In-network Sensor Data Modelling Methods for Fault Detection

Lei Fang and Simon Dobson

School of Computer Science, University of St Andrews UK
lf28@st-andrews.ac.uk

Abstract. Wireless sensor networks are attracting increasing interest but suffer from severe challenges such as low data reliability. To improve the data reliability, many sensor fault detection techniques have been proposed. Behind these methods, mathematical models are usually employed to serve as comparing metric to find faulty data in the absence of ground truth. In this paper, we firstly discuss sensor data features and their relevance to fault detection. Criteria that should be met to become a competent data model for the purpose of fault detection is summarised. Some existing sensor data modelling methods for fault detection are presented and qualitatively compared.

1 Introduction

Wireless sensor networks (WSNs) typically consist of multiple battery powered sensor nodes, distributed over a large area, measuring and reporting real-world quantities through one or more powerful sink nodes. With the maturation of sensor network software, WSNs applications, which now range from scientific exploration [1], home and health control [2], [3], habitat monitoring [4] and environment monitoring [5] to infrastructure protection [6], have been attracting growing interests from both academia and industry. However, one problem that still prevents the further commercialising WSN technology is the low reliability of data gathered by sensors. It has been found that a substantial portion of the data gathered in real monitoring applications is actually faulty [7]. For example, 51% of the data collected in [1] was faulty; 3-60 % of data collected in the Great Duck Island experiment was incorrect [8]. Other data series [9],[10] collected by WSNs also have been found faulty.

Faulty data occurs when the data gathered and reported by the WSN application deviates from the true sample of the physical environment being measured [8]. Given the criticality of the applications envisioned for WSNs, the data collected by them should be accurate and reliable. To improve the data reliability, many solutions featuring different techniques have been put forward to calibrate sensor readings by filtering out faulty data in a on-line and in-network fashion.

Although server-side error filtering in which filters clean the received data at network sink is always an option, we believe on-line and especially in-network solutions have certain advantages over the traditional method. On-line solutions can provide in-time alerts when things go wrong. Therefore, timely remedies can be given, like replacing the faulty sensors, to avoid the collected data set being completely useless. Moreover, in-network solutions are more scalable and flexible than their server side counterparts. For large scale sensor deployments, centralised solutions usually causes big overheads. And in-network solution is more flexible in that it can carry out casual sensor health inspection without sending all the data entries back to the sink. Furthermore, for some applications, in-network error filter may be the only feasible solution especially for those event driven deployments in which sensors do not send every data entry back to the sink [11]. Last but not least, the low yield of WSNs applications makes server side error filtering not practical. For example, the Redwood project [1] reported only 49 % of expected points are finally received at sink. The error filtering becomes challenging, if not impossible, with incomplete data set.

Most of the sensor fault detectors spot data faults by modelling historical sensor data as a norm and future data series are checked against the models to be classified as either normal readings or faulty data. This model-based solution is widely used because firstly it is a data centric method that usually does not require prior field expert knowledge and human intervention, which are not generally available or applicable for most WSNs applications. In other words, in the absence of ground truth, data model provides the metric for sensor data to be measured against their degree of being a fault. Secondly, the solution also fits the context of WSNs well due to its relative simplicity so that the whole operation can be carried out in a distributed, on-site and on-line manner, as sensor nodes with restricted processing power and memory space usually cannot cope with sophisticated machine learning methods. Thirdly, since mathematical models are usually employed, formal reasoning and inference can be naturally incorporated to the solution so that the solution is sound and accurate. For example, via a sound inference, not only faulty data can be detected but missing data or erroneous data can be reconstructed.

Numerous data modelling methods have been proposed for the task of fault detection. This paper provides a study on these different methods and qualitatively assess them in one metric. Section 3 introduces some sensor data features and their relationships with fault detection. Section 2 gives the definition of sensor data faults and as well as different types of sensor faults. Section 4 presents an evaluation metric for modelling method of sensor fault detection. Different modelling methods are then introduced and qualitatively compared in detail in Section 5. The paper concludes with future work and some research directions of data modelling methods.

