

Unifying sensor fault detection with energy conservation

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 power constraints and low data reliability. Sensors are often energy-hungry and cannot operate over the long term, and the data they gather are frequently erroneous in complex ways. The two problems are linked, but existing work typically treats them independently: in this paper we consider both side-by-side, and propose a self-organising solution for model-based data collection that reduces errors and communications in a unified fashion.

1 Introduction

Wireless Sensor Network (WSN) applications have been attracting growing interest from both academia and industry. For environmental monitoring, distributed nodes are deployed to sense the environment and return sampled data to the sink, typically at a regular frequency. However, sensors are energy hungry. Various energy-saving techniques [1,2] have been proposed, such as sleep/wakeup protocols, model-based data collection, and adaptive sampling.

Another, related problem is the low reliability of data gathered by sensors. A substantial portion of the data gathered in real monitoring applications is actually faulty [3]. A data-fault can be defined as reading entries, sampled and reported by the sensor nodes, deviating from the true representation of the physical phenomenon to be measured. For example, many data series, such as [4], have been found to be faulty. Deployed WSNs are expected to operate in remote areas for long periods of time without human intervention: manual fault filtering is not an option. To increase data reliability, automatic data fault detection methods become imperative. However, most existing work adopts a centralised and off-line approach to checking the validity of data, which does not validate the sensed data until it has been collected. Recent work [5] has proposed a framework to increase sensor data reliability in a real time.

However, the two domains – energy and error – are typically excluded from each other: sensor fault detection has always been separated from sensor data collection techniques. Existing sensor fault detection methods do not care how the data is collected and whether the collection is energy-efficient or not. Conversely, energy-efficient data collection methods usually assume the validity of sensor data and so do not incorporate sensor fault detection into their collection method. In

reality, these two problems both coexist and co-relate. To achieve energy-efficient, minimum-error data collection, we require a solution combining fault detection and energy-aware data collection.

In this paper, we propose an initial framework addressing both features in a self-organising fashion. We use only local data and communications to learn and autonomously maintain a model of the data expected to be observed by the network, and then uses this model both to pre-filter data to remove probable errors and to reduce the amount of raw data exchanged to reduce energy usage.

Section 2 presents our framework, which is evaluated in section 3 by means of some preliminary simulation results. Section 4 concludes with some directions for future work.

2 A proposed framework

In our proposed framework, models describing the expected behaviour of the sensed phenomena are maintained symmetrically in both network nodes and sink(s). The model is used to predict the measurements sampled by the nodes within some error bounds. Essentially when a data entry sensed is within a predefined error-band (ϵ) of the value calculated by the model, the data entry is exempt from sending to the sink. However, a disagreement between the data and the model may either mean the model is stale, making the model no longer fits the changing phenomenon, or the data is faulty, resulting in it deviating from the model. Therefore, a fault-detection technique is needed to filter out faulty data from “normal” readings so that necessary updates can take place.

The whole life cycle of the framework can be described as follows. The algorithm starts with a learning phase, in which the data model (we use ARIMA model in this work, which is presented in detail in 2.1) and a spatial model (Algorithm 1) is learnt and synced between nodes and sink. Also the fault detection model, presented in detail in 2.2, is also constructed in this phase. Following the learning phase, a error-aware data collection phase starts. In each remote node, it compares data entries sensed against the model. When a disagreement occurs, the data entry will go through the fault detector first, and different actions will be taken according to the result. If the result is positive, i.e. faulty, the data entry will be ignored; otherwise, it will be sent to the sink. The whole process will go back to the learning phase again, when a large number of data entries have to be sent.

Another way to look at this approach is that the WSN develops a model of the data it is observing in a manner akin to what a scientist does when evaluating a data set – but performs this analysis automatically and on-the-fly as data is being collected. The data being collected is therefore being used to maintain and refine the model rather than being treated as uninterpreted by the network, with the model being updated dynamically to reflect the data. While not appropriate for all domains, we believe that this approach is well-suited to systems with missions to detect anomalies, systematic or other deviations from an expected

observational baseline, and this class of systems has wide applicability in domains including environmental monitoring and human safety.

2.1 The ARIMA model

The ARIMA model is widely used for univariate time series modelling, and has been used [2] to form an energy-efficient information-collection algorithm.

