An Adaptive Method for Data Reduction in the Internet of Things

Yasmin Fathy, Payam Barnaghi and Rahim Tafazolli

Institution for Communication Systems (ICS), Electrical and Electronic Engineering Department, University of Surrey
Guildford, Surrey, GU2 7XH, United Kingdom
{*y.fathy, p.barnaghi, r.tafazolli*}@surrey.ac.uk

Abstract—**Enormous amounts of dynamic observation and measurement data are collected from sensors in Wireless Sensor Networks (WSNs) for the Internet of Things (IoT) applications such as environmental monitoring. However, continuous transmission of the sensed data requires high energy consumption. Data transmission between sensor nodes and cluster heads (sink nodes) consumes much higher energy than data sensing in WSNs. One way of reducing such energy consumption is to minimise the number of data transmissions. In this paper, we propose an Adaptive Method for Data Reduction (AM-DR). Our method is based on a convex combination of two decoupled Least-Mean-Square (LMS) windowed filters with differing sizes for estimating the next measured values both at the source and the sink node such that sensor nodes have to transmit only their immediate sensed values that deviate significantly (with a pre-defined threshold) from the predicted values. The conducted experiments on a real-world data show that our approach has been able to achieve up to 95% communication reduction while retaining a high accuracy (i.e. predicted values have a deviation of ± 0.5 from real data values).**

Keywords-**Internet of Things (IoT);Wireless Sensor Networks (WSN);data reduction;Least-Mean-Square (LMS)**

I. INTRODUCTION

Recent advances in sensing and actuator technologies offer possibilities for continuously monitoring real-world phenomena in distributed networks. Sensors are often deployed in distributed and dynamic environments over a large (dense or sparse) geographical area (i.e. sensing field). WSNs enables the development of a broad range of potential IoT applications including environmental monitoring (e.g. temperature, humidity, light, and voltage). In IoT networks, sensor nodes require a continuous reporting of their immediate readings to base stations (i.e. sink nodes). However, data transmission is a dominant factor of communication overhead and energy consumption in WSNs [1].

To address this problem, several data-driven approaches for minimising energy consumption in WSNs have been proposed in the literature such as [2], [1] and [3]. Moreover, data reduction strategies have gained significant attention for effectively lowering the number of transmissions and extending the network lifetime [4]. The main goal of data reduction schemes is that sink nodes are capable of reproducing the complete data stream sensor readings (with a reasonable high accuracy) from a particular set of elements of data streams that are transmitted from sensor nodes. Data reduction includes different approaches such as in-network processing, data compression, and data prediction [4]. On the other hand, in prediction-based approaches, a model is built for describing a particular phenomenon such that user queries can be answered through constructed models instead of accessing the sensors for getting their actual sensed data.

There are several existing work on prediction-based methods and solutions [5, 6]. Some of these approaches such as [7] and [8] assume that sensor observations are highly correlated. For example, spatial and temporal correlations have been proposed in [7] to selectively transmit a subset of the data sensors from which the whole sensor readings can be reproduced based on a predictive model and consequently reduce the number of data transmissions. Although such approaches are able to lower significantly the number of transmissions, they suffer from performance loss in terms of accuracy such that the predictive models have to be updated regularly to include fined-grained changes [1]. To address this problem, dual prediction schemes have been proposed. Dual prediction minimises the number of data transmissions between sensor and sink nodes at the finest level such that sensor nodes transmit only a set of their sensed values without affecting the quality of the actual measurement values [6]. For example, Jain et al. [8] propose a dual prediction scheme using Kalman filters for reducing the number of transmissions. However, Kalman filters require a priori knowledge (e.g. statistical data properties).

In this paper, we propose an Adaptive Method for Data Reduction (AM-DR). Our method is a prediction-based data reduction that exploits LMS adaptive filters. More specifically, our method is based on a convex combination of two decoupled LMS windowed filters with differing sizes for estimating the next measured values both at the source and the sink node such that sensor nodes have to transmit only their immediate sensed values that deviate significantly ($> e_{max}$, a pre-defined threshold) from the predicted values.

The paper is structured as follows. The problem formulation is explained in Section II. Section III provides the required background and the related work. Our proposed algorithm is demonstrated and discussed in Section IV. The performance evaluation, parameter settings, and reproducibility descriptions are included in Section V. Moreover, the proposed algorithm is evaluated and analysed on a real-world

dataset against the state-of-the-art in the same section. We conclude the paper and explain the future directions of our research in Section VI.