2 Faulty Sensor Data

2.1 Data Faults and Data Outliers

Data-faults emerge when a node performs, or is forced to perform, a sensing task in an erroneous way resulting in faulty data which deviates from a true sample of the physical context to be measured. Data-faults in general are generated by either internal (i.e. system faults) or external factors. External source usually involves various kinds of malicious attacks, like unauthorised message spoof or node tampering [12], which lead to the received data altered therefore faulty. Intrusion detection or sensor network security [12], [13], [14], however, is another research topic which is beyond the scope of this paper. On the other side, internal sources include battery failure, weakening battery supply, connection failure, sensor hardware malfunctioning, calibration error, short-circuited connections and so on [15], [7]. Though different the sources, either external or internal, are, they all lead to faulty readings which do not agree with the ground truth of the interest.

One should also note the difference between data outlier and data-fault. Data-fault should be considered as a special kind of outlier. Outliers are data entries that deviate from expected normal patterns, which may either be caused by an unexpected genuine event, for example rainfalls, or other sources like malicious attacks or sensor failures. Sudden changes of the environment may cause turbulent sensor readings, which, however, usually is the main interest of a WSN application. For example, volcano eruption will give rise to radical sensor readings including temperature, humidity, and light; however, monitoring the eruption is the main purpose of the deployment. Therefore, separating real data faults caused by malfunctioning sensors from outliers is crucial for WSN applications as data fault detector may discard valuable information as data faults. But drawing a fine line between them is difficult especially for resource constrained sensors, which requires the data model adaptive and responsive to the changing underlying environment. The relationship between data-fault and outlier, and their corresponding causes are summarized in Fig. 1.

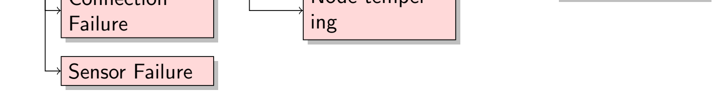


Fig. 1. The relationships between data-outlier and data-fault and their sources.

2.2 Data Fault Types

By analysing the real world sensor data, sensor data faults can be categorised into four categories according to [7], [15]. The four types of faults: short, constant, noise and calibration have been constantly found in different WSNs deployments [1], [9], [10], [16], [17]. The definitions of the three types faults are listed below.

NOISE Sensor readings exhibit an unexpectedly high amount of variation for a period of time. The noisy variance is beyond the expected variation of the underlying phenomenon. Usually high noise is due to a hardware failure or low batteries [15].

SHORT A sharp momentary change in the measured value between normal consecutive readings. Hardware failures like fault in the analog-to-digital convert board may lead to short faults [7].

CONSTANT Also known as "Stuck-at" fault. The readings remain constant for a period of time greater than expected. The reported constant value usually is out of the possible range of the expected normal readings and uncorrelated to the underlying physical phenomena [7].

CALIBRATION Sensor readings may have offsets or incorrect gain, rendering reported data deviating from the true value. Drift faults occur when the offset or gain change with time.

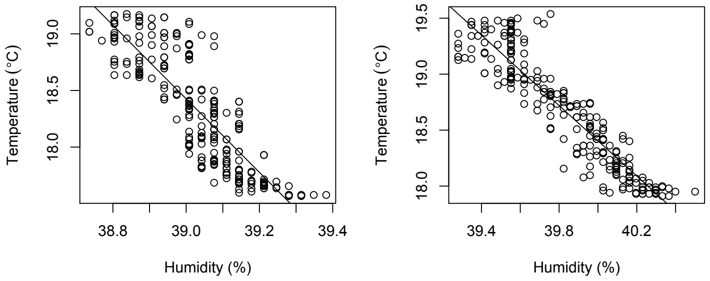
3 Sensor Data Features

3.1 Multi-dimensionality

Sensor data gathered at the sink can be viewed as data stream indexed by time and their locations. For each deployed node, it usually has more than one type of sensor incorporated. The most commonly deployed classes of sensors include temperature, humidity, light, and chemical ones. Each data stream of the on-board sensor classes becomes a univariate time series. Therefore, the ensemble of the data streams becomes multivariate time series. Although different classes of sensor readings exhibit varying statistical features, they share similar fault types listed in Section 2.2; their relative frequencies present in different classes of sensor readings may vary though [15].

