Real-time Edge Analytics for Cyber Physical Systems using Compression Rates

Sokratis Kartakis and Julie A. McCann, ***Imperial College London***

https://www.usenix.org/conference/icac14/technical-sessions/presentation/kartakis

This paper is included in the Proceedings of the 11th International Conference on Autonomic Computing (ICAC '14).

June 18–20, 2014 • Philadelphia, PA

ISBN 978-1-931971-11-9

Open access to the Proceedings of the 11th International Conference on Autonomic Computing (ICAC '14) is sponsored by USENIX.

Real-time Edge Analytics for Cyber Physical Systems using Compression Rates

Sokratis Kartakis and Julie A. McCann
Department of Computing, Imperial College London, UK
{*s.kartakis13, j.mccann*}*@imperial.ac.uk*

Abstract

There is a movement in many practical applications of Cyber-Physical Systems to push processing to the edge. This is particularly important were the CPS is carrying out monitoring and control, where the latency between the decision making and control message reception should be minimal. However, CPS are limited by the capabilities of the typically battery powered low resourced devices. In this paper we present a self-adaptive scheme that both reduces the amount of resources required to store high sample rate data at the edge and at the same time carries out initial data analytics. Using out Smart Water datasets, plus a selection from other real world CPS applications, we show that our algorithm reduces computation by 98%; data volumes by 55%; while requiring only 11KB of memory at runtime (including the compression algorithm). In addition we show that our system supports self-tuning and automatic reconfiguration which means that manual tuning is alleviated and the scheme can be both applied to any kind of raw data automatically and is able self-optimize as the nature of the incoming data changes over time.

1 Introduction

The work presented in this paper is part of a Smart Water project that both monitors water distribution networks (WDN) and controls its valves to optimize water network performance and lifetime over varying demands. ICT to support WDN typically consist of remote or online battery-powered telemetry units (data loggers) that record water data such as flow and pressure etc periodically over numbers of minutes and aggregate this data and send to a server periodically; typically via the mobile phone networks or 3G. Contemporary approaches use Wireless Sensor Network (WSN) [1], [2], [3], [4], [5] technologies to monitor the status of the water network and detect leakage or water bursts. The main drawbacks of these approaches are: (a) the analysis of the data takes place off-line, in base stations or servers meaning that optimal real time decision-making for control would be unrealistic and (b) the sensor nodes require a lot of energy, which places upper bounds on the amounts of data that can be sensed and relayed for analysis. There is a move to make WDN more dynamic and intelligent using wireless sensors and actuation effecting a CPS to monitor and optimally control the water network in real time, by pushing analytics to the edge and increasing the decision-making capacities of energy-constrained sensor nodes.

Typically such CPS projects monitor the dynamical conditions of the water distribution network. Traditionally this data is sensed at the edge of the network then sent to off-line servers to identify potential failures. Here further analysis via fusion with other data sets, such as customer data may take place. To do this, high precision pressure and flow data, at rates that can exceed 100 samples a second per sensor which can equate to high-precision data averaging at over 512bytes per sec or 0.45Mbytes per 15 minutes. If the system has to transmit this amount of data in 15-minute intervals then the communication process alone will drain the battery of the sensor node rapidly. Therefore, our aim is to reduce the energy cost related to the communication without sacrificing the precision of the data. To this end we evaluated a number of lossless compression algorithms.

During this evaluation, using real data, we observed a correlation between compression rate and data value fluctuation, and from this derived a scheme that enables the identification of transients or failures in the WDN. This means that instead of compressing raw sensor data and sending it to servers to be decompressed and then analyzed for anomalies, we can use the compression rate to detect anomalies and outliers directly on the sensor node. This is faster, more lightweight and provides early indications of an issue, which can be fed directly into the control function without having to communicate via servers saving time and energy. Furthermore, we have

expanded the system using ideas inspired from active learning to support optimal selection of the algorithms input parameters to enable self-tuning and automatic reconfiguration.