II. PROBLEM FORMULATION

Consider a network of N sensor nodes and M sink nodes (e.g. gateways) that are placed in a monitoring region/space. At each time instance $t > 0$, each sensor node s_i $(i = 1, 2, ...N)$ publishes dynamic data streams $x(t)$ for monitoring applications. Data streams are a sequence of numerical data-points in a consecutive order.

We are interested in reducing data transmissions for sensor platforms such that minor perturbations from actual sensor readings are acceptable. In other words, sensor nodes do not require to send their actual readings unless there is a deviation ($>$ a predefined threshold) between the predicted sensor readings and their actual readings. This approach can be used in both of critical and not critical applications by customising the deviation value. For instance, in critical healthcare monitoring applications, accurate sensor readings are required. In such case, a small deviation value is necessary, however, it has an impact on energy consumption. On the other hand, some other environmental monitoring applications (i.e. temperature monitoring) can consume less energy (i.e. reduce the number of transmitted sensor readings) by allowing a reasonable deviation of predicted values compared to actual sensor values. For instance, in fire detection and warning systems, temperature sensors' values are used as an indicator of a fire, in such case, however, the temperature values will have a higher value than a predefined threshold, which, in turn, requires sensor nodes to send their immediate readings to a sink node and consequently, a fire will be detected.

Overall, we are interested in reducing the communication between the sensor and sink nodes such that:

- Sensor nodes need to transmit only their immediate observation and measurement values if the predicted values at sink nodes deviate significantly (with a predefined threshold e_{max}) from real observed values.
- Predicting the future sensor readings at both source and sink nodes with a high accuracy (i.e. predicted values have a small deviation e_{max} value from real data values) and minimal communication between nodes.

Addressing these two concerns allows us to reduce the number of data transmissions between sensor and sink nodes by avoiding unnecessary transmissions without affecting the quality of sensor measurements. Figure 1 gives an illustrative example of our problem formulation for prediction-based data reduction based on dual prediction and LMS adaptive filters. There are three main components: query processing, a set of N sensor nodes and M sink nodes (M is not necessarily equal to N).

We assume that a set of user queries are submitted to query processing component which contain users' requested absolute error $\pm e_{max}$. Each of M sink nodes should have predictive models for its connected sensors S. The model can then predict the next reading value within a certain absolute error $\pm e_{max}$. The prediction is based on using two LMS adaptive filters with differing sizes w_s and w_f such that the filters are able to estimate the next sensor reading based on its previous observation values (i.e. window size). If the estimated values are not a good approximation to actual values, sensor nodes have to explicitly transmit their actual values to sink nodes. Overall, this approach reduces the number of data transmissions and consequently the communication overhead.
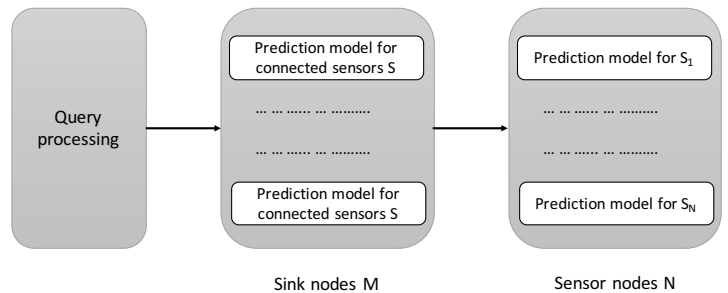

Figure 1. An illustrative example for problem formulation

It is worth mentioning that we refer to accuracy in the paper as a high degree of data quality such that predicted values have a minimal deviation of e_{max} value (e.g. 0.5) from actual data values.

III. BACKGROUND AND RELATED WORK

This section briefly discusses some of the existing work in this area and describes the background information.

A. LMS algorithm and adaptive filters

Least-Mean-Square (LMS) is a de facto adaptive filtering algorithm that has a set of filtering weights that are estimated continuously to minimise the least mean square error (i.e. the difference between the desired and the estimated data streams). LMS has a low computational overhead [1]. It relies on a stochastic gradient descent approach in which coefficients are updated iteratively to minimise the least mean square error $e(t)$ of the filter at the current time t.

$$e(t) = x(t) - y(t) \tag{1}$$

where $x(t)$ is a data stream at time t and $y(t)$ is the output of applying an adaptive filter to the input $x(t)$ such that

$$y(t) = w(t)x(t) \tag{2}$$

$w(t)$ is the filter weight that is adapted to minimise the error $e(t)$ with a step size α (i.e. learning rate) using standard LMS rule.

