On the Prevalence of Sensor Faults in Real-World Deployments

Abhishek Sharma*, Leana Golubchik † and Ramesh Govindan*
* Computer Science Department
† Computer Science and EE-Systems Departments, IMSC
University of Southern California, Los Angeles, CA-90089
(Email: absharma,leana,ramesh@usc.edu)

Abstract—**Various sensor network measurement studies have reported instances of transient faults in sensor readings. In this work, we seek to answer a simple question: How often are such faults observed in real deployments? To do this, we first explore and characterize three qualitatively different classes of fault detection methods. Rule-based methods leverage domain knowledge to develop heuristic rules for detecting and identifying faults. Estimation methods predict "normal" sensor behavior by leveraging sensor correlations, flagging anomalous sensor readings as faults. Finally, learning-based methods are trained to statistically identify classes of faults. We find that these three classes of methods sit at different points on the accuracy/robustness spectrum. Rule-based methods can be highly accurate, but their accuracy depends critically on the choice of parameters. Learning methods can be cumbersome, but can accurately detect and classify faults. Estimation methods are accurate, but cannot classify faults. We apply these techniques to four real-world sensor data sets and find that the prevalence of faults as well as their type varies with data sets. All three methods are qualitatively consistent in identifying sensor faults in real world data sets, lending credence to our observations. Our work is a first-step towards automated on-line fault detection and classification.**

I. Introduction

With the maturation of sensor network software, we are increasingly seeing longer-term deployments of wireless sensor networks in real world settings. As a result, research attention is now turning towards drawing meaningful scientific inferences from the collected data [1]. Before sensor networks can become effective replacements for existing scientific instruments, it is important to ensure the quality of the collected data. Already, several deployments have observed faulty sensor readings caused by incorrect hardware design or improper calibration, or by low battery levels [2], [1], [3].

Given these observations, and the realization that it will be impossible to always deploy a perfectly calibrated network of sensors, an important research direction for the future will be automated detection, classification, and root-cause analysis of sensor faults, as well as techniques that can automatically scrub collected sensor data to ensure high quality. A first step in this direction is an understanding of the prevalence of faulty sensor readings in existing real-world deployments. In this paper, we take such a step.

We start by focusing on a small set of sensor faults that have been observed in real deployments: single-sample spikes in sensor readings (we call these SHORT faults, following [2]), longer duration noisy readings (NOISE faults), and anomalous constant offset readings (CONSTANT faults). Given these fault models, our paper makes the following two contributions.

Detection Methods. We first explore three qualitatively different techniques for automatically detecting such faults from a trace of sensor readings. Rule-based methods leverage domain knowledge to develop heuristic rules for detecting and identifying faults. Estimation methods predict "normal" sensor behavior by leveraging sensor correlations, flagging deviations from the normal as sensor faults. Finally, learning-based methods are trained to statistically detect and identify classes of faults.

By artificially injecting faults of varying intensity into

This research has been funded by the NSF DDDAS 0540420 grant. It has also been funded in part by the NSF Center for Embedded Networked Sensing Cooperative Agreement CCR-0120778. This work made use of Integrated Media Systems Center Shared Facilities supported by the National Science Foundation under Cooperative Agreement No. EEC-9529152. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect those of the National Science Foundation.

1-4244-1268-4/07/$25.00 ©2007 IEEE

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE SECON 2007 proceedings.

sensor datasets, we are able to study the detection performance of these methods. We find that these methods sit at different points on the accuracy/robustness spectrum. While rule-based methods can detect and classify faults, they can be sensitive to the choice of parameters. By contrast, the estimation method we study is a bit more robust to parameter choices but relies on spatial correlations and cannot classify faults. Finally, our learning method (based on Hidden Markov Models) is cumbersome, partly because it requires training, but it can fairly accurately detect and classify faults. Furthermore, at low fault intensities, these techniques perform qualitatively differently: the learning method is able to detect more NOISE faults but with higher false positives, while the rule-based method detects more SHORT faults, with the estimation method's performance being intermediate. We also propose and evaluate hybrid detection techniques, which combine these three methods in ways that can be used to reduce false positives or false negatives, whichever is more important for the application.

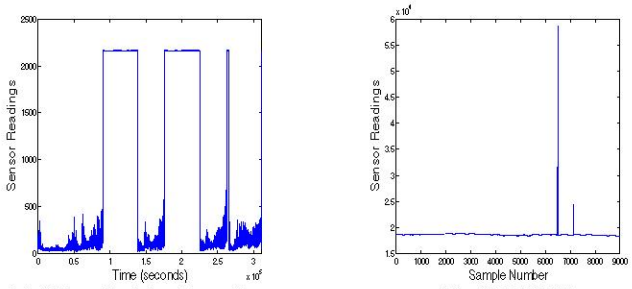
Evaluation on Real-World Datasets. Armed with this evaluation, we apply our detection methods (or, in some cases, a subset thereof) to four real-world data sets. The largest of our data sets spans almost 100 days, and the smallest spans one day. We examine the frequency of occurrence of faults in these real data sets, using a very simple metric: the fraction of faulty samples in a sensor trace. We find that faults are relatively infrequent: often, SHORT faults occur once in about two days in one of the data sets that we study, and NOISE faults are even less frequent. We find no spatial or temporal correlation among faults. However, different data sets exhibits different levels of faults: for example, in one month-long dataset we found only six instances of SHORT faults, while in another 3-month long dataset, we found several hundred. Finally, we find that our detection methods incur false positives and false negatives on these data sets, and hybrid methods are needed to reduce one or the other.