This paper is organized as follows: Section 2 contains our evaluation setup of the compression algorithm and presents our correlation observations. Section 3 describes our anomaly detection algorithm showing that the compression rate can be used in the indirect analysis of raw data. Section 4 presents the cross-evaluation system for the selection of optimal parameters. Section 5 describes the execution of the system using other kind of datasets, and section 6 discusses future work and concludes the paper.

2 Compression Rate and Raw Data Correlation

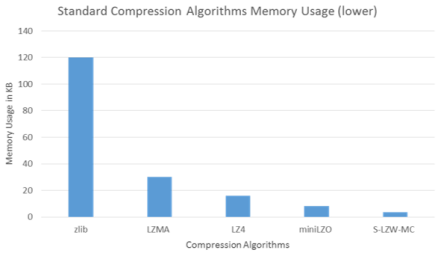

Figure 1: Compression algorithms evaluation.

In order to reduce the energy consumption that would be consumed by high data rate transmissions, while maintaining a high data precision, each sensor node uses lossless compression. Our choice takes memory and energy constraints of our devices into account, so computation and memory intensive algorithms are inappropriate, in spite of their potentially better compression rates. For example, current ultra low power MCUs have 64Kbytes memory[6], therefore we limit compression to 10K.

MiniLZO [7] (coding method is sliding window - LZ77), requires 8.192KB memory at runtime, and S-LZW-MC [8] (coding method is dictionary - LZ78), requires 3.250KB of memory (Figure 1). We evaluated the compression rate of each algorithm which we adapted to use in embedded platforms. Three different real datasets (Datasets A, B, and C - Figure 2 black line), provided by a large UK water company from their loggers were used [1]. Each data set consists of 5.5 million data pairs. In the evaluation the input stream is converted into 512-byte packets with the following structure: (a) timestamp (8-byte double data type), (b) 62 measurements (8-byte double * 62 = 469 bytes), and (c) CRC (8-byte double data type).

[1] We anonymize the company and dataset names for privacy reasons.

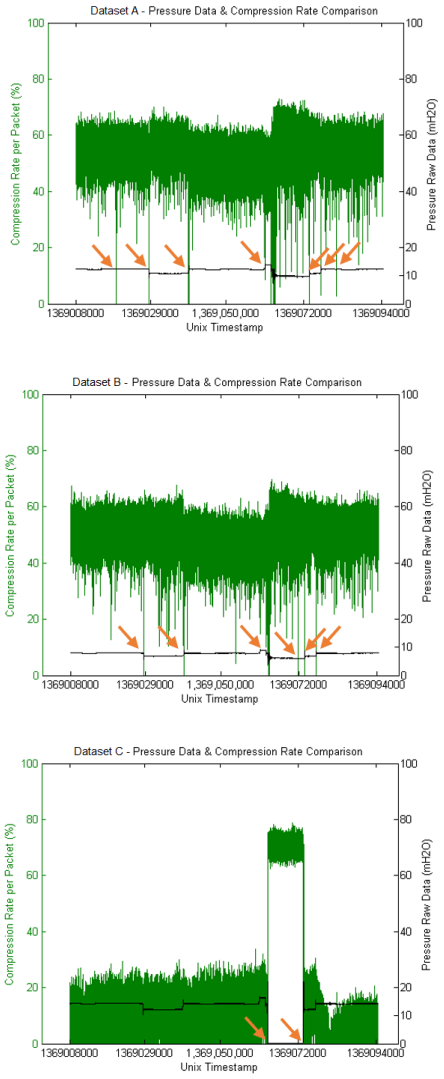

Figure 2: Compression rate & raw data comparison of Dataset A, B, and C.

Using the miniLZO compression algorithm, the results of the compression rate per packet can be seen in the three charts of Figure 2. Note that the original water pressure data is also overlaid on the same graphs in black. It is clear from Figure 2, that these traces highlight data anomalies as indicated by the orange arrows. From this, we formed the hypothesis that we could use the correlation of the compression rate and raw data.