The multi-variate nature of sensor readings bring both benefits and difficulties with regard to fault detection for WSNs. The different classes of sensor readings tend to be correlated and such local correlations can be exploited to find outlier data entries in an energy efficient way as no data transmission is required. For example, the readings of humidity and barometric pressure sensors are related to the readings of the temperature sensors [18], as shown in Fig. 2. Capturing this correlation helps to improve the detection accuracy [19].

However, on the other side, higher dimensions bring higher computational complexity as well as modelling difficulty, whose impact on resource constrained sensors is apparent. Moreover, outlier detection may have to check multi-variate outliers as well. Because, as pointed out by Sun [20], occasionally, while each individual attribute reading appears normal, the ensemble of the attributes may display anomaly.



(a) Node 1

(b) Node 2

Fig. 2. Correlation between temperature and humidity sensors by real world sensor data [10]

3.2 Non-stationarity of Sensor Data

Stationarity is an important concept in statistical time series analysis [21]. Roughly speaking, a data series is stationary if its behaviour is self-similar therefore its statistical properties, like the first two moments mean and variance, does not vary according to time. Such an assumption is vital for all fault detection techniques as it provides the legitimacy to apply historic model learnt based on past data to future data series.

However, such an assumption is usually a "fair tale invented for the amusement of undergraduates" [22]. There is a clear loophole in this assumption: if the phenomena of interest were stationary, there would be no need to deploy sensors to monitor the ongoing changes whatsoever as the underlying process had been assumed to remain constant. The non-stationarity can be seen clearly from sensor data plots. As shown in Fig. 3a, the temperature readings change radically along the time stamps. In terms of data modelling for fault detection, dealing with the non-stationary sensor data is crucial as it may decide the detection accuracy of the technique: using a wrong historic model will lead to completely nonsense results.

3.3 Correlation of Sensor Data

Due to the fact that the underlying phenomenon usually is dominated by a smooth continuous process, sensor data tend to be correlated in both time and space, especially for those data collected from environmental monitoring applications [23].

Temporal Correlation Temporal correlation means sensor readings sampled at closer time stamps tend to be similar. In other words, the readings observed at one time instant are related to the readings observed at the previous time instants [24]. Such an assumption is valid for most WSNs deployments because the underlying physical process usually evolves continuously and the sampling frequencies set for WSN applications are usually at sufficient granularity to capture the process smoothly.

Spatial Correlation Spatial correlation implies that the readings from sensor nodes geographically close to each other are expected to be similar, i.e. correlated [24]. This assumption is usually held true because typical node-to-node spacings, usually in the range of 100-200 meters or less, are close enough to measure similar underlying evolving phenomena. Fig. 3a shows spatial correlation as the two data series collected at adjacent nodes exhibit the same pattern.

Capturing the spatio-temporal correlations can not only be used to filter out outlier readings from normal data but also can be used to further distinguish between faulty data and event outliers [18], [25], [19].

4 Evaluation Features of Modelling Methods

In general, a good fault detector should take both detection accuracy and complexity into account. First of all, to form an on-line solution, the cost of constructing and maintaining the model behind the detector should be within the storage and computational capabilities of regular sensor nodes. On the other hand, detection accuracy is the main metric to compare the performance of a fault detector. The faults reported by a detector can be categorised into the following four classes: data points correctly detected as faulty (true positive); data points correctly detected as non-faulty (true negatives); data points incorrectly detected as faulty (false positives); and data points incorrectly detected as non-faulty (false negatives). Good detection accuracy implies the method should be able to filter out the exactly amount of faulty data, i.e. achieve high true positive rate but keep false negative rate low.

