

The Perils of Detecting Measurement Faults in Environmental Monitoring Networks

(Invited Paper)

Jayant Gupchup*, Abhishek Sharma†, Andreas Terzis*, Randal Burns*, Alex Szalay‡

*Johns Hopkins University
Computer Science Department
Email: gupchup,randal,terzis@jhu.edu

†University of Southern California
Computer Science Department
Email: abhishek@enl.usc.edu

‡Johns Hopkins University
Physics and Astronomy Department
Email: szalay@jhu.edu

Abstract—Scientists deploy environmental monitoring networks to discover previously unobservable phenomena and quantify subtle spatial and temporal differences in the physical quantities they measure. Our experience, shared by others, has shown that measurements gathered by such networks are perturbed by sensor faults. In response, multiple fault detection techniques have been proposed in the literature. However, in this paper we argue that these techniques may mis-classify events (e.g. rain events for soil moisture measurements) as faults, potentially discarding the most interesting measurements. We support this argument by applying two commonly used fault detection techniques on data collected from a soil monitoring network. Our results show that in this case, up to 45% of the event measurements are misclassified as faults. Furthermore, tuning the fault detection algorithms to avoid event misclassification, causes them to miss the majority of actual faults. In addition to exposing the tension between fault and event detection, our findings motivate the need to develop novel fault detection mechanisms which incorporate knowledge of the underlying events and are customized to the sensing modality they monitor.

I. INTRODUCTION

Wireless sensor networks have been used in a number of environmental monitoring applications [1–3], offering scientists the ability to observe physical phenomena in spatial and temporal granularities not previously possible. In turn, these observations reveal previously unknown physical phenomena and subtle variations (e.g. micro-climates) that scientists could not previously measure.

Alas, environmental monitoring networks introduce their own set of problems: results from early deployments have shown that sensor faults occur occasionally, causing faulty data to be recorded and collected [3–5]. The underlying cause of these faults include incorrect hardware and software design, malfunctioning transducers and low battery levels. Irrespective of their origin, faults need to be detected so the network does not consume its resources in delivering corrupted measurements and these measurements do not pollute the experiment. Given the importance of this problem, a number of *fault detection* techniques have been proposed in the literature (e.g., [6, 7]

among others). While each technique uses a different statistical method to detect faults, they all rely on the assumption that faulty data are inherently *different* from so-called *normal* data.

In this paper we argue that the blanket assumption that all measurements which do not conform to some notion of normalcy are due to faults and thus should be discarded, is a precarious one. One of the goals of environmental monitoring networks are to detect rare and subtle *events*. We buttress this argument by employing two fault detection techniques, initially proposed in [6], to detect faults in a dataset collected from a soil monitoring network we deployed. We then measure how many events (in this context rainfall instances) were classified as faults. Our results show that these techniques can misclassify up to 45% of the events as faults. Moreover, tuning the techniques' parameters such that events are no longer misclassified, leads to a large number of false negatives, that is failing to detect actual faults.

In addition to identifying and quantifying the danger of misclassifying events as faults using specific fault detection algorithms, we provide a list of directions for developing novel fault detection algorithms that are sensitive to events. Specifically, we stress the importance of leveraging the *signatures* of events, as reflected by different modalities, in reducing the number of misclassifications. We observe that the onset of an event can be indistinguishable from a fault. Furthermore, different sensors register delayed versions of the same underlying events, provide another argument for temporarily storing collected measurements before relaying them to the back-end.

This paper has five sections. In the section that follows we review related work in the area of fault and event detection in wireless sensor networks. Section III summarizes the two fault detection techniques, originally presented in [6], we use in this study and describes the types of faults they are designed to detect. In Section IV we present the results of applying these algorithms to data gathered from a soil monitoring network and quantify the percentage of events that are misclassified as



Fig. 1. Environmental monitoring mote deployed in the Jug Bay wetlands sanctuary to monitor soil moisture and soil temperature.

faults. Finally, we close in Section V with a discussion about the requirements for future fault detection algorithms.

