UNIVERSITÀ DEGLI STUDI DI PALERMO

NDS NETWORKING AND DISTRIBUTED SYSTEMS RESEARCH GROUP

# *Detecting faulty wireless sensor nodes through stochastic classification*

Article

Accepted version

A. Farruggia, G. Lo Re, M. Ortolani

In Proceedings of the IEEE International Conference on Pervasive Computing and Communications Workshops, 2011, pp. 148-153

It is advisable to refer to the publisher's version if you intend to cite from the work.

Publisher: IEEE

http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5766858

NDS LAB - Networking and Distributed Systems
http://www.dicgim.unipa.it/networks/

# Detecting Faulty Wireless Sensor Nodes through Stochastic Classification

Alfonso Farruggia, Giuseppe Lo Re, and Marco Ortolani

*Department of Computer Engineering*
*University of Palermo*
*Viale delle Scienze, ed. 6 – Palermo*
*Email: {alfarruggia, lore, ortolani}@unipa.it*

DRAFT

***Abstract*—In many distributed systems, the possibility to adapt the behavior of the involved resources in response to unforeseen failures is an important requirement in order to significantly reduce the costs of management. Autonomous detection of faulty entities, however, is often a challenging task, especially when no direct human intervention is possible, as is the case for many scenarios involving Wireless Sensor Networks (WSNs), which usually operate in inaccessible and hostile environments. This paper presents an unsupervised approach for identifying faulty sensor nodes within a WSN. The proposed algorithm uses a probabilistic approach based on Markov Random Fields, requiring exclusively an analysis of the sensor readings, thus avoiding additional control overhead. In particular, abnormal behavior of a sensor node will be inferred by analyzing the spatiotemporal correlation of its data with respect to its neighborhood. The algorithm is tested on a public dataset, over which different classes of faults were artificially superimposed.**

***Keywords*-Autonomic Computing; Markov Random Fields; Wireless Sensor Networks.**

## I. Introduction

Wireless Sensor Networks (WSN) are nowadays increasingly gaining popularity, also in challenging scenarios such as pollution control, intrusion detection, healthcare monitoring [1]. Although such considerations clearly show that it is mandatory to achieve a high degree of robustness, and tolerance to unpredictable faults, such goals are somewhat conflicting with the typical constraints of WSNs, which usually operate in hostile environments, and are supposed to require as little manual intervention as possible.

The general topic of self management, and self healing in complex systems has been addressed in the context of Autonomic Computing [2], [3], whose main goal is the development of simple, and clever software tools for automating all the administrative tasks that are so far delegated to skilled personnel, in order to reduce the overall maintenance costs of the system.

Our work presents an algorithm for modeling the behavior of sensor nodes, whose main task consists in monitoring typical indoor environmental quantities, in order to detect potential faults, and our approach will rely exclusively on the analysis of sensed data, with no additional control overhead. The proposed algorithm considers each sensor on board of the nodes as the generator of a stochastic variable, whose value is to be reliably estimated. Assuming that the considered physical quantities are correlated both in time and in space, our aim is to detect a faulty sensor node (i.e. one whose sensors behave as outliers with respect to its neighboring nodes) through an approach based on probabilistic graphical models, namely Markov Random Fields (MRF) [4], [5]. To this aim, groups of nearby sensors are represented as nodes of an undirected graph, as shown in Figure 1; starting with an approximation of the health status of each sensor, our method will provide a more reliable estimate for the same status.

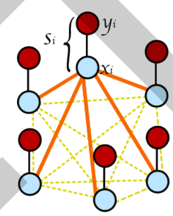


Figure 1. Representation of a group of sensors as nodes of a MRF; the health status of each sensor $s_i$ is represented by an observable variable $y_i$, and a hidden variable $x_i$.

Unlike other works, our approach specifically attempts to minimize the additional required control overhead, since it purely relies on the analysis of sensed data. Although a simpler version of our algorithm might run directly on the host where all data is collected, we will show that in fact only local information about the neighborhood of a node is necessary in order to infer its health status, so that it is straightforward to devise a distributed version of the algorithm running remotely within a cluster of nodes.

The remainder of the paper is organized as follows. Section II briefly presents some other works about autonomic approaches to WSN management, and to probabilistic data analysis in a WSN context. In Section III, we present the details of our work. Section IV shows some experimental results on a widely known public dataset; finally, Section V contains some concluding remarks.

