Energy-Efficient Data Acquisition By Adaptive Sampling for Wireless Sensor Networks

Yee Wei Law[1] Supriyo Chatterjea[2] Jiong Jin[1] Thomas Hanselmann[1]
Marimuthu Palaniswami[1]*
[1]Department of EEE, The University of Melbourne, Parkville, VIC 3010, Australia
{y.law, j.jin, t.hanselmann, m.palaniswami}@ee.unimelb.edu.au
[2]Faculty of EEMCS, University of Twente, P.O. Box 217, 7500AE Enschede, The Netherlands
supriyo@cs.utwente.nl

ABSTRACT

Wireless sensor networks (WSNs) are well suited for environment monitoring. However, some highly specialized sensors (e.g. hydrological sensors) have high power demand, and without due care, they can exhaust the battery supply quickly. Taking measurements with this kind of sensors can also overwhelm the communication resources by far. One way to reduce the power drawn by these high-demand sensors is adaptive sampling, i.e., to skip sampling when data loss is estimated to be low. Here, we present an adaptive sampling algorithm based on the Box-Jenkins approach in time series analysis. To measure the performance of our algorithms, we use the ratio of the reduction factor to root mean square error (RMSE). The rationale of the metric is that the best algorithm is the algorithm that gives the most reduction in the amount of sampling and yet the the smallest RMSE. For the datasets used in our simulations, our algorithm is capable of reducing the amount of sampling by 24% to 49%. For seven out of eight datasets, our algorithm performs better than the best in the literature so far in terms of the reduction/RMSE ratio.

Categories and Subject Descriptors

G.3 [**PROBABILITY AND STATISTICS**]: Time series analysis

General Terms

Algorithms

Keywords

Adaptive sampling, Box-Jenkins, ARIMA

*The authors are supported by the Australian Research Council Research Network on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), and the DEST International Science and Linkage Grant.

IWCMC '09, June 21-24, 2009, Leipzig, Germany

1. INTRODUCTION

In many monitoring applications, WSNs are often used to sample environment variables of unknown distributions, i.e., if we denote one such environment variable by Z_t, a random variable subscripted by time t, then $\Pr[Z_t = z]$ is unknown for all possible t's and z's. One practical problem is that some sensors, e.g. the EXCELL salinity sensor by Falmouth (`http://www.falmouth.com/products`), consume so much energy that if the sampling frequency is set too high, the sensor nodes would be depleted of energy too soon.

One option is to set the sampling frequency low but this is not always possible. The Nyquist-Shannon sampling theorem states that if a function $f(t)$ contains no frequencies higher than ω, it is completely determinable by a sampling process of frequency 2ω, i.e., the Nyquist frequency. Since it is impossible to determine the Nyquist frequency of an unknown function, the sampling frequency has to be set high.

An alternative approach is *adaptive sampling*, i.e., to let the sensors skip sampling whenever, based on existing samples, we can estimate the future readings we intend to skip accurately enough (to avoid confusion henceforth, we use *samples* to mean samples in the normal statistical context, and *readings* to mean the samples collected by a sensing process, but still use the continuous verb *sampling* to refer to the action or process of sensing). There are two issues to consider in the preceding proposal: (1) how do we estimate the samples that we intend to skip, and (2) how can we be sure we can estimate the samples accurately enough? These are the two problems we address in this paper.

Our contribution is an adaptive sampling algorithm that is capable of reducing the amount of sampling by 24% to 49% for the datasets used in our simulations. Our algorithm is based on the Box-Jenkins approach in time series analysis [6], and some heuristic improvements that might be useful for guiding the development of future adaptive sampling algorithms. To measure the performance of our algorithms, we use the ratio of the reduction factor to root mean square error (RMSE). The rationale of the metric is that the further the amount of sampling is reduced and the smaller the RMSE, the better an algorithm is. For seven out of eight datasets, our algorithm performs better than the best in the literature so far in terms of the reduction/RMSE ratio.

The rest of the paper is organized as follows. Section 2 discusses related work. Section 3 elaborates on the problem statement and outlines our solution. Section 4 lays out the essential definitions for later sections. Section 5 describes the algorithm and its improved variants in detail. Section 6

gives the simulation results. Finally Section 7 concludes.