Our study informs the research on ensuring data quality. Even though we find that faults are relatively rare, they are not negligibly so, and careful attention needs to be paid to engineering the deployment and to analyzing the data. Furthermore, our detection methods could be used as part of an on-line fault detection and remediation system, i.e., where corrective steps could be taken during the data collection process based on the diagnostic system's results.

II. Sensor Faults

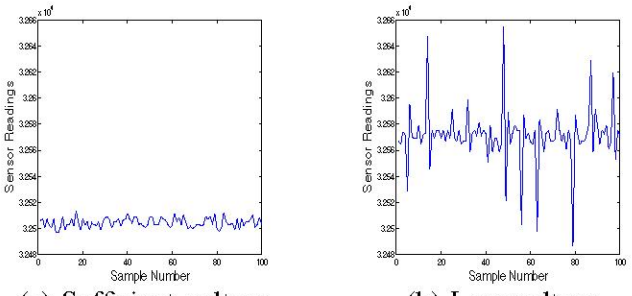
In this section, we visually depict some faults in sensor readings observed in real datasets. These examples are drawn from the same real-world datasets that we use to evaluate the prevalence of sensor faults; we describe details about these datasets later in the paper. These examples give the reader visual intuition for the kinds of faults that occur in practice, and motivate the fault models we use in this paper.

Before we begin, a word about terminology. We use the term *sensor fault* loosely. Strictly speaking, what we call a sensor fault is really a visually or statistically anomalous reading. In one case, we have been able to establish that the behavior we identified was indeed a fault in the design of an analog-to-digital converter. That said, the kinds of faults we describe below have been observed by others as well [2], [1], and that leads us to believe that the readings we identify as faults actually correspond to malfunctions in sensors. Finally, in this paper we do not attempt to precisely establish the *cause* of a fault.



(a) Unreliable Readings (b) SHORT

Fig. 1. Errors in sensor readings



(a) Sufficient voltage (b) Low voltage

Fig. 2. Increase in variance

Figure 1.a shows readings from a sensor reporting chlorophyll concentration measurements from a sensor network deployment in lake water. Due to faults in the analog-to-digital converter board the sensor starts

reporting values $4-5$ times greater than the actual chlorophyll concentration. Similarly, in Figure 1.b, one of the samples reported by a humidity sensor has a value that is roughly 3 times the value of the rest of the samples, resulting in a noticeable spike in the plot. Finally, Figure 2 shows that the variance of the readings from an accelerometer attached to a MicaZ mote measuring ambient vibration increases when the voltage supplied to the accelerometer becomes low.

The faults in sensor readings shown in these figures characterize the kinds of faults we observed in the four data sets from wireless sensor network deployments which we analyze in this paper. We know of two other sensor network deployments [1], [2] that have observed similar faults.

In this paper, we explore the following three fault models motivated by these examples (further details are given in Section IV):

1) **SHORT**: A sharp change in the measured value between two successive data points (Figure 1.b).
2) **NOISE**: The variance of the sensor readings increases. Unlike SHORT faults that affect a single sample at a time, NOISE faults affect a number of successive samples (see Figure 2).
3) **CONSTANT**: The sensor reports a constant value for a large number of successive samples. The reported constant value is either very high or very low compared to the "normal" sensor readings (Figure 1.a) and uncorrelated to the underlying physical phenomena.

SHORT and **NOISE** faults were first identified and characterized in [2] but only for a single data set.

III. Detection Methods

In this paper, we explore and characterize three qualitatively different detection methods – Linear Least-Squares estimation (LLSE), Hidden Markov Models (HMM), and a Rule-based method which leverages domain knowledge (the nature of faults in sensor readings) to develop heuristic rules for detecting and identifying faults. The Rule-based methods analyzed in this paper were first proposed in [2].

Our motivation for considering three qualitatively different detection methods is as follows. As one might expect, and as we shall see later in the paper, no single method is perfect for detecting the kinds of faults we consider in this paper. Intuitively, then, it makes sense to explore the space of detection techniques to understand the trade-offs in detection accuracy versus the robustness to parameter choices and other design considerations. This is what we have attempted to do in a limited way, and our choice of qualitatively different approaches exposes differences in the trade offs.

A. Rule-based (Heuristic) Methods

Our first class of detection methods uses two intuitive heuristics for detecting and identifying the fault types described in Section II.

NOISE Rule: Compute the standard deviation of sample readings within a window N. If it is above a certain threshold, the samples are corrupted by the NOISE fault. To detect CONSTANT faults, we use a slightly modified NOISE rule where we classify the samples as corrupted by CONSTANT faults if the standard deviation is zero. The window size N can be in terms of time or number of samples. Clearly, the performance of this rule depends on the window size N and the threshold.

