Cost-Aware Dual Prediction Scheme for Reducing Transmissions at IoT Sensor Nodes

Victor Wattin Håkansson*, Naveen K. D. Venkategowda*, Frank Alexander Kraemer†, Stefan Werner*
*Department of Electronic Systems, †Department of Information Security and Communication Technology
NTNU - Norwegian University of Science and Technology
E-mail: victor.haakansson@ntnu.no, kraemer@ntnu.no, naveen.dv@ntnu.no, stefan.werner@ntnu.no

***Abstract*—This paper develops a method for deciding when to update the prediction model or transmit a set of measurements from the sensor to the fusion centre (FC) to achieve minimal data transmission in a dual prediction scheme (DPS). The proposed method chooses a transmission strategy that results in the lowest expected future transmission cost among a given set of strategies. In a practical IoT setting, statistical information of the measurements might be limited. Hence, without assuming any distribution for the measurements, the proposed method estimates the transmission cost for each strategy through bootstrap data where associated model residuals are resampled using the maximum entropy bootstrap algorithm, which preserves several stochastic properties of the empirical distribution. Numerical results with simulated and real world data shows that the proposed method results in significant reduction in the transmitted data.**

I. Introduction

The Internet-of-things (IoT) holds great potential for infrastructure monitoring, smart cities, and digital society due to the paradigm of connecting myriads of sensors that monitor various physical phenomena and analyzing the vast amount of data acquired. The utility of IoT depends on the ability of the sensors to deliver relevant and accurate measurements to the destination or fusion center (FC) on time. However, data communication consumes significant portion of the energy resources at the sensor nodes. The sensor nodes have limited energy storage capabilities and extensive communication severely reduces their lifetime [1]. Therefore, we need approaches that reduce the number of data transmissions from the sensors without compromising the data accuracy, and thereby enhance utility and prolong lifetime of sensor nodes.

The sensor measurements in IoT tend to be spatio-temporally correlated, which can be exploited to reduce the number of transmitted measurements. This has motivated solutions such as adaptive sensing [2], [3], data compression [4], and data aggregation [5] to reduce the communication from sensor nodes. However, in delay-sensitive and safety-critical applications [6], it is important to fulfill real-time monitoring requirements, where compressing or aggregating data is not an option.

These shortcomings can be overcome by prediction based data-reduction methods [7]–[9], in which the FC replaces missing transmissions with predictions. One such method is the dual prediction scheme (DPS) [9]–[16], which can lead to sparse transmissions in time, low latency, and bounded error at the FC. The DPS is based on the idea of employing an identical prediction model for the sensor measurement process at both the sensor and FC. The model is used to predict the future sensor measurement, and if the prediction error is below a predefined threshold, the sensor node does not transmit the measurement and the FC registers the locally predicted value instead of the true measurement. The number of transmission instances, for a given threshold, depends on the prediction accuracy, which is influenced by the choice of prediction model, model parameter estimates and input data. In literature, the prediction models suggested are in the form of time series models [12], adaptive filters [13], [14], a weighted average [17], or a hybrid of two prediction models [15].

In practice, the sensors observe processes that are often dynamic and non-stationary [18]. Hence, to maintain an acceptable prediction accuracy, the model parameters need to be re-estimated and transmitted to the FC on a regular basis. In addition, if the input variables to the predictor, i.e., predictions and/or true measurements, get outdated, the sensor and FC predictors need to update their inputs with more recent measurements, which incur sensor-FC communication. Although the works in [7], [12]–[16] consider protocols for updating the prediction model to account for changes in the measurement process, they do not evaluate whether such update improves the prediction accuracy. Therefore, energy efficiency for communication can be enhanced by protocols that are not only selective in transmission of measurement data, but also in transmission of model parameter updates.