2. RELATED WORK

An important energy conservation technique for WSNs is to approximate the time series captured by a sensor and synchronize the approximation with the sink. This is useful for answering queries, because instead of streaming back raw data, precious bandwidth and energy can be saved by sending back approximations. For example, a scheme by Olston et al. [17] answers queries by *bounded approximate answers*, which are essentially real-value pairs $[L, H]$ within which the exact answers are guaranteed to lie. Other approximation techniques include using Kalman filters to compute Markovian transition models of the time series [9], and deriving relatively coarse *piecewise constant approximations* of the time series [13]. The idea behind Jain et al.'s *dual Kalman filter* scheme [11] is to execute a Kalman filter at the sink, and another Kalman filter at a remote sensor. Both filters are used to predict future samples, and when the remote filter fails to predict a future sample within a certain precision constraint, an update is sent to the sink so that the sink-side filter can be updated accordingly. This scheme is probably the first known instance of the paradigm called *dual prediction* [14]. The potential drawback of dual Kalman filter is that a-priori knowledge about the time series is required. Chu et al. [8] extend Jain et al.'s idea by taking spatial correlation into account. Like dual Kalman filter, a Markovian model is maintained at the sink, and another is distributed in the network (not in a single node). The pair of Markovian models are synchronized whenever a reading deviates significantly from its forecast. Later advances focus on what model and how the model can be built. AR(3) models [21] [20], ARIMA models (coupled with a custom model selection criterion) [15] and least-mean-square adaptive filtering have been proposed [19]. In a recent work, the racing algorithm [16] has been suggested as an efficient way of selecting the best among AR(p) ($1 \leq p \leq 5$) models that describe a time series [14].

The problem addressed here, namely the problem of adaptive sampling, is different from the problem addressed by dual prediction. Dual prediction is aimed at reducing the transmissions of readings to the sink, while the readings are acquired at full sampling rate. Adaptive sampling is about reducing the amount of sampling independent of the dual prediction scheme employed. Adaptive sampling is important as some sensors are very demanding in terms of power consumption. Moreover, many sophisticated sensors used for environmental monitoring also have long start-up and sampling duration, thereby compounding the importance of reducing the amount of sampling. We elaborate on the difference between dual prediction and adaptive sampling from a data viewpoint. Dual prediction works by collecting readings, comparing the readings with the forecasts, and if the forecasts differ enough from the readings, updating the model. *Our problem – and we cannot stress this enough – is to determine when the forecasts might deviate significantly from the readings without actually acquiring the readings.* Due to the uncertainty involved with not having actual readings to compare the forecasts with, the problem we are addressing inevitably requires some level of heuristics.

The first proposal of an adaptive sampling scheme is probably by Chatterjea et al. [7]. Their scheme, written in pseudocode and labeled Algorithm 0 for ease of later discussion, is as follows:

Algorithm 0

```
Comment: CSSL = CurrentSkipSampleLimit, SS = SkipSamples,
MSSL = MaximumSkipSamplesLimit
Collect b samples
CSSL ← SS ← 0
repeat {
  Acquire 1 reading
  Use this new reading and the previous reading to interpolate
    samples skipped in the previous round, if any
  Make 1 forecast using all but the latest reading
  if (|reading - forecast| < ε)
    SS ← CSSL ← min(CSSL + 1, MSSL)
  else
    SS ← CSSL ← 0
  while (SS > 0) {
    Skip 1 reading
    SS ← SS − 1
  }
}
```

The justification for this algorithm is completely heuristic, that is, if after SS readings have been skipped, and the next reading is close to the next forecast, then the next $SS+1$ readings can be skipped (but at most $MSSL$ readings should be skipped); otherwise, we should resume acquiring every reading until the reading and the forecast are close to each other again. The algorithms we propose in this paper are built on a firmer theoretical foundation, although not without some minor heuristic adjustments.

3. PROBLEM STATEMENT AND SOLUTION OUTLINE

Recall that the two problems addressed by this paper are: (1) how do we estimate the samples that we intend to skip, and (2) how can we be sure that we can estimate the samples accurately enough? Note that we do not consider the problem of "how accurate is accurate enough" an issue because the required level of accuracy is often dependent on the requirements of domain experts, and it is usually represented by a user-specified error tolerance threshold, denoted ϵ.

To solve the first problem, we need to estimate future samples, based on existing samples. This is a problem of time series forecasting. The standard workflow consists of model identification, parameter estimation, model selection and forecasting. In the model identification phase, several candidate models are identified. Then for each candidate model, the parameters of the model are estimated. Lastly, the model among these candidates that satisfies some specified criteria best is selected to provide forecasts of future samples.