SHORT Rule: Compute the rate of change of the physical phenomenon being sensed (temperature, humidity etc.) between two successive samples. If the rate of change is above a threshold, it is an instance of a SHORT fault.

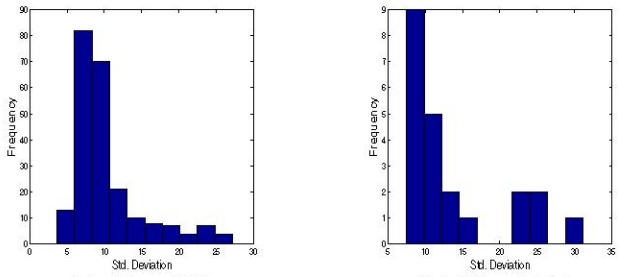
For well-understood physical phenomena like temperature, humidity etc., the thresholds for the NOISE and SHORT rules can be set based on domain knowledge. For example, [2] uses feedback from domain scientists to set a threshold on the rate of change of chemical concentration in soil.

For automated threshold selection, [2] proposes the following technique:

- **Histogram method**: Plot the histogram of the standard deviations or the rate of change observed for the entire time series (of sensor readings) being analyzed. If the histogram is multi-modal, select one of the modes as the threshold.

For the NOISE rule, the Histogram method for automated threshold selection will be most effective when, in the absence of faults, the histogram of standard deviations is uni-modal and sensor faults affect the measured values in such a way that the histogram becomes bi-modal. However, this approach is sensitive to the choice of N; the number of modes in the histogram of standard deviations depends on N. Figure 3 shows the effect of N on the number of modes in the histogram computed for sensor measurements taken from a real-world deployment. The measurements do not contain a sensor fault, but choosing $N = 1000$ gives a multi-modal histogram, and would result in false positives.

Selecting the right parameters for the rule-based methods requires a good understanding of *reasonable* sensor readings. In particular, a domain expert would have



(a) $N = 100$ (b) $N = 1000$

Fig. 3. Histogram Shape

suggested that $N = 1000$ in our previous example was an unrealistic choice.

B. An Estimation-Based Method

Is there a method that perhaps requires less domain knowledge in setting parameters? For physical phenomena like ambient temperature, light etc. which exhibit a diurnal pattern, statistical correlations between sensor measurements can be exploited to generate estimates for the sensed phenomenon based on the measurements of the same phenomenon at other sensors. Regardless of the cause of the statistical correlation, we can exploit the observed correlation in a reasonably dense sensor network deployment to detect anomalous sensor readings.

More concretely, suppose the temperature values reported by sensors s_1 and s_2 are correlated. Let $\hat{t_1}(t_2)$ be the estimate of temperature at s_1 based on the temperature t_2 reported by s_2. Let t_1 be the actual temperature value reported by s_1. If $|t_1 - \hat{t_1}| > \delta$, for some threshold δ, we classify the reported reading t_1 as erroneous. If the estimation technique is robust, in the absence of faults, the estimate error ($|t_1 - \hat{t_1}|$) would be small whereas a fault of the type SHORT or CONSTANT would cause the reported value to differ significantly from the estimate.

In this paper we consider the Linear Least-Squares Estimation (LLSE) [4] method as the estimation technique of choice. In the real-world, the value t_2 at sensor s_2 might itself be faulty. In such situations, we can estimate $\hat{t_1}$ based on measurements at more than one sensor.

In general, the information needed for applying the LLSE method may not be available *a priori* . In applying the LLSE method to the real-world data sets, we divide the data set into a training set and a test set. We compute the mean and variance of sensor measurements, and the covariance between sensor measurements based on the training data set and use them to detect faulty samples in the test data set. This involves an assumption that, in the absence of faults or external perturbations, the physical phenomenon being sensed does not change dramatically between the time when the training and test samples were collected. We found this assumption to hold for many of the data sets we analyzed.

Finally, we set the threshold δ used for detecting faulty samples based on the LLSE estimation error for the training data set. We use the following two heuristics for determining δ:

- *Maximum Error*: If the training data has no faulty samples, we can set δ to be the maximum estimation error for the training data set, i.e. $\delta = \max\{|t_1 - \hat{t_1}| : t_1 \in TS\}$ where TS is the set of all samples in the training data set.
- *Confidence Limit*: In practice, the training data set will have faults. If we can reasonably estimate, e.g., from historical information, the fraction of faulty samples in the training data set, (say) $p\%$, we can set δ to be the upper confidence limit of the $(1 - p)\%$ confidence interval for the LLSE estimation errors on the training data set.

Finally, although we have described an estimation-based method that leverages spatial correlations, this method can equally well be applied by only leveraging temporal correlations at a single node. By extracting correlations induced by diurnal variations at a node, it might be possible to estimate readings, and thereby detect faults, at that same node. We have left an exploration of this direction for future work.

C. A Learning-based Method

For phenomena that may not be spatio-temporally correlated, a *learning-based* method might be more appropriate. For example, if the pattern of "normal" sensor readings and the effect of sensor faults on the reported readings for a sensor measuring a physical phenomenon is well understood, then we can use learning-based methods, for example Hidden Markov Models (HMMs) and neural networks, to construct a model for the measurements reported by that sensor. In this paper we chose HMMs because they are a reasonable representative of learning based methods that can simultaneously detect and classify sensor faults. Determining the most effective learning based method is left for future work.