A. Events in Environmental Monitoring Networks

We present our work using data from a soil monitoring network we deployed at the Jug Bay wetlands sanctuary. This sanctuary is located along the Patuxent river in Maryland and serves as the habitat for a variety of turtle species, including the Eastern Box turtle. These turtles are of scientific interest because their sex is not determined by sex genes but by the incubation temperature. It has been shown in the lab that a difference of two degrees centigrade is enough to produce male instead of female offsprings. On the other hand, the *in vivo* conditions of box turtle nests have not been observed in the wild. Considering environmental conditions in turtle nests are currently unknown, correlating rare events with nest conditions could reveal valuable information. The network we deployed in Spring 2007 continuously monitored the conditions of three turtle nests until the eggs hatched in September of the same year. We use Tmote Sky motes [8], coupled with ECH₂O EC-5 soil moisture sensors from Decagon and custom soil temperature sensors (see Fig.1). We also measure the temperature inside each mote’s enclosure using the mote’s on-board temperature sensor. We term this reading *box temperature* to differentiate it from soil temperature.

Figure 2 presents an example of how box temperature

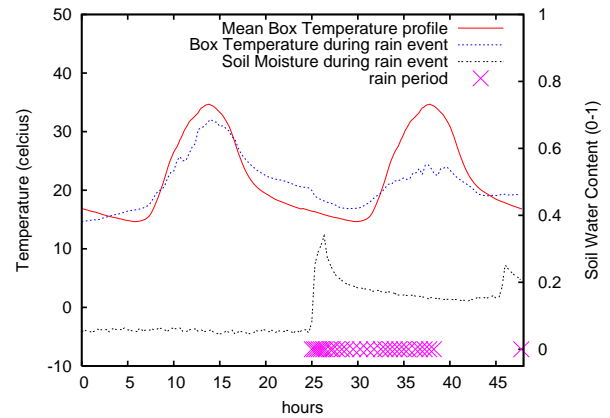


Fig. 2. Reaction of box temperature and soil moisture modalities to a rain event shown on the x-axis. Soil water pressure is measured as the ratio of water volume to the total soil volume. Also shown is the normal box temperature profile, generated by averaging the box temperature at the same time of day during all non-event days.

and soil moisture register a rainfall event. The duration and magnitude of these events are recorded by a weather station co-located with the soil monitoring network. The figure shows a non-event day followed by a day with considerable rain. One can notice that box temperature during the second day clearly differs from the normal diurnal temperature pattern. The reaction of soil moisture is distinctly different—the event’s onset causes a sudden increase in the value recorded by the soil moisture sensor followed by a period of gradual drying of the soil and corresponding decrease in soil moisture. The magnitude of the increase and the duration of the decay period are controlled by the amount and the duration of the rainfall.

II. RELATED WORK

Fault characterization and detection has received significant attention in the sensor network community, starting with the work of Koushhanfar et al. who proposed a cross validation procedure to detect generalized sensor faults in real time [9]. More recently, Ramanathan et al. provided an account of the types and the underlying causes of sensor faults they encountered in three soil sensor deployments [5]. Partially motivated by these findings, Sharma et al. provided a taxonomy of sensor faults and proposed multiple approaches to detect these faults in real and simulated datasets [6]. In this paper, we focus on understanding how existing fault detection mechanisms perform in datasets that contain events which deviate from the norm.

Abadi et al. introduced a declarative approach for detecting sensor events [7]. Specifically, they proposed distributing and storing “event predicates” on a network’s sensor nodes. The nodes then compute in-network joins of the collected data and notify the user when one of the described event predicates are satisfied. We are interested in understanding whether events can be misclassified as faults using fault detection techniques proposed in the literature.

III. METHODOLOGY

We describe the two fault detection techniques we use and present the faults they are designed to detect.