$$w(t) = w(t-1) + \alpha e(t)x(t) \tag{3}$$

Combination scheme of two filters instead of using one filter has been investigated to improve the steady-state characteristics and performance of LMS [9]. Following the work in [10, 9, 11, 12], a convex combination employs two filters that are decoupled and simultaneously applied to the same input. Their weights are adjusted to minimise the overall errors of the filters. To this end, a convex combination scheme is used to combine the weights of the two filters using a parameter $\lambda(t)$. $\lambda(t)$ is a mixing scalar parameter ($0 \leqslant \lambda(t) \leqslant 1$) to preserve the convexity of this combination [12]. In this case, the overall weight $w(t)$ which is the mixture filter weight is represented as follows:

$$w(t) = \lambda(t)w_1(t) + (1 - \lambda(t))w_2(t) \qquad (4)$$

where $w_1(t)$ and $w_2(t)$ are the weights of the first filter and the second filter at a time instant t, respectively. $\lambda(t)$ is updated as a convex combination parameter with a step size of α using the standard LMS adaptation rule [13, 14] similar to equ. 3 as follows:

$$\lambda(t + 1) = \lambda(t) + \alpha e(t)x(t) \qquad (5)$$

In this paper, we consider the case where LMS algorithm has a constant step size α. It must be noted, however, that there are LMS-type algorithms with a variable step size such that individual step sizes for each of the filter weights are used. Interested readers can refer to the review of Bismor et al. [15] for a detailed discussion about different LMS approaches with a variable step size.

B. Dual prediction

There are several data-driven approaches for minimising energy consumption in WSNs in the literature such as [2], [1] and [3]. Interested readers can refer to the survey of Anastasi et al. [4] for a detailed discussion about other approaches for energy saving in sensor networks. Moreover, dual prediction scheme is a potential candidate to optimise the data transmission between sensor and sink nodes [16]. Jain et al. [8] propose a dual prediction scheme using Kalman filters for predicting sensor readings. However, Kalman filters require feeding with a pre-defined data streams model (e.g. statistical data properties should be known). To address this issue, Santini and Romer [1] propose a dual prediction approach based on LMS filters that requires no prior knowledge. The experiments have been conducted on temperature dataset and achieved up to 92% communication reduction such that the predicted sensor readings have a deviation of ± 0.5 from actual readings.

IV. THE PROPOSED APPROACH

In this paper, we propose an Adaptive Method for Data Reduction (AM-DR) based on dual prediction scheme using a convex combination of two LMS adaptive filters (Algorithm 1). Our dual prediction scheme is similar to [1]. Identical predictive filters are employed at both sink and

sensor nodes such that sensor nodes require transmitting an observed value $x(t)$ at time t if its estimated value $y(t)$ has a significant deviation ($> e_{max}$) from the real observation $x(t)$, where e_{max} is the maximum absolute deviation from actual observation values.

Using two adaptive filters (with a short/long observation memory) explained in the previous section and moving average estimator for LMS filters are the main key differences here compared to the existing approaches including [1]. Our approach does not only require any prior knowledge but also achieves a better communication reduction while retaining a high accuracy (i.e. small e_{max} value) for estimation next observations using LMS windowed filters with differing sizes. As mentioned before, using a combination scheme of two filters instead of using one provides an enhancement of both convergence and steady-state accuracy of the convex weight parameter for a better prediction of next data observation [9].

In equ. 4, we consider one of the adaptive filters is fast while the other is slow. The fast filter w_1 has a short-term observation memory based on using a relatively small fixed window size w_f while the slow filter w_2 has a long-term observation memory based on an increasingly large window size w_s [17]. It is worth noting when λ is near 1, LMS weight is updated based on w_1 and when it is near 0, the weight is updated based on w_2.

Running moving average with a fixed window $\hat{y}_f$ and an increasing window $\hat{y}_s$ for fast and slow adaptive filters, respectively, is to have a good combination for estimating the next observation value based on the previous set of values (i.e. window size). To this end, the outputs of the fast $\hat{y}_f$ and slow $\hat{y}_s$ filters are as follows:

$$\hat{y}_f = \frac{1}{w_f} \sum_{i=t-w_f}^{t} x(t) \qquad (6)$$

$$\hat{y}_s = \frac{1}{w_s} \sum_{i=t-w_s}^{t} x(t) \qquad (7)$$

where w_f is a fixed window size for fast filter and w_s is an increasing window size for slow filter such that $w_f < w_s$. Similar to [10], the overall output for filters $\hat{y}(t)$ is a convex combination of the outputs of both filters mentioned above.