In this paper, we present a method for DPS that reduces the amount of data transmissions in the sensor-to-FC communication link. Our method decides on a transmission strategy that minimizes the expected future transmission cost. In particular, we generate measurement trajectories by bootstrapping model residuals and estimate the future transmission costs for each strategy. Our approach of estimating the transmission cost is similar to pricing financial derivatives in quantitative finance, where the future pay-out is a path-dependent function of an underlying asset [19], [20].

In large IoT networks sensor measurement distributions tend to be heterogeneous and non-stationary, and it is appropriate to use robust estimation techniques [21] where the performance is less sensitive to inaccurate assumptions of measurement distributions. The bootstrap method is an efficient tool to avoid using estimators derived from assumed distributions,

This work was partly supported by the Research Council of Norway.

by resampling from the empirical distribution [22], [23]. To create the simulated trajectories we use model-based bootstrap and resample the model residuals using the maximum entropy bootstrap algorithm [24], [25], which preserves the mean, variance and time dependence structure of the empirical distribution. We test our approach on simulated as well as real data and observed that the proposed method yields transmission reductions up to 92%.

II. Background and Problem Formulation

A dual prediction scheme (DPS) exploits correlation in consecutive measurements to reduce the amount of communication between sensors and fusion centre (FC). In DPS sensors selectively transmit the measurements to the FC that replaces missing measurements by local predictions. For this purpose, identical predictors are employed at both the sensor and FC. Consequently there is an associated tracking error at the FC, since stored predicted values differ from true measurements. Since the sensor can calculate the prediction error, it transmits information to the FC only when the prediction error exceeds a predefined threshold. This ensures that the data at the FC differs from the true observation at most by the predefined error threshold.

Let $\hat{x}_t$ denote the prediction of measurement x_t at time t and γ represent the accuracy threshold. The measurement registered for traditional DPS at the FC can be expressed as

$$x_t^{FC} = \begin{cases} \hat{x}_t, & \text{if } |x_t - \hat{x}_t| \leq \gamma, \\ x_t, & \text{if } |x_t - \hat{x}_t| > \gamma. \end{cases} \tag{1}$$

The predicted value $\hat{x}_t$ at the FC and sensor can be obtained by using the prediction model given by

$$\hat{x}_t = h(\boldsymbol{\theta}_t, \boldsymbol{x}_{t-1}^{FC}), \tag{2}$$

where the function $h : \mathbb{R}^p \times \mathbb{R}^n \to \mathbb{R}$ defines the model, $\boldsymbol{\theta}_t = [\theta_{t,1}, \theta_{t,2}, \ldots, \theta_{t,p}]^T \in \mathbb{R}^p$ denotes the prediction model parameter of dimension p and $\boldsymbol{x}_t^{FC} = [x_{j_1}^{FC}, x_{j_2}^{FC}, \ldots, x_{j_n}^{FC}]^T \in \mathbb{R}^n$ is the data available at the FC for predictions, where $j_i \in \{1, 2, \ldots, t\}$ and n is number of values. Note that $\boldsymbol{x}_t^{FC}$ can consist of both measurements and predictions according to (1).

From (1) and (2), we see that a measurement x_t is transmitted due to: volatility of the process x_t, outdated model parameter $\boldsymbol{\theta}_t$, or, inaccurate predictor input $\boldsymbol{x}_t^{FC}$. Therefore, the model parameters $\boldsymbol{\theta}_t$ and input variables $\boldsymbol{x}_{t-1}^{FC}$ must occasionally be updated to reduce the number of future transmissions. The model parameters are continuously re-estimated at the sensor node and can be transmitted to the FC to replace the model parameters currently used by predictor. The input variables can be updated by having the sensor transmit a set $\tilde{\boldsymbol{x}}_t$ of k, $k \leq n$, of missing measurements, such that all the input variables for the upcoming prediction are true measurements.

