Predicting battery depletion of neighboring wireless sensor nodes

DOI:
10.1007/978-3-319-03889-6_32

Document status and date:
Published: 01/01/2013

Document Version:
Accepted manuscript including changes made at the peer-review stage

Predicting Battery Depletion of Neighboring Wireless Sensor Nodes

Roshan Kotian[1], Georgios Exarchakos[1], Decebal Constantin Mocanu[1], and Antonio Liotta[1]

Eindhoven University of Technology, Den Dolech 2, 5612 AZ Eindhoven
{r.kotian,g.exarchakos,d.c.mocanu,a.liotta}@tue.nl

Abstract. With a view to prolong the duration of the wireless sensor network, many battery lifetime prediction algorithms run on individual nodes. If not properly designed, this approach may be detrimental and even accelerate battery depletion. Herein, we provide a comparative analysis of various machine-learning algorithms to offload the energy-inference task to the most energy-rich nodes, to alleviate the nodes that are entering the critical state. Taken to its extreme, our approach may be used to divert the energy-intensive tasks to a monitoring station, enabling a cloud-based approach to sensor network management. Experiments conducted in a controlled environment with real hardware have shown that RSSI can be used to infer the state of a remote wireless node once it is approaching the cutoff point. The ADWIN algorithm was used for smoothing the input data and for helping a variety of machine learning algorithms particularly to speed up and improve their prediction accuracy.

1 Introduction

When sensor nodes operate in harsh environments, there are many points of failures. They need to have enough computational intelligence to cope with failures [8]. One of the main causes of failure could be fast, unpredictable battery depletion of the nodes. Failure of strategic nodes can bring down the entire network and is not favorable for the end users who depend on it for their day-to-day operation. Besides the support of critical applications, battery level prediction is important for self-organization of wireless sensor networks (WSN). An important action for topology control in a WSN is the power scaling of the transmitters. The aim of such action is to improve the connectivity of the transmitter and reduce the interference in a highly dense wireless network. However, a node is not aware of how the transmission power should be scaled in order to avoid shadowing the neighbors. Symmetrically, in self-organized networks, nodes may react on behalf of their neighbors to report a critical state to a monitoring system.

Within that context, this paper analyses the possibilities of using machine learning algorithms to infer the critical state of the battery of a neighboring node.

Inference of that state can be used in both transmitter power scaling or collaborative cloud-based monitoring. Failing nodes which lack the power to transmit their state can be reported by neighboring nodes to a cloud service with a global overview of the network status. The cloud aggregates multiple data about the failing node from neighbors. It is, hence, more safely inferred whether that node is a strategic node and whether it can easily be assumed that the reported node is reaching the cutoff point.

The approach presented is a two-step processing of Received Signal Strength Indicator (RSSI) values. RSSI was chosen as it is an already available indicator in every sensor node and provides some indirect information about the remote transmitting node. The RSSI values are filtered at the node level with a fast inexpensive data smoothing algorithm. Then, the smoothened values are submitted to prediction algorithms running in the cloud for estimating the voltage level those values correspond to.

The contribution of this work lies on the comparative analysis of various well-established machine learning algorithms for predicting the voltage level of a remote node using exclusively RSSI values. Although our experimentations show that, the nature of RSSI values does not allow for an early and accurate inference of the nodes current voltage level, we found that the cutoff point is very quickly detectable by many algorithms. However it is essential that the chosen data smoothing algorithm (ADWIN) [3] does not only prune the outliers but also significantly reduces the amount of necessary data points for training the learning algorithms.

The remaining parts of this study are as follows. Section 2 provides a focused criticism on similar efforts to estimate the battery depletion rate. Section 3 is a description of the envisioned cloud-based system; and section 4 describes our experimental setup. Section 5 analyses the conducted experiments; and section 6 concludes and provides suggestions for further research steps on this topic.

2 Related Work

In the literature, we find two broad categories of techniques to maximize the lifetime of a sensor network based, respectively on 1) the prediction of the energy consumption in the WSN and 2) the prediction of the battery depletion of the sensor network. The latter one is an indirect way in the sense that knowing how fast the energy is depleting can help the network engineer replace the dying batteries of the node and thereby extend the operation time of the network. Further, the battery depletion techniques can be further classified into: a) battery life modeling and b) estimating techniques.

