Cleaning Environmental Sensing Data Streams Based on Individual Sensor Reliability

Yihong Zhang, Claudia Szabo, and Quan Z. Sheng

School of Computer Science
The University of Adelaide, SA 5005, Australia
{yihong.zhang,claudia.szabo,michael.sheng}@adelaide.edu.au

Abstract. Environmental sensing is becoming a significant way for understanding and transforming the environment, given recent technology advances in the Internet of Things (IoT). Current environmental sensing projects typically deploy commodity sensors, which are known to be unreliable and prone to produce noisy and erroneous data. Unfortunately, the accuracy of current cleaning techniques based on mean or median prediction is unsatisfactory. In this paper, we propose a cleaning method based on incrementally adjusted individual sensor reliabilities, called *influence mean cleaning* (IMC). By incrementally adjusting sensor reliabilities, our approach can properly discover latent sensor reliability values in a data stream, and improve reliability-weighted prediction even in a sensor network with changing conditions. The experimental results based on both synthetic and real datasets show that our approach achieves higher accuracy than the mean and median-based approaches after some initial adjustment iterations.

Keywords: Internet of Things, data stream cleaning, sensor reliability.

1 Introduction

In environmental sensing, sensors are deployed in physical environments to monitor environmental attributes such as temperature, humidity, water pressure, and pollution gas concentration. With the emergence of the Internet of Things (IoT), which connects billions of small devices such as sensors and RFID tags to the Internet, environmental sensing is becoming a significant means towards understanding and transforming the environment [12]. Many IoT-inspired environmental sensing projects have emerged recently, including the Air Quality Egg[1] and the Cicada Tracker[2].

In most environmental sensing projects, commodity sensors are deployed to minimize the cost. Commodity sensors, however, are widely known to be unreliable and prone to producing noisy and erroneous data [2, 8]. Data cleaning is therefore an important issue in environmental sensing, especially when critical realtime decisions need to be made based on the collected data. Recent works

[1] http://airqualityegg.com/
[2] http://project.wnyc.org/cicadas/

B. Benatallah et al. (Eds.): WISE 2014, Part II, LNCS 8787, pp. 405–414, 2014.

have proposed solutions to extract the truthful information from noisy sensor data [8, 14, 16]. A common approach to automatically predict truthful readings is by aggregating spatially correlated readings, using either mean [8, 16] or median [14]. However, it is documented that such approaches have not achieved satisfying accuracy [4].

Intuitively, knowing individual sensor reliability can improve prediction accuracy by, for example, giving unreliable sensors less weight when aggregating the readings. In this paper, we propose a sensor data cleaning technique based on incrementally adjusted individual sensor reliabilities. We adopt a *data-centric* approach to sensor reliabilities, and identify potential sensor malfunctioning through *faulty data*. There are two types of sensor malfunctioning exist, namely, *systematic* and *random* [3]. Faulty data caused by *systematic malfunctioning* typically can be fixed by a single change in the calibration parameter, as proposed by several works [3, 7]. In this paper, we focus on the *random malfunctioning*, which can be caused by unpredictable issues such as sensor damage or battery exhaustion.

Our proposed reliability-based sensor data cleaning method, called *influence mean cleaning* (IMC), weights the mean prediction based on individual sensor reliabilities, and incrementally updates sensor reliabilities based on the readings in each data collecting iteration. We validate our approach extensively by using both synthetic and real datasets. The experimental results show that IMC can significantly improve prediction accuracy over the traditional mean and median methods. When there are sensor condition changes in the network, our method also accurately captures different types of changes, and allows the predictions to adjust to new sensor conditions quickly.

The remainder of this paper is organized as follows. In Section 2, we overview the related work. In Section 3, we present the proposed IMC, which consists of a weighted mean prediction and an incremental reliability update model. We report the experimental results with the simulated and real datasets in Section 4. In Section 5, we provide some concluding remarks.

2 Related Work

A data-centric approach to detect sensor faults has been studied in several research projects. Ni et al. [10] investigated different types of sensor malfunctioning (e.g., battery exhaustion and hardware malfunction) and associated faulty data patterns with them. Sharma et al. [11] identified three faulty data patterns in a number of real datasets, and proposed techniques for their detection. In the evaluation of their techniques, they injected faulty data patterns into known clean data, and used the original clean data as the ground truth. We adopted this data synthesis method when designing our experiments.