Good data modelling methods should possess the following merits to form a accurate but cost effective fault detector.

Lightweight As said before, a lightweight model is essential to produce an online and in-network solution. To be more specific, the model construction process should be lightweight enough to take place in local sensor nodes. Moreover, the learnt model should be lightweight enough to store locally as well.

Accurate Prediction Range Each sensor data model, when applied to future data series, will produce a prediction range as the expected normal data limits; data entries outside this normal range is considered as faults. To achieve good detection accuracy, the range should be carefully selected so that it is neither too wide (lead to low true positive rate), nor too narrow (lead to high false positive rate).

Non-stationarity Resilient As mentioned in Section 3.2, sensor data, by its nature, is non-stationary. Updating the stale model is an option to make the model commensurate with the stochastic phenomena being measured, but it incurs extra computation or communication cost. Ideally speaking, a constant data model, or a model with minimal updates, is the best option for sensor fault detector.

Robust Learning The learning data to construct the model at the first place is unreliable as the rest; therefore, the data modelling method should be robust to the errors present in the learning data. Otherwise, erroneous models may be obtained, leading to detection accuracy degradation.

4.1 Model Data

Models can be constructed upon either raw sensor data or transformed data. Most existing solutions use raw sensor data as learning series, like [7], [8], [26]. However, modelling on pre-processed data shows merits like making the model resilient to non-stationary sensor data [19], [25].

The solutions presented in [19], [25] use synchronised difference between adjacent sensor readings as learning data. After the difference, the new data becomes partially stationary or partially self-similar, i.e. the model learnt by historic data remains true for most future data series. Fig. 3a shows temperature data series from two correlated sensors. It is obvious that, comparing with the original data, the absolute difference, shown in Fig. 3b, is more self-similar. It is shown that the modified data stream becomes stationary in the sense it passes stationarity statistical test and also the majority of future data series agrees upon the historic model learnt by the first 150 data entries [19]. One should, however, note that models built on spatial data differences requires data sharing among local neighbouring nodes at model construction phase, incurring extra data transmissions. Also when it comes to operation phase, sensor data under test also needs to be shipped among neighbouring nodes.

Bettencourt et al. also build statistical models on difference between each node's own measurements at different times by making use of the temporal correlation [25]; and similar results are found.

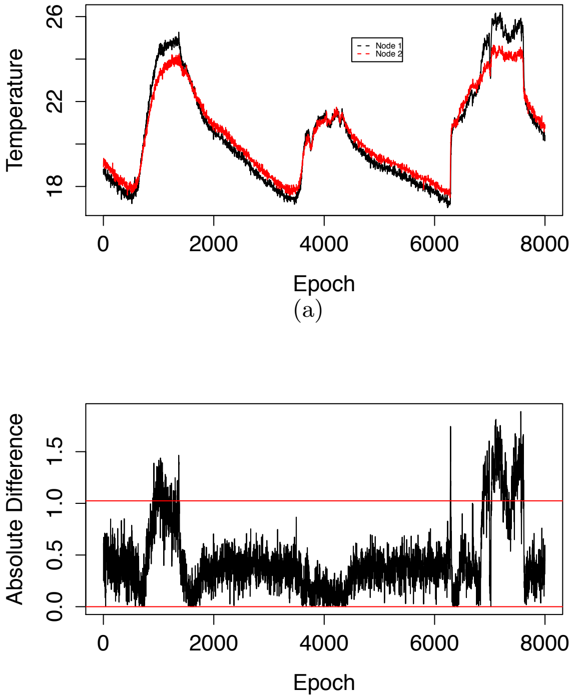
5 Data Modelling Methods

5.1 Regression Based Modelling

Statistical correlations among data attributes can be modelled by regression. In sensor data context, simple linear models are commonly used. For example, the spatial correlation between neighbouring temperature sensors, s_1 and s_2, is modelled as:

$$X = \beta_0 + \beta_1 Y + u_i, \tag{1}$$

where temperature readings from s_1, X, is modelled as a linear combination of its correspondent Y plus some random error u_i [27]. Similarly, linear model between temperature and humidity can also be constructed.