Let $d_t \in \mathcal{S}$ denote the decision by sensor at time t and $\mathcal{S} = \{0, 1, 2, 3, 4\}$ denote the set of strategies available at the sensor. If the prediction error magnitude is below the accuracy tolerance level γ no transmission takes place and $d_t = 0$. Otherwise, the sensor either decides to transmit the measurement x_t, i.e., $d_t = 1$, or to improve the prediction accuracy by transmitting a set of missing measurements $\tilde{\boldsymbol{x}}_t$ i.e. $d_t = 2$, or re-estimated model parameters $\hat{\boldsymbol{\theta}}_t$ to the FC i.e. $d_t = 3$ or transmit both $\tilde{\boldsymbol{x}}_t$ and $\hat{\boldsymbol{\theta}}_t$, which is represented by $d_t = 4$. The transmission strategies available at the sensor is summarized in Table I.

TABLE I
Transmission Strategies and Cost for Current Time.

Strategy s	Transmit	Current Transmission Cost $M_t(s)$
0	-	0
1	x_t	1
2	$\tilde{\boldsymbol{x}}_t$	k
3	$\hat{\boldsymbol{\theta}}_t$	p
4	$\tilde{\boldsymbol{x}}_t$ and $\hat{\boldsymbol{\theta}}_t$	$p + k$

After a sensor transmission is triggered according to (1), the sensor-to-FC communication protocol decide on the strategy that leads to the lowest number of transmissions in future. The transmission cost for the current time $M_t(s)$ that is associated with each strategy is summarized in Table I. In previous literature [7], [12]–[16], the protocols for updating model parameters or inputs are ad-hoc, and do not consider whether such updates reduce future transmission cost. The energy-efficiency for communication can, therefore, be enhanced by a more selective transmission of $(x, \hat{\boldsymbol{\theta}}_t, \tilde{\boldsymbol{x}}_t)$.

In the following section we propose a transmission strategy that projects the least future transmission instances within a given prediction horizon. We estimate the expected number of future transmission instances given each strategy by simulating the number of future transmission instances. For the simulation we use model-based bootstrap, where we resample from the empirical distribution. We choose this approach to avoid relying fully on an assumed parametric distribution.

III. Bootstrap based Cost-Aware Dual Prediction Scheme

This section presents a method for deciding which transmission strategy should be chosen at a transmission instance in DPS. The proposed cost-aware dual prediction scheme (CA-DPS) relies on model-based bootstrap to estimate the future transmission cost associated with each strategy in DPS.

A. Decision criterion for transmission reduction

Assuming that a transmission is triggered at time t according to (1), the expected number of transmission instances in future $N_t(m, s)$ during time period $(t, t+m]$ is given by

$$N_t(m, s) = \sum_{k=t+1}^{t+m} \mathbb{E}_x[\mathbb{1}(|x_k - \hat{x}_k| > \gamma) \mid d_t = s], \tag{3}$$

where $\mathbb{1}(\cdot)$ is an indicator function having value 1 if the condition in the argument is true and 0 otherwise, and the expectation over the distribution of x_t.

The expected total *transmission cost* for time period $[t, t+m]$ in future due to choosing strategy $d_t = s$ at time t is given by

$$C_t(m, s) = M_t(s) + N_t(m, s), \quad (4)$$

where $M_t(s)$ is the current transmission cost at time t as described in Table I. The transmission strategy chosen at triggering time t, $d_t = s^*(m)$, is the one that minimizes the expected total transmission cost

$$s^*(m) = \underset{s \in \mathcal{S}}{\arg\min}\ M_t(s) + N_t(m, s). \quad (5)$$

Since $M_t(s)$ is known apriori, we only need to determine $N_t(m, s)$ from (3).

However, evaluating (3) is intractable and requires knowledge of the distribution of x_t, which is seldom available for real-world measurements in IoT. To overcome this, we find an estimate of $N_t(m, s)$ that is subsequently used to determine the decision criterion as $s^*(m) = \arg\min_{s \in \mathcal{S}} M_t(s) + \hat{N}_t(m, s)$. In the next section, we present the approach for estimating $N_t(m, s)$.