## II. Related Work

The concept of Autonomic Computing proposed by IBM in 2003 [2] has since inspired many authors, who have applied the autonomic approach to various scenarios. For instance, the authors of [3] survey several projects of autonomic systems, including a WSN-based system specifically targeting the power management issue; in [6], a comprehensive autonomic approach for a WSN-based application is presented, where the authors discuss an intrusion detection scenario.

For the purpose of the present work, we specifically focus on monitoring wireless sensor nodes in order to provide failure detection [7]. The failure detectors presented in literature fall into two main categories: the ones directly exploiting the sensed data in order to assess the status of the monitored element, and those that rely on some other external control mechanism. For instance, in [8], the authors present a failure detector based on data mining, able to detect faults in an electric power system, whereas [9] presents a failure detector based on a handshake mechanism evaluating the delays of messages exchanged between the monitored and the managed element.

In our work, we adopt the former approach, and we aim to assess the health status of a sensor node, exclusively by analyzing the sensor readings; in detail, we will use a probabilistic approach to estimate the status of our monitored element. Similar approaches have been presented in literature, with a focus on different application scenarios. In [10], a stochastic recursive identification algorithm is presented which can be implemented in a fully distributed and scalable manner within the network. The authors demonstrate that it consumes modest resources as compared to a centralized estimator, while still being stable, unbiased, and asymptotically efficient; in the considered scenario, sensors classify the presence or absence of an effluent released from a chemical plant into a river. In [5] a framework for distributed signal processing in sensor network environments is discussed, in which sensor nodes collect noisy readings, and classify them by using a Markov Random Field; unlike our approach, however, the authors propose a static calibration of the necessary MRF parameters.

In [11] MRFs are used to identify the most relevant collected data, in order to implement an algorithm for aggregation of large amounts of data originating from diverse sources; unlike our approach, however, the possibility of anomalies introduced by faulty sensors is not taken into account.

Finally, the authors of [12] present an efficient collaborative sensor-fault detection (CSFD) scheme, where the health status of a sensor node is inferred via a homogeneity test. Similarly to our work, CSFD implements a probabilistic approach, although it relies on specific control messages thus causing additional overhead, not required by our approach.

## III. Identifying Faulty Sensors through MRF

This Section presents the proposed method for assessing the health status of a sensor node; specifically, we intend to infer the general functioning of a node by analyzing the data produced by the individual sensors on board of the node itself. We use a probabilistic approach based on a particular instance of graphical models, namely the Markov Random Fields (MRF) [4], [5], in order to classify each sensor according to a binary label representing its state in terms of spatial correlation with respect to sensors for the same physical quantity, on board of nearby nodes. MRFs are a mathematical tool that allows to exploit spatial information in a classification process, where the considered stochastic variables are assumed to have Markov properties, and have been widely used in the classification of data from spatial databases [13]. MRFs allow to reduce a global model of a wide dataset into an equivalent model based only on the local properties of data.

Our approach is based on the assumption that sensory measurements collected by nearby nodes are similar to each other, due to the intrinsic nature of the considered physical quantities; such similar measurements are then expected to show sufficiently high *spatial* statistical correlation when all sensors are correctly functioning. We will classify the health status of each sensor obeying this rule as GOOD, and otherwise we will assume that sensor to be DAMAGED.

In the present work we are considering data collected through a WSN; for the sake of simplicity, we will focus here on a single physical environmental quantity. Let $S = \{s_1, \ldots, s_n\}$ denote the set of sensors located in the considered area, and let us represent the health status of each sensor by means of two stochastic binary variables: $Y_i$, representing its *observable* health status at a given moment, and $X_i$, representing its estimated (*hidden*) status. Intuitively, the former represent the (possibly imprecise) information about a sensor status that can be computed based on the collected measurements, whereas the latter represents the true status, which may not be directly derived from the physical evidence. Our aim is to provide a reasonable initialization for all the $Y_i$; the values for the corresponding $X_i$ will be inferred by minimizing a globally-defined entropy function.

In order to build the MRF for the considered physical quantity we will work on the undirected graph representing the corresponding sensors; the set of vertices is clearly $S$, and we will assume that an edge between any two sensors exists if they are "sufficiently close"; the precise definition of closeness is heavily dependent on the chosen scenario; in our case, it will be detailed in Section IV.

