The 10th Conference for Informatics and Information Technology (CIIT 2013)

COMPARISON OF DIFFERENT DATA PREDICTION METHODS FOR WIRELESS SENSOR NETWORKS

Biljana Risteska Stojkoska
Faculty of Computer Science and Engineering
Skopje, Macedonia

Kliment Mahoski
Faculty of Computer Science and Engineering
Skopje, Macedonia

ABSTRACT

Different data reduction strategies have been developed in order to reduce the energy consumption in wireless sensor networks (WSN). Most of them reduce the amount of sent data by predicting the measured values both at the source and the sink, requiring transmission only if a certain reading differs by a given margin from the predicted values. The subject of this paper is comparison of a few different techniques for prediction of time series data in WSN. While these strategies often provide great reduction in power consumption, they don't need a priori knowledge of the explored domain in order to correctly model the expected values.

I. INTRODUCTION

Distributed WSN provide the ability to make temporal and spatial progression of the quantitis they measure. If the nodes report sensed data at each interval, it will vastly reduce the network lifetime and will create sufficient communication overhead. There are several techniques that have been developed to overcome these problems, i.e. to lower the communication overhead and to increase the energy savings.

Data-reduction techniques can be basically divided into three main groups: data compression, data prediction and in-network processing [1]. Data compression is applied to reduce the amount of information sent by source nodes. This scheme involves coding strategy used to represent data regardless of their semantics and is very suitable if the WSN application doesn't require the most recent measurements. In-network processing performs data aggregation while data is routed towards the sink node. This paradigm aims to transform the raw data into less voluminous refined data using summarization functions (minimum, maximum and average). For applications that require original and accurate measurements, such a summarization may be inappropriate since it brings loss of the accuracy [2].

Data prediction techniques usually maintain two instances of a prediction model, one residing at the sink and the other at the sensor. To avoid a rapid deterioration in the predicted values, such approaches need to periodically validate and update their models. Data prediction techniques can be divided into three subclasses: stochastic approaches, time series forecasting and algorithmic approaches. The last are application-specific and usually apply some heuristics about the domain they explore. Stochastic approaches are used when sensed phenomena can be modeled with probability density function. These algorithms provide acceptable predictions but usually are inappropriate due to its computational overhead. Data prediction models for WSN are those based on time series forecasting. Moving Average (MA), Autoregressive (AR) or Autoregressive Moving Average (ARMA) models are simple, easy for implementation and provide acceptable accuracy [3][4]. In this paper, we investigate and compare time-series forecasting techniques for WSN based on these three algorithms.

The rest of the paper is organized as follows: the next section presents a brief overview of related work. The third section of this paper describes the process models used for data prediction - MA, AR and ARMA. The fourth chapter covers the simulation results. Finally, we conclude this paper in section five.

II. RELATED WORK

Time series forecasting in WSN is still not enough explored, beside the attractiveness of WSN in the last decade. Only a few well known techniques from time series analyses have been implemented and appropriately evaluated on different WSN datasets.

The most popular paradigm is Dual Prediction Scheme (DPS) [3][5][6][7] (formerly known as Dual Kalman Filter). Here, each node runs a filter (or a model) that estimates the next measurement. The sink (or the base station) runs exactly the same models for each sensor in the network and makes the same predictions. Since the sensor makes measurements of the sensed quantity, it can check whether the predicted value differs from the sensed value above the predefined threshold E_{max}. If the difference is below the threshold, both the sensor and the sink accept the predicted value and store it in the memory instead of measured value. Otherwise, the sensor sends the actual value to the sink node. Both the sensor and sink use this value and simultaneously estimate the prediction model and update the filter weights.