Besides time series, a rich variety of tools exist at our disposal. Table 1 lists some of the most well-known methods to date, their advantages and disadvantages. We choose to use the method of time series analysis – the Box-Jenkins approach in particular – because it has relatively low complexity and resource requirement (for example, it is practical to implement ARIMA algorithms on MICA2 motes [15]), and having a long history dating back to the 1970's, it is probably the best understood among the listed methods [6].

We now discuss how we use the Box-Jenkins approach. Let us denote by $\hat{Z}_n(l)$ the l-step ahead forecast of sample

Table 1: Time series forecasting methods

Method	Advantages	Disadvantages
Time series analysis	Relatively low complexity and resource requirement	No incremental update mechanism
State space methods	Numerical stability; insensitivity to small specification errors, statistical properties of parameter estimates; ease of handling for vector-valued or nonstationary time series	Relies on parameters whose identification proves to be difficult in settings where no a-priori knowledge on the signals is available [14]
Adaptive filters	Some of the most used approaches, e.g. Kalman filters, recursive least square filters	Stability is difficult to prove for arbitrary input sequence, but for practical applications they often work well
Support vector machines (classified by some authors under neural networks)	Quick for small to moderate-size problems; simple to use; optimization of constraint quadratic function with global minima	For large-scale problems, special tricks need to be applied to have tractable algorithms; query times depend on the number of support vectors
Neural networks	Many variants of algorithms; easy to use on difficult problems without much prior knowledge; quick query times	Training in general problem is difficult as cost function is non-convex in general; many local minima; slow to train; large data sets are often required

Table 2: Partial list of symbols

Symbol	Semantics	Symbol	Semantics
$\{Z_t\}$	Time series	Z_n	Sample at time $t = n$
$\lvert \cdot \rvert$	Cardinality of set $\cdot$	l	Forecasting horizon
ϵ	User-specified error tolerance threshold	Φ	Unit normal survival function
$N_{\alpha/2}$	$\Phi^{-1}(\alpha/2)$	b	Buffer size
$p_{\max}$, $q_{\max}$	Specified maximum values of p and q in ARIMA(p, d, q)	$\hat{Z}_n(l)$	Forecast of sample Z_{n+l} based on $Z_n, Z_{n-1}, \ldots$

Z_{n+l} (for the discussion henceforth, the symbols in Table 2 will be used). We mentioned that when $|Z_{n+l} - \hat{Z}_n(l)| < \epsilon$, we can skip sampling for Z_{n+l}. The problem is of course that, without actually knowing Z_{n+l}, it is impossible to know if $|Z_{n+l} - \hat{Z}_n(l)| < \epsilon$. However, for every $\hat{Z}_n(l)$ ($l \geq 1$), an associated confidence interval can be calculated. A confidence interval for $\hat{Z}_n(l)$ is a random interval, calculated from the samples, that contains Z_{n+l} with some specified probability. For example, we are interested in the confidence interval for $\hat{Z}_n(l)$ that contains Z_{n+l} at a probability of 90%. Intuitively, the further ahead in time we forecast, the bigger the confidence interval will be, i.e., the less we can be certain of the forecasted value. When the confidence interval is less than 2ϵ (2ϵ, because the forecast can either be lower or higher than the actual value), it is probably safe to skip sampling; otherwise, sampling should be continued or resumed.

Naturally, if the model used for forecasting is not accurate enough, we may never get a confidence interval that is actually smaller than ϵ. Hence, it is vital that the model used for forecasting is identified as accurately as possible under the physical constraints of the sensor nodes. Futhermore, the more samples we skip, the less actual data we will end up using to update the model, so with time, the confidence interval becomes only a heuristic indicator of when sampling can be skipped.

At this point, we have sketched the answers to the problems: (1) how we can estimate the future samples that we are intend to skip, and (2) how we can be sure that we can estimate the samples accurately enough. Namely, in a nutshell, we use the Box-Jenkins approach to forecast the next sample we intend to skip, and if the confidence interval of our forecast is less than 2ϵ, we deem the forecast accurate enough, skip acquiring the next reading and take the forecast as the next sample; otherwise, we proceed to acquire the next reading as usual. More detailed description of the algorithm is given in Section 5.

4. PRELIMINARIES

This section serves as a brief introduction to the components of time series analysis that are used in this paper.

A series of samples $\{Z_t\}$ is called a time series. $\{Z_t\}$ is covariance-stationary, or weakly stationary, or simply stationary, when (1) the mean $E[Z_t] = \mu$ is constant for all values of t, and (2) the jth covariance $E[(Z_t - \mu)(Z_{t-j} - \mu)] = \gamma_i$ only depends on j. Stationary series can be modeled using autoregressive (AR) models or moving average (MA) models, but a lot of time series in the real world are nonstationary by nature. For these time series, autoregressive integrated moving average (ARIMA) models can be applied. We now introduce ARIMA models via the following series of definitions.