We will start by defining a *clique* $C_{s_i} \subseteq S$, which will contain the sensors that most influence the behavior of $s_i$.

If $c_{s_i}$ indicates the spatial coordinates of sensor $s_i$, a clique of size $\omega_{s_i}$ and composed of sensors distant at most $\vartheta_{s_i}$ from $s_i$ will be defined as:

$$\begin{aligned} C_{s_i} = \{s_1, s_2, \ldots, s_{\omega_{s_i}} \quad &: \quad \|c_{s_j} - c_{s_i}\| \leq \vartheta_{s_i} \\ &\wedge \quad \|c_{s_j} - c_{s_k}\| \leq \vartheta_{s_i}, \\ & \qquad \forall j, k = 1, \ldots, s_{\omega_{s_i}}\} \end{aligned}$$

DRAFT

where $\| \cdot \|$ defines the Euclidean distance.

In order for such a clique to exist, we need to determine $\vartheta_{s_i} : C_{s_i} \neq \emptyset$.

In the context of MRFs, the concept of a clique is useful in that the probability, conditioned by the variables for all the other sensors in the field, that the health status of a sensor is $X_i = x_i$ actually depends only on the sensors within its own clique, as expressed by the following equation:

$$p(x_i \,|\, \{x_j\}_{s_j \in S - \{s_i\}}) = p(x_i \,|\, \{x_j\}_{s_j \in C_{s_i}}). \qquad (1)$$

Each of the hidden variables will also depend on its respective observable variable; in order to provide a meaningful initialization for each of the latter, we introduce the notion of *temporal* correlation into our method. In detail, we make the additional assumption that nearby sensors are to show similar trends for their respective measurements across reasonably small time periods.

For each of the sensors in the clique of $s_i$ we will thus consider the window $W(s_j)$ containing the last $w$ readings, where $w$ is dynamically adapted to the considered scenario, as will be explained in Section IV. In order to carry out the computations, we assigned the values -1 and +1 to the labels DAMAGED and GOOD respectively, so we will initialize the observable variable for $s_i$ as follows:

$$Y_i = \begin{cases} -1 \quad (\text{DAMAGED}) & \text{if } avgCorr(s_i) \leq 0.3, \\ \;\;1 \quad (\text{GOOD}) & \text{otherwise;} \end{cases} \qquad (2)$$

where $avgCorr(s_i)$ represents the average correlation between the samples sensed by $s_i$ and those of each of the other sensors in its clique, computed as follows:

$$avgCorr(s_i) = \frac{\sum\limits_{s_j \in C_{s_i}} corr(W(s_i), W(s_j))}{|C_{s_i}|} \qquad (3)$$

We are assuming that a weak correlation (expressed by the 0.3 threshold, according to the Pearson coefficient) denotes an abnormal behavior for the sensor, or in other words, a damaged status.

The estimated health status of a sensor $s_i$ is however represented by its hidden variable $X_i$; we can make use of the *Hammersley-Clifford* theorem [4] in order to express the probability density function according to the well-known Markov-Gibbs equivalence, as in the following equation:

$$p(x_i, y_i) = \frac{1}{Z} \exp\{\frac{-E(x_i, y_i)}{T_i}\} \qquad (4)$$

where $Z$ is the *partition function* used for normalization, which may be computed as follows:

$$Z = \sum_i \exp\{\frac{-E(x_i, y_i)}{T_i}\} \qquad (5)$$

and $E(x_i, y_i)$ is a Hamiltonian function which represents the *energy* of the MRF, following the concept of *Ising Model* [14], and may be computed as follows:

$$E(\mathbf{x}, \mathbf{y}) = -\beta \sum_{i,j} x_i x_j - \eta \sum_i x_i y_i + h \sum_i x_i. \qquad (6)$$

In Equation 6, $\mathbf{x}$ and $\mathbf{y}$ are the sets of all the hidden and observable variables respectively; $\beta$ and $\eta$, are the *coupling parameters* weights between the random variables of the field; namely, the former influences the interaction among "nearby" hidden variables, whereas the latter controls the relationship between each hidden variable and its observable variable; the last parameter, $h$, weighs the previous status of the hidden variables.