$$\hat{y}(t) = \lambda(t)\ \hat{y}_f + [1 - \lambda(t)]\ \hat{y}_s \qquad (8)$$

Where the mixing parameter λ of their combination is adaptively updated in an on-line manner with the aim of minimising the error of overall filters $e(t)$ between the desired signal $d(t)$ and overall output of both filters $\hat{y}(t)$

$$e(t) = [d(t) - \hat{y}(t)] \qquad (9)$$

The motivation of our proposed approach is to extract the best properties of the independent and decoupled fast $\hat{y}_f$

and slow $\hat{y}_s$ filters by assigning and updating λ that is a combination of both filters at each time t (Please refer to Appendix A for more details) with the aim to minimise the mean squared error.

$$\lambda(t+1) = \lambda(t) + \alpha \ e(t) \ [\hat{y}_f - \hat{y}_s] \qquad (10)$$

where α is the learning rate (i.e. step size) parameter. The learning rate influences the stability and the convergence of the model. It was noted that LMS filters do not converge if $\alpha > 1.0$ [18].

Algorithm 1: Adaptive Method for Data Reduction (AM-DR)

Input : Input data streams x
    Initialisation of $start = 1$, w_s, w_f, α, λ, e_{max}
    Initialisation mode

1 **while** $t < T$ **do**
2   $\hat{y}_f \leftarrow \frac{1}{w_f} \sum_{i=t-w_f+1}^{t-1} x_e(i)$
3   $\hat{y}_s \leftarrow \frac{1}{w_s} \sum_{i=start}^{t-1} x_e(i)$
4   $\hat{y}(t) \leftarrow \lambda(t) \ \hat{y}_f + [1 - \lambda(t)] \ \hat{y}_s$
5   $e(t) \leftarrow d(t) - \hat{y}(t)$
6   $w \leftarrow \hat{y}_f - \hat{y}_s$
7   **if** $|e(t)| < e_{max}$, *(for w_s consecutive steps)* **then**
8   stand-alone mode
9    $x_e(t) = y(t)$
10    $\lambda(t+1) \leftarrow \lambda(t)$
11   **else**
12   normal mode
13    $x_e(t) = x(t)$
14    $\lambda(t+1) \leftarrow \lambda(t) + \alpha \ e(t) \ w$
15    $start \leftarrow t$
16   **end**
17   $t \leftarrow t + 1$
18 **end**

We have summarised our proposed algorithm (AM-DR) in Algorithm 1 and the parameters that are used for the equations in the paper in Table I. Similar to [1], nodes have three main modes:

- initialisation mode: the nodes have this mode only once at the beginning of constructing prediction models at both sensor and sink nodes. During the initialisation mode, sink nodes receive user queries for the first time. Sensor nodes then start transmitting the first w_s observations. During this mode, the learning rate α and e_{max} have to be initialised at both of sensor and sink nodes with same values (Fig. 1).
- normal mode: the node work in a normal mode when the predicted values do not have a good approximation of the actual measurements. In other words, when the predicted model is not accurate enough to predict the

Table I
SUMMARY OF PARAMETERS

Parameter	Definition
N	Total number of sensor Nodes
M	Total number of sink Nodes
i	Sensor index, $n = 1, 2, .., N$
T	Total time duration
t	Time index, $t = 1, 2, ...T$
$x(t)$	data streams that are published by one of N sensors at a time t
w_f	A relatively small fixed window size for a short-term observation memory based filter (i.e. fast filter)
w_s	An increasingly large window size for a long-term observation memory based on filter (i.e. slow filter)
x_e	The available data streams either the predicted values or actual values (i.e. based on normal/stand-alone mode)
α	Learning rate (i.e. step size)
$\lambda(t)$	Mixing weight parameter of the combination of both filters at time t
$start$	The initialise of slow filter window
w	The difference between output filters $\hat{y}_f - \hat{y}_s$
$\hat{y}_s$	Output of slow filter (i.e. moving average with an increasing window size w_s) at time t
$\hat{y}_f$	Output of fast filter (i.e. moving average with a fixed window size w_f) at time t
$e(t)$	Error between the desired signal $d(t)$ and overall output of both filters $\hat{y}(t)$
$\hat{y}(t)$	A convex combination of the outputs of both filters
e_{max}	Maximum absolute deviation from actual observation values

upcoming measurement, sensor nodes have to send their immediate sensor values ($x(t)$). In such a case, the convex combination filter weight (λ) has to be updated. If there is a number of w_s consecutive steps such that the prediction error $e(t) < e_{max}$. The node should switch to stand-alone mode because it has now enough sensor observations to predict the upcoming measurement.

- stand-alone mode: the node works in a stand-alone mode when the prediction model is good enough to predict upcoming measurement with a deviation of $< |e_{max}|$. To this end, the convex combination filter weight (λ) is to be left unchanged.