B. Estimating the number of future transmissions

We propose to find an estimate $\hat{N}_t(m, s)$ of the number of future transmissions $N_t(m, s)$ using the bootstrap paradigm [26]. In this approach, we generate L future trajectories of the measurements $[x_{t+1}^{(l)}, x_{t+2}^{(l)}, \ldots, x_{t+m}^{(l)}]^T$, $l = 1, 2, \ldots, L$, in the time interval $[t+1, t+m]$ by drawing resamples from the empirical distribution of measurement process x_t. Thereafter, DPS described in Section II is applied to each trajectory given the model parameters and input variables corresponding to each strategy $s \in \mathcal{S}$ to compute the number of transmissions for the lth resample, which is denoted by $N_t^{(l)}(m, s)$. We then estimate $N_t(m, s)$ by taking the average of $N_t^{(l)}(m, s)$ i.e.,

$$\hat{N}_t(m, s) = \frac{1}{L} \sum_{l=1}^{L} N_t^{(l)}(m, s),\ \forall s \in \mathcal{S}. \quad (6)$$

We use model-based bootstrap [27] to draw the resamples of x_t. The future trajectories of x_t are obtained by sequentially using the estimated prediction model in (2), with re-estimated model parameters $\hat{\boldsymbol{\theta}}_t$, and adding resampled residuals from the empirical distribution of the estimated model residuals.

Define the model residual e_i as

$$e_i = x_i - h(\hat{\boldsymbol{\theta}}_t, \boldsymbol{x}_{i-1}), \quad (7)$$

where $\boldsymbol{x}_i$ is a set $\boldsymbol{x}_i = [x_{j_1}, x_{j_2}, \ldots, x_{j_n}]^T \in \mathbb{R}^n$ and $j_i \in \{1, 2, \ldots, i\}$. Starting from measurement x_t, we generate the next time step $x_{t+1}^{(l)}$ of the lth simulated trajectory as

$$x_{t+1}^{(l)} = h(\hat{\boldsymbol{\theta}}_t, \boldsymbol{x}_t) + e_{t+1}^{(l)}, \quad (8)$$

where $e_{t+1}^{(l)}$ is a resampled from the empirical distribution of model residuals. The model residuals are resampled using the maximum entropy bootstrap algorithm (with scale adjustment) described in [24], [25], which preserves several stochastic properties such as temporal correlation, variance and mean from the empirical distribution. We repeat the same procedure to create a full trajectory $[x_{t+1}^{(l)}, x_{t+2}^{(l)}, \ldots, x_{t+m}^{(l)}]^T$.

To capture changes in the measurement distribution of x_t we estimate $\hat{\boldsymbol{\theta}}_t$ from a sliding time window of w previous measurements and compute the model residuals for the empirical residual distribution.

IV. Evaluation

To evaluate the performance, we test the proposed cost aware dual prediction scheme (CA-DPS) using a linear prediction model on both synthetic and real-world data.

A. Synthetic data − Gaussian random walk with drift

The synthetic data set follows a Gaussian random walk with drift, $x_{t+1} = x_t + \theta_t + z_t$, where $z_t \sim N(0, \sigma^2)$ and $\sigma = 0.1$. The drift parameter θ_t gives the linear trend of the process and was simulated as constant until it made random normal distributed jumps $\Delta\theta_t \sim N(0, 1)$ with a probability of 0.02. The assigned predictor h has the form $\hat{x}_{t+m} = x_t + m\hat{\theta}_t^{FC}$, where $\hat{\theta}_t^{FC}$ is the estimate of θ_t available at FC. The maximum likelihood estimate of θ_t, computed at the sensor whenever a transmission is triggered, is given by [28]

$$\hat{\theta}_t = \frac{x_t - x_{t-w+1}}{w-1}. \quad (9)$$

where w is the window size. Consequently, we have $\hat{\theta}_t^{FC} = \hat{\theta}_t$ for strategies $s = 3$ and $s = 4$ whenever the error bound is violated, and $\hat{\theta}_t^{FC} = \hat{\theta}_{t-1}^{FC}$ otherwise.