A. Types of Faults

We focus on two types of sensor faults that have been experimentally observed by a number of environmental monitoring networks. Using the terminology coined by Sharma et al. in [6], we consider SHORT and NOISE faults. SHORT faults are characterized by a drastic difference between the current and the previous sensor measurement. On the other hand, a NOISE fault is characterized by a period during which the data samples exhibit larger than normal variations. Sharma et al. also defined the CONSTANT fault type, in which case the standard deviation of the collected samples is (almost) zero. Instead of defining a third category, we expand the definition of NOISE faults to include sets of measurements whose standard deviation is significantly *higher or lower* compared to the overall standard deviation.

B. Fault detection techniques

In order to detect the fault types described above and to study the prevalence of event misclassifications, we implement the heuristic-based and estimation-based techniques presented in [6].

The heuristic-based techniques consist of the SHORT rule and the NOISE rule. In the SHORT rule, whenever the absolute difference between the current and the last measurement is larger than δ , the current measurement is classified as a fault. The appropriate value of δ is obtained from leveraging domain knowledge. The NOISE rule declares a fault whenever the standard deviation (σ_{sample}) of a set of N successive measurements exceeds a threshold. Specifically, if σ_{sample} is not within $\sigma_{train} \pm \sigma_{allow}$, we consider all N samples as faulty. We compute σ_{train} by dividing the training data into sets of N consecutive samples and compute the standard deviation for each of these sets. We then plot the histogram of all standard deviation values and set σ_{train} to be the mean value of the histogram. Furthermore, σ_{allow} is set as an integer multiple of the standard deviation of the histogram. In Section IV, we present the effect of varying σ_{allow} on the misclassification error. Finally, we found empirically that setting the number of samples, N , to the equivalent of a 6-hour time window gave the best results.

The estimation-based technique is an application of linear least-square estimation (LLSE) [10] and leverages any correlations between the measurements collected by spatially distributed sensors. Let i and j be two sensors whose measurements $s_i(t)$ and $s_j(t)$ are correlated. We assume that the correlation can be represented by a linear model and thereby the estimate of $s_i(t)$ based on $s_j(t)$ can be written as:

$$\hat{s}_{ij}(t) = \beta_{0,j} + \beta_{1,j} * s_j(t)$$

Equivalently, in matrix notation,

$$\mathbf{S}_{ij} = \tilde{\mathbf{S}}_j \cdot \beta$$

where \mathbf{S}_{ij} is the vector of \hat{s}_{ij} estimates, $\tilde{\mathbf{S}}_j = [\mathbf{1} \mid \mathbf{S}_j]$ where \mathbf{S}_j is the vector of s_j measurements, and $\beta = [\beta_{0,j} \ \beta_{1,j}]^T$. Using the LLSE formulation, we set β to $(\tilde{\mathbf{S}}_j^T \tilde{\mathbf{S}}_j)^{-1} \tilde{\mathbf{S}}_j^T \mathbf{S}_{ij}$ using measurements from a training set.

Then, the estimation error is $\epsilon_{ij}(t) = \hat{s}_{ij}(t) - s_i(t)$. If $\epsilon_{ij}(t)$ is greater than a threshold, T_{ij} , we consider $s_i(t)$ as faulty. The threshold T_{ij} is set such that $p\%$ of the estimation errors are below T_{ij} when the model is applied to the training set. In practice, we compute the threshold T_{ij} for each of i 's k neighbors and declare the reading $s_i(t)$ as faulty if more than q neighbors have $\epsilon_{ij} > T_{ij}$.

IV. EVALUATION

A. Evaluation Metrics

In order to study the misclassification of events as faults, we need to establish appropriate metrics. The misclassification metric we use has two variants depending on the method under evaluation. First, the SHORT-rule and the LLSE methods classify individual sensor readings as faulty and therefore the misclassification error μ can be defined as

$$\mu = \frac{\text{number of event measurements tagged as faults}}{\text{total number of event measurements}}$$

In this case, an *event measurement* is a sensor reading (e.g. box temperature) during an event (rainfall).