[7] points out that both the Received Signal Strength Indicator (RSSI) and the Link Quality Indicator (LQI) become unstable shortly before the depletion of the

nodes battery. Based on the fact that as RSSI values deteriorate, Inacio et al [12] used six mathematical models such as Simple Average, linear regression, Auto regressive, etc. They found that auto regression could adequately represent the charge depletion process thereby permitting to predict the node behavior and to detect the moment to replace its batteries.

However, our experimental results point out that RSSI values are so unpredictable that the accuracy of most of the classification algorithms are not plausible to claim that battery depletion can be predicted by the RSSI parameter alone.

3 System Overview

Figure 1 represents the system overview.

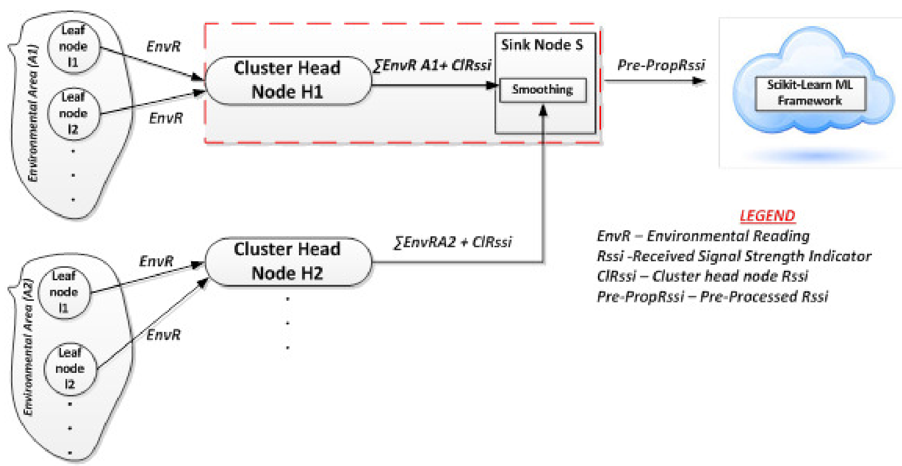

Fig. 1: System Overview

$\sum_{i=1}^{n} l_i$ represents the end nodes that are responsible for collecting the domain specific readings such as temperature, humidity, etc.
$\sum_{i=1}^{n} C_i$ represents the cluster head nodes that aggregates the data at local level. Once the head nodes collect aggregated data, it transmits it to the sink node S. The sink node S contains the ADWIN algorithm that does the pre-processing of the data and sends the pre-processed data to the cloud, which runs the popular scikit-Learn Machine Learning framework [10].

Due to environmental conditions such as interference or temperature, the RSSI values are non-linear in nature. Providing the Machine Learning algorithms with data having sharp variations can give less accurate predictions. In order to increase the prediction accuracy of the machine learning algorithms and to reduce the number of outliers, we need to smooth the data. Due to sharp variations in the RSSI values, it is not possible to classify whether the battery level is good, average, or bad with single RSSI value. Hence, we need to maintain a window that keeps the most recently read RSSI values. Furthermore, since machine learning is a time consuming process it should be triggered only when the average of the sequence of RSSI values in the window crosses the sensitivity threshold set by the network engineer. To meet the above-mentioned requirements, the algorithm of choice for our experiments is ADWIN as it uses the concept of sliding window allows for the engineers to set the sensitivity threshold an a priori parameter.

We used the scikit-Learn Machine learning framework to check how feasible it is to predict the battery depletion level classes (good, average, and bad), based on various popular classification algorithms.

4 Experimental Setup

Since RSSI is the sum of the pure received signal and the noise floor [1], it is important to reduce the noise floor to get accurate received signal strength readings. Therefore, the experimentation was conducted inside an anechoic chamber that is an interference free room.

The noise in the sensor node communication is introduced due to co-location of 802.11b network [11]. In addition to this, the presence of Bluetooth network and domestic appliances can significantly affect the transmission in the IEEE 802.15.4 network [1, 13].

Since the concurrent transmission from other nodes in the network can introduce the noise in the communication channel [9] and for the sake of simplicity, only the communication between one cluster head node (transmitting) and sink node (receiving) was performed.

We conducted two sets of experiments using CrossBows TelosB motes. In the first setup, the distance between the transmission node and the receiving node was set to 2 meters. In the second setup, the distance was increased by 5 meters. The battery depletion of the transmitting cluster head node was emulated using Benchmark power supply.