We compare the method to an oracle solution (ORACLE), where the sensor knows the distribution of x_t and transmits the model parameters every time the distribution changes. We also compare the proposed method to a constant prediction model (CM), where the prediction is the last transmitted measurement for which the registered measurement at the FC is [29]

$$x_t^{FC} = \begin{cases} x_{t-1}^{FC}, & \text{if } |x_t - x_{t-1}^{FC}| \leq \gamma, \\ x_t, & \text{if } |x_t - x_{t-1}^{FC}| > \gamma. \end{cases} \quad (10)$$

Figure 1 shows the percentage of data transmitted versus accuracy γ for ORACLE, CM, and CA-DPS with window size $w = 10$ to estimate θ_t, $m = 10$ time steps, and $L = 50$ resamples. We see that CM performs poorly since the model does not capture the drift component θ_t in the process x_t. For CA-DPS, the FC model parameter is updated in line with the ORACLE method.

Since the decision criterion is based on the estimate $\hat{N}_t(s, m)$, we compared the distribution of simulated transmission instances to the realizations of $N_t(s, m)$ at each transmission time instant. Table II shows the percentage of data transmitted (DT) and percentage of transmission strategies picked, for which strategies $s = 1$ and $s = 3$ were most frequent. To evaluate the efficacy of the proposed method, in Table II we present the mean absolute difference (MAD) between estimates $\hat{N}_t(s, m)$ and realized transmission instances $N_t(s, m)$, coverage probability (CP), defined as the percentage of realized transmission instances $N_t(s, m)$ included in the 90%-confidence interval of the empirical distribution of the

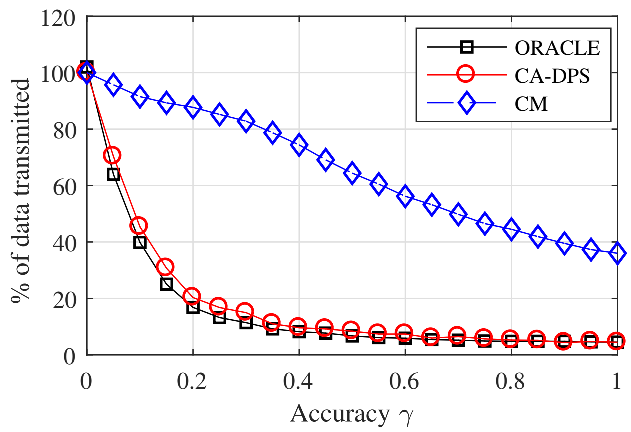

Fig. 1. Percentage of data transmitted versus accuracy γ for CM and CA-DPS with sliding window length $w = 10$, forecast horizon $m = 10$ and $L = 50$ trajectory simulations.

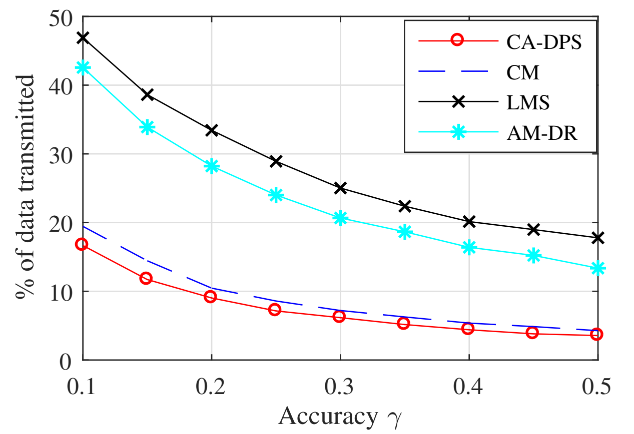

Fig. 2. Percentage of data transmitted versus accuracy γ for Mote ID 30 for CA-DPS with sliding window length $w = 30$, forecast horizon $m = 20$, and $L = 40$ trajectory simulations, LMS with filter length $N = 4$, and step-size $\mu = 10^{-5}$ and AM-DR with fixed window size $w_f = 4$, slow window size $w_s = 8$, and learning rate $\alpha = 10^{-7}$.