(a)

(b)

Fig. 3. Stationarity of Real World Sensor Data. The top shows the temperature series from two co-located and correlated sensors; The bottom shows the absolute difference between the two series. Over 87% of the future data agree upon the historic model.

Model parameters, β_0, β_1, may be learnt by the ordinary least-squares (OLS) estimation. The OLS estimators have closed-form solutions, as shown in (2).

$$\begin{aligned} \hat{\beta_1} &= \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{\sum_{i=1}^{n}(X_i - \bar{X})^2} \\ \hat{\beta_0} &= \bar{Y} - \hat{\beta_1}\bar{X} \end{aligned} \tag{2}$$

Threshold for fault detection According to the linear model, a prediction interval, which sets the boundary values for the value of interest, can be calculated formally at specific confidence interval:

$$\begin{aligned} X_{new} &\in \hat{\beta_0} + \hat{\beta_1}Y_{new} \pm \varepsilon, \\ \varepsilon &= t_{n-2,\alpha}\hat{\sigma}(1 + \frac{1}{n} + \frac{(Y_{new} - \bar{Y})^2}{S_Y})^{1/2} \end{aligned} \tag{3}$$

where $\hat{\sigma}^2 = \frac{1}{n-1} \sum_{i=1}^{n} u_i^2$ is the residual sum of squares; and $t_{n-2,\alpha}$ is the significance test coefficient obtained from a t-table. When a new pair of observations, say (T_2, T_1), is sampled, the prediction interval can be calculated according to Eqn. 3. Any data entry which is outside of the interval is marked as a fault.

However, some works [27] use user specified error band, $\hat{\varepsilon}$ instead of the regressor-specific error ε. $\hat{\varepsilon}$ may be set as the maximum estimation error, i.e. $max\{u_i\}$, in learning data under the assumption the training data is error free.

Evaluation

Lightweight The solution usually is **lightweight** enough to be carried out at local nodes, as the learning phase only involves the calculation of mean, variance and covariance [19], [28]; and the operational phase requires a storage of three floating numbers.

Accurate Prediction Range The precision largely depends on the selection of error band ε, which involves some domain knowledge. One should also note regression-based solutions can be used to detect multi-variate outliers, as they naturally model the correlation relationship among the multi-dimensional attributes.

Non-stationarity Resilient The model however is not resilient to the non-stationarity of sensor data. As pointed out in [19], the correlation model needs to be updated as the underlying correlation changes from time to time.

Robust Learning Erroneous learning data's effect may be minimised by applying robust regression [19], [29].

5.2 Parametric Statistical Modelling

Parametric method is a statistical modelling method under the assumption that the modelling data has come from a specific type of probability distribution. It then models the data by estimating the distribution parameters. In WSNs applications, Gaussian model is the most commonly assumed distribution. Gaussian model is used because of its computational convenience and also its small model parameter size (a Gaussian distribution can be completely specified by its mean and variance [30]).

The model parameters are usually learnt through the maximum likelihood method [31]. The model parameters are selected based on their corresponding likelihoods towards the data. The maximum likelihood estimators for a Gaussian model are:

$$\begin{aligned} \hat{\mu} &= \bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i \\ \hat{\sigma}^2 &= \frac{1}{n} \sum_{i=1}^{n} (x_i - \bar{x})^2. \end{aligned} \tag{4}$$

Threshold for fault detection Based on the learnt Gaussian distribution, a new data entry, d_i, can be tested by comparing its p-value against some pre-specified significance level α. The p-value is simply the probability of observing a data as or more extreme than d_i, as shown in (5). Commonly used significance level is 0.05 or 0.01 [25].

$$p_i = min\{P(d \leq d_i), P(d \geq d_i)\} \tag{5}$$

Evaluation

Lightweight Parametric solutions with Gaussian assumption is **lightweight** enough to be carried out at local nodes, as the learning phase only involves the calculation of mean, variance; and the operational phase requires a storage of two floating numbers, i.e. mean and variance.