We verified offline that the anomalies indicated on our graphs were true. From water technician logs we observed that they were valve position changes which were used to simulate water bursts, causing significant pressure data fluctuation. At these points the compression algorithm is unable to compress the data so the compression rate falls to 0%. In Figure 2, the drop in compression

rate isolates the areas of raw data where the fluctuation pattern is changeable. The Dataset C is more problematic and the water pressure fluctuation is quite high (see Figure 2c) resulting in the compression rate averaging at 16%. Further, in the same dataset a great drop in water pressure occurs (because the local valve was closed for a small period) which impacts compression rate, which increases to 70%.

Confident that our hypothesis was confirmed and we could use compression rate fluctuation to detect instabilities in high sample rate data, we derived an analytics algorithm that is performed at the end of the compression stage. This has the added advantage that the analytics cost m times less in terms of scale and complexity, where m is the number of measurements per packet. Because each packet contains 62 measurements (m = 62), the produced compression rates are approximately 89,000 (5,518,000 total measurements / 62 measurements per packet). Thus, the analysis is applied to 98% less values.

3 Anomaly Detection Algorithm

We produce a scheme to automatically detect significant changes in compression rate and therefore identify the timestamps of anomalies. To maximize the anomaly detection while minimizing the number of false-positive results, noise is removed from the compression rate stream using a one-dimensional Kalman Filter [9], [10] indicated in Figure 3b with a blue line. The use of Kalman filters is motivated by: (a) its support of streaming analysis using only the current input measurement (and therefore is memory efficient), (b) no matrix calculations are required (therefore it is computationally efficient), (c) ease of the algorithm tuning process, and (d) implementation simplicity.

For every new data value input, the Kalman Filter algorithm uses and updates the Kalman state. The Kalman state consists of the process noise covariance q, the measurement noise covariance r, the actual value x after noise removal, the estimation error covariance p, and the Kalman gain k. During the initialization process the parameters which need tuning are the noise q, the sensor noise r, the initial estimated error p and the initial value of x. The Kalman filter was manually initialized using the following parameters: q = 0.005, r = 25, p = 0, and x = the first compression rate measurement. In every new measurement, the algorithm updates the Kalman state using the following steps:

```
1: x = x
2: p = p + q
3: k = p / (p + r)
4: x = x + k * (measurement  x)
5: p = (1  k) * p
```

After noise removal, the anomalies can be detected accurately because according to Figure 3b (which presents

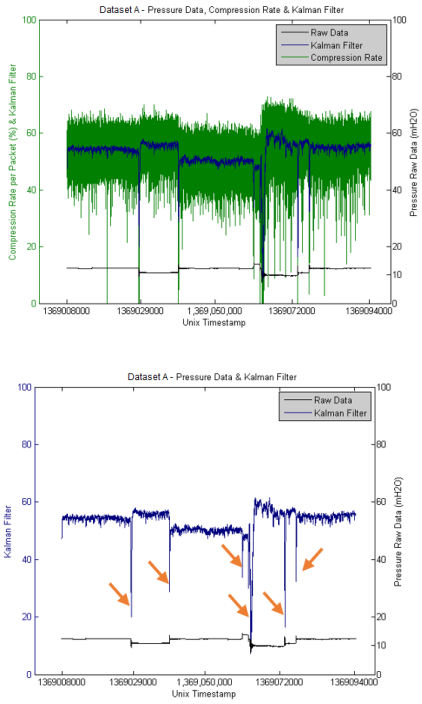

Figure 3: Apply Kalman Filter to Dataset A and drop detection - (w = 128, q = 0.005, r = 25).