TABLE II
PERFORMANCE OF FUTURE TRANSMISSION COST ESTIMATION ($\gamma = 0.5, L = 40$).

m	w	DT	$s = 1$	$s = 3$	$s = 4$	MAD	CP	BOS
10	10	8%	45%	51%	4%	0.7	78%	82%
10	20	9%	51%	49%	0%	0.9	72%	93%
20	30	11%	51%	47%	2%	2.4	47%	83%
30	40	13%	55%	39%	5%	4.0	42%	74%
40	80	23%	61%	24%	16%	6.7	24%	85%
50	80	23%	62%	22%	16%	9.4	17%	72%
80	100	27%	70%	19%	12%	18.8	9%	59%

simulations of $N_t(s, m)$, and the best-strategy-selected (BOS), denoting the percentage of times the picked strategy resulted in the minimum realized transmission cost $C_t(s, m)$. We see that the estimates of $\hat{N}_t(s, m)$ and the empirical distribution of simulated transmission instances are in line with the realized transmission instances as the process evolves over time. As the length of the sliding window w and the forecast horizon m increases so does the probability of a shift in the measurement distribution, i.e., a jump in the parameter θ_t during the time span $[t - w, t + m]$. Therefore, the simulated trajectories and estimates $\hat{N}_t(s, m)$ diverge more from the realizations for larger values of w and m.

B. Real world data and benchmarks

CA-DPS was also tested on a real-world data set, containing the readings from 54 Mica2Dot sensors in the Intel Berkeley Research lab between March 6 and 9, 2004[1]. Each sensor is identified by a Mote ID number and has readings of humidity, temperature, light and voltage, taken every 31 seconds. This data set has repeatedly been used as test data in the literature of prediction based data reduction [13], [14], [16], [30].

[1]Available at http://db.lcs.mit.edu/labdata/labdata.html

To compare the performance, we used three different DPS algorithms as benchmarks. The first benchmark was again the CM model. The second benchmark was the least-mean-square (LMS) algorithm presented in [13], which achieved a transmission rate of 8% at an acceptance tolerance level of $\gamma = 0.5$. The third benchmark is the proposed method in [14], referred to as adaptive method for data reduction (AM-DR). All benchmark methods use linear models.

Figure 2 shows the percentage of data transmitted for CA-DPS and benchmark methods when applied to the temperature data of Mote ID 30, as was also done in [13], [14]. We see that CA-DPS achieves the lowest transmission rate for γ between 0.1 and 0.5. At $\gamma = 0.1$ it achieves a transmission rate of 17% with a mean absolute error (MAE) of 0.05 between the true and predicted readings $|x_t^{FC} - x_t|$.

CA-DPS achieves a lower transmission rate for all 54 mote IDs, at $\gamma = 0.1$, as all sensors are below the $45°$ line in Fig. 3. It was able to achieve an average transmission rate of 8%, at $\gamma = 0.1$, for all sensors using the same settings $(w = 30, m = 20, L = 40)$.

V. CONCLUSION

This paper presented a cost-aware dual prediction scheme for reducing the amount of data transmitted between IoT sensor nodes and fusion center. In particular, we proposed the CA-DPS method that chooses among a set of strategies the one that achieves minimum expected transmission cost within a given forecast horizon. The transmission cost incurred in future is estimated using bootstrapping model residuals associated with each strategy. Simulations results with both synthetic and real measurement data confirmed that the proposed method achieves a significant reduction in the communication requirement for a given error constraint.