Accurate Prediction Range The precision largely depends on the selection of significance level and also the validity of the Gaussian assumption. For example, the Gaussian assumption is widely made for sensor data; however, its validity remains uncertain.

Non-stationarity Resilient Depends on the model data. If modified data is used, like difference between neighbouring sensors, the model usually is robust to the stochastic evolution [25]. However, if raw sensor data is used, the model needs frequent updates to adapt itself to the changing phenomena.

Robust Learning Robust estimators of model parameters, like median and median absolute deviation (MAD), may be used to counteract the effects of faulty learning data [32].

5.3 Non-Parametric Statistical Modelling

One drawback of parametric modelling is its immature assumption of the data distribution. However, in reality, this priori knowledge is not always available and it may not be even possible to conjecture a good distribution for some data sets. For example, To solve this problem, non-parametric methods model the data without pre-fixing a distribution model and the model is determined from the input data. Histograms [25], [33] and Kernel density methods [34] are the two most widely used approaches in this category [18].

Histogram Modelling The method usually involves two steps. First, a histogram, or a frequency table, is constructed based on the input learning data. Parameters like bin size and number of bins are needed to specify the model. During the following fault detection phase, a new data entry is examined against the histogram. The corresponding frequency of the data entry can be served as an indicator of being an outlier.

Kernel Method This method uses kernel density estimators to approximate the underlying distribution of the data. In essence, the method treat an observed data as an indicator of high probability density in its surrounding region so that data entries close to an observed data are of higher probability densities. After the distribution of the data is approximated, the following fault detection process is done by checking its corresponding probability against the kernel function. A threshold is needed to classify data with low estimated probability as faults.

Evaluation

Lightweight Comparing with parametric methods, non-parametric methods in general involves more calculation to estimate model parameters and occupies larger memory space to store the parameters. For example, the model learning cost for kernel estimation is up to quadratic, comparing with a linear complexity for a parametric method [31].

Accurate Prediction Range The precision largely depends on the selection of parameters both for the model construction and fault threshold. Both [31] and [25] found detection results of histogram modelling largely rely on a good guess of model parameters like bin size and bin numbers. For kernel method, the threshold for outlier detection is also crucial [31].

Non-stationarity Resilient Depends on the model data. If modified data like spatial or temporal data differences are used, the model usually is robust to the stochastic evolution [25]. Otherwise, the model needs to be updated frequently.

Robust Learning Faults in learning data may produce noisy model which later will lead to poor fault detection accuracy. Fault pre-filter may be useful to clean learning data but may rendering in incomplete learning data.

6 Conclusion

In this paper, we firstly investigated sensor data features and their association with fault detection. Sensor data faults were examined with regard to their causes and different types. After the discussion of the desired attributes of a good data modelling method for sensor data fault detection, various existing modelling methods were presented and qualitatively examined against the four attributes. In future work, we plan to carry out a quantitative comparison study of the different modelling methods regarding both their model sizes and fault detection accuracies.

We find all the three categories of modelling methods provide a distance metric to classify data outliers from normal data in the absence of ground truth. However, during both the model construction and fault detection phases, the methods require certain level of user involvement, which largely determines the detection performance. For example, model parameter selection for non-parametric methods and fault threshold selection for all the methods. Selecting appropriate or even adaptive model parameters for different applications to different situations and in different contexts becomes imperative to further improve the detection accuracy. Other research challenges include the detector's ability to draw a fine line between data faults and event outliers; timely but on-demand update of stale model to commensurate the non-stationary sensor data.