The following settings were kept constant for the entire experiment.

1. The transmitting node was configured to send the data to the receiving node every 250 ms.
2. The receiving node connected to the laptop was our sink node.
3. The position of the transmitting and the receiving node was not changed during the entire experimentation process.
4. The amps were set at 0.25mA.
5. For every voltage ranging from 3V to 1.5V, 1000 RSSI reading were taken.

5 System Evaluation

This section presents the experimental results from conducting the aforementioned experiments. The section is split in two parts: data smoothing and battery level prediction. Data smoothing is executed at the sensor node level and aims at reducing either the processing or the communication or both. Moreover, it contributes on the efficiency of the prediction algorithm by reducing the outlier data points and, hence, the overlap of the classes used at classifiers or reducing the bias in the regression models. Smoothened data are the input to the prediction algorithms which are running in the cloud. The algorithms are evaluated based on their accuracy and speed.

Fig. 2 presents the raw input RSSI data in the two datasets as well as the

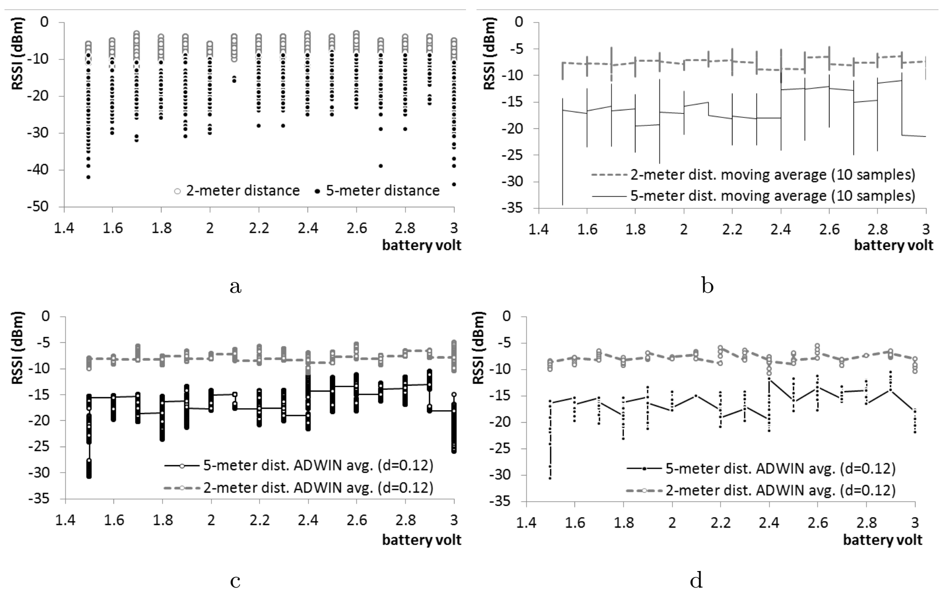

Fig. 2: Raw and pre-processed input data from the two monitored network conditions (two datasets for 2-meter and 5-meter distance between two sensors). Fig. 2a illustrates all raw datapoints, Fig. 2b presents the moving average of those datasets with a sliding window of 10 samples, Fig. 2c depicts the output of ADWIN algorithm in verbose mode and Fig. 2d illustrates the ADWIN output when the window size changes. you need to add the labels a to d to the actual plots

smoothing of that data using a naive moving average (Fig. 2b) and the ADWIN (Fig. 2c and Fig. 2d) algorithms. The moving average algorithm outputs the average value of a window of the 10 latest samples for every new RSSI value received. The ADWIN algorithm is used in two modes:

Table 1: Legends for Table 2, 3, 4, and 5

SVM-RBK:	Support Vector Machine with Radial Basis Kernel[6]
SVM-PK:	Support Vector Machine with Polynomial Kernel[6]
GMM:	Gaussian Mixture Model[4]
RFT:	Random Forest Trees[5]
KNN:	K Nearest Neighbors[4]
LogR+RBM:	Logistic Regression[4] built on a top of a Restricted Boltzmann Machine[2]
LogR:	Logistic Regression[4]
LR:	Linear Regression[4]
RC:	Random Classifier
Not available(NA):	Algorithm was halted if being executed for more than 1 minute.

– *Verbose*: for every RSSI value received, ADWIN outputs the average value of the current window (Fig. 2c).
– *Change-detection*: ADWIN provides the average value of the last window just on moments of a change at the window size (Fig. 2d).