REFERENCES

[1] M. A. Alsheikh, S. Lin, D. Niyato, and H. P. Tan, "Machine learning in wireless sensor networks: algorithms, strategies, and applications," *IEEE*

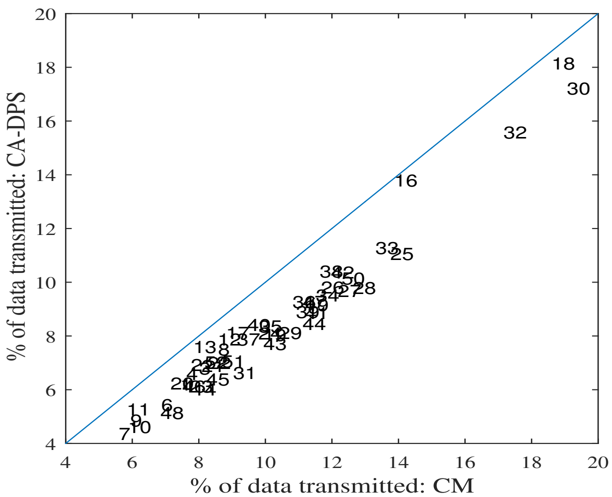

Fig. 3. Percentage of data transmitted with accuracy set to $\gamma = 0.1$ for all temperature sensors in the Intel office data set, between March 6 and 9, for CM and CA-DPS with sliding window length $w = 30$, forecast horizon $m = 20$ and $L = 40$ trajectory simulations. Numbers represent Mote IDs.

Communications Surveys and Tutorials, vol. 16, no. 4, pp. 1996–2018, 2014.

[2] Z. Chen, J. Ranieri, R. Zhang, and M. Vetterli, "DASS: Distributed adaptive sparse sensing," *IEEE Transactions on Wireless Communications*, vol. 14, no. 5, pp. 2571–2583, 2015.

[3] M. Mathew and N. Weng, "Quality of information and energy efficiency optimization for sensor networks via adaptive sensing and transmitting," *IEEE Sensors Journal*, vol. 14, no. 2, pp. 341–348, 2014.

[4] M. Abu Alsheikh, S. Lin, D. Niyato, and H. P. Tan, "Rate-distortion balanced data compression for wireless sensor networks," *IEEE Sensors Journal*, vol. 16, no. 12, pp. 5072–5083, 2016.

[5] H. Harb, A. Makhoul, S. Tawbi, and R. Couturier, "Comparison of Different Data Aggregation Techniques in Distributed Sensor Networks," *IEEE Access*, vol. 5, pp. 4250–4263, 2017.

[6] D. J. McCorrie, E. Gaura, K. Burnham, N. Poole, and R. Hazelden, "Predictive data reduction in wireless sensor networks using selective filtering for engine monitoring," in *Wireless Sensor and Mobile Ad-Hoc Networks: Vehicular and Space Applications*. Springer New York, 2015, pp. 129–148.

[7] U. Raza, A. Camerra, A. L. Murphy, T. Palpanas, and G. P. Picco, "What does model-driven data acquisition really achieve in wireless sensor networks?" *2012 IEEE International Conference on Pervasive Computing and Communications*, pp. 85–94, 2012.

[8] M. Wu, L. Tan, and N. Xiong, "Data prediction, compression, and recovery in clustered wireless sensor networks for environmental monitoring applications," *Information Sciences*, vol. 329, pp. 800–818, Feb. 2016.

[9] G. M. Dias, B. Bellalta, and S. Oechsner, "A survey about prediction-based data reduction in wireless sensor networks," *ACM Computing Surveys*, vol. 49, no. 3, pp. 1–35, Nov. 2016.

[10] S. Werner and J. Lundén, "Smart load tracking and reporting for real-time metering in electric power grids," *IEEE Transactions on Smart Grid*, vol. 7, no. 3, pp. 1723–1731, 2016.

[11] H. Liazid, M. Lehsaini, and A. Liazid, "An improved adaptive dual prediction scheme for reducing data transmission in wireless sensor networks," *Wireless Networks*, Feb 2019.