DEFINITION 1. *A backward shift or lag operator, denoted B, when applied to Z_t j times, gives $B^j Z_t = Z_{t-j}$.*

DEFINITION 2. *An ARIMA(p, d, q) process is characterized by*

$$\phi_p(B)(1 - B)^d Z_t = \theta_0 + \theta_q(B)a_t,$$

where $\phi_p(B) = 1 - \phi_1 B - \phi_2 B^2 - \ldots - \phi_p B^p$, $\theta_q(B) = 1 - \theta_1 B - \theta_2 B^2 - \ldots - \theta_q B^q$, a_t is zero-mean white noise process, d is the order of differencing, and θ_0 is called the deterministic trend term when $d \geq 1$.

ARIMA(p, 1, q) can be used to model series whose *level* is continuously updated by random shocks, whereas ARIMA(p, 2, q) processes can be used to model series whose *level and slope* are continuously updated by random shocks. In practice, differencing a series twice ($d = 2$) is more than enough to transform a nonstationary series to a stationary series [4].

DEFINITION 3. *Let $\sigma_e^2 = \mathrm{Var}(\hat{Z}_n(l))$ and $N_{\alpha/2} = \Phi^{-1}(\alpha/2)$. If we assume the random variable Z_{n+l} to be normally distributed, then the confidence interval for $\hat{Z}_n(l)$ is given by*

$$[\hat{Z}_n(l) - N_{\alpha/2}\sigma_e, \hat{Z}_n(l) + N_{\alpha/2}\sigma_e].$$

To measure how our algorithms perform, we need a distance measure to quantify the distance between the original time series and the time series that is the output of our algorithms. Many distance measures have been proposed in the literature. In the time series literature, *Kullback-Leibler divergence* is widely used to measure the distance between two models (probability distributions), but requires the true model to be known, which is impractical. In the data mining

literature, a desired property of a distance measure is that it is insensitive to noise (see [12] for an extensive survey of distance measures). In our case, we need a distance measure that *is* sensitive to noise, because any noise/error that is due to our adaptive sampling algorithms must be taken into account. For this reason, a Euclidean distance measure is sufficient. We could use mean absolute error, or mean percentage absolute error, but we choose root mean square error (RMSE) because unlike the rest, it can be used as an unbiased estimator of the standard error.

5. THE ALGORITHM AND ITS IMPROVED VARIANTS

As outlined in Section 3, we use the Box-Jenkins approach to forecast the next sample we intend to skip, and if the confidence interval of our forecast is less than 2ϵ, we deem the forecast accurate enough, skip acquiring the next reading and take the forecast as the next sample; otherwise, we proceed to acquire the next reading as usual. The result of refining upon this sketch is Algorithm 1:

Algorithm 1

Initialize system parameters: b, $p_{\max}$, $q_{\max}$, l, ϵ
Collect samples $Z \leftarrow \{Z_1, Z_2, ..., Z_b\}$
Let pointer $n \leftarrow b$
repeat {
  Let $Z'' \leftarrow (1-B)^2 Z$
  Identify the best ARIMA(p, 2, q) model for Z'', where $0 \le p \le p_{\max}$, $0 \le q \le q_{\max}$, and $q \ne 0$ when $p = 0$
  Make l forecasts $\{\hat{Z}_n(1), ..., \hat{Z}_n(l)\}$ with corresponding confidence intervals $\{\tau_1, ..., \tau_l\}$
  $Z'' \leftarrow \{Z'', \hat{Z}_n(1), ..., \hat{Z}_n(l)\}$
  $Z \leftarrow (1-B)^{-2} Z''$

▶ Comment: Determine how many future readings we can skip
▶ $skip \leftarrow 0$
▶ **for** $i = 1$ to l {
▶ **if** $(\tau_i < 2\epsilon)$ $skip$++
▶ **else break**
▶ }
  Discard the first $skip$ and the last $(l - skip)$ members of Z s.t. $|Z| = b$
  Skip acquiring $skip$ samples

  Comment: For as many readings we have skipped, we should collect that many more readings again
  if $(skip > 0)$ $noskip \leftarrow skip$
  else $noskip \leftarrow l$
⋆ Discard the first $noskip$ members of Z s.t. $|Z| = b - noskip$
⋆ Collect samples $\{Z_{n+skip+1}, ..., Z_{n+skip+noskip}\}$
⋆ $Z \leftarrow \{Z, Z_{n+skip+1}, ..., Z_{n+skip+noskip}\}$
⋆ $n \leftarrow n + skip + noskip$
}

The user-configurable parameters are b, $p_{\max}$, $q_{\max}$, l and ϵ. b is the buffer size, or equivalently the number of samples in a time series. As dictated by majority of the experience reported in the literature, b should be set to 50 or more. The other parameters are sufficiently explained in Table 2.