Romer and Santini in [5] choose Least Mean Square (LMS) over Kalman Filter since it doesn't require a priori knowledge of the desired measurements, which implies that the sink and the sensors don't need to agree on a predefined model. In [6][7], the authors propose a modification of LMS that uses variable step size parameter for fine tuning the filter weights. Le Borgne and Santini in [3] present a general framework for DPS in which sensor nodes using racing mechanism [8] autonomously select prediction model among K candidate prediction models: constant prediction model (CM) and AR models of orders 1–5. The results obtained from 14 different datasets show that CM outperforms AR models for time series with sudden and sharp changes, while in all other cases AR performs significantly better than CM.

In [4] a more complex prediction model is used based on ARIMA technique. In the first phase, sink receives n sensor readings from each sensor, builds an appropriate ARIMA model for each sensor and sends the ARIMA parameters back to the sensors. In the second phase, the prediction is performed on both sides. If the readings significantly differ from the predicted values, the sensor should send the latest p readings to the sink so it can update the dataset. An adaptive ARMA (A-ARMA) technique with moving window in [9] employs low-order AR term and MA term. Each node locally computes the parameters of A-ARMA model using the last W sensor readings and propagates the model parameters to the sink node. Choosing the right order for AR and MA component in [4][9] model is a tradeoff between forecasting accuracy and energy efficiency.

In [10] the authors introduce a hybrid model based on Grey-Model-based approach [11] and Kalman Filter. Evaluation done on real datasets shows that the proposed model outperforms others in terms of energy consumption. In [12], DPS for Wireless Body Sensor Network is presented using proportional–integral–derivative (PID) based algorithm for data prediction. Other approaches include “send on delta” technique [13], which calculates the difference between the current value and the predicted value.

Different metrics have been purposed for measuring algorithms performance. One metric is the reduction of transmissions in percentage [5]. Other metric used for evaluation is by measuring the difference between the predicted and the true value, i.e. mean square error (MSE) [6][7] or root mean square error (RMSE) [4][9]. These two metrics can be integrated into one, which is the ratio reduction/RMSE.

In our work, we implement Moving Average and Autoregressive Moving Average models as a predictive filter. The evaluation was done with MSE and number of transmissions using real datasets from Intel Berkeley Research Laboratory [14] and NDBC dataset [15].

III. Algorithms for Data Prediction

In this section we are going to give a brief explanation of each of the prediction filters.

A. Moving Average process model

A moving average process of order p is denoted by MA(p) and is defined as

$$X_t = \sum_{r=0}^{p} \theta_r \, \epsilon_{t-r} \tag{1}$$

where $\theta_1, \dots, \theta_p$ are fixed constants, $\theta_0 = 1$ holds, and $\{\epsilon_t\}$ is the white noise (an array of independent random variables) with mean value of 0 and variance σ^2.

By definition, processes of this class are second-order stationary and it holds that the autocovariance function has value of

$$\gamma_k = \begin{cases} 0, & |k| > p \\ \sigma^2 \sum_{r=0}^{p-|k|} \theta_r \theta_{r+k}, & |k| \le p \end{cases} \tag{2}$$

It is possible for two MA processes to have the same autocorrelation function (defined as $\rho_k = \gamma_k/\gamma_0$), for example

$$X_t = \epsilon_t + \theta\epsilon_{t-1}, \text{ and} \tag{3}$$

$$X_t = \epsilon_t + \frac{\epsilon_{t-1}}{\theta} \tag{4}$$

both have $\rho_1 = \theta/(1+\theta^2)$ and $\rho_k = 0$ (for $|k| > 1$). However, equation (3) yields

$$\begin{aligned} \epsilon_t &= X_t - \theta\epsilon_{t-1} = X_t - \theta(X_{t-1} - \theta\epsilon_{t-2}) \\ &= X_t - \theta X_{t-1} + \theta^2 X_{t-2} - \cdots \end{aligned} \tag{5}$$

for $|\theta| < 1$, which is a characteristic of an invertible process. Two different invertible processes can never have the same autocorrelation function [16-20].

B. Autoregressive process model

An autoregressive process of order p, similarly, is denoted by AR(p), and is defined as

$$X_t = \sum_{r=1}^{p} \phi_r X_{t-r} + \epsilon_t \tag{6}$$

where $\phi_1, \dots, \phi_r$ are fixed constants and $\{\epsilon_t\}$ again is a sequence of independent random variables with mean value of 0 and variance σ^2.