A large number of works exist on data cleaning in wireless sensor networks [8, 14–16]. Most of the proposed techniques are based on the assumption that sensor readings are aggregated when transmitted, and individual sensor readings are not available or difficult to obtain. The IoT inspired environmental sensors,

however, are assumed to be connected to the Internet directly, like those used by Devarakonda et al. in [4]. Such direct Internet connection of individual sensors allows individual sensor readings to be accessed and preserved, which creates an opportunity for studying individual sensor behaviors.

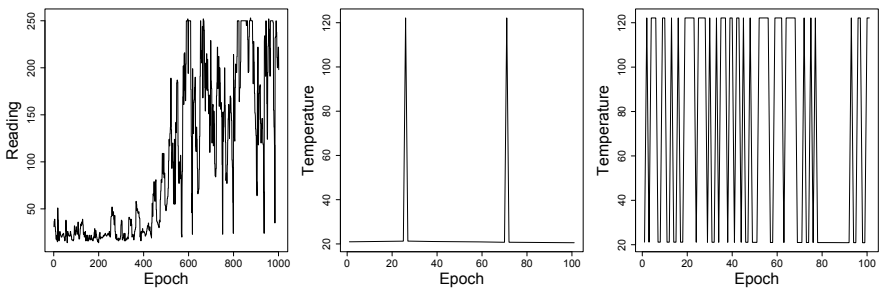
Data source reliability has appeared in truth prediction in information retrieval. In the Web environment, it is not unusual to have multiple data sources that may have different views on a same fact. Most of the existing works are based on probabilistic inference [6, 13]. The probability-based solutions for truth finding, however, are ineffective for environmental sensing data, where sensor reliabilities can be influenced by unpredictable external factors over time. We argue that our incremental update approach is more effective for reflecting unpredictable changes of sensor reliabilities in continuous sensing data streams.

3 Reliability-Weighted Prediction

In this section, we first discuss generic faulty data patterns in real sensor datasets and introduce our proposed data cleaning procedure, which we will explain in two parts: the reliability-based prediction called *influence mean cleaning* (IMC), and the incremental reliability update model.

3.1 The Faulty Data Patterns and the Cleaning Procedure

Sensors can produce noisy and erroneous data when operating in less than ideal working conditions. Fig. 1 shows three patterns of faulty data found in real sensor data that may be caused by sensor malfunctioning. According to the research by Ni et al. [10], *high volatility*, characterized by a sudden rise of variance in the data, can be caused by hardware failure or a weakening in battery supply. *Single spikes*, occasional unusually high or low readings occurred in a series of otherwise normal reading, can be caused by battery failure. *Intense* single spikes that occur with high frequency, may indicate hardware malfunction.



(a) High Volatility (b) Single Spike (c) Intense Spikes

Fig. 1. Faulty sensor data patterns

Our intention is not to detect the type and cause of sensor faults, but to calculate a representative reliability value for each individual sensor that can be used to improve prediction accuracy. In engineering, reliability is defined as "*the probability that a device will perform its intended function during a specified period of time under stated conditions*" [5]. The intended function of an environmental sensor is to generate readings according to the environmental feature that it is monitoring. Consequently, when a sensor produces a reading, the sensor reliability indicates the probability that this reading is the same as the presumed true value.

Our proposed *influence mean cleaning* (IMC) predicts true readings based on incrementally updated individual sensor reliabilities. The general procedure of applying our approach to a sensing data stream is depicted in Fig. 2. Following the data-centric approach, we do not assume any prior hardware information that can be used to infer the reliability of individual sensors, and our approach allows initial reliabilities to be set arbitrarily. The continuous operation of the cleaning method consists of iterations. In each iteration, new sensing data are collected, predictions of true readings are made, and the reliabilities are updated by comparing individual readings to the prediction. The procedure repeats as the data being continuously collected.

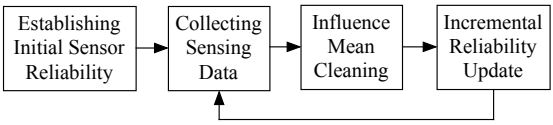


Fig. 2. Continuous cleaning procedure on a sensing data stream

3.2 Influence Mean Cleaning

In an environmental sensing application, the true reading value can be predicted as the mean of the readings made by a group of spatially correlated sensors:

$$P_{MEAN}(R) = \frac{1}{k} \sum R \tag{1}$$

where $R = \{r_1, r_2, ..., r_k\}$ is the set of k readings produced by the spatially correlated sensors.