The salient features of Algorithm 1 are:

1. The time series is always differenced twice to be converted from a potentially nonstationary series to a stationary one. The number of differencing is fixed at two because this is what is needed for most time series, and superfluous differencing does not do any harm [22].

2. Instead of making just one forecast, we make l forecasts, and if for $skip$ out of l forecasts, the corresponding confidence intervals are smaller than 2ϵ, we skip acquiring the next $skip$ readings, and use the $skip$ forecasts as the next $skip$ samples.

3. After skipping $skip$ readings, we compensate for the loss of actual data by acquiring $skip$ more readings. As such, 50% is the asymptotic upper limit of the reduction in sampling. This intentional limitation is meant to be an insurance against over-aggressive reduction.

However, Algorithm 1 may not work well for rapidly changing time series, in which case the confidence interval may be large – intuitively, the more rapidly a time series changes, the less confident we would be in forecasting future values – and easily larger than the user-specified ϵ, resulting in no reduction in sampling. To overcome this weakness, we apply a heuristic rule whereby if the *very first* confidence interval is larger than 2ϵ, we set ϵ to half of that confidence interval, in effect increasing our tolerance for uncertainty. We only look at the very first confidence interval, because this confidence interval, being based on the largest number of actual readings in the time series, is of the highest quality. This heuristic addition yields Algorithm 1a, which is now capable of handling rapidly changing time series:

Algorithm 1a

... Same as Algorithm 1 ...

▶ Comment: Determine how many future readings we can skip;
▶ the check $n == b$ ensures we only do this at the very start
▶ **if** $(n == b$ **and** $\epsilon < \tau_1/2)$ $\epsilon \leftarrow \tau_1/2$
▶ $skip \leftarrow 0$
▶ **for** $i = 1$ to l {
▶ **if** $(\tau_i < 2\epsilon)$ $skip$++
▶ **else break**
▶ }

... Same as Algorithm 1 ...

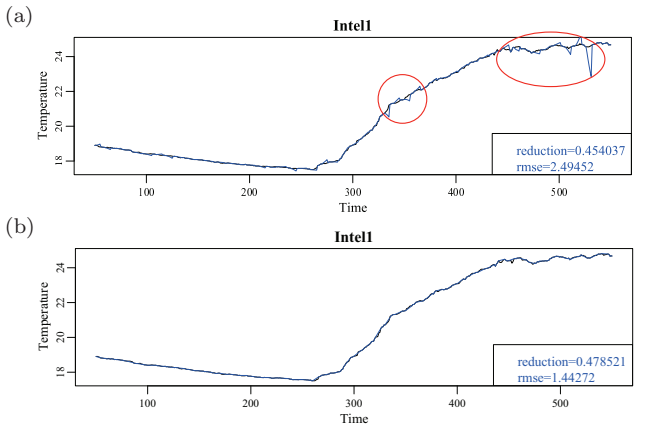

Figure 1: (a) Example of spikes due to bad forecasts by Algorithm 1a; (b) the smoothening effect of Algorithm 1b results in smaller RMSE. Note: black curves represent the original time series, whereas blue curves represent the time series generated by Algorithm 1a or 1b.

While Algorithm 1a can now handle rapidly changing time series, a weakness shared by both Algorithm 1 and Algorithm 1a is that spikes may arise due to bad forecasts (Figure 1(a)). To remedy this situation, we add another heuristic rule to Algorithm 1a to smoothen out these spikes. The smoothening device we use is interpolation, that is, we override the forecasted samples in the time series with values interpolated from (1) the sample immediately before the skip, and (2) the sample immediately after the skip. The result is Algorithm 1b, the smoothening effect of which can be seen in Figure 1(b).

Algorithm 1b

... Same as Algorithm 1a ...