the x value of Kalman Filter state and raw data) the anomalies are presented as great drops (orange arrows). The drops are being detected by using the average and the standard deviation of the compression rate moving average for a predefined window size w. We use this because it smoothes the states for easier analysis and reduces threshold computation to window sizes. Specifically, the algorithm computes the moving average of compression rate data with a window size w = 128 Kalman Filter x measurements, with the average *avg* and the standard deviation *std* of the moving average. In every Kalman state update, the algorithm checks if:

```
(Kalman state x) > (avg + std * l) ||
(Kalman state x) < (avg - std * l)
```

Where l represents the elasticity of the outlier detection (smaller values mean that the system is more sensitive - in Figure 4a l=3 and in b l=1.5). As can be observed in Figure 4a (Dataset C), the algorithm suffers from a cold start effect (it identifies the first values to be outliers because the moving average is not calculated). To solve this problem, the algorithm initializes the *avg* and computes *std* by using the current compression rate value. Furthermore, another problem occurs when a significant variation of compression rate data is detected (Figure 4a). In that case, because the standard deviation has a high value, the algorithm needs more intervals for the moving

average calculations to detect the outliers or anomalies. The solution is to reset the values, that is to initialize the *avg* and *std*, every time the distance between the boundaries created by the standard deviation become greater than a specific threshold t (in our system the threshold t = 35).

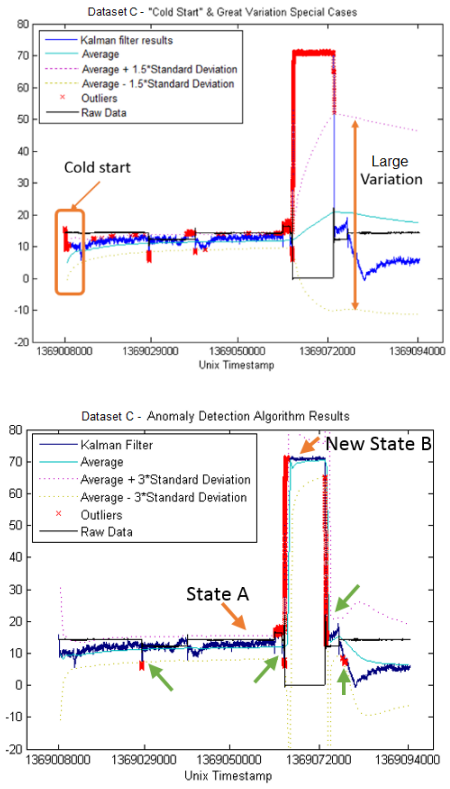

Figure 4: Dataset C - (a) "Cold Start", Large Variation and unelastic outliers detection (l = 1.5), and (b) fixed algorithm results (l=3) - (w = 128, q = 0.005, r = 25).

Figure 4b shows that this solves the cold start and large variation problems and illustrates the anomalies based on normalized data (green arrows). Furthermore, another benefit is that the algorithm can be adapted to changes in the behavior of the data stream. For example, Figure 4b, the algorithm detected the anomaly (red markers x value = timestamp) when the compression rate changes from 10% to 70% as there is no immediate drop (State A to new State B), the algorithm recognizes that the system has a new steady state (B) until the next drop from 70% to 10%. Therefore, this shows that the algorithm adapt extremely fast to new conditions/states.

We applied this approach to Dataset A and B, and Figure 4 presents the results for Dataset A(raw data = black line). The red x markers are the anomalous values detected; the green arrows illustrate the process of matching the timestamps between compressed and raw data.