ARIMA consists of three terms: the *auto-regressive* (AR), the *moving average* (MA), and an optional *integration* term (I). The AR term is a linear regression of the current data value against prior historical data entries. The MA term captures the influence of random “shocks” to the model. The I term makes the process stationary by differencing the original data series. A first-order differencing of data series X_t , for example, is defined as $X_t^1 = X_t - X_{t-1}$. We use the notation $\text{ARIMA}(p, d, q)$ to define a data series modelled by a ARIMA model with p -lagged historic data entries, q most recent random shocks, and d -order difference:

$$X_t^d = \phi_1 \times X_{t-1}^d + \dots + \phi_p \times X_{t-p}^d + \varepsilon_t + \theta_1 \times \varepsilon_{t-1} + \dots + \theta_q \times \varepsilon_{t-q} \quad (1)$$

In the learning phase, the parameter set, including $\phi_1, \dots, \phi_p; \theta_1, \dots, \theta_q$, is learnt from the learning data by either conditional least squares method or maximum likelihood method. However, if the model include only AR terms, then ordinary least square method can be used, which means the whole computation can be placed in network. To sync the model, only an array of floating numbers are required to send.

2.2 Data fault detection

In this work, in order to place the detector into resource-constrained sensors, we use a simple but effective method to filter out erroneous readings. Spatial correlation is exploited to further validate an observation. To quantify the spatial correlation, each node i maintains a $2 \times |nbr_i|$ spatial matrix, where nbr_i denotes the set of neighbouring nodes of i . The spatial matrix simply stores the maximum and minimum difference in observation between node i and its neighbours¹. We use a voting mechanism to validate a suspect data entry. When a data entry does not match the statistical model, the hosting node will send the data to its neighbours. Upon receiving a validation request, each node will consult the spatial matrix and send back a boolean result accordingly. The final decision will be made according to the results solicited from the neighbours. The data will be marked as faulty if a larger number of neighbours believe so. The spatial correlation between nodes is assumed to be stable, meaning that the spatial matrix remains valid during the whole data collection process, which is a common (although not unassailable) assumption [5].

¹ We assume for simplicity that all data is represented by a continuous range of values. Categorical data requires a slightly different treatment.

```

input :  $nbr_i, x_t^i$ 
output: spatialMatrix
initialise spatialMatrix of size  $2 * |nbr_i|$  ;
for data series  $d_t$  in  $nbr_i$  do
    | spatialMatrix(0,  $i$ )  $\leftarrow \max(|d_t - x_t^i|)$  ;
    | spatialMatrix(1,  $i$ )  $\leftarrow \min(|d_t - x_t^i|)$  ;
end

```

Algorithm 1: Learning Spatial Model Algorithm

3 Preliminary evaluation

We evaluate our proposed framework by simulation over an accepted benchmark real-world data set [4], collected by 54 sensors in Intel’s Berkeley Laboratory. We examine both the fault-detection accuracy and energy-saving features respectively using our time-series and spatial correlation models.

3.1 Fault model

Injecting artificial faults into real data set is a common approach to measure the accuracy of a detector [3]. We consider four particular kinds of faults: short, constant, noise, and drift. The parameters for the fault modelling, which are selected based on existing literatures [3] [5], are provided as well.

SHORT Data $d_{i,t}$ is replaced by $d_{i,t} + d_{i,t} * f$ where f is a random multiplicative factor. f ranging from 0.1 to 10.0 is used here.

CONSTANT Data $d_{i,t}$ for $t \in [t_s, t_e]$ is replaced by some random constant c , and c is randomly selected from 30 to 999.

NOISE Data $d_{i,t}$ for $t \in [t_s, t_e]$ is replaced by $d_{i,t} + x$, where x is a Gaussian random variable, whose distribution is $N(0, \sigma^2)$, and σ is set to be 2.

DRIFT Data $d_{i,t}$ for $t \in [t_s, t_e]$ is replaced by $d_{i,t} + a^{t+1-t_s}$, where $a > 1$. We use natural number e here.

3.2 Results

Table 1 shows the behaviour of the model to the different injected faults as described in the previous section. ARIMA(4,2,0) model is used here with the first 100 data entries as learning data. As there is no MA terms, the whole learning process is tractable to be carried out in remote sensor mote. The error-band, ϵ , is set 0.4 Celsius degree.

It is clear that the proposed solution has high classification accuracy while keeping false alarm rate low. Note that although there is no specific rule to each fault type, for example constant or drift, the solution still exhibits good detection performance. It is also interesting to see that the false alarms rate is zero for all four cases. This is mainly because of the stationarity of the spatial correlation, making neighbouring nodes become good reference models for error detection.