Following from (6), the AR(1) process is defined as

$$X_t = \phi_1 X_{t-1} + \epsilon_t \tag{7}$$

By making successive substitutions, we find that

$$\begin{aligned} X_t &= \epsilon_t + \phi_1\big(\epsilon_{t-1} + \phi_1(\epsilon_{t-2} + \cdots)\big) \\ &= \epsilon_t + \phi_1\epsilon_{t-1} + {\phi_1}^2\epsilon_{t-2} + \cdots \end{aligned} \tag{8}$$

It can be observed that $E(X_t)$ is 0 and the autocovariance function can be calculated as follows:

$$\gamma_k = \frac{\sigma^2 {\phi_1}^k}{1-{\phi_1}^2} \tag{9}$$

which leads to the conclusion that $\{X_t\}$ is second order stationary. The autocorrelation function is

$$\rho_k = \phi_1\rho_{k-1} + \phi_2\rho_{k-2} + \cdots + \phi_p\rho_{k-p}, k = 1,2, \dots \tag{10}$$

also known as the Yule-Walker equations [16-20].

C. Autoregressive Moving Average process model

An autoregressive moving average process of order (p, q), meaning it has p autoregressive and q moving average terms, is noted as ARMA(p, q) and is defined as:

$$X_t = \sum_{r=1}^{p} \phi_r X_{t-r} + \sum_{s=0}^{q} \theta_s \epsilon_{t-s} \qquad (11)$$

where $\{\epsilon_t\}$ is white noise. This process is stationary for the appropriate values for ϕ and θ.

Let us consider the state space model given by:

$$\begin{aligned} X_t &= \phi X_{t-1} + \epsilon_t, \\ Y_t &= X_t + \eta_t \end{aligned} \qquad (12)$$

If we suppose that $\{X_t\}$ is unobserved, $\{Y_t\}$ is observed and $\{\epsilon_t\}$ and $\{\eta_t\}$ are independent white noise sequences, while $\{X_t\}$ is AR(1), we can write that

$$\xi_t = Y_t - \phi Y_{t-1} = \epsilon_t + \eta_t - \phi \eta_{t-1} \qquad (13)$$

where ξ_t is stationary and $cov(\xi_t, \xi_t + k)$ is 0 for any $k \geq 2$. That way, ξ_t can be modeled as a MA(1) process and $\{Y_t\}$ as ARMA$(1, 1)$ process [16-20].

IV. Simulation Results

In this section the simulation results from the MA(2), MA(4) and ARMA algorithms are going to be presented. The efficiency of the algorithms evaluated with respect to two metrics: **percentage of insufficiently correct predictions** (PICP) and **mean square error** (MSE). Lower values for PICP and MSE are desirable (mean better performance). The algorithms were simulated in MATLAB [21].

On the following graphs, the results of the algorithms are represented by different colors: MA(2) – blue, MA(4) – yellow, MA(10) – red and ARMA – green. The upper graphs of each picture represent the PICP values and the lower graphs represent the MSE. The horizontal axis represents the value of the threshold E_{max}.

A. Evaluation of the Intel experimental dataset

For the evaluation of the algorithms, the first set that was used is the experimental dataset from Intel Lab [14]. The 54 Mica2Dot sensors deployed in the laboratory were equipped with weather boards and measured temperature once every 31 seconds. The measurements were collected between February 28th and April 5th, 2004. The simulations were run for 50 different error margins E_{max} (ranging from 0.1°C to 5°C).

The results (Figure 1 and Figure 2) show that, concerning the PICP, the ARMA algorithm is constantly better than the other two. MA(4) is second in this regard, and MA(2) is slightly behind. In most of the results from the simulations we have done, the difference is greater for threshold values in the [0, ~1.5] range. However, there is very little difference for values greater than ~1.5.