We want to find out the values for the hidden variables that, given the chosen initial conditions for the observable variables, have the highest probability to minimize the energy function. To this aim we use the algorithm known as Iterated Conditional Mode (ICM) proposed by Besag [15], i.e. a deterministic algorithm which maximizes local conditional probabilities sequentially. It uses the *greedy* strategy in the iterative local maximization to approximate the maximal joint probability of a Markov Random Field. In our case, the ICM sequentially converges to a local maximum of the conditional probability of $p(x_i \,|\, y_i, \{x_j\}_{s_j \in C_{s_i}})$.

We solved the system of equations using the Lagrange multipliers, after imposing the constraint $\beta^2 + \eta^2 + h^2 = 1$, similarly to what suggested by [16], where an optimization technique to automatically select a set of control parameters for a MRF is presented. In our method, such parameters are recomputed at each iteration in order to increase adaptability.

Finally, we need to compute the value for the $T_i$ parameter, which represents the *temperature* of the Boltzmann distribution. Since maximum variation for the energy, computed as in Equation 6, occurs when a single variable is surrounded by variables of the opposite sign (e.g., a DAMAGED sensor within a clique of GOOD sensors) we will use the following formula for $T_i$:

$$T_i = \frac{var(E_{x_i})}{e^{\theta_i}} \qquad (7)$$

where $var(E_{x_i})$ is the variance of the energy relative to the latest $w$ readings of sensor $s_i$, and the parameter $\theta_i$ is computed as:

$$\theta_i = \begin{cases} \theta_D & \text{if} \quad x_i = \text{DAMAGED}, \\ \theta_G & \text{if} \quad x_i = \text{GOOD}; \end{cases} \qquad (8)$$

with $\theta_D < \theta_G$, so that the DAMAGED label is preferred in case of higher variance.

The algorithm may thus be summarized by the following steps, representing one iteration of the ICM method:

For each sensor $s_i$:

1) consider the sensor $s_i$ and its clique $C_{s_i}$;

DRAFT

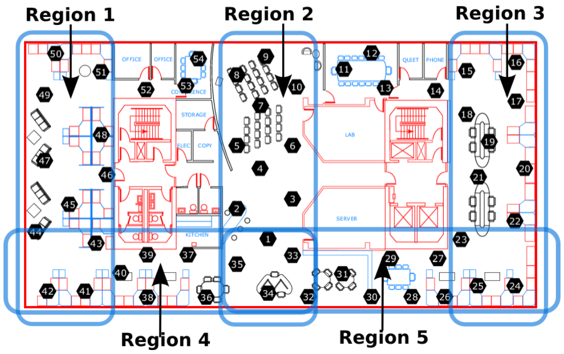


Figure 2. Map of the sensor field from Intel Berkeley Research Lab [17], highlighting regions of correlated sensor readings.

2) compute the value of its observable variable $y_i$ according to Equation 2;
3) compute the parameters $\beta, \eta, h$ of the energy function;
4) compute the value $x_i$ that currently maximizes Equation 4.

At the end of all iterations each sensor is labeled according to the latest $x_i$.

## IV. Experiments

In order to assess the validity of our method, we used a publicly available dataset provided by the Intel Berkeley Research Lab [17], which contains readings collected from 54 sensors between February 28th and April 5th, 2004, via a network of Mica2Dot sensor nodes equipped with weather boards measuring temperature, relative humidity, and ambient light. The authors of [18] report that such data do show significant spatial correlation, and they accordingly divided the nodes into 5 main regions as shown in Figure 2.

We tested our algorithm on the readings relative to temperature for a period of 6 days (from March 1st to March 6th, 2004), and considering only 45 sensors, after eliminating those with an insufficient number of readings or falling out of the mentioned regions. In order to build the topology of the MRF for our scenario, we constructed a clique for each sensor, formed at least by four neighbors located within the same regions. Finally, we grouped the considered samples into time slots of 15 minutes, and taking the average value as representative of each slot, in order to disregard the differences in sampling times for various sensors; this resulted in 570 available samples for each sensor.

In order to test our algorithm, we superimposed different artificially created faults to a subset of the available sensors. A general classification of potential faults is reported in [19]; for our purposes, the actual cause of the fault is not relevant, and rather only the actual resulting trend of the considered quantity over time needs to be taken into account, so we consider here two types of faults (namely, *continuous*, and *discontinuous* faults) obtained by aggregating some of the original classes. *Continuous* faults occur for the entire duration of the experiment; for instance, a sensor can simply produce a constant output, or the sensor readings may happen to be altered by Gaussian noise. *Discontinuous* faults occur at specific time intervals only; we assume that in those intervals the faulty sensors produce a constant output, while returning to normal functioning otherwise. Discontinuous faults are characterized by two parameters: the *duration* of the fault, and the total *number* of its occurrences during the experiment.