The states in an HMM mirror the characteristics of both the physical phenomenon being sensed as well as the sensor fault types. For example, based on our characterization of faults in Section II, for a sensor measuring ambient temperature, we can use a 5-state HMM with the states corresponding to day, night, SHORT faults, NOISE faults and CONSTANT faults. Such an HMM

can capture not only the diurnal pattern of temperature but also the distinct patterns in the reported values in the presence of faults. For brevity, we omit a formal definition of HMMs; the interested reader is referred to [5].

D. Hybrid Methods

Finally, observe that we can use combinations of the Rule-based, LLSE, and HMM methods to eliminate/reduce the false positives or negatives. In this paper, we study two such schemes:

- Hybrid(U): Over two (or more) methods, this method identifies a sample as faulty if at least one of the methods identifies the sample as faulty. Thus, Hybrid(U) is intended for reducing false negatives (it may not eliminate them entirely, since all methods might incorrectly flag a sample to be faulty). However, it can suffer from false positives.
- Hybrid(I): Over two (or more) methods, this method identifies a sample as faulty only if both (all) the methods identify the sample as faulty. Essentially, we take an intersection over the set of samples identified as faulty by different methods. Hybrid(I) is intended for reducing false positives (again, it may not eliminate them entirely), but it can suffer from false negatives.

Several other hybrid methods are possible. For example, Hybrid(U) can be easily modified so that results from different methods have different weights in determining if a measurement is faulty. This would be advantageous in situations where a particular method or heuristic is known to be better at detecting faults of a certain type. In situations where it is possible to obtain a good estimate of the correct value of an erroneous measurement, for example with LLSE, we can use different methods in sequence- first correct all the faults reported by one method and then use this modified time series of measurements as input to another method. We have left the exploration of these methods for future work.

IV. Evaluation: Injected Faults

Before we can evaluate the prevalence of faults in real-world datasets using the methods discussed in the previous section, we need to characterize the accuracy and robustness of these methods. To do this, we artificially injected faults of the types discussed in Section II into a real-world data set. Before injecting faults, we verified that the real-world data set did not contain any faults.

This methodology has two advantages. First, injecting faults into a data set gives us an accurate "ground truth" that helps us better understand the performance of a detection method. Second, we are able to control the *intensity* of a fault and can thereby explore the limits of performance of each detection method as well as comparatively assess different schemes at low fault intensities. Many of the faults we have observed in existing real data sets are of relatively high intensity; even so, we believe it is important to understand behavior across a range of fault intensities, since it is unclear if faults in future data sets will continue to be as pronounced as those in today's data sets.

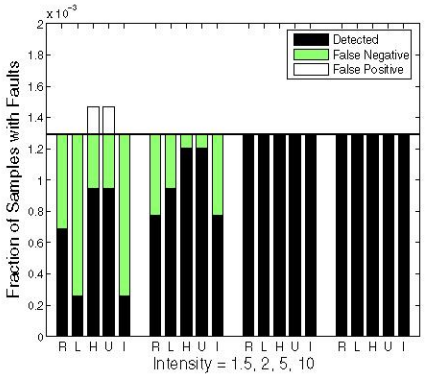


Fig. 4. Injected SHORT Faults

Below, we discuss the detection performance of various methods for each type of fault. We describe how we generate faults in the corresponding subsections. We use three metrics to understand the performance of various methods: the number of faults detected, false negatives, and false positives. More specifically, we use the fraction of samples with faults as our metric, to have a more uniform depiction of results across the data sets. For the figures pertaining to this section and Section V, the labels used for different detection methods are:**R**: Rule-based, **L**: LLSE, **H**: HMM, **U**:Hybrid(U), and **I**: Hybrid(I).

A. SHORT Faults

To inject SHORT faults, we picked a sample i and replaced the reported value v_i with $\hat{v_i} = v_i + f \times v_i$. The multiplicative factor f determines the intensity of the SHORT fault. We injected SHORT faults with intensity $f = \{2, 5, 10\}$. Injecting SHORT faults in this manner (instead of just adding a constant value) does not require knowledge of the range of "normal" sensor readings.

Figure 4 compares the accuracy of the SHORT rule, LLSE, HMM, Hybrid(U), and Hybrid(I) for SHORT faults. The horizontal line in the figure represents the actual fraction of samples with injected faults. The four

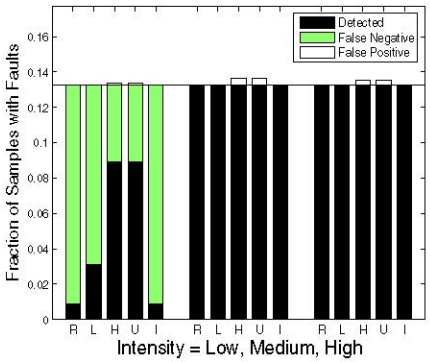


Fig. 5. Injected NOISE Fault: 3000 samples with errors

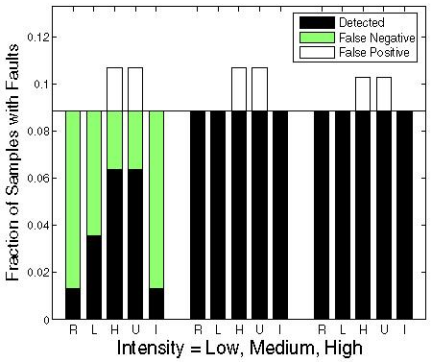


Fig. 6. Injected NOISE Fault: 2000 samples with errors

sets of bar plots correspond to increasing intensity of SHORT faults (left to right).