It is worth noting that since the w_s is a filter with an increasing window, it has to be reinitialised (i.e. to get a fresh start) with the current t value ($start = t$) only during the normal mode. This is because the combination filter weight (λ) has to be updated during the normal mode.

V. EVALUATION

As discussed in the previous section, the proposed solution is composed of dual prediction scheme with a convex combination of two adaptive filters. We have compared our algorithm with the state-of-the-art algorithm in [1]. Our motivation of comparison between our proposed method and the selected state-of-the-art approach (baseline) is that both of them use a dual prediction scheme with LMS adaptive

filters. However, using a combination scheme of two filters instead of using one provides an enhancement of both convergence and steady-state accuracy of the convex weight parameter for a better prediction of next data observation which is the key difference here compared with the state-of-the-art.

A. Dataset

We have conducted our experiments on a real-world dataset that is available at (http://db.lcs.mit.edu/labdata/labdata.html). The dataset is for 54 Mica2Dot sensors [1] with weather boards. Each sensor has the following parameters: temperature, humidity, light, voltage values, data and time at which a sensor reading is obtained and a sensor identifier (i.e. moteid). A clustered view for Mica2Dot sensors with weather boards at Intel Berkeley Research lab is shown in Figure. 2. During our experiments, we have used data reported by the temperature sensors of the same motes (1, 11, 13, 49) between March 6 and 9 to have a fair comparison and consistency with the baseline approach in [1].
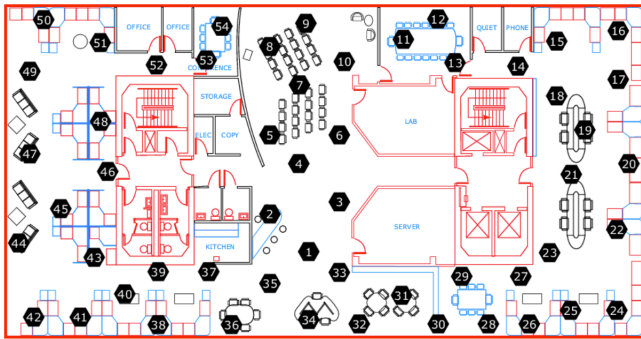
Figure 2. A clustered view for Mica2Dot sensors with weather boards at Intel Berkeley Research lab

B. Parameter settings and reproducibility

We evaluate the performance of our method (AM-DR) against the approach in [1]. The following are the specific default values that we have used for each of the parameters ($e_{max} = 0.5$, $w_f = 5$, $w_s = 10$, $\alpha = 1.0e - 007$). On the other hand, we have used ($N = 5$, $e_{max} = 0.5, \mu = 10^{-5}$) for the baseline approach (as reported in [1]). It is worth noting that we have selected a deviation value of 0.5 to have consistency with the baseline approach [1] such that if the predicted temperature sensor value is, for instance, 20 and the actual value for the same sensor is > 20.5 or less than 19.5, the sensor node has to send the sensed value to the sink node.

During empirical experimentation, we have noticed that w_s should be doubled the value of w_f. Since N parameter

value of the baseline approach in [1] is 5, we have also used $w_f = 5$ and $w_s = 10$. Another observation during the empirical experimentation is that α should be a value of $e - 007$ (verified experimentally).

To ensure the reproducibility of our results, we have made the code and dataset of our implementation and baseline available and have also provided details of a configurable experimental set-up at (http://github.com/YasminFathy/AMDRIoT)

C. Results and discussion

We evaluate the performance of our approach (AM-DR) against the baseline mentioned above. We have implemented the baseline and we have been able to reproduce similar results as in [1]. Figures 3 and 4 demonstrate the results of AM-DR and baseline approaches. The red cross indicates the sensor readings that have to be transmitted to sink nodes. The baseline and AM-DR approaches achieve 92% and 95% communication reduction with an accuracy $e_{max} = \pm 0.5$, respectively. Moreover, Figures 5 and 6 show the error of both approaches of Figures 3 and 4, respectively. The figures indicate that when the prediction error exceeds $|e_{max}|$, the sensor nodes have to transmit immediate readings to the sink, as soon as the prediction error decreases at least w_s times, the node switches to a stand-alone mode.
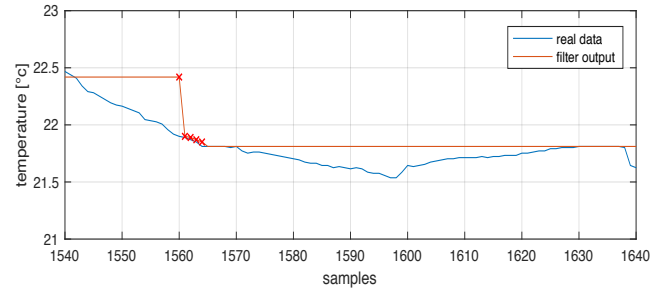