Suppose the set of the sensors are $\{s_1, s_2, ..., s_l\}$. Let $\{srlb_1, srlb_2, ..., srlb_l\}$ be each sensor's reliability. We can define the *reliability of a reading* as the reliability of the sensor that produced it:

$$rlb(r) = srlb_i, \quad \text{if } r \text{ was produced by } s_i \tag{2}$$

Consequently, $rlb(R) = \{rlb(r_1), rlb(r_2), ..., rlb(r_k)\}$ is the reliability of each reading. The reliability of a reading indicates the probability of the reading being the true reading. Thus we can use a weighted prediction formulated as:

$$P_{IMC}(R) = \frac{\sum R \times rlb(R)}{\sum rlb(R)} \tag{3}$$

We call the prediction defined by Equation (3) *influence mean*, in the sense that it does not aggregates specific readings, but the $influences$ of the sensors on the prediction, which are determined by their reliabilities.

3.3 Incremental Reliability Update

After the prediction is made, the reliability update compares individual readings with the prediction. Since the reading value is typically a real number, it is rare to get two readings exactly the same. Therefore, to compare a reading value with the prediction, we use a tolerance threshold tol. If the difference between the reading value and the prediction is within the threshold, the reading is considered as *consistent* with the prediction. We define the *consistency* of a reading $r \in R$ as the following:

$$cons(r) = \begin{cases} 1, & |r - P_{IMC}(R)| \leq tol \\ 0, & otherwise \end{cases} \tag{4}$$

We calculate the reliability of a sensor as the percentage of the readings made by the sensor that are consistent with the prediction, from the total number of readings it has made during an observation period:

$$srlb = \frac{1}{n} \sum_{i=1}^{n} cons(r_i) \tag{5}$$

where $\{r_1, r_2, ..., r_n\}$ are the readings made by the sensor. When applying the method to continuous streams, the observation period is usually a moving time window with a fixed length. In practice, the choice of observation period length usually depends on the type of temporary interference that can occur in the deployment.

Now we can derive an incremental reliability update formula. Suppose that after making n readings, the reliability calculated for a sensor using Equation (5) is $srlb$. If the sensor has made another reading since then, the new reliability $srlb'$ can be calculated as:

$$srlb' = \frac{1}{n+1} \sum_{i=1}^{n+1} cons(r_i)$$

Substituting Equation (5) into above formula will give:

$$srlb' = srlb \times \frac{n}{n+1} + \frac{1}{n+1} cons(r_{n+1}) \tag{6}$$

Equation (6) can be used as an incremental formula for calculating the new reliability given the current reliability and a new reading. Substituting Equation (4) into the formula gives a *reward or penalty function*, which lets the sensor gain or lose some reliability based on its new reading:

$$srlb' = \begin{cases} srlb + \dfrac{1 - srlb}{n+1}, & \text{if } cons(r_{n+1}) = 1 \\ srlb - \dfrac{srlb}{n+1}, & \text{if } cons(r_{n+1}) = 0 \end{cases} \tag{7}$$

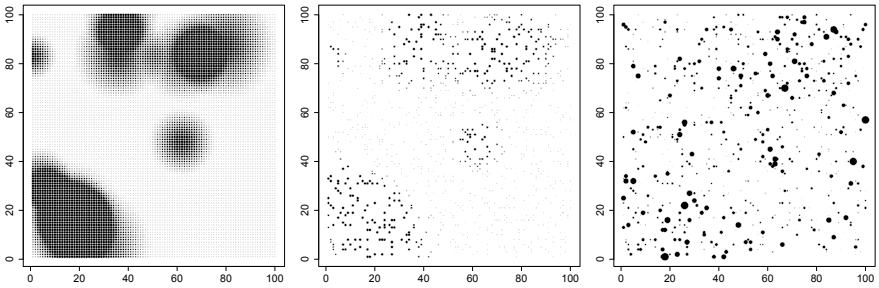
4 Experimental Analysis on Synthetic and Real Dataset

In this section, we describe our experiments for testing our approach on synthetic and real datasets. In both cases, we first obtained a set of clean data, then injected faulty data to simulate sensor malfunctions.