⋆ Discard the first *noskip* members of Z s.t. $|Z| = b - noskip$
⋆ Collect samples $\{Z_{n+skip+1}, ..., Z_{n+skip+noskip}\}$
⋆ **if** $(skip > 0)$ {
⋆ $(x_0, x_2) \leftarrow (|Z| - skip, |Z| + 1)$
⋆ $(y_0, y_2) \leftarrow (Z_{x_0}, Z_{n+skip+1})$
⋆ **for** $x_1 = |Z| - skip + 1$ to $|Z|$
⋆ $Z_{x_1} \leftarrow y_0 + (x_1 - x_0)(y_2 - y_0)/(x_2 - x_0)$
⋆ }
⋆ $Z \leftarrow \{Z, Z_{n+skip+1}, ..., Z_{n+skip+noskip}\}$
⋆ $n \leftarrow n + skip + noskip$

... Same as Algorithm 1a ...

6. SIMULATION

There are two metrics by which the performance of the algorithms can be measured: reduction factor and RMSE. Reduction factor measures the fraction of sampling that can be avoided, whereas RMSE measures the discrepancy between the actual time series and the adaptively sampled time series. Instead of looking at the two metrics separately, we use the ratio reduction/RMSE to measure the performance of the algorithms. When there is no reduction, RMSE is 0, in which case we set reduction/RMSE to 0. Conversely, when RMSE is 0, there is almost always no reduction, in which case we also set reduction/RMSE to 0.

We compare Algorithms 1, 1a and 1b with Algorithm 0 (Section 2). We parameterize Algorithm 1 (also 1a and 1b) according to the values of $p_{\max}$ and $q_{\max}$. For example, Algorithm 1b(3,0) refers to an instantiation of Algorithm 1b that chooses the best model among ARIMA(1,2,0), ARIMA(2,2,0) and ARIMA(3,2,0). We choose $(p_{\max}, q_{\max}) \in \{(1,0), (3,0), (5,0), (3,3)\}$ because the first three combinations are reported in the literature [7] [21] [20] [14], and the last combination is meant to provide new perspectives on how using ARIMA instead of pure AR models might improve forecasting. In our simulations, we choose the model that gives the lowest value of Akaike's Information Criterion (AIC) [3] as the best model, although the racing algorithm [14] should give better efficiency. We set $b = 50$ and $l = 5$ for all simulations. We vary ϵ according to the datasets as listed in Table 3. Our simulations are scripted in R, and for each simulation, 5000 samples are processed. Figure 2 shows the result.

An analysis of the result follows. As shown in Figure 2, Algorithm 1 fails to reduce the amount of sampling for 41001h2007WSPD and 41001h2007GST at all. The reason is discovered to be that the confidence intervals turn out to be constantly larger than ϵ. Algorithm 1a rectifies this shortcoming by taking the first confidence interval as ϵ, if the first confidence interval is larger than ϵ. As a result, the reduction factor improves. Algorithm 1b further improves on Algorithm 1a by smoothening out the spikes. Figure 3 shows how closely the curves generated by Algorithm 1b approximate the original curves of 41001h2007WSPD and 41001h2007GST.

We see that for most cases, providing more models to choose from (e.g., $(p_{\max}, q_{\max}) = (3,3)$ compared with ($p_{\max}$, $q_{\max}$)=(1,0)) does not necessarily improve the reduction/RMSE ratio of the algorithm. One explanation is that due to the smoothening effect of Algorithm 1b(1,0), the contribution of Z_{n-2}, Z_{n-3}, ... to Z_n is greatly diminished compared to the contribution of Z_{n-1} to Z_n. In fact, with respect to all these datasets except Olga, Algorithm 1b(1,0) emerges as the best overall performer, and notably better than the benchmark Algorithm 0. The above observation can be gleaned from Table 4, which also shows that Algorithm 1b(1,0) is capable of reducing the amount of sampling by 24% to 49% for these particular datasets.

Table 3: Datasets and the corresponding values of ϵ used in the simulations

Dataset	ϵ	Dataset	ϵ
Olga [5]	0.3	41001h2007WSPD [1, wind speed]	0.3
Intel1 [2, temperature]	0.3	41001h2007GST [1, gust speed]	0.3
Intel2 [2, humidity]	0.3	41001h2007WVHT [1, wave height]	0.3
Intel3 [2, light]	3.0	41001h2007PRES [1, pressure]	1.0