[12] G. Li and Y. Wang, "Automatic ARIMA modeling-based data aggregation scheme in wireless sensor networks," *Eurasip Journal on Wireless Communications and Networking*, vol. 2013, no. 1, pp. 1–13, Dec. 2013.

[13] S. Santini and K. Römer, "An adaptive strategy for quality-based data reduction in wireless sensor networks," in *Proc. International Conference on Networked Sensing Systems (INSS)*, Jun. 2006, pp. 29–36.

[14] Y. Fathy, P. Barnaghi, and R. Tafazolli, "An adaptive method for data reduction in the internet of things," in *Proc. IEEE 4th World Forum on Internet of Things*, May. 2018.

[15] R. Askari Moghadam and M. Keshmirpour, "Hybrid ARIMA and neural network model for measurement estimation in energy-efficient wireless sensor networks," in *Informatics Engineering and Information Science*. Springer Berlin Heidelberg, 2011, pp. 35–48.

[16] U. Raza, A. Camerra, A. L. Murphy, T. Palpanas, and G. P. Picco, "Practical data prediction for real-world wireless sensor networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 8, pp. 2231–2244, Aug. 2015.

[17] F. A. Aderohunmu, G. Paci, D. Brunelli, J. D. Deng, L. Benini, and M. Purvis, "An application-specific forecasting algorithm for extending WSN lifetime," in *Proc. IEEE International Conference on Distributed Computing in Sensor Systems*, May 2013, pp. 374–381.

[18] G. Ditzler, M. Roveri, C. Alippi, and R. Polikar, "Learning in nonstationary environments: a survey," *IEEE Computational Intelligence Magazine*, vol. 10, no. 4, pp. 12–25, Nov. 2015.

[19] P. Rebentrost, B. Gupt, and T. R. Bromley, "Quantum computational finance: Monte Carlo pricing of financial derivatives," *Physical Review A*, vol. 98, no. 2, pp. 1–15, Aug. 2018.

[20] R. W. Shonkwiler, "Pricing exotic options," in *Finance with Monte Carlo*. Springer New York, 2013, pp. 117–134.

[21] A. M. Zoubir, V. Koivunen, Y. Chakhchoukh, and M. Muma, "Robust estimation in signal processing: A tutorial-style treatment," *IEEE Signal Processing Magazine*, vol. 29, no. 4, pp. 61–80, 2012.

[22] A. M. Zoubir and B. Boashash, "The bootstrap and its application in signal processing," *IEEE Signal Processing Magazine*, vol. 15, no. 1, pp. 56–76, Jan. 1998.

[23] M. Golz, V. Koivunen, and A. Zoubir, "Nonparametric detection using empirical distributions and bootstrapping," *25th European Signal Processing Conference, EUSIPCO 2017*, pp. 1450–1454, Jan. 2017.

[24] J. Lopez-de Lacalle and H. Vinod, "Maximum entropy bootstrap for time series: The meboot R package," *Journal of Statistical Software*, vol. 29, no. 05, 2009.

[25] H. D. Vinod, "Maximum Entropy Bootstrap Algorithm Enhancements," *Ssrn*, pp. 1–15, 2013.

[26] B. Efron, "Bootstrap methods: Another look at the jackknife," *The Annals of Statistics*, vol. 7, no. 1, pp. 1–26, 1979.

[27] S. N. Lahiri, *Resampling methods for dependent data*. Springer-Verlag New York, 2003.

[28] K. Bertin, S. Torres, and C. A. Tudor, "Maximum-likelihood estimators and random walks in long memory models," *Statistics*, vol. 45, no. 4, pp. 361–374, Aug. 2011.

[29] J. Lundén and S. Werner, "Real-time smart metering with reduced communication and bounded error," in *IEEE International Conference on Smart Grid Communications*, 2014, pp. 326–331.

[30] L. Tan and M. Wu, "Data reduction in wireless sensor networks: a hierarchical LMS prediction approach," *IEEE Sensors Journal*, vol. 16, no. 6, pp. 1708–1715, Mar. 2016.