Concerning the MSE, the simulation results (Figure 1 and Figure 2) are a bit less consistent, but still certain patterns emerge. Firstly, one can observe that there are two main intervals of threshold values, range [0, ~3.5] and range [~3.5, 5]. In the first interval the results are relatively consistent – ARMA has the best MSE, MA(4) is second and MA(2) third. However, in the second interval, the results for the MSE from simulations done on some datasets are still relatively consistent (Figure 1), while other simulations give inconsistent, seemingly chaotic results (Figure 2).

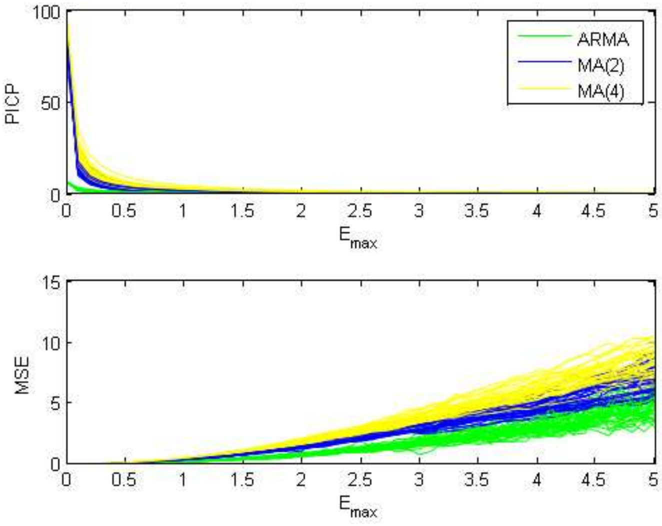

Figure 1: Results from simulations done on 35 different nodes that yield relatively consistent performance.

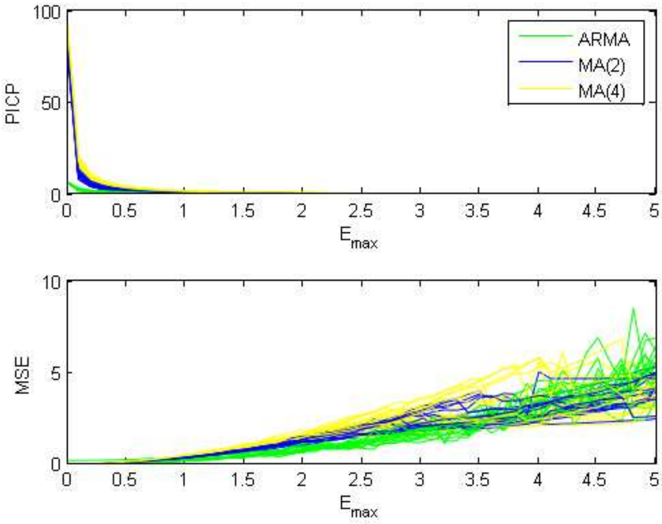

Figure 2: Results from simulations done on 17 different nodes that yield inconsistent performance.

For some datasets, the order of the algorithms (sorted by their MSE) changes in the second interval. This is caused by the difference in the values of the data elements in the sets, i.e. the entropy of the datasets. We observed that the nodes which have relatively consistent MSE in the second interval as well, have small differences between the values of two neighboring data elements, while those datasets for which the order changes have larger differences among their data elements. Although the results for such datasets for the most part in the [~3.5, 5] range are intertwined, there is a tendency for the algorithms to swap their places in regard of their MSE – it can be observed that, in Figure 2, MA(4) gradually becomes the best of the three, while ARMA becomes worst. From that we can conclude that:

- When the E_{max} threshold value is in the range of [0, ~3.5] the ARMA algorithm performs best in both PICP and MSE.
- When the E_{max} threshold value is above ~3.5, ARMA is still the first choice for datasets with smaller entropy, but for datasets with higher entropy, MA(4) is the first choice.