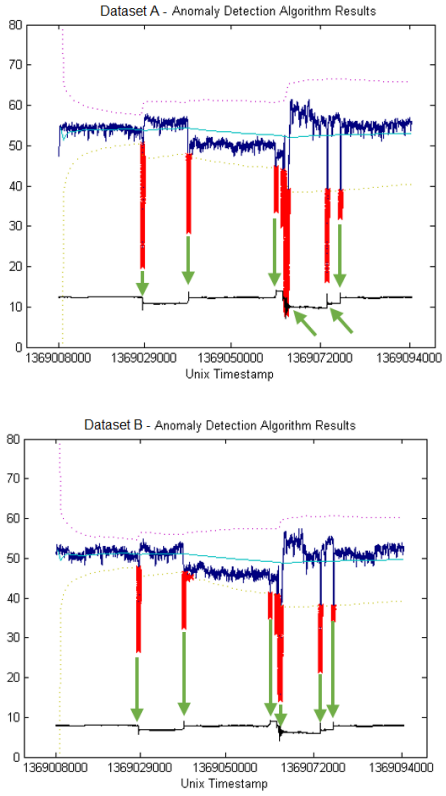

Figure 5: Algorithm results (l=3) for (a) Dataset A and (b) Dataset B - (w = 128, q = 0.005, r = 25).

4 Input Parameters Optimal Tuning

According to the above analysis, by tuning the input parameters, our algorithm can be applied to any case of high sample rate anomaly detection in hardware-constrained sensor nodes. However, to maximise the performance of this, an element of tuning is required as observed in the previous section. Table 1 aggregates all the tuning parameters required by our algorithm.

Table 1: Algorithm input parameters

Process	Parameters
Input stream split	Packet size m
Input stream data precision	Measurement bytes
Kalman Filter initialization	Noise q Sensor noise r Initial estimated error p
Moving average computation	Window size w
Boundaries creation	Elasticity l
Great variation threshold	Threshold t

The initialization of input parameters using a manual approach is inappropriate because it requires permutations of all the different combinations of parameters val-

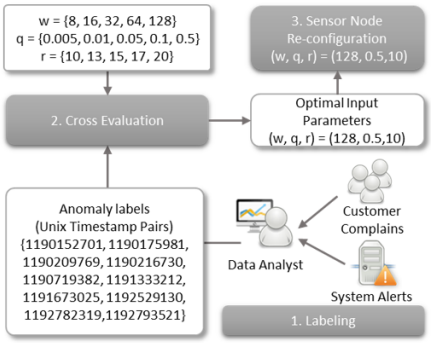

Figure 6: Sensor node re-configuration process.

ues. In order to optimize the algorithms tuning process, we borrow ideas from active learning techniques [11]. This approach requires feedback from real users or an offline system to identify true anomalies, but this is not onerous[2]. Here true anomalies for a single representative training dataset are labeled.

For the results that we present here, we applied the active learning idea by asking water data technicians to manually label anomalies on a subset of our evaluation data. Then, we created an offline cross-evaluation system of Figure 6, which uses our algorithm and calculates the correct, false/positive (FP), and true/negative (TN) anomaly detections based on the initial labeling. This establishes the optimal input parameters as the combination that maximizes the following distance:

```
Distance D = [Correct - (FP + TN)] Detections
```

Using this, one can imagine that a system would update parameters to re-configure the in-node anomaly detection algorithm over time. The data analysis component could recognize that the system requires reconfiguration using customer complains (anomalies are being missed) and system alarms (which increase when the water network is unstable).

Before the creation of the cross-evaluation system, the algorithm's input parameters for datesets were selected manually. In the previous section we show that correct anomalies were identified however here we show that the cross-evaluation system further improves the algorithms accuracy significantly. The reason is that manual observation of high sample rate data is difficult because the high density data. For example, Figure 7 presents the results of the Distance D of each different combination of the following parameter sets for Dataset A:

```
w = {64, 128, 512, 1024}
q = {0.001, 0.005, 0.05}
```

[2]Data anomaly detection can be confirmed off-line automatically by correlating candidate stream data anomalies with other data sets such as customer or water technician records.