Acknowledgements

Lei Fang is supported by a studentship from Scottish Informatics and Computer Science Alliance (SICSA).

References

1. Tolle, G., Polastre, J., Szewczyk, R., Culler, D., Turner, N., Tu, K., Burgess, S., Dawson, T., Buonadonna, P., Gay, D., et al.: A macroscope in the redwoods. In: Proceedings of the 3rd International Conference on Embedded Networked Sensor Systems. (2005) 51–63
2. Noury, N., Hervé, T., Rialle, V., Virone, G., Mercier, E., Morey, G., Moro, A., Porcheron, T.: Monitoring behavior in home using a smart fall sensor and position sensors. In: 1st Annual International, Conference on Microtechnologies in Medicine and Biology. (2000) 607–610
3. Herring, C., Kaplan, S.: Component-based software systems for smart environments. Personal Communications, IEEE **7**(5) (2000) 60–61
4. Szewczyk, R., Mainwaring, A., Polastre, J., Anderson, J., Culler, D.: An analysis of a large scale habitat monitoring application. In: Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems. (2004) 214–226
5. Ingelrest, F., Barrenetxea, G., Schaefer, G., Vetterli, M., Couach, O., Parlange, M.: SensorScope: Application-specific sensor network for environmental monitoring. ACM Transactions on Sensor Networks (TOSN) **6**(2) (2010) 17
6. Xu, N., Rangwala, S., Chintalapudi, K., Ganesan, D., Broad, A., Govindan, R., Estrin, D.: A wireless sensor network for structural monitoring. In: Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems. (2004) 13–24
7. Sharma, A., Golubchik, L., Govindan, R.: Sensor faults: Detection methods and prevalence in real-world datasets. ACM Transactions on Sensor Networks **6**(3) (2010) 23–33
8. Kamal, A.R.M., Bleakley, C., Dobson, S.: Packet-level attestation (pla): A framework for in-network sensor data reliability. ACM Trans. Sen. Netw. **9**(2) (April 2013) 19:1–19:28
9. SensorScope: EPFL SensorScope Project. http://sensorscope.epfl.ch (2008)
10. INTEL: Intel Berkeley Laboratory sensor data set. http://db.csail.mit.edu/labdata/labdata.html (2004)
11. Buratti, C., Conti, A., Dardari, D., Verdone, R.: An overview on wireless sensor networks technology and evolution. Sensors **9**(9) (2009) 6869–6896
12. Pires, W.R., J., de Paula Figueiredo, T., Wong, H., Loureiro, A.A.F.: Malicious node detection in wireless sensor networks. In: Parallel and Distributed Processing Symposium, 2004. Proceedings. 18th International. (2004) 24–