The SHORT rule and LLSE do not have any false positives; hence, the Hybrid(I) method exhibits no false positives. However, for faults with low intensity ($f = 2$), the SHORT rule as well as LLSE have significant false negatives. The choice of threshold used to detect a faulty sample governs the trade-off between false positives and false negatives; reducing the threshold would reduce the number of false negatives but increase the number of false positives. For the SHORT rule, the threshold was selected automatically using the histogram method and for LLSE the threshold was set using the *Maximum Error* criterion.

The HMM method has fewer false negatives compared to SHORT rule and LLSE but it has false positives for lowest intensity ($f = 2$). While training the HMM for detecting SHORT faults, we observed that if the training data had a sufficient number of SHORT faults (on the order of 15 faults in 11000 samples), the intensity of the faults did not affect the performance of HMMs.

In these experiments, Hybrid(U) performs like the method with more detections and Hybrid(I) performs like the method with fewer detections (while eliminating the false positives). However, in general this does not have to be the case: in the absence of false positives, Hybrid(U) could detect more faults than the best of the methods and Hybrid(I) could detect fewer faults than the worst of the methods (as illustrated on the real data sets in Section V).

B. NOISE Faults

To inject NOISE faults, we pick a set of successive samples W and add a random value drawn from a normal distribution, $N(0, \sigma^2)$, to each sample in W. We vary the intensity of NOISE faults by choosing different

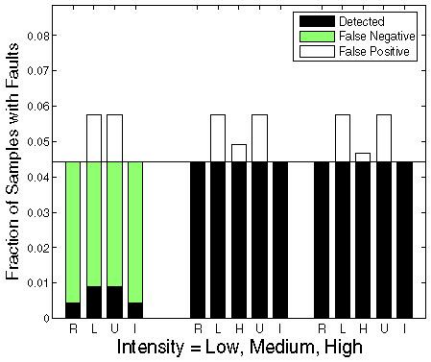


Fig. 7. Injected NOISE Fault: 1000 samples with errors

values for σ. The Low, Medium, and High intensity of NOISE faults correspond to $0.5x$, $1.5x$, and $3x$ increase in standard deviation of the samples in W, respectively. Apart from varying the intensity of NOISE faults, we also vary its duration by considering different numbers of samples in W. The total number of samples in the time series into which we injected NOISE faults was $22,600$.

To train the HMM, we injected NOISE faults into the

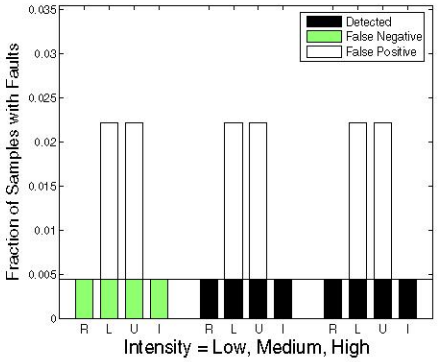


Fig. 8. Injected NOISE Fault: 100 samples with errors

training data. These faults were of the same duration and intensity as the faults used for comparing different methods. There were no NOISE faults in the training data for LLSE. For the NOISE rule, $N = 100$ was used.

Figures 5 ($|W| = 3000$), 6 ($|W| = 2000$), 7 ($|W| = 1000$) and 8 ($|W| = 100$) show the performance of different methods for NOISE faults with varying intensity and duration. The horizontal line in each figure corresponds to the number of samples with faults.

Impact of Fault Duration: The impact of fault duration is most dramatic for the HMM method. For $|W| = 100$, regardless of the fault intensity, there were not enough samples to train the HMM model. Hence, Figure 8 does not show HMM results. For $|W| = 1000$ and low fault intensity, we again failed to train the HMM model. This is not very surprising because for short duration (e.g., $|W| = 100$) or low intensity faults, the data with injected faults is very similar to the data without injected faults. For faults with medium and high intensity or faults with sufficiently long duration, e.g., $|W| \geq 1000$, performance of the HMM method is comparable to those of the NOISE rule and LLSE.

The NOISE rule and LLSE methods are more robust to fault duration than HMMs in the sense that we were able to derive model parameters for those cases. However, for $|W| = 100$ and low fault intensity, both the methods fail to detect any of the samples with faults. The LLSE also has a significant number of false positives for $|W| = 100$ and fault intensity $0.5x$. The false positives were eliminated by the Hybrid(I) method.