On the other hand, the NOISE-rule method declares sets of N successive samples as faulty and therefore the metric must account for the misclassification duration. Let us say that the i -th event spans E_i samples and let F_i be the number of successive samples that are declared as faulty. Then, all sets of samples D_i within F_i that overlap with E_i contribute to the misclassification error. One can then obtain the total misclassification error by summing over all misclassification instances:

$$\mu = \frac{\sum_i D_i}{\sum_i E_i}$$

Having established a misclassification metric, we need a metric to study the efficacy of the fault detection method itself. The reason is that one can set a method's parameters to minimize the number of misclassifications. Doing so however, might cause the method to fail to detect the actual faults. We use the false negative ratio, defined as the fraction of faults that were not detected by the method to the total number of faults, for this purpose.

B. Data

We apply the fault detection techniques presented in Section III-B to the data obtained from the Jug bay turtle monitoring sensing network [11]. Specifically, we use the box temperature and soil water content (i.e., soil moisture) modalities collected by three motes at the deployment site. The raw data series consists of measurements taken at ten minute intervals, but we use a smoother version by calculating the average of every two sensor readings.

Approximately five months of data was collected from the Jug bay sensor deployment, from 2007/06/22 to 2007/11/27.

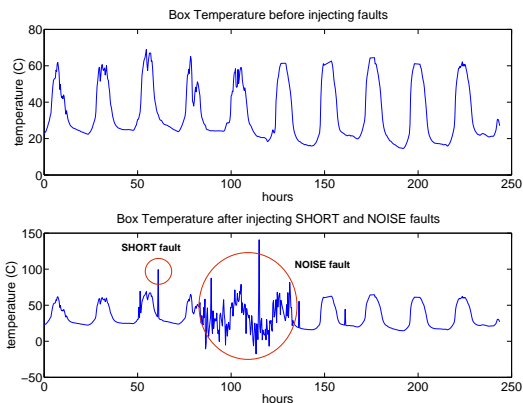


Fig. 3. Box temperature test data after injecting SHORT and NOISE faults.

We use one month of data from each of the sensors for training purposes and the rest as test data. The training data was thoroughly cleaned using a median filter. Moreover, we visualized the data and manually removed any faulty readings to make the training set devoid of faults. The set of events that occurred during the deployment period is gathered from a weather station located approximately 700 meters away from the monitoring site, which records precipitation data at 15-minute intervals [12]. Twenty one major events occurred during the measurement period, spanning a total of 9,480 rainfall minutes (158 hours).

C. Fault Injection

As we mentioned in Section IV-A, we are also interested in the percentage of real faults that the detection algorithms miss. However, in order to calculate this ratio we need to know which measurements correspond to actual faults. Considering that we do not know which actual sensor readings are faulty, we resort to artificially injecting SHORT and NOISE faults. To do so, we use the procedure outlined by Sharma et al. [6].

To inject a SHORT fault, a sample v_i is picked at random and is replaced by the value $\hat{v}_i = v_i + f * v_i$. SHORT faults with intensities $f = \{0.5, 1, 2\}$ and $f = \{0.1, 0.2, 0.5\}$ were injected in the test set for box temperature and soil moisture respectively. To inject a NOISE fault, a set of W successive samples is randomly chosen and random values drawn from the distribution $\sim N(0, \sigma^2)$ are added to the test set. NOISE faults causing an increase of $0.5\times$, $1.5\times$, and $3\times$ in standard deviation (σ) were injected in the box temperature and soil moisture test data sets. The fraction of SHORT faults in the data was set at 1.5%. We inject NOISE faults consisting of 144 and 360 consecutive samples, such that the total number of NOISE faults samples in the data is 6.5%. Note that SHORT faults are ephemeral and are thus much more in number compared to the NOISE faults which last for longer periods of time. Figure 3 provides an example of the artificially injected fault data for box temperature.