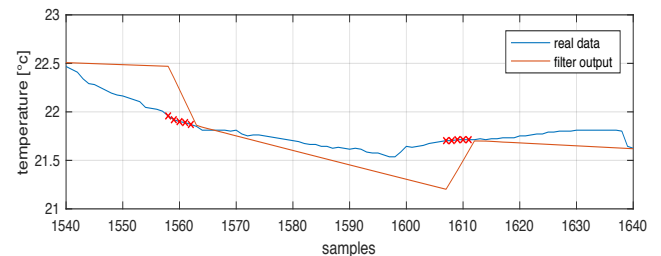Figure 3. AM-DR: real and predicted sensor readings of mote 11

Figure 4. Baseline: real and predicted sensor readings of mote 11

We have conducted another set of experiments and we have reported the simulation results in Figure 7. The figure shows the percentage of transmitted data by mote 11 with
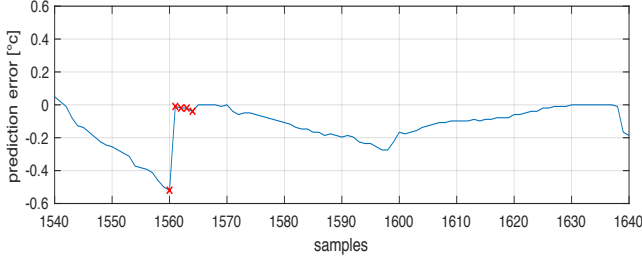
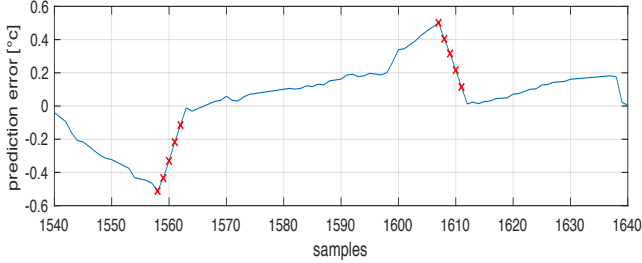Figure 5. AM-DR: prediction error of mote 11

Figure 6. Baseline: prediction error of mote 11

different w_s, w_f and α values (using the same values reported in (Figure 4) in [1]). Figure 7 shows having different window sizes (i.e. filter lengths) for both of adaptive filters can guarantee up to 95% communication reduction (i.e transmitting only about 5% of the data) with an accuracy of 0.5. On the other hand, the baseline approach has been able to transmit about 10% of the collected sensor data. To this end, we have been able to transmit a lower number of data transmissions (i.e. half the number of data transmissions compared with the baseline) while retaining the same accuracy.
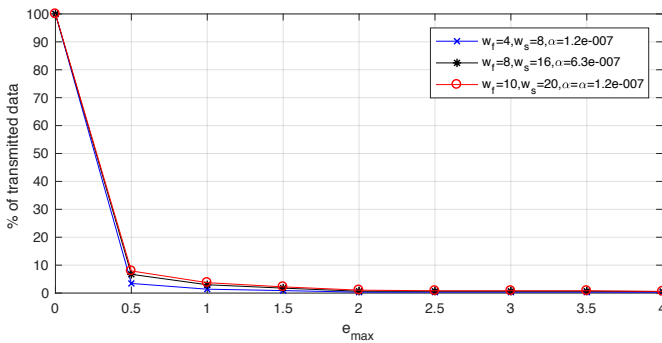

Figure 7. AM-DR: percentage of transmitted data by mote 11

It is worth noting that AM-DR has a higher number of data transmissions than the baseline (i.e. the number of w_s readings is doubled the number of w_f readings such that $w_f = N$ where $N = 5$ readings in the baseline approach during the first set of simulation results). However, AM-DR has been able to achieve a better performance in terms of

communication overhead and accuracy.

The performance of all selected motes for our experiments (1, 11, 13, 49) is shown in Figure. 8. The results show a significant data reduction for all motes. For instance, mote 11 is able to transmit only 7% of the sensor readings while reattaining 0.5 accuracy (i.e. e_max is ± 0.5).