We assessed the approach by computing two performance metrics for each experiment: *sensitivity* ($Se$) and *specificity* ($Sp$). Sensitivity measures the ability of the algorithm, in a particular test, to detect a faulty sensor when it really is, while specificity analogously applies to healthy sensors. They are computed as follows:

$$Se = \frac{Tp}{Tp + Fn}, \tag{9}$$

$$Sp = \frac{Tn}{Tn + Fp}; \tag{10}$$

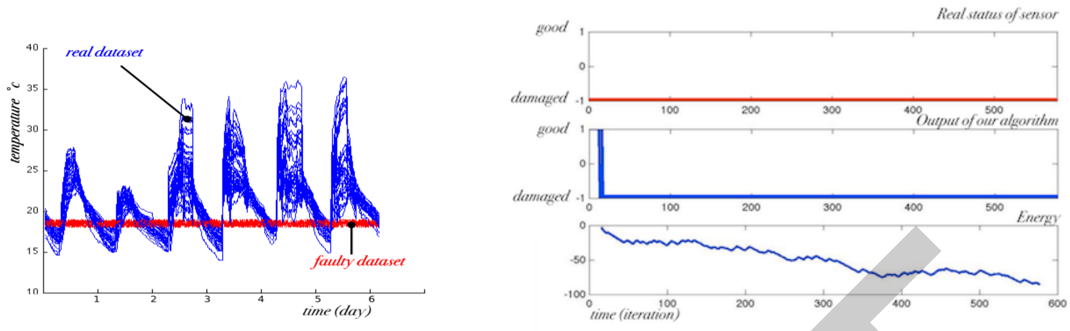
where the $Tp$ parameter measures the amount of sensors whose health status is Damaged and are actually detected as such (i.e. true positives), the $Fp$ parameter measures the amount of sensors whose health status is Good, but are erroneously detected as Damaged (i.e. false positives), and analogously for the two remaining parameters.

In order to assess our algorithm, we corrupted 5% of the total available sensors by applying one of the previously mentioned faults at a time; "faulty" sensors were chosen randomly, according to a uniform distribution. We executed each experiment 10 times, and computed the average values for specificity and sensitivity.

Our method needs the preliminary setting of two parameters, as explained in Section III: the initial size for the window storing the last $w$ samples for each sensor, and the value indicating the maximum number of allowed iterations for the ICM maximization process. The value of $w$ will be dynamically adapted during each run; namely, it will be linearly increased after each fault detection, and reset to the initial value $w_0$ when the sensor status becomes Good again; in our experiments we set $w_0$ to 4. The maximum number of iterations was set to 10.

Figure 3(a) shows the plots representing the trend for temperature measured by healthy sensors, and artificially faulty ones in one run of our algorithm, when considering *continuous* faults (i.e. we assumed that faulty nodes continuously reported a constant value of 18°C corrupted by white noise, with a small variance of 1°C). The three plots reported in Figure 3(b) show the true health status for one of the faulty nodes (constantly Damaged in this case), the detected status according to our algorithm, and the corresponding trend for the energy function, as computed according to Equation 6. The first column of Table I shows

DRAFT




(a) (b)

Figure 3. (a) Sample of the real dataset, with 5% of the nodes corrupted by *continuous* constant fault. (b) Plots showing how the algorithm detects the health status for a faulty sensor.

Table I
CONTINUOUS FAULTS

| | **Constant** | **Gaussian** $\mathcal{N}(\mu = 0, \sigma = \sigma^*)$ | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $\sigma^*$ | n/a | 0.4 | 0.8 | 1.2 | 1.6 | 2.0 | 2.4 | 2.8 | 3.2 | 3.6 | 4.0 |
| **Specificity** | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 |
| **Sensitivity** | 0.98 | 0.044 | 0.41 | 0.48 | 0.80 | 0.85 | 0.66 | 0.97 | 0.93 | 0.98 | 0.97 |

the resulting performance metrics, which are unsurprisingly good, considering the easily recognizable fault type.