TABLE I
MISCLASSIFICATIONS AND FALSE NEGATIVES FOR THE LLSE METHOD

Modality	Misclassification error	False Negatives
Box Temperature	0.3%	77.19%
Soil Moisture	46.3%	50.03%

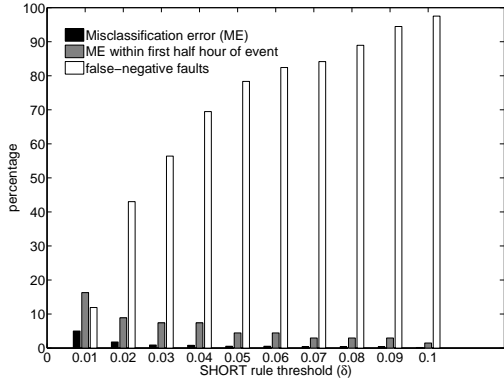
D. Results

In order to study the effect of SHORT faults on misclassification, we evaluate the misclassification error and false negative fraction for the SHORT rule as a function of increasing δ . Figure 4(a) and Figure 4(b) show the results on soil moisture and box temperature respectively. As one would expect, SHORT faults have a higher impact on soil moisture misclassification compared to box temperature because rain events generate measurement spikes that can be misinterpreted as faults. For this reason, we find that a significant proportion of the misclassifications occur in the first half hour of the event period, jeopardizing the most valuable part of the event data. Even though the misclassification error decreases as δ increases, one still observes considerable misclassification errors during the events' first half hour. It is clear that this loss can be mitigated by buffering suspicious data, and leveraging the soil moisture event signature to discriminate them from faults. The SHORT rule works well for box temperature data since box temperature does not show an equivalent leading edge behavior during an event's onset.

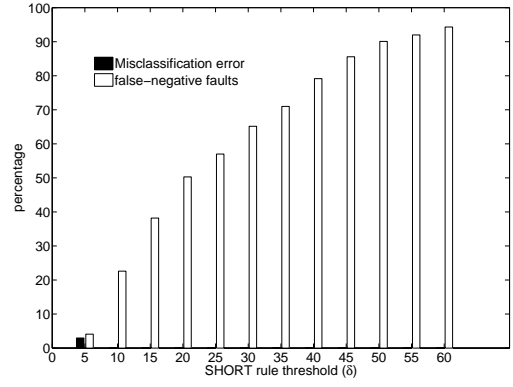
Next, we study the misclassification error for the NOISE rule as a function of σ_{allow} . Figures 4(c) and 4(d) show the performance of the NOISE rule. The persistent misclassification error for soil moisture across all σ_{allow} values can be explained by the observation that soil moisture does not show any variation unless when it spikes in reaction to rain events. Therefore, avoiding misclassifications requires large σ_{allow} values which in turn leads to missing most actual faults.

On the other hand, as Figure 4(d) indicates, increasing σ_{allow} does lead to a sharp decrease in misclassification error for box temperature. Moreover, we found that most of the misclassifications are caused due to the lower side of the rule ($\sigma_{train} - \sigma_{allow}$), which is not surprising as box temperature is known to drop before, during and after a rain event (cf. Fig. 2). At the same time, increasing σ_{allow} has the undesired side effect of increasing the ratio of false negatives by threefold.

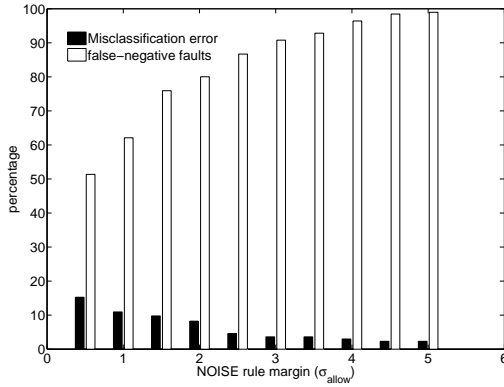
For the LLSE method, we set the confidence interval, p , to 95% and the neighbor-opinion, q , to two. Table I presents our findings for the two sensing modalities. The higher misclassification error associated with soil moisture can be attributed to the observation that soil moisture sensors react differently depending on their location, due to soil's heterogeneity. Figure 5 presents an example of phenomenon, by plotting the reaction of three soil moisture sensors to a rain event. Node 5 and Node 6 tend to move together, whereas the reaction of Node 2 is much lower in magnitude.