In order to study the comparison between our method (AM-DR) and the baseline in more detail, we have conducted an experiment with additional sensor measurements obtained from mote 30 (mote 30 has more spiky data than other tested motes). The results are reported in Figure. 9. The figure shows how our method is able to perform well with more spiky data than the baseline approach. The baseline requires transmitting 18% of sensor data to obtain 0.5 accuracy, while our method requires only about 13% to retain the same accuracy. To this end, it is clear that our method adapts well to the changes in sensor measurements compared with the baseline method and has a high predictability for upcoming sensor reading.
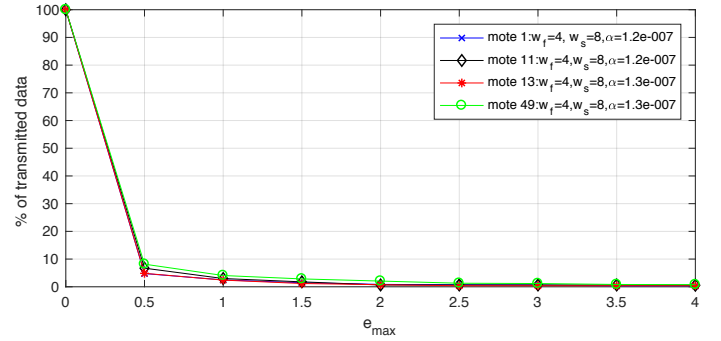

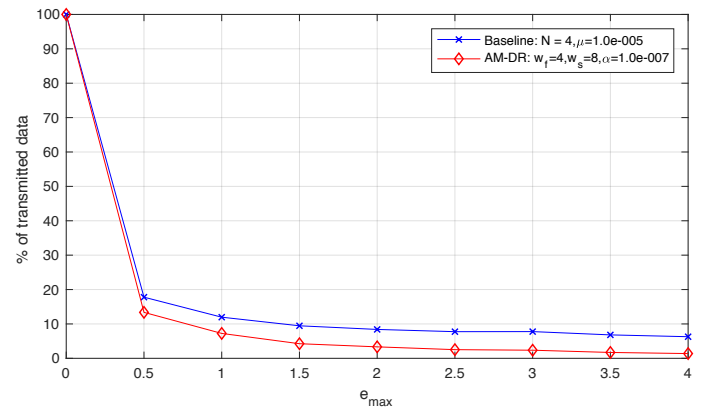Figure 8. AM-DR: percentage of transmitted data by mote 1, 11, 13, 49

Figure 9. Baseline Vs AM-DR :percentage of transmitted data by mote 30

VI. CONCLUSIONS AND FUTURE WORK

We have introduced a novel Adaptive Method for Data Reduction (AM-DR). The proposed approach employs dual

prediction scheme based on a convex combination of two adaptive filters with differing window sizes for predicting upcoming measurements. The main goal is to lower the number of data transmissions between sensor and sink nodes and consequently reducing the communication overhead.

We have then provided a comparison between our proposed algorithm a selected baseline approach on a real-world temperature dataset. Through our experiments, our algorithm has provided a high communication reduction (up to 95%) while retaining a high accuracy (i.e. with a deviation of ±0.5 from actual observation values).

Although our approach has shown a better performance for having a good approximation for estimating the next observation values compared with the baseline, the future work will focus on investigating the sensitivity of the different choices for (AM-DR) parameters. It will include using a modified variable step size of the current algorithm. It will also consider implementing the proposed approach with different network models (e.g. clustered, star, tree). Although we believe that different network models can be easily extended in our approach, message and communication loss have not been considered in the current work.

REFERENCES

[1] S. Santini and K. Romer, "An adaptive strategy for quality-based data reduction in wireless sensor networks," in *Proceedings of the 3rd international conference on networked sensing systems (INSS 2006)*, 2006, pp. 29–36.

[2] A. Deshpande, C. Guestrin, S. R. Madden, J. M. Hellerstein, and W. Hong, "Model-driven data acquisition in sensor networks," in *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*. VLDB Endowment, 2004, pp. 588–599.

[3] B. Stojkoska, D. Solev, and D. Davcev, "Data prediction in WSN using variable step size LMS algorithm," in *Proceedings of the 5th International Conference on Sensor Technologies and Applications*, 2011.

[4] G. Anastasi, M. Conti, M. Di Francesco, and A. Passarella, "Energy conservation in wireless sensor networks: A survey," *Ad hoc networks*, vol. 7, no. 3, pp. 537–568, 2009.

[5] S. Goel and T. Imielinski, "Prediction-based monitoring in sensor networks: taking lessons from MPEG," *ACM SIGCOMM Computer Communication Review*, vol. 31, no. 5, pp. 82–98, 2001.

[6] G. M. Dias, B. Bellalta, and S. Oechsner, "The impact of dual prediction schemes on the reduction of the number of transmissions in sensor networks," *Computer Communications*, vol. 112, pp. 58–72, 2017.