4.1 Influence Meaning Cleaning in a Synthetic Dataset

Our first experiment simulated a scenario of attaching sensors to motor vehicles to monitor air pollution in urban areas. Such a scenario has been run in several projects such as OpenSense, which put air quality sensors on trams in Zurich [9], and Common Sense, which put air quality sensors on street sweepers in San Francisco [1]. The dataset in such projects usually contains sensor readings and time and location of the sensor readings. In addition, each reading is associated with a sensor, which changes its location frequently.

We first simulated a pollution map. The pollution map consists of 100×100 location points, and the corresponding pollution information at each point, as shown in Fig. 3a. The size of a dot on the map indicates the pollution level: the larger the dot, the higher the pollution level at the corresponding location. The maximum pollution level is 1, and the minimum pollution level is 0. We then simulated 20 mobile sensors. In each data collection iteration, a sensor made 50 readings at random locations on the map, and a total of 1,000 readings were made, similar to the readings shown in Fig. 3b. These are clean readings, as they are exactly the same as the ground truth pollution level at their report locations. We used the faulty data injection method introduced in [11] to simulate sensor malfunctioning. We injected *high volatility* faults, similar to those shown in Fig. 1a, which are commonly found in air quality sensors.



(a) Ground (b) Clean Readings (c) Noisy Readings

Fig. 3. The ground truth pollution map and generated readings

We ran the IMC procedure shown in Fig. 2. First we set an initial reliability of 1 for all sensors. In each iteration, we generated a set of noisy readings similar to the one shown in Fig. 3c. To divide the readings into spatially correlated groups, we divided the map into 100 10×10 blocks, and the readings

whose coordinates fall within the same block were grouped. In each iteration, one prediction was made for each block. The ground truth pollution level of each block is calculated as the mean of pollution levels of all location points in the block. We also recorded the predictions of mean and median methods in each iteration. The mean prediction is defined in Equation (1). The median is defined as $P_{MED}(R) = median(R)$, where R is the set of readings in one block.

We measured the precision and the mean square error for the predictions made by three methods in each iteration, as shown in Fig. 4. We notice that the performance of the mean and median methods remain stable over the iteration. The performance of IMC, however, quickly improves in the first 20 iterations, before it becomes stable. The reason is that the reliability update process is picking up appropriate individual reliabilities, thus allowing the reliability-weighted method to become more accurate. After 20 iterations, IMC steadily outperforms the other two methods. For instance, in the last ten iterations of the 50 iteration run, the average precisions for mean, median and IMC are 0.6, 0.78 and 0.85, respectively, while the average mean square error are 0.017, 0.011, and 0.006, respectively. In the long run, the IMC has the potential to have nearly a 10 percent higher precision than the mean and median methods.

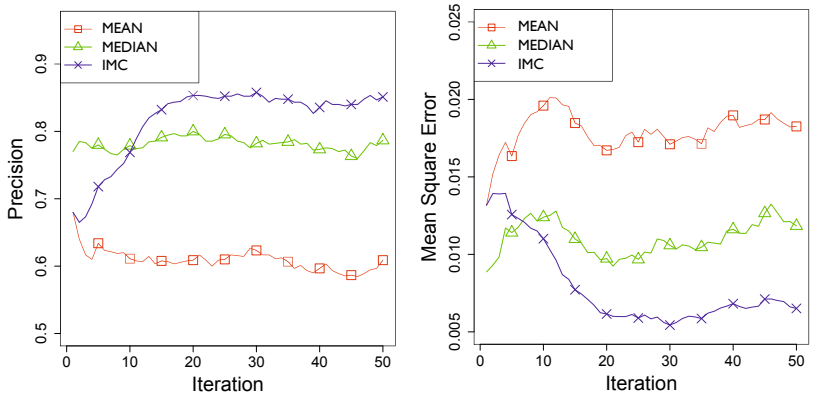


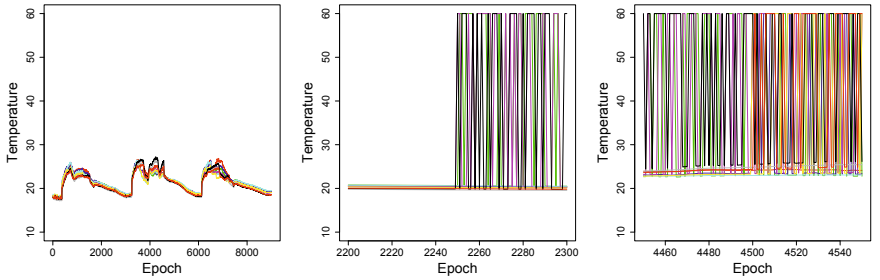
Fig. 4. The precision and mean square error of three prediction methods