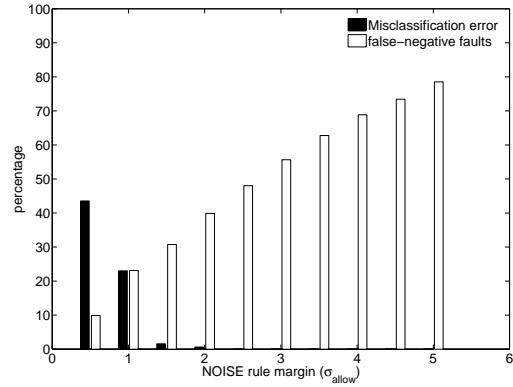
(a) Soil Moisture SHORT-rule



(b) Box Temperature SHORT-rule



(c) Soil Moisture NOISE-rule



(d) Box Temperature NOISE-rule

Fig. 4. Misclassification error and fault negative ratios for soil moisture and box temperature using the SHORT and NOISE rules. Sub-figure (a) also shows the misclassification error during the first half hour of rain events.

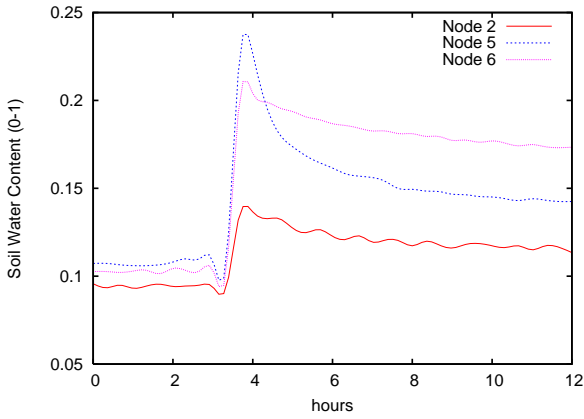


Fig. 5. Reactions of different soil moisture sensors to a rain event.

V. DISCUSSION

The results from Section IV indicate that some fault detection techniques are susceptible to misclassifying events as faults. While more sophisticated techniques might be able to reduce the percentage of misclassifications, the point we want

to raise is that we need a new acid test for *all* fault detection techniques. This test will evaluate their performance in the presence of events. Doing so is crucial, because environmental monitoring networks are in many cases deployed for the purpose of detecting rare and subtle events that deviate from the norm.

As we noted earlier, events have characteristic signatures that are specific to each sensing modality. For example, a rain event causes a sudden increase in the value of measured soil moisture followed by a period of gradual decay. The decay rate is a function of multiple factors, including the soil type, amount of rain, and duration of rain. Furthermore, the *onset* of a rain event is indistinguishable from a SHORT fault. This observation suggests that motes should not prematurely characterize measurements as faults but should rather buffer enough data points to be able to compare *suspicious* measurements against event signatures. The same argument applies to comparing measurements across different motes, because these motes might be registering the same event with different time lags.

While different from the 'baseline' signal, we conjecture that, as the rain example implies, events follow distinct and

common patterns that can be identified and exploited to reduce misclassifications. As part of our previous work, we used a Principal Component Analysis (PCA) based technique to identify the most significant characteristics of the baseline signal (*i.e.* the daily, seasonal cycles) [13]. We believe that a similar methodology can be used to discover the common characteristics of event signals. This information can be then encoded and used to differentiate events from true faults.

Our experience has shown that a high percentage of normal periods (*i.e.* periods during which no events occurred) can be misclassified as event periods [13]. We found that a small fraction of these misclassification were due to faulty sensors that did not accurately measure the underlying physical phenomena. It is therefore desirable to unify the problems of *event* and *fault detection* as a classification problem, in which the algorithm assigns one of three labels to each time period: *normal*, *faulty*, *event*. Furthermore, the assignment of labels should be driven by the application-specific misclassifications costs. For example, misclassifying a normal period as an event period hurts an event-detection application far less than failing to detect an event completely. On the other hand, misclassifying events as faulty data would hurt the application the most.