[7] M. C. Vuran, Ö. B. Akan, and I. F. Akyildiz, "Spatio-temporal correlation: theory and applications for wireless sensor networks," *Computer Networks*, vol. 45, no. 3, pp. 245–259, 2004.

[8] A. Jain, E. Y. Chang, and Y.-F. Wang, "Adaptive stream resource management using kalman filters," in *Proceedings of the 2004 ACM SIGMOD international conference on Management of data*. ACM, 2004, pp. 11–22.

[9] J. Arenas-García, A. R. Figueiras-Vidal, and A. H. Sayed, "Steady state performance of convex combinations of adaptive filters," in *Acoustics, Speech, and Signal Processing, 2005. Proceedings.(ICASSP'05). IEEE International Conference on*, vol. 4. IEEE, 2005, pp. iv–33.

[10] M. Martinez-Ramon, J. Arenas-Garcia, A. Navia-Vázquez, and A. R. Figueiras-Vidal, "An adaptive combination of adaptive filters for plant identification," in *Digital Signal Processing, 2002. DSP 2002. 2002 14th International Conference on*, vol. 2. IEEE, 2002, pp. 1195–1198.

[11] Y. Zhang and J. A. Chambers, "Convex combination of adaptive filters for a variable tap-length LMS algorithm," *IEEE Signal Processing Letters*, vol. 13, no. 10, pp. 628–631, 2006.

[12] J. Arenas-Garcia, A. R. Figueiras-Vidal, and A. H. Sayed, "Mean-square performance of a convex combination of two adaptive filters," *IEEE transactions on signal processing*, vol. 54, no. 3, pp. 1078–1090, 2006.

[13] R. K. Martin, W. A. Sethares, R. C. Williamson, and C. R. Johnson, "Exploiting sparsity in adaptive filters," *IEEE Transactions on Signal Processing*, vol. 50, no. 8, pp. 1883–1894, 2002.

[14] D. Mandic, P. Vayanos, C. Boukis, B. Jelfs, S. L. Goh, T. Gautama, and T. Rutkowski, "Collaborative adaptive learning using hybrid filters," in *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, vol. 3. IEEE, 2007, pp. III–921.

[15] D. Bismor, K. Czyz, and Z. Ogonowski, "Review and comparison of variable step-size LMS algorithms," *International Journal of Acoustics and Vibration*, vol. 21, no. 1, pp. 24–39, 2016.

[16] G. M. Dias, B. Bellalta, and S. Oechsner, "A survey about prediction-based data reduction in wireless sensor networks," *ACM Computing Surveys (CSUR)*, vol. 49, no. 3, p. 58, 2016.

[17] D. Kalus, M. Muma, and A. M. Zoubir, "Distributed robust change point detection for autoregressive processes with an application to distributed voice activity detection," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 3906–3910.

[18] Y. Wu, R. M. Rangayyan, Y. Zhou, and S.-C. Ng, "Filtering electrocardiographic signals using an unbiased and normalized adaptive noise reduction system," *Medical Engineering & Physics*, vol. 31, no. 1, pp. 17–26, 2009.

APPENDIX A.
UPDATE EQUATION FOR λ

LMS uses a stochastic gradient descent to find the optimal weights obtained from adaptive filter in order to minimise a cost function. The weights/coefficients are updated iteratively at every time step t with an aim to minimise the signal error (i.e. the difference between the desired and the actual/estimated signal). The formula for gradient descent is as follows:

$$\lambda_{t+1} = \lambda_t - \alpha \frac{\partial f(\lambda_t)}{\partial \lambda_t} \tag{11}$$

Where λ_t is the coefficient (i.e. weight), t is the index of the time step $\{t = 0, 1, 2, 3, ..T\}$, f is the cost function of e_t that needs to be minimised (i.e. error of the model) and α is the learning rate (i.e. step size).

$$e_t = [d_t - y_t] \tag{12}$$

where y_t the output of applying the filter on an input x_t ($y_t = \lambda_t x_t$). By taking the first derivative of the function with respect to the weight λ

$$\frac{\partial e_t}{\partial \lambda_t} = \frac{[d_t - y_t]}{\partial \lambda_t} \tag{13}$$

$$\frac{\partial e_t}{\lambda_t} = [d_t - \lambda_t x_t] \tag{14}$$

$$\frac{\partial e_t}{\partial \lambda_t} = e_t \ x_t \tag{15}$$

We can show the updated coefficient formula of LMS as:

$$\lambda_{t+1} = \lambda_t + \alpha \ e_t \ x_t \tag{16}$$