Another conclusion, concerning the performance of the MA algorithm that can be drawn from the results (Figure 1, 2) is that both the PICP and the MSE are positively correlated with the order of the MA algorithm in question. This is more clearly visible when a third MA algorithm – MA(10) is introduced (Figure 3).

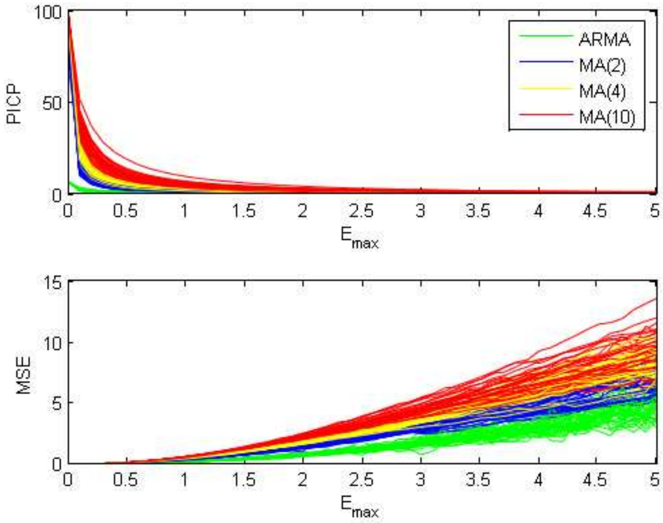

Figure 3: Results from simulations done on 30 different nodes demonstrating the correlation between the performance and the order of the MA algorithm.

B. Evaluation of the NDBC dataset

The second set of measurement data that was used is the dataset of the National Data Buoy Center (NDBC) of the National Oceanic and Atmospheric Administration of the USA [15]. The results of a simulation run on a dataset of wind direction measurements is shown on Figure 4. Wind directions are taken as an example of a dataset with suddenly changing element values. The ARMA algorithm is not appropriate for making predictions of such data and is excluded from this comparison because it consistently has a high value for its MSE.

In regard of the PICP, the results show that it is inversely correlated with the order of the MA algorithm.

Although the results for each of the algorithms are close to one another in regard of their MSE metric, MA(2) has the best performance again, but for higher values for E_{max}, the values for their MSE are sometimes intertwined.

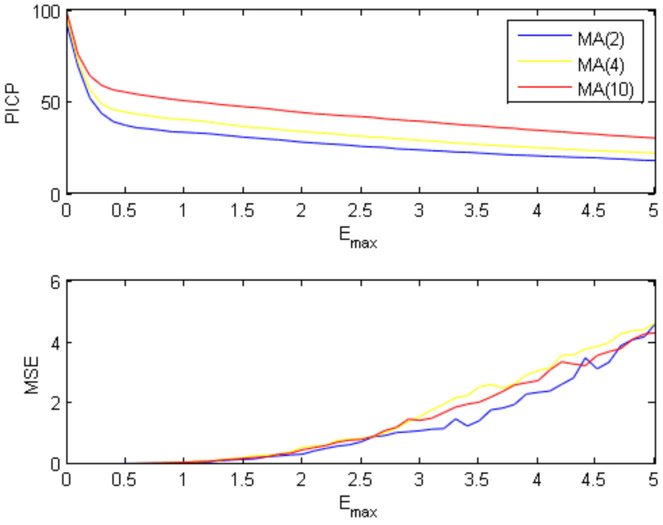

Figure 4: Results for the MA algorithms of different order of a simulation run on a wind direction dataset.

V. Conclusion

In this paper we compare different algorithms for making predictions in time series data acquired from WSN. The results of the simulations we have done show that the nature of data (mainly their entropy) influences the performance of a certain algorithm. We have come to the conclusion that for gradually changing data – water temperature, water level etc. (data with lower entropy), ARMA performs best; for data with sudden or sharp changes in the values (higher entropy), MA(4) has the best ratio between the performance and complexity.