13. da Silva, A.P.R., Martins, M.H.T., Rocha, B.P.S., Loureiro, A.A.F., Ruiz, L.B., Wong, H.C.: Decentralized intrusion detection in wireless sensor networks. In: Proceedings of the 1st ACM international workshop on Quality of service & security in wireless and mobile networks. Q2SWinet '05, New York, NY, USA, ACM (2005) 16–23
14. Bhuse, V., Gupta, A.: Anomaly intrusion detection in wireless sensor networks. Journal of High Speed Networks **15** (2006) 33–51
15. Ni, K., Ramanathan, N., Chehade, M.N.H., Balzano, L., Nair, S., Zahedi, S., Kohler, E., Pottie, G., Hansen, M., Srivastava, M.: Sensor network data fault types. ACM Transactions on Sensor Networks **5**(3) (June 2009)
16. Ramanathan, N., Balzano, L., Burt, M., Estrin, D., Harmon, T., Harvey, C., Jay, J., Kohler, E., Rothenberg, S., Srivastava, M.: Rapid deployment with confidence: Calibration and fault detection in environmental sensor networks. Technical report, Center for Embedded Networked Sensing, UCLA and Department of Civil and Environmental Engineering, MIT (2006)
17. Mainwaring, A., Culler, D., Polastre, J., Szewczyk, R., Anderson, J.: Wireless sensor networks for habitat monitoring. In: Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications. WSNA '02, New York, NY, USA, ACM (2002) 88–97
18. Zhang, Y., Meratnia, N., Havinga, P.: Outlier detection techniques for wireless sensor networks: A survey. Communications Surveys Tutorials, IEEE **12**(2) (2010) 159–170
19. Fang, L., Dobson, S.A., Hughes, D.: An error-free data collection method exploiting hierarchical physical models of wireless sensor networks. In: Proceedings of the 10th ACM symposium on Performance evaluation of wireless ad hoc, sensor, and ubiquitous networks. PE-WASUN '13, New York, NY, USA, ACM (2013) "To Appear".
20. Sun, P.: Outlier detection in high dimensional, spatial and sequential data sets. PhD thesis, Citeseer (2006)
21. Box, G., Jenkins, G.: Time series analysis: forecasting and control. Prentice Hall (1994)
22. Thomson, D.: Jackknifing multiple-window spectra. In: Acoustics, Speech, and Signal Processing, 1994. ICASSP-94., 1994 IEEE International Conference on. Volume vi. (1994) VI/73–VI/76 vol.6
23. Elnahrawy, E., Nath, B.: Context-aware sensors. In: Wireless Sensor Networks. Springer (2004) 77–93
24. Jeffery, S.R., Alonso, G., Franklin, M.J., Hong, W., Widom, J.: Declarative support for sensor data cleaning. In: Proceedings of the 4th international conference on Pervasive Computing. PERVASIVE'06, Berlin, Heidelberg, Springer-Verlag (2006) 83–100
25. Bettencourt, L., Hagberg, A., Larkey, L.: Separating the wheat from the chaff: Practical anomaly detection schemes in ecological applications of distributed sensor networks. In Aspnes, J., Scheideler, C., Arora, A., Madden, S., eds.: Distributed Computing in Sensor Systems. Volume 4549 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2007) 223–239
26. Fang, L., Dobson, S.A.: Unifying sensor fault detection with energy conservation. In: Proceedings of the 7th International Workshop on Self-Organizing Systems. IWSOS '13, Springer (2013) "To Appear".
27. Sharma, A., Golubchik, L., Govindan, R.: On the prevalence of sensor faults in real-world deployments. In: Sensor, Mesh and Ad Hoc Communications and Net-

works, 2007. SECON '07. 4th Annual IEEE Communications Society Conference on. (2007) 213–222

28. Kamal, A.R.M., Bleakley, C.J., Dobson, S.: Congestion mitigation using in-network sensor datasummarization. In: Proceedings of the 9th ACM symposium on Performance evaluation of wireless ad hoc, sensor, and ubiquitous networks. PE-WASUN '12, New York, NY, USA, ACM (2012) 93–100
29. Myers, R.: Classical and modern regression with applications. Volume 2. Duxbury Press Belmont, CA (1990)
30. Ross, S.M.: Introduction to probability models. Academic Press (2006)
31. Han, J., Kamber, M., Pei, J.: Data Mining: Concepts and Techniques. 3rd edn. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (2011)
32. Maronna, R.A., Martin, R.D., Yohai, V.J.: Robust statistics. J. Wiley (2006)
33. Sheng, B., Li, Q., Mao, W., Jin, W.: Outlier detection in sensor networks. In Kranakis, E., Belding, E.M., Modiano, E., eds.: MobiHoc, ACM (2007) 219–228
34. Subramaniam, S., Palpanas, T., Papadopoulos, D., Kalogeraki, V., Gunopulos, D.: Online outlier detection in sensor data using non-parametric models. In: Proceedings of the 32nd international conference on Very large data bases. VLDB '06, VLDB Endowment (2006) 187–198