4.2 Influence Mean Cleaning in a Real Dataset

We tested our approach on a real dataset provided by the Intel Berkeley Research Lab, called *Intel Lab data*[3]. The data contains environmental readings, such as humidity and temperature, reported by 52 Mica2Dot sensors. The sensors were installed in an indoor area, and had the same reporting frequency of once per 31 seconds. In our experiment, we chose a portion of temperature data in the Intel Lab data made by nine motes with ID 1, 2, 3, 4, 6, 7, 8, 9, 10. These nine

[3] `http://db.csail.mit.edu/labdata/labdata.html`

sensors were installed next to each other in a continuous open area, and can be considered as spatially correlated. We chose a study period of roughly 75 hours, between March 2 and March 5, 2004. There were 9,000 report epochs in this period. We visually confirmed that the readings produced by the nine sensors for these epochs are clean, as shown in Fig. 5a.



(a) Clean Data (b) First Stage Injection (c) Second Stage Injection

Fig. 5. The data used for testing. (b) and (c) show the injected faulty data

To generate the noises, we injected faulty data into the dataset. We injected *intense spikes*, which can be found in other parts of the Intel Lab data. To imitate sensor condition changes over time, we injected the faults in three stages. First we grouped the sensors into three groups: the first group contained sensors 1, 4, 8, the second group contained sensors 2, 6, 9, and the third group contained sensors 3, 7, 10. We made the second group of sensors fail in stage one and two, and the third group of sensors fail in stage two and three. So in the first fault stage, which lasted from epoch 2250 to 4500, readings from the sensors in the second group were injected with faulty data, as shown in Fig. 5b. In the second fault stage, which lasted from epoch 4500 to 6750, readings from the sensors in the second and third groups were injected with faulty data, as shown in Fig. 5c. In the third stage with remaining epochs, only readings from the sensors in the third group were injected with faulty data. When being injected with faulty data, each reading had a probability of 0.5 to be replaced by a spike sensor value (60 in our case).

Similar to our experiments with the synthetic dataset, we ran the IMC procedure with the generated noisy data, as well as the mean and median prediction. Since these nine sensors were considered as a single spatially correlated group, only one prediction was generated in each iteration. We recorded the predictions made by three methods in each iteration. When updating reliabilities, we used a modified version of reward or penalty function, by adding a constant value of 0.01 to the penalty defined in Equation (7). The higher penalty is chosen to mitigate the effect of extremeness of faulty values. How to dynamically change the reward or penalty amount in the case of unpredictable extreme faulty values is a topic of future work.

We measured the performance of the three prediction methods as the square error of the prediction, given the ground truth as the mean of the clean data. Fig. 6a shows predictions of the three methods and the ground truth over 9,000 epochs. Fig. 6b shows the square error of the three methods over 9,000 epochs. To avoid showing high volatility in the graph, the data was smoothed by 100 epochs before plotting.

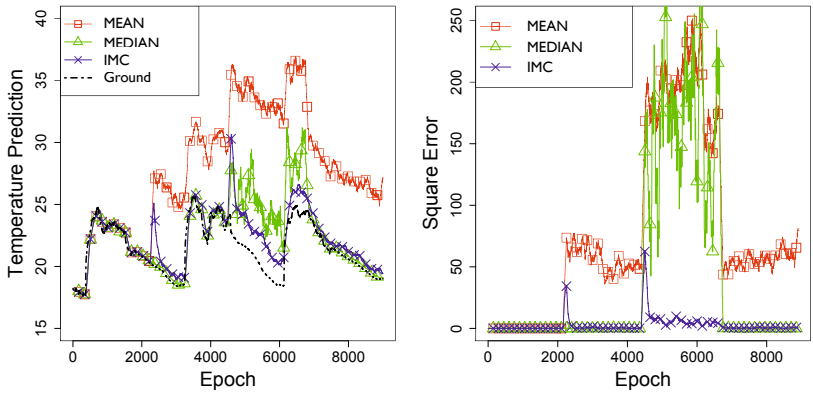


Fig. 6. The prediction and square error of three prediction methods

As shown in the figures, IMC is affected the least by the intense faults in the second stage, and produces only small errors comparing to other methods. At the beginning of the first and second fault stages where the portion of faulty sensors increases, the performance of IMC experiences sudden declines, but can always recover in a short time. This was because our method adjusted the sensor reliability to the new sensor conditions. In the third fault stage, the performance of IMC improves from the second stage, and becomes similar to what it is in the first stage. This adjustment shows that our reliability update process not only detects sensor faults, but also captures sensors' recovery from the faults.