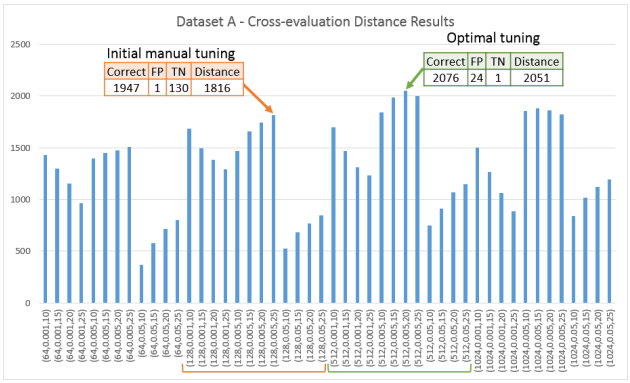

Figure 7: Cross evaluation distance calculation results - Dataset A.

```
r = {10, 15, 20, 25}
```

The orange arrow on Figure 7 are showing input parameters we derived manually, which were $w = 128$, $q = 0.005$, $r = 25$ (Figure 7 orange combinations). The initial selection of parameters was made on a moving average window w = 128 because our intuition was that a smaller window used to calculate the thresholds, would provide greater accuracy. However, the cross-evaluation system shows that the optimal combination would have a window of $w = 512$, $q = 0.005$, $r = 20$ (Figure 7 green arrow) where the distance D indicates that accuracy will be increased by 15%. Because the cross-evaluation system uses an exhaustive approach, it always returns the optimal combination of input parameters reducing the effort and time to find the optimal combination manually.

5 Using Different Datasets

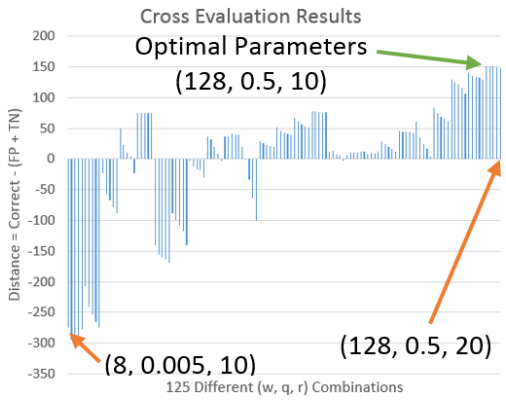

Figure 8: Cross evaluation distance calculation results - Temperature Dataset.

To understand the generality of the work beyond water applications, we applied our cross-evaluation system to datasets from St Bernard Mountain Pass sensor nodes

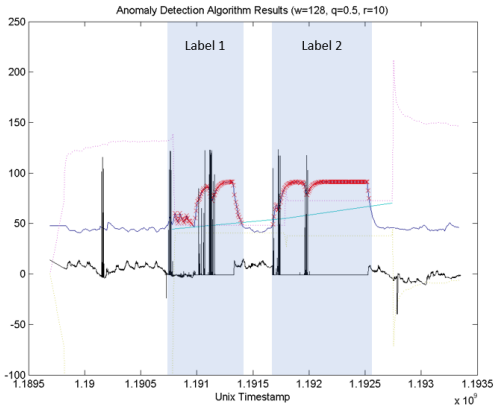

Figure 9: Temperature data anomaly detection results - (w = 128, q = 0.5, r = 10).

in Switzerland [12] reporting temperature, soil moisture and watermark measurements.

Figure 8 and Figure 9 illustrates the results of our cross-evaluation system for the temperature dataset. During the labeling process, we defined two periods as anomalies and we executed our cross evaluation process which was initiated using the following:

```
w = {8, 16, 32, 64, 128}
q = {0.005, 0.01, 0.05, 0.1, 0.5}
r = {10, 13, 15, 17, 20}
```

Figure 8 presents the results of distance D per combination of input parameters, where the optimal combination (w = 128, q = 0.5, r = 10) with 170 correct and 21 error anomaly detection (D = 151). Furthermore, from Figure Figure 9 we can infer that the anomalies are detected precisely and in order to verify our results we manually checked 125 cross evaluation runs. We repeated the same process for soil moisture, and watermark sensor measurements from ten different sensor nodes and we achieved similar results, but due to lack of space we do not include them in this paper.