REFERENCES

[1] G. Anastasi, M. Conti, M. Di Francesco, A. Passarella, Energy Conservation in Wireless Sensor Networks: A survey, Ad Hoc Networks, vol.7 n.3, pp. 537-568, May, 2009.

[2] E. F. Nakamura, A. A. F. Loureiro, A. C. Frery, Information Fusion for Wireless Sensor Networks: Methods, models, and classifications, ACM Computing Surveys (CSUR), vol.39 n.3, pp. 9-55, 2007.

[3] Y. L. Borgne, S. Santini, G. Bontempi, Adaptive Model Selection for Time Series Prediction in Wireless Sensor Networks, Signal Processing, vol.87 n.12, pp. 3010-3020, December, 2007.

[4] C. Liu, K. Wu, M. Tsao, Energy Efficient Information Collection with the ARIMA Model in Wireless Sensor Networks, Proceedings from IEEE Globecom, vol. 5, pp. 2470–2474, 2005.

[5] S. Santini and K. Römer, An Adaptive Strategy for Quality-Based Data Reduction in Wireless Sensor Networks, Proceedings of the 3rd International Conference on Networked Sensing Systems (INSS 2006), pp. 29-36, Chicago, IL, USA. June 2006.

[6] B. Risteska Stojkoska, D. Solev, and D. Davcev, Variable Step Size LMS Algorithm for Data Prediction in Wireless Sensor Networks, Sensors & Transducers 14 (2012).

[7] B. Stojkoska, D. Solev, and D. Davcev, Data Prediction in WSN using Variable Step Size LMS Algorithm, In SENSORCOMM 2011, The Fifth International Conference on Sensor Technologies and Applications, pp. 191-196. 2011.

[8] O. Maron, A.W. Moore, The racing algorithm: model selection for lazy learners, Artificial Intell. Rev. 11 (1–5) (1997) 193–225.

[9] J. Lu, F. Valois, M. Dohler, M.-Y. Wu, Optimized Data Aggregation in WSNs using Adaptive ARMA, Proceeedings Sensorcomm 2010, 2010, pp. 115–120.

[10] G. Wei, Y. Ling, B. Guo, B. Xiao, A. V. Vasilakos, Prediction-based Data Aggregation in Wireless Sensor Networks: Combining Grey Model and Kalman Filter, Computer Communications 34(6): 793-802, 2011.

[11] J. L. Deng, Introduction to Grey system theory, Journal of Grey System 1 (1) (1989) 1‐24.

[12] F. Xia, Z. Xu, L. Yao, W. Sun, M. Li, Prediction-Based Data Transmission for Energy Conservation in Wireless Body Sensors, The Int Workshop on Ubiquitous Body Sensor Networks (UBSN), in conjunction with the 5th Annual Int Wireless Internet Conf (WICON), Singapore, March 2010.

[13] Y. S. Suh, Send-On-Delta Sensor Data Transmission with a Linear Predictor, Sensors, 2007, 7(4): 537-547.

[14] Intel Lab data. Web page (accessed on 06/06/2011) http://db.lcs.mit.edu/labdata/labdata.html.

[15] National Oceanic and Athmospheric Administration’s (of the USA) National Data Buoy Center. Web page (accessed on 06/06/2011) http://www.ndbc.noaa.gov/historical_data.shtml.

[16] A. M. Alonso, C. García-Martos, Time Series Analysis - Autoregressive, MA and ARMA processes, Universidad Carlos III de Madrid, Universidad Politécnica de Madrid, June – July 2012.

[17] P. J. Brockwell and R.A. Davis, Time Series: Theory and Methods, Springer Series in Statistics (1986).

[18] C. Chatfield, The Analysis of Time Series: Theory and Practice, Chapman and Hall (1975).

[19] P. J. Diggle, Time Series: A Biostatistical Introduction, Oxford University Press (1990).

[20] M. Kendall, Time Series, Charles Griffin (1976).

[21] MATLAB Web page (accessed on 28/01/2013) http://www.mathworks.com/help/matlab/.