Impact of Fault Intensity: For medium and high intensity faults, there are no false negatives for any method. For low intensity faults, all the methods have significant false negatives. For fault duration and intensities for which the HMM training algorithm converged, the HMM method gave lower false negatives as compared to the NOISE rule and LLSE. However, most of the time the HMM method gave more false positives. Hybrid methods are able to reduce the number of false positives or negatives, as intended. High false negatives for low fault intensity arise because the data with injected faults is very similar to the data without faults.

V. Faults in Real-World data sets

We analyze four data sets from real-world deployments for prevalence of faults in sensor traces. The sensor traces contain measurements from a variety of phenomena – temperature, humidity, light, pressure, and chlorophyll concentration. However, all of these phenomena exhibit a diurnal pattern in the absence of outside perturbation or sensor faults.

A. Great Duck Island (GDI) data set

We looked at data collected using 30 weather motes on the Great Duck Island over a period of 3 months [6]. Attached to each mote were temperature, light, and pressure sensors, and these were sampled once every 5 minutes. Of the 30 motes, the data set contained sampled readings from the entire duration of the deployment for only 15 motes. In this section, we present our findings on the prevalence of faults in the readings for these 15 motes.

The predominant fault in the readings was of the type SHORT. We applied the SHORT rule, the LLSE method, and Hybrid(I) to detect SHORT faults in light, humidity, and pressure sensor readings. Figure 9 shows the overall prevalence (computed by aggregating results from all 15 nodes) of SHORT faults for different sensors in the GDI data set. The Hybrid (I) technique eliminates all false positives reported by the SHORT rule or the LLSE method. The intensity of SHORT faults was high enough to detect by visual inspection. This ground-truth is included for reference in the figure under the label **V**.

It is evident from the figure that SHORT faults are relatively infrequent. They are most prevalent in the light sensor data (approximately 1 fault every 2000 samples). Figure 10 shows the distribution of SHORT faults in light sensor readings across various nodes. SHORT faults do not exhibit any discernible pattern in the prevalence of these faults across different sensor nodes; the same holds for other sensors, but we have omitted the corresponding graphs for brevity.

In this data set, NOISE faults were infrequent. Only two nodes had NOISE faults with a duration of about 100 samples. The NOISE rule detected it, but the LLSE method failed primarily because its parameters had been optimized for SHORT faults.

B. INTEL Lab, Berkeley data set

54 Mica2Dot motes with temperature, humidity and light sensors were deployed in the Intel Berkeley Research Lab between February 28th and April 5th, 2004 [7]. In this paper, we present the results on the prevalence of faults in the temperature readings (sampled on average once every 30 seconds).

This dataset exhibited a combination of NOISE and CONSTANT faults. Each sensor also reported the voltage values along with the samples. Inspection of the voltage values reported showed that the faulty samples were well correlated with the last few days of the

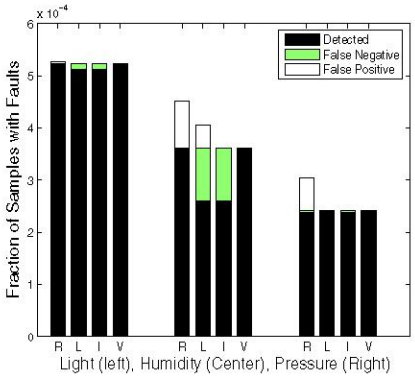


Fig. 9. SHORT Faults in GDI data set

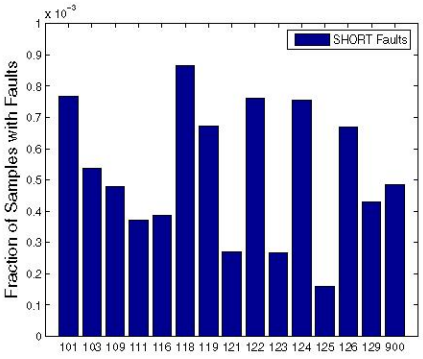


Fig. 10. SHORT Faults: Light Sensor, x-axis: Node ID

deployment when the lithium ion cells supplying power to the motes were unable to supply the voltage required by the sensors for correct operation.

The samples with NOISE faults were contiguous in time, and both the NOISE rule and a simple two-state HMM model identified most of these samples. Figure 11 shows that close to 20% of the total temperature samples collected by all the motes were faulty. Both the NOISE

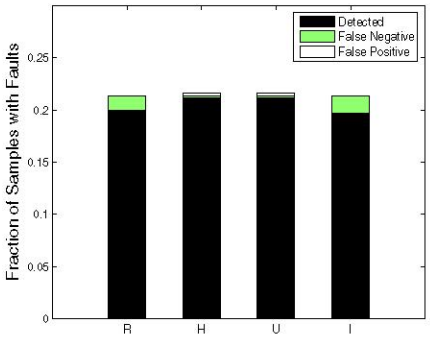


Fig. 11. Intel data set: Prevalence of Noise faults

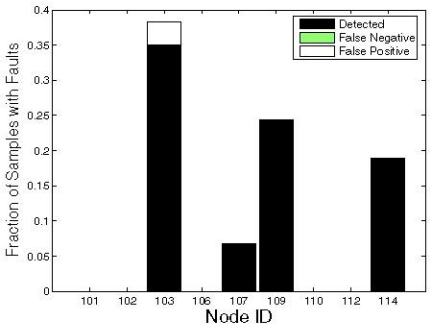


Fig. 12. NAMOS data set (August): NOISE/CONSTANT faults

rule and the HMM have some false negatives while the HMM also has some false positives. For this data set, we could eliminate all the false positives using Hybrid(I) with the NOISE rule and the HMM. However, the Hybrid(I) incurred more false negatives.