ACKNOWLEDGMENTS

We extend our gratitude to Chris Swarth, the director of the Jug Bay wetlands sanctuary, and all the staff at the sanctuary for their help and support in deploying the sensor network described in this work. We would also like to thank the members of the LifeUnderYourFeet project (<http://www.lifeunderyourfeet.org/>) for their technical and domain expertise and their constructive criticism.

This research was supported in part by NSF grant CNS-0720730 and by grants from Microsoft Research, the Seaver Institute, and the Moore Foundation. Any opinions, findings, conclusions or recommendations expressed in this publication are those of the authors and do not represent the policy or position of the NSF.

REFERENCES

- [1] R. Musaloiu-E., A. Terzis, K. Szlavecz, A. Szalay, J. Cogan, and J. Gray, "Life Under your Feet: A Wireless Sensor Network for Soil Ecology," in *Proceedings of the 3rd EmNets Workshop*, May 2006.
- [2] L. Selavo, A. Wood, Q. Cao, T. Sookoor, H. Liu, A. Srinivasan, Y. Wu, W. Kang, J. Stankovic, D. Young, and J. Porter, "LUSTER: Wireless Sensor Network for Environmental Research," in *Proceedings of the 5th ACM Sensys Conference*, Nov. 2007.
- [3] G. Tolle, J. Polastre, R. Szewczyk, N. Turner, K. Tu, P. Buonadonna, S. Burgess, D. Gay, W. Hong, T. Dawson, and D. Culler, "A Macroscopic in the Redwoods," in *Proceedings of the 3rd ACM SenSys Conference*, Nov. 2005.
- [4] G. Werner-Allen, K. Lorincz, J. Johnson, J. Lees, and M. Welsh, "Fidelity and Yield in a Volcano Monitoring Sensor Network," in *Proceedings of the 7th USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, 2006.
- [5] N. Ramanathan, T. Schoellhammer, D. Estrin, M. Hansen, T. Harmon, E. Kohler, and M. Srivastava, "The final frontier: Embedding networked sensors in the soil," UCLA, Center for Embedded Networked Computing, Tech. Rep. CENS-TR-68, November 2006.
- [6] A. Sharma, L. Golubchik, and R. Govindan, "On the prevalence of sensor faults in real world deployments," in *IEEE Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*, 2007.

- [7] D. J. Abadi, S. Madden, and W. Lindner, "REED: Robust, Efficient Filtering and Event Detection in Sensor Networks," in *Proceedings of the 31st VLDB Conference*, 2005.
- [8] J. Polastre, R. Szewczyk, and D. Culler, "Telos: Enabling Ultra-Low Power Wireless Research," in *Proceedings of the Fourth International Conference on Information Processing in Sensor Networks: Special track on Platform Tools and Design Methods for Network Embedded Sensors (IPSN/SPOTS)*, Apr. 2005.
- [9] F. Koushanfar, M. Potkonjak, and A. Sangiovanni-Vincentelli, "On-line fault detection of sensor measurements," *Proceedings of IEEE Sensors*, vol. 2, pp. 974–979, Oct. 2003.
- [10] R. Hutchison, *Linear Least-Square Estimation*. Stroudsburg, PA, 1977.
- [11] K. Szlavecz, A. Terzis, R. Musaloiu-E., C.-J. Liang, J. Cogan, A. Szalay, J. Gupchup, J. Klofas, L. Xia, C. Swarth, and S. Matthews, "Turtle Nest Monitoring with Wireless Sensor Networks," in *Proceedings of the American Geophysical Union, Fall Meeting*, 2007.
- [12] National Estuarine Research Reserve, "Jug Bay weather station (cb-mjwbwq)," Available at <http://cdmo.baruch.sc.edu/QueryPages/anychart.cfm>.
- [13] J. Gupchup, A. Terzis, R. Burns, and A. Szalay, "Model-Based Event Detection in Wireless Sensor Networks," in *Proceedings of the Workshop for Data Sharing and Interoperability on the World Wide Web (DSI 2007)*, Apr. 2007.