6 Future Work & Conclusion

This paper presents a scheme that combines lightweight compression and anomaly detection for Cyber-Physical Systems. The work has been developed as part of a Smart Water project and we show that it not only significantly reduces the amount of communications between sensor devices and the cloud, but also that early transient or event (such as water bursts) detection can run on low-resourced sensor nodes meaning that local control functions can occur with minimal latency. The main contribution is the innovative approach to analyzing high sample rate data by using compression rate rather than raw data. The main benefits of our system are: (a) the size of the program at run time, which can be applied in embedded systems, (b) data reduction (and proportional communication and energy costs) by 55%, (c) computation reduction by 98%, (d) the algorithm can be applied independently of the content of the raw data with an appropriate initial tuning, (e) the adaptation of our method to match trend/state changes in the raw data. We extend the system to be able to derive initialization parameters for self-configuration and to adjust said parameters as the nature of the underpinning data changes over time thus showing significant performance improvements over manual tuning by a further 15%.

Future work of our approach is to examine the effect of changes in data precision (e.g. by using float instead of double values), to test our algorithm with other lightweight compression algorithms.

7 Acknowledgments

This work forms part of the Big Data Technology for Smart Water Nets research project funded by NEC Corporation, Japan.

The hydraulic pressure datasets used in this paper have been provided by Dr Ivan Stoianov and Mr Asher Hoskins, Dept Civil Engineering, Imperial College London, UK.

References

[1] Babak Aghaei. Using wireless sensor network in water, electricity and gas industry. In *Electronics Computer Technology (ICECT), 2011 3rd International Conference on*, volume 2, pages 14–17. IEEE, 2011.

[2] Alexandre Santos and Mohamed Younis. A sensor network for non-intrusive and efficient leak detection in long pipelines. In *Wireless Days (WD), 2011 IFIP*, pages 1–6. IEEE, 2011.

[3] WANG Zhu, HAO Xiao-qiang, and WEI De-bao. Remote water quality monitoring system based on wsn and gprs [j]. *Instrument Technique and Sensor*, 1:018, 2010.

[4] Michael Allen, Ami Preis, Mudasser Iqbal, and Andrew J Whittle. Water distribution system monitoring and decision support using a wireless sensor network. In *Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), 2013 14th ACIS International Conference on*, pages 641–646. IEEE, 2013.

[5] Ivan Stoianov, Lama Nachman, Sam Madden, Timur Tokmouline, and M Csail. Pipenet: A wireless sensor network for pipeline monitoring. In *Information Processing in Sensor Networks, 2007. IPSN 2007. 6th International Symposium on*, pages 264–273. IEEE, 2007.

[6] Atmel. Atmel AVR 8-bit and 32-bit Microcontrollers. `http://www.atmel.com/products/microcontrollers/avr/default.aspx`, 2014. [Online; accessed 20-March-2014].

[7] Jan Kraus and Viktor Bubla. Optimal methods for data storage in performance measuring and monitoring devices. In *Proceedings of Electronic Power Engineering Conference*, 2008.

[8] Christopher M Sadler and Margaret Martonosi. Data compression algorithms for energy-constrained devices in delay tolerant networks. In *Proceedings of the 4th international conference on Embedded networked sensor systems*, pages 265–278. ACM, 2006.

[9] Interactive Matter Lab. Filtering Sensor Data with a Kalman Filter. `http://interactive-matter.eu/blog/2009/12/18/filtering-sensor-data-with-a-kalman-filter/`, 2009. [Online; accessed 20-March-2014].

[10] Reza Olfati-Saber. Distributed kalman filter with embedded consensus filters. In *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC'05. 44th IEEE Conference on*, pages 8179–8184. IEEE, 2005.

[11] Burr Settles. Active learning literature survey. *University of Wisconsin, Madison*, 52:55–66, 2010.

[12] Guillermo Barrenetxea, François Ingelrest, Gunnar Schaefer, and Martin Vetterli. Wireless sensor networks for environmental monitoring: the sensorscope experience. In *Communications, 2008 IEEE International Zurich Seminar on*, pages 98–101. IEEE, 2008.