The other type of *continuous* fault we considered was a *Gaussian* error added to a subset of the sensors; in particular we corrupted their original readings with a Gaussian with 0 mean, and increasing variance. The remaining part of Table I contains the values of the corresponding performance metrics; we can see that *sensitivity* tends to 1 with increasing values for the variance. It is worth noting that significant values for *sensitivity* occur when the variance is greater than $1.6°$C, i.e. faulty sensors are correctly detected as soon as the additional error may be distinguished from natural, intrinsic variations of the considered quantity.

Faults belonging to the *discontinuous* class cause sensors to produce a constant value of temperature (between $0°$C and $5°$C in our experiments) during some time intervals, regardless of the natural trend. Figure 4(a) highlights the effect of this type of fault on the dataset.

We tested our algorithm by varying the length of such intervals, and the number of times that a sensor assumes such behavior during the experiment. Table II reports the performance of our algorithm in two different scenarios: in the former one, we decrease the duration of each fault, while progressively increasing the number of occurrences; we measure the duration of faults as a multiple of the time slots we used, so 96 corresponds to a duration of 1 day (with 6 days corresponding to 570 samples); the latter scenario dually increases the duration of the faults, while their occurrences decreases.

Considering the second scenario, it is relevant to highlight that *sensitivity* keeps approaching to 1.0, while the duration of the faults increases; this may be intuitively explained by considering that, as soon as our algorithm identifies a "steadily" faulty sensor, the energy function shows a higher variance than when the sensor behaves correctly, as can be intuitively recognized by considering the parts of the energy plot highlighted by the two dashed rectangles in Figure 4(a).

Also considering Equations 4 and 7, we can see that the parameter $\theta$ indirectly influences the final probability $p(x_i, y_i)$; for our experiments, we choose the two possible values for $\theta$ so that:

$$\frac{\theta_D}{\theta_G} = 0.1$$

which results in preferring the DAMAGED label.

## V. CONCLUSION

This paper discussed a stochastic approach to fault detection for wireless sensor nodes, through Markov Random Fields. The classifier does not need supervision, and may be easily implemented in a distributed scenario, without requiring additional control overhead. Our experiments, carried out on a publicly available dataset, showed promising results in the presence of various types of faults, and encourage us to explore possible future developments in the context of a more comprehensive autonomic framework.

## VI. ACKNOWLEDGEMENTS

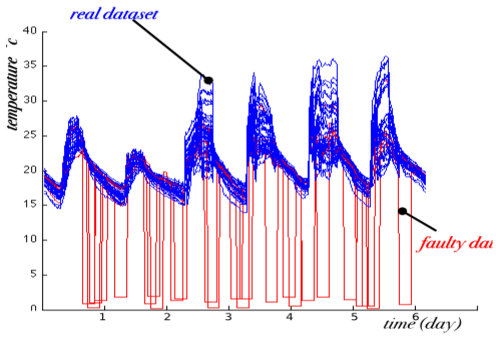
This work has been partially supported by the "FRASI – Framework for Agent-based Semantic-aware Interoperability" project.

## REFERENCES

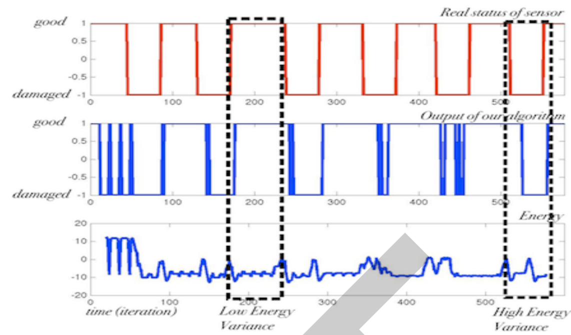
[1] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Computer Networks*, vol. 52, no. 12, pp. 2292–2330, 2008.

[2] Kephart, J. O. and Chess, D. M., "The vision of autonomic computing," in *IEEE Computer*, vol. 36. Los Alamitos, CA, USA: IEEE Computer Society Press, 2003, pp. 41–50.

DRAFT




(a)




(b)

Figure 4. (a) Sample of the real dataset, with 5% of the nodes corrupted by *discontinuous* fault. (b) Plots showing how the algorithm detects the health status for a faulty sensor.