Table 4: Comparison of Algorithm 0 and Algorithm 1b(1,0) in detail

Dataset	Algorithm 0			Algorithm 1b(1,0)		
	Reduc.	RMSE	$\frac{Reduc.}{RMSE}$	Reduc.	RMSE	$\frac{Reduc.}{RMSE}$
Olga	0.49	0.15	3.37	0.49	0.17	2.93
Intel1	0.80	4.09	0.20	0.47	2.03	0.23
Intel2	0.72	1.50	0.48	0.37	0.75	0.49
Intel3	0.62	5.26	0.12	0.24	0.18	1.36
41001h2007WSPD	0.20	0.34	0.59	0.43	0.59	0.73
41001h2007GST	0.17	0.37	0.46	0.49	0.81	0.61
41001h2007WVHT	0.62	0.12	2.32	0.28	0.06	2.60
41001h2007PRES	0.57	0.25	2.32	0.43	0.17	2.60

7. CONCLUSION AND FUTURE WORK

We have developed an adaptive sampling algoritm based on the Box-Jenkins approach in time series analysis. After observing some shortcomings of the base algorithm with respect to the rigidity of the error tolerance threshold and the existence of undesirable spikes, we have incorporated some heuristic adjustments that drastically improve the base algorithm. The final best overall performer, Algorithm 1b(1,0), for seven out of eight datasets used in the simulations, performs better than the best in the literature so far in terms of the reduction/RMSE ratio. Algorithm 1b(1,0) is capable of reducing the amount of sampling by 24% to 49%, with respect to all the datasets.

There are still a host of other methods (Table 1) to explore. For the near future work, compressive sensing [10] is next on our agenda. Another near future work is to integrate dual prediction with adaptive sampling in a seamless architecture that conserves energy not only in sampling but also in communication. We also have not addressed the issue of outliers in adaptive sampling. Some existing work in robust regression analysis [18] could be carried over without much difficulty but this has yet to be investigated.

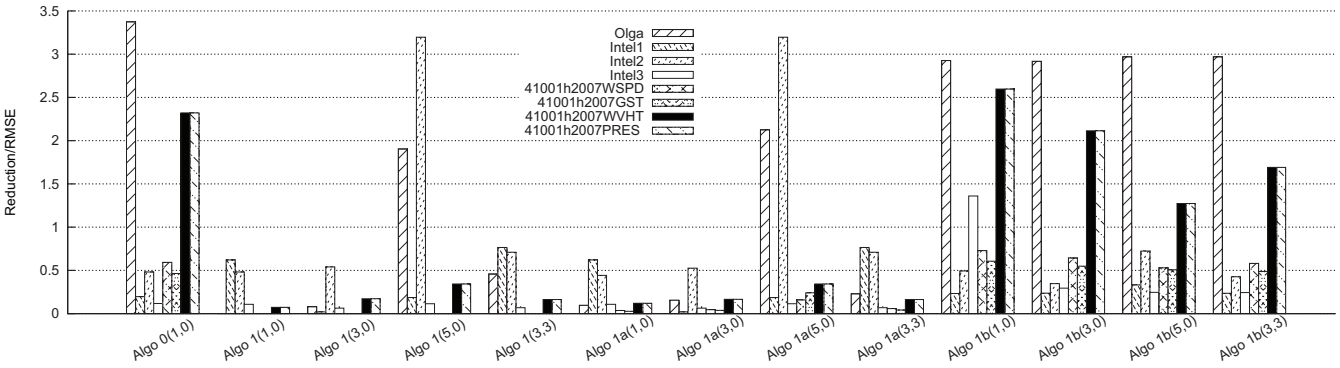

Figure 2: Reduction/RMSE ratios of various algorithms with respect to various datasets. "Algo $i(j,k)$" refers to Algorithm i with $p_{\max} = j$ and $q_{\max} = k$.

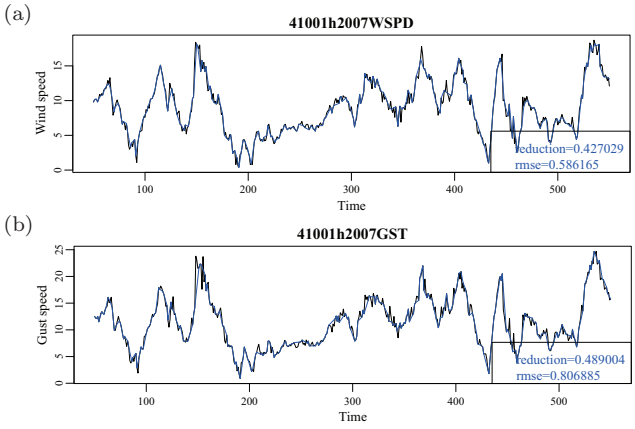

Figure 3: Algorithm 1b(1,0) produces curves that closely approximate the curves of (a) 41001h2007WSPD and (b) 41001h2007GST respectively. Without the improvement introduced by Algorithm 1b, Algorithm 1 cannot reduce sampling of these datasets at all. Note: black curves represent the original time series, whereas blue curves represent the time series generated by Algorithm 1b(1,0).