5 Conclusion and Future Work

In this paper, we propose a sensor reliability-based method for sensor data cleaning, called influence mean cleaning (IMC). Our experiment results show that for noisy datasets with different types of faulty data patterns, IMC can achieve higher accuracy than mean and median methods over time. By updating the reliability incrementally, our method can properly discover the latent sensor reliability values. The experimental results with the real dataset from Intel Lab show that our method can capture both sensor malfunctioning and recovery. While individual sensor reliability is largely overlooked in current sensor network research, we show that individual sensor reliabilities can be leveraged to create positive impacts. In the future, we plan to investigate the performance of our approach in datasets with mixed or changing faulty data patterns.

References

1. Aoki, P.M., Honicky, R.J., Mainwaring, A., Myers, C., Paulos, E., Subramanian, S., Woodruff, A.: A vehicle for research: Using street sweepers to explore the landscape of environmental community action. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (2009)
2. Buonadonna, P., Gay, D., Hellerstein, J.M., Hong, W., Madden, S.: Task: Sensor network in a box. In: Proceeedings of the Second European Workshop on Wireless Sensor Networks (2005)
3. Bychkovskiy, V., Megerian, S., Estrin, D., Potkonjak, M.: A collaborative approach to in-place sensor calibration. In: Zhao, F., Guibas, L.J. (eds.) IPSN 2003. LNCS, vol. 2634, pp. 301–316. Springer, Heidelberg (2003)
4. Devarakonda, S., Sevusu, P., Liu, H., Liu, R., Iftode, L., Nath, B.: Real-time air quality monitoring through mobile sensing in metropolitan areas. In: Proceedings of the 2nd ACM SIGKDD International Workshop on Urban Computing (2013)
5. Enrick, N.L.: Quality, reliability, and process improvement. Industrial Press Inc. (1985)
6. Galland, A., Abiteboul, S., Marian, A., Senellart, P.: Corroborating information from disagreeing views. In: Proceedings of the Third ACM International Conference on Web Search and Data Mining (2010)
7. Hasenfratz, D., Saukh, O., Thiele, L.: On-the-fly calibration of low-cost gas sensors. In: Picco, G.P., Heinzelman, W. (eds.) EWSN 2012. LNCS, vol. 7158, pp. 228–244. Springer, Heidelberg (2012)
8. Jeffery, S.R., Alonso, G., Franklin, M.J., Hong, W., Widom, J.: Declarative support for sensor data cleaning. In: Fishkin, K.P., Schiele, B., Nixon, P., Quigley, A. (eds.) PERVASIVE 2006. LNCS, vol. 3968, pp. 83–100. Springer, Heidelberg (2006)
9. Li, J.J., Faltings, B., Saukh, O., Hasenfratz, D., Beutel, J.: Sensing the air we breathe-the opensense zurich dataset. In: Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence (2012)
10. 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), 25:1–25:29 (2009)
11. Sharma, A.B., Golubchik, L., Govindan, R.: Sensor faults: Detection methods and prevalence in real-world datasets. ACM Transactions on Sensor Networks 6(3) 23, 23:1–23:39 (2010)
12. Sheng, Q.Z., Li, X., Zeadally, S.: Enabling next-generation RFID applications: Solutions and challenges. IEEE Computer 41(9), 21–28 (2008)
13. Wang, D., Kaplan, L., Le, H., Abdelzaher, T.: On truth discovery in social sensing: A maximum likelihood estimation approach. In: Proceedings of the 11th International Conference on Information Processing in Sensor Networks (2012)
14. Wen, Y.J., Agogino, A.M., Goebel, K.: Fuzzy validation and fusion for wireless sensor networks. In: Proceedings of the ASME International Mechanical Engineering Congress (2004)
15. Zhang, Y., Meratnia, N., Havinga, P.: Outlier detection techniques for wireless sensor networks: A survey. IEEE Communications Surveys Tutorial 12(2), 159–170 (2010)
16. Zhuang, Y., Chen, L., Wang, X., Lian, J.: A weighted moving average-based approach for cleaning sensor data. In: Proceedings of the 27th International Conference on Distributed Computing Systems (2007)