Interestingly, for this dataset, we could not apply the LLSE method to detect NOISE faults. NOISE faults across various nodes were temporally correlated, since all the nodes ran out of battery power at approximately the same time. This breaks an important assumption underlying the LLSE technique, that faults at different sensors are uncorrelated.

Finally, in this data set, there were surprisingly few instances of SHORT faults. A total of 6 faults were observed for the entire duration of the experiment (Table I). All of these faults were detected by the HMM method, LLSE, and the SHORT rule.

ID	# Faults	Total # Samples
2	1	46915
4	1	43793
14	1	31804
16	2	34600
17	1	33786

TABLE I
INTEL LAB: SHORT FAULTS, TEMPERATURE

C. NAMOS data set

Nine buoys with temperature and chlorophyll concentration sensors (fluorimeters) were deployed in Lake Fulmor, James Reserve for over 24 hours in August 2006 [8]. Each sensor was sampled every 10 seconds. We analyzed the measurements from chlorophyll sensors for the prevalence of faults.

The predominant fault was a combination of NOISE and CONSTANT caused by hardware faults in the ADC

Station ID	# Faults	Total # samples
3	86	45396
21	1	84818
41	1	60680

TABLE II
SHORT FAULTS IN SENSORSCOPE DATA SET

(Analog-to-Digital Converter) board. Figure 1.a shows the measurements reported by buoy 103. We applied the NOISE Rule to detect samples with errors. Figure 12 shows the fraction of samples corrupted by faults. The sensor measurements at 4 (out of 9) buoys were affected by faults in the ADC board and resulted in more than 15% of erroneous samples at three of them. Buoy 103 was affected the worst, with 35% erroneous samples. We could not apply LLSE and HMM methods because there was not enough data to train the models (data was collected for 24 hours only).

D. SensorScope data set

The SensorScope project is an ongoing outdoor sensor network deployment consisting of weather-stations with sensors for sensing several environmental quantities such as temperature, humidity, solar radiation, soil moisture, and so on [9]. We analyzed the temperature measurements reported once every 15 seconds during November, 2006 by 31 weather stations deployed on a university campus.

We found the occurrence of faults to be the lowest for this data set. The data from only 3 out of the 31 stations that we looked at contained instances of SHORT faults. We identified the faulty samples using the SHORT rule and the LLSE method. Neither of the methods generated any false positives. We did not find any instances of NOISE and CONSTANT faults. Table (II) presents our findings from the SensorScope data set.

VI. RELATED WORK

Two recent papers [10], [11] have proposed a declarative approach to erroneous data detection and cleaning. StreamClean [10] provides a simple declarative language for expressing integrity constraints on the input data. Samples violating these constraints are considered faulty. StreamClean uses a probabilistic approach based on entropy maximization to estimate the correct value of an erroneous sample. The evaluation in [10] is geared towards a preliminary feasibility study and does not use any real world data sets. Extensible Sensor stream Processing (ESP) framework [11] provides support for specifying the algorithms used for detecting and cleaning erroneous samples using declarative queries. This approach works best when the types of faults that can occur and the methods to correct them are known *a priori*. The declarative queries used for outlier detection in [11] are similar to our Rule-based method for SHORT faults. The ESP framework is evaluated using the INTEL Lab data set [7] and a data set from an indoor RFID network deployment.

Koushanfar et al. [12] propose a real-time fault detection procedure that exploits multi sensor data fusion. Given measurements of the same source(s) by n sensors, the data fusion is performed $(n+1)$ times–once with measurements from all the sensors and in the rest of the iterations the data from exactly one sensor is excluded. Measurements from a sensor are classified as faulty if excluding them improves the consistency of the data fusion results significantly. This approach is similar to our HMM model based fault detection because it requires a sensor data fusion model. However, it cannot be used for applications such as volcano monitoring [3] where sensor data fusion is not used; but the HMM based method can be. Simulations and data from a small indoor sensor network are used for evaluation in [12]. Data from real world deployment are not used.

Elnahrawy et al. [13] propose a Bayesian approach for cleaning and querying noisy sensors. However, using a Bayesian approach requires prior knowledge of the probability distribution of the true sensor readings and the characteristics of the noise process corrupting the true readings. In terms of the prior knowledge and the models required, the Bayesian approach in [13] is similar to the HMM based method we evaluated in this paper. The evaluation in [13] does not use any real world data set.

