87–90. The Handbook of Brain Theory and Neural Networks. MIT Press, 2003.

[4] E. Catterall, K. Van Laerhoven, and M. Strohbach. Self organization in ad hoc sensor networks: An empirical study. In *The 8th International Conference on the Simulation and Synthesis of Living Systems*, Sydney, Australia, 2002.

[5] A. K. Fletcher, S. R., and V. K. Goyal. Estimation from lossy sensor data: jump linear modeling and kalman filtering. In *The Third International Symposium on Information Processing in Sensor Networks (IPSN)*, pages 251–258, New York, NY, 2004.

[6] E. Granger, M. A. Rubin, S. Grossberg, and Pierre Lavoie. Classification of incomplete data using the fuzzy artmap neural network. In *The IEEE-INNS-ENNS International Joint Conference on Neural Networks (IJCNN)*, volume 6, pages 35–40, Washington, DC, 2000.

[7] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-efficient communication protocols for wireless microsensor networks. In *International Conference on System Sciences*, Maui, HI, January 2000.

[8] A. Kulakov and D. Davcev. Tracking of unusual events in wireless sensor networks based on artificial neural-networks algorithms. *Information Technology: Coding and Computing (ITCC)*, 2:534–539, 2005.

[9] Y. Li and L. E. Parker. Intruder detection using a wireless sensor network with an intelligent mobile robot response. *To appear in IEEE SoutheastCon*, April 2008.

[10] R. J. Little and D. B. Rubin. *Statistical analysis with missing data*. John Wiley & Sons, Inc., New York, NY, 1986.

[11] L. Reznik, G. V. Pless, and T. A. Karim. Signal change detection in sensor networks with artificial neural network structure. In *IEEE International Conference on Computational Intelligence for Homeland Security and Personal Safety (CIHSPS)*, pages 44–51, Orlando, Florida, March 2005.

[12] E. Sapojnikova. *ART-based Fuzzy Classifiers: ART Fuzzy Networks for Automatic Classification*. PhD thesis, University of Tubingen, Tubingen, Germany, April 2003.

[13] J. Zhao, R. G., and D. Estrin. Computing aggregates for monitoring wireless sensor networks. In *The 1st IEEE International Workshop on Sensor Network Protocols and Applications (SNPA)*, Anchorage, Alaska, May 2003.