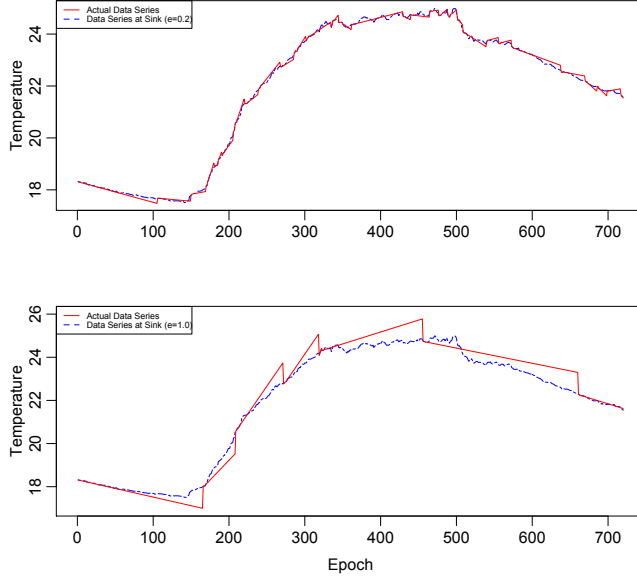


Figure 1. Actual *versus* restored data at the sink node with different error-band (up ($\epsilon = 0.2^\circ\text{C}$), bottom ($\epsilon = 1.0^\circ\text{C}$))

Table 1. Fault Detection Accuracy

SHORT CONSTANT NOISE DRIFT				
True Positive	99.5%	100%	83%	100%
False Positive	0.0%	0.0%	0.0%	0.0%

In assessing the energy efficiency, in the interests of space we only list the performance of one sensor, Sensor 1 from the data set [4]. ARIMA(4, 2, 0) is still used. Figure 1 shows both actual data series sampled at a remote node and the data series restored at the sink as error-band is set 0.2 and 1.0 Celsius degree. It is clear that both data series restored by the ARIMA model closely follows the actual data series but at different granularities. It is obvious that the data from upper graph fits the actual data better. The mean squared error is 0.0068 and 0.2053 respectively. However the accuracy comes at the cost of higher energy consumption: only 76% of the data are exempted from sending back to the sink, while the bottom one saves 95%. But both of them save great amount of energy from sending data back to sink. Note the overall energy saving also depends on the size of data faults. As more data faults present, more extra efforts are paid to clean them out by local communication.

In general, there is a trade-off between energy saving and data integrity and granularity. Better restored data and data integrity is achieved at the price of relative frequent communication, or higher energy cost. The trade-off can be set

by the error-band parameter. Users can choose its value according to specific application requirements.

4 Conclusion

We have presented a framework which features couples fault detection and energy efficiency in data collection. We learn and maintain statistical models of the expected data stream and use these models to reduce communications and improve bad-point rejection. Potentially they may also be used to answer queries at the sink without recourse to the WSN.

We believe that our preliminary results show that these two significant challenges to WSN deployments can be addressed using a unified approach, in a way that maintains the statistical and properties of the data collected. We believe that this suggests a practical approach to further improving WSN longevity in the field.

The work presented here remains proof-of-concept. We are currently implementing our framework on real sensors to perform a more thorough evaluation on the method, and also to test its feasibility in a real-world deployment. Other data modelling and fault detection methods are also being investigated, and we hypothesise that a range of techniques can be incorporated into the same framework to target different classes of sensor network mission. We would also like to move beyond model-based data collection methods to investigate other energy-saving techniques that have the potential to integrate further with the fault detector.

Acknowledgements

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

References

1. Schurgers, C., Tsiatsis, V., Srivastava, M.: STEM: Topology management for energy efficient sensor networks. In: Proceedings of the 3rd IEEE Aerospace Conference Proceedings. (2002)
2. Liu, C., Wu, K., Tsao, M.: Energy efficient information collection with the ARIMA model in wireless sensor networks. In: Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM'05). (2005)
3. 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
4. INTEL: Intel Berkeley Laboratory sensor data set. <http://db.csail.mit.edu/labdata/labdata.html> (2004)
5. Kamal, A.R.M., Bleakley, C., Dobson, S.A.: Packet-Level Attestation: a framework for in-network sensor-data reliability. *ACM Transactions on Sensor Networks* To appear.