Table II
DISCONTINUOUS FAULTS

| | Scenario 1 | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Duration** | 96 | 48 | 32 | 24 | 19 | 16 | 13 | 12 | 10 |
| **#faults** | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| **Specificity** | 0.98 | 0.97 | 0.96 | 0.95 | 0.93 | 0.93 | 0.92 | 0.92 | 0.91 |
| **Sensitivity** | 0.94 | 0.88 | 0.83 | 0.77 | 0.72 | 0.69 | 0.67 | 0.64 | 0.61 |

| | Scenario 2 | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Duration** | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| **#faults** | 96 | 48 | 32 | 24 | 19 | 16 | 13 | 12 | 10 |
| **Specificity** | 0.65 | 0.68 | 0.77 | 0.81 | 0.85 | 0.87 | 0.88 | 0.89 | 0.91 |
| **Sensitivity** | 0.28 | 0.28 | 0.24 | 0.36 | 0.44 | 0.55 | 0.57 | 0.57 | 0.71 |

[3] M. C. Huebscher and J. A. McCann, "A Survey of Autonomic Computing — degrees, models, and applications," *ACM Computer Survey*, vol. 40, no. 3, pp. 1–28, 2008.

[4] Frank, Ove and Strauss, David, "Markov Graphs," *Journal of the American Statistical Association*, vol. 81, no. 395, pp. pp. 832–842, 1986.

[5] Dogandzic, A. and Zhang, B., "Distributed Estimation and Detection for Sensor Networks Using Hidden Markov Random Field Models," *IEEE Transactions on Signal Processing*, vol. 54, no. 8, pp. 3200–3215, 2006.

[6] David Marsh and Richard Tynan and Donal O'Kane and Gregory M. P. O'Hare, "Autonomic wireless sensor networks," *Engineering Applications of Artificial Intelligence*, vol. 17, no. 7, pp. 741 – 748, 2004.

[7] T. D. Chandra and S. Toueg, "Unreliable failure detectors for reliable distributed systems," *Journal of the ACM*, vol. 43, pp. 225–267, 1995.

[8] Yagang Zhang and Jinfang Zhang and Jing Ma and Zengping Wang, "Fault Detection Based on Data Mining Theory," in *International Workshop on Intelligent Systems and Applications.*, 2009, pp. 1–4.

[9] Nicole Sergent and Xavier Défago and Andre Schiper, "Failure Detectors: implementation issues and impact on consensus performance," École Polytechnique Fédérale de Lausanne, Switzerland, Tech. Rep., 1999.

[10] Oka, Anand and Lampe, Lutz, "Incremental distributed identification of Markov random field models in wireless sensor networks," *IEEE Transactions on Signal Processing*, vol. 57, no. 6, pp. 2396–2405, 2009.

[11] R. Brooks, P. Ramanathan, and A. Sayeed, "Distributed target classification and tracking in sensor networks," *Proceedings of the IEEE*, vol. 91, no. 8, pp. 1163–1171, 2003.

[12] T.-Y. Wang, L.-Y. Chang, and P.-Y. Chen, "A collaborative sensor-fault detection scheme for robust distributed estimation in sensor networks," *IEEE Transactions on Communications*, vol. 57, no. 10, pp. 3045 –3058, 2009.

[13] Shekhar, S. and Schrater, P.R. and Vatsavai, R.R. and Weili Wu and Chawla, S., "Spatial contextual classification and prediction models for mining geospatial data," *IEEE Transactions on Multimedia*, vol. 4, no. 2, pp. 174 – 188, 2002.

[14] R. Kindermann, J. Snell, and A. M. Society, *Markov random fields and their applications*. American Mathematical Society Providence, Rhode Island, 1980.

[15] Besag, J., "On the Statistical Analysis of Dirty Pictures," *Journal of the Royal Statistical Society*, vol. B-48, pp. 259–302, 1986.

[16] U. Castellani, A. Fusiello, R. Gherardi, V. Murino, "Automatic selection of MRF control parameters by reactive tabu search," *Image and Vision Computing*, vol. 25, 2007.

[17] S. Madden, "Intel lab data," http://db.csail.mit.edu/labdata/labdata.html.

[18] Guestrin, C. and Bodik, P. and Thibaux, R. and Paskin, M. and Madden, S., "Distributed regression: an efficient framework for modeling sensor network data," *3rd Intl Symposium on Information Processing in Sensor Networks.*, pp. 1 – 10, 2004.

[19] S. Zug, A. Dietrich, and J. Kaiser, "An architecture for a dependable distributed sensor system," *Instrumentation and Measurement, IEEE Transactions on*, vol. 60, no. 2, pp. 408 –419, 02 2011.

DRAFT