8. REFERENCES

[1] Standard meterological data of 2007 from the NOAA's National Data Buoy Center, `http://www.ndbc.noaa.gov/view_text_file.php?filename=41001h2007.txt.gz&dir=data/historical/stdmet/`.

[2] Intel Lab Data, `http://db.csail.mit.edu/labdata/data.txt.gz`.

[3] H. Akaike. A new look at the statistical model identification. *IEEE Trans. Automat. Contr.*, 19(6):716–723, Dec. 1974.

[4] O. Anderson, editor. *Forecasting*. North-Holland Publishing Company, 1979.

[5] O. Bondarenko, S. Kininmonth, and M. Kingsford. Underwater sensor networks, oceanography and plankton assemblages. In *Proc. ISSNIP 2007*, pages 657–662. IEEE, 2007.

[6] G. E. P. Box, G. M. Jenkins, and G. C. Reinsel. *Time Series Analysis: Forecasting and Control*. Wiley Series in Probability and Statistics. Wiley, 4th edition, 2008.

[7] S. Chatterjea and P. Havinga. An adaptive and autonomous sensor sampling frequency control scheme for energy-efficient data acquisition in wireless sensor networks. In *Proc. Distributed Computing in Sensor Systems (DCOSS)*, volume 5067 of *LNCS*, pages 60–78. Springer-Verlag, 2008.

[8] D. Chu, A. Deshpande, J. Hellerstein, and W. Hong. Approximate data collection in sensor networks using probabilistic models. In *Proceedings of the 22nd International Conference on Data Engineering (ICDE '06)*, pages 48–59, Apr. 2006. IEEE.

[9] A. Deshpande, C. Guestrin, S. Madden, J. Hellerstein, and W. Hong. Model-driven data acquisition in sensor networks. In *Proc. VLDB'04*, pages 588–599, 2004.

[10] D. L. Donoho. Compressed sensing. *IEEE Trans. Inform. Theory*, 52(4):1289–1306, Apr. 2006.

[11] A. Jain, E. Y. Chang, and Y.-F. Wang. Adaptive stream resource management using Kalman Filters. In *Proc. SIGMOD '04*, pages 11–22, 2004. ACM.

[12] E. Keogh and S. Kasetty. On the need for time series data mining benchmarks: a survey and empirical demonstration. *Data mining and knowledge discovery*, 7:349–371, 2003.

[13] I. Lazaridis and S. Mehrotra. Capturing sensor-generated time series with quality guarantee. In *Proc. ICDE'03*, pages 429–440, 2003. IEEE.

[14] Y.-A. Le Borgne, S. Santini, and G. Bontempi. Adaptive model selection for time series prediction in wireless sensor networks. *Signal Processing*, 87(12):3010–3020, Dec. 2007.

[15] C. Liu, K. Wu, and M. Tsao. Energy efficient information collection with the ARIMA model in wireless sensor networks. In *Proc. GLOBECOM '05*, volume 5, pages 2470–2474, 2005. IEEE.

[16] O. Maron and A. W. Moore. The racing algorithm: Model selection for lazy learners. *Artificial Intelligence Review*, 11:193–225, 1997.

[17] C. Olston, J. Jiang, and J. Widom. Adaptive filters for continuous queries over distributed data streams. In *Proc. SIGMOD '03*, pages 563–574, 2003. ACM.

[18] P. J. Rousseuw and A. M. Leroy. *Robust Reression and Outlier Detection*. Wiley, 2003.

[19] S. Santini and K. Römer. An adaptive strategy for quality-based data reduction in wireless sensor networks. In *Proc. INSS 2006*, 2006.

[20] D. Tulone and S. Madden. An energy-efficient querying framework in sensor networks for detecting node similarities. In *Proc. MSWiM '06*, pages 191–300, 2006. ACM.

[21] D. Tulone and S. Madden. *Wireless Sensor Networks*, volume 3868 of *LNCS*, chapter PAQ: Time Series Forecasting for Approximate Query Answering in Sensor Networks. Springer-Verlag, 2006.

[22] W. W. Wei. *Time Series Analysis: Univariate and Multivariate Methods*. Addison-Wesley Publishing Company, 1990.