Several papers on real-world sensor network deployments [1], [6], [2], [3] present results on meaningful inferences drawn from the collected data. However, to the best of our knowledge, only [2], [3], [1] do a detailed analysis of the collected data. The aim of [2] is to do root cause analysis using Rule-based methods for on-line detection and remediation of sensor faults for a specific type of sensor network monitoring the presence of arsenic in groundwater. The SHORT and NOISE detection rules analyzed in this paper were proposed in [2]. Werner et al. [3] compare the fidelity of data collected using a sensor network monitoring volcanic activity to the data collected using traditional equipment used for monitoring volcanoes. Finally, Tolle et al. [1] examine spatio-temporal patterns in micro-climate on a single redwood

tree. While these publications thoroughly analyze their respective data sets, examining fault prevalence was not an explicit goal. Our work presents a thorough analysis of four different real-world data sets. Looking at different data sets also enables us to characterize the accuracy and robustness of three qualitatively different detection methods.

VII. Summary, Conclusions, and Future Work

In this paper, we focused on a simple question: How often are sensor faults observed in real deployments? To answer this question, we first explored and characterized three qualitatively different classes of fault detection methods (Rule-based, LLSE, and HMMs) and then applied them to real world data sets. Several other methods—based on time series, Bayesian filter, neural networks etc.— can be used for sensor fault detection. However, the three methods discussed in this paper are representatives of the larger class of these alternate techniques. For example, a time series method would rely on temporal correlations in measured samples at the same node whereas the LLSE method relies on temporal as well as spatial correlation across different nodes. Hence, an analysis of the three methods with injected faults presented in Section IV, not only demonstrates the differences, in terms of accuracy and robustness, between these methods but can also help make an informed opinion about the efficacy of several other methods for sensor fault detection.

We know summarize our main findings. SHORT faults in real data sets were relatively infrequent but of high intensity. In the GDI data set SHORT faults occurred once in two days but the faulty sensor values were often orders of magnitude higher than the correct value. CONSTANT and NOISE faults were relatively infrequent too, but in the INTEL Lab and NAMOS data sets a significant percentage (between $15-35\%$) of samples were affected. Such a high percentage of erroneous samples highlights the importance of automated, on-line sensor fault detection. Except in the INTEL Lab data set, we found no spatial or temporal correlation among faults. In that data set, the faults across various nodes were temporally correlated because all the nodes ran out of battery power at approximately the same time. Finally, we found that our detection methods incurred false positives and false negatives on these data sets, and hybrid methods were needed to reduce one or the other.

Even though we analyzed most of the publicly available real world sensor data sets for faults, it is hard to make general statements about sensor faults in real world deployments based on just four data sets. However, our results raise awareness of the prevalence and severity of the problem of data corruption and can inform future deployments. Overall, we believe that our work opens up new research directions in automated high-confidence fault detection, fault classification, data rectification, and so on. More sophisticated statistical and learning techniques than those we have presented can be brought to bear on this crucial area.

References

[1] G. Tolle, J. Polastre, R. Szewczyk, D. Culler, N. Turner, K. Tu, S. Burgess, T. Dawson, P. Buonadonna, D. Gay, and W. Hong, " A Macroscope in the Redwoods," in *SenSys '05: Proceedings of the 2nd international conference on Embedded networked sensor systems*. New York, NY, USA: ACM Press, 2005, pp. 51–63.

[2] N. Ramanathan, L. Balzano, M. Burt, D. Estrin, E. Kohler, T. Harmon, C. Harvey, J. Jay, S. Rothenberg, and M. Srivastava, " Rapid Deployment with Confidence: Calibration and Fault Detection in Environmental Sensor Networks," CENS, Tech. Rep. 62, April 2006.

[3] 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.

[4] *Linear Least-Squares Estimation*. Hutchison & Ross, Stroudsburg, PA, 1977.

[5] L. Rabiner, " A tutorial on Hidden Markov Models and selected applications in speech recognition," *Proceedings of IEEE*, vol. 77(2), pp. 257–286, 1989.

[6] A. Mainwaring, J. Polastre, R. Szewczyk, and D. C. J. Anderson, " Wireless Sensor Networks for Habitat Monitoring ," in *the ACM International Workshop on Wireless Sensor Networks and Applications (WSNA)*, 2002.

[7] " The Intel Lab, Berkeley data set," http://berkeley.intel-research.net/labdata/.

[8] " NAMOS: Networked Aquatic Microbial Observing System," http://robotics.usc.edu/ namos.

[9] " The SensorScope project," http://sensorscope.epfl.ch.

[10] N. Khoussainova, M. Balazinska, and D. Suciu, " Towards Correcting Inpur Data Errors Probabilistically Using Integrity Constraints," in *Fifth International ACM Workshop on Data Engineering for Wireless and Mobile Access (MobiDE)*., 2006.

[11] S. R. Jeffery, G. Alonso, M. J. Franklin, W. Hong, and J. Widom, " Declarative Support for Sensor Data Cleaning ," in *Fourth International Conference on Pervasive Computing (Pervasive)*, 2006.

[12] F. Koushanfar, M. Potkonjak, and A. Sangiovammi-Vincentelli, " On-line Fault Detection of Sensor Measurements," in *IEEE Sensors*, 2003.

[13] E. Elnahrawy and B. Nath, " Cleaning and Querying Noisy Sensors," in *the ACM International Workshop on Wireless Sensor Networks and Applications (WSNA)*, 2003.