As shown in Fig. 2a, the two RSSI raw datasets are overlapping. On the one hand, the moving average algorithm filters out many outliers that were causing that overlap. On the other hand, ADWIN has reduced significantly the variance of the two datasets and has increased the gap in between. As expected, the data-points generated by ADWIN in Fig. 2d are significantly fewer than those in Fig. 2c as data are submitted to the prediction algorithms in the cloud solely upon a considerable change to the ADWIN window size.
An ADWIN window changes upon a shift of the estimated voltage level, i.e. concept, based on the received RSSI values. Had such concept shift not been present, there would also be no need for triggering the battery voltage level prediction algorithm. Therefore, ADWIN on change-detection mode reduces the communication overhead for the sensor nodes and the processing overhead for the prediction algorithms.
The input data shown in Fig. 2 are the training data for the prediction algorithms. Every training data-point in those datasets is classified to one of the 16 voltage levels (1.5v-3v). Therefore, any RSSI value from the testing datasets has to be fed into the prediction algorithm and classified to one of those levels i.e. classes. The output of ADWIN algorithm in both modes was used for the classification process. Tables 2 and 3 (please see Table 1 for the acronyms description) present the evaluation of various algorithms with regards to their accuracy (percentage of input data-points classified in the correct class) and execution time (seconds spent during training phase). From Table 2, it becomes clear that all tested algorithms perform at most twice as good as a random classifier. On the other hand, Table 3 demonstrates a slightly improved situation when the classifiers use ADWIN output exclusively when the window adapts to the concept drifting. However, even in that case (ADWIN in change-detection mode) their

Table 2: Prediction algorithms evaluation. Input data come from the output of ADWIN in verbose mode. RSSI values are classified to one of the 16 voltage levels i.e. classes

Classification Algorithms	**2-meter distance dataset**		**5-meter distance dataset**	
	Accuracy (%)	Time (sec)	Accuracy (%)	Time (sec)
SVM-RBK	10.76%	3.7464	10.51%	3.7499
SVM-PK	NA	> 1 minute	NA	> 1 minute
GMM	5.51%	2.3817	7.27%	3.2728
RFT	10.23%	0.3366	9.76%	0.3472
KNN	10.19%	0.0108	9.89%	0.0107
LogR+RBM	12.86%	4.1106	13.99%	4.0429
LogR	12.71%	0.1648	12.36%	0.1727
LR	5.62%	0.0277	10.56%	0.0221
RC	6.25%		6.25%	

Table 3: Prediction algorithms evaluation. Input data come from the output of ADWIN in change-detection mode. RSSI values are classified to one of 16 voltage levels i.e. classes

Classification Algorithms	**2-meter distance dataset**		**5-meter distance dataset**	
	Accuracy (%)	Time (sec)	Accuracy (%)	Time (sec)
SVM-RBK	7.86%	0.0010	12.10%	0.00309
SVM-PK	17.97%	0.4803	NA	> 1 minute
GMM	8.98%	0.0523	8.28%	0.0630
RFT	17.97%	0.0050	15.92%	0.0050
KNN	12.35%	0.0007	12.10%	0.0005
LogR+RBM	12.35%	0.0492	14.01%	0.0799
LogR	11.23%	0.0492	13.37%	0.0034
LR	7.86%	0.0019	12.74%	0.0003
RC	6.25%		6.25%	

performance is limited. Therefore, the results in tables 2 and 3 are inconclusive with regards to the inference of the battery level of a neighboring sensor node using only received RSSI values.

There are various reasons behind this inaccuracy. The input raw RSSI values have very high variance for each voltage level. This variance, in a well-controlled environment like the anechoic chamber, might be caused by the inaccuracy of RSSI register at the receiver, which, in TelosB nodes, varies for 6dBm. Moreover, the average RSSI value of any voltage level differs maximum 3dBm from any other level. These two issues create a very wide overlapping among the voltage classes that all the tested classifiers cannot easily detect.

However, during the experiments above we noticed that two voltage levels were more accurately inferred than others. As shown in Table 4 and Table 5, the classifiers can perform much better when just two classes are considered. Instead of 16 classes, the classifiers were trained with the same input data to classify

Table 4: Prediction algorithms evaluation. Input data come from the output of ADWIN in verbose mode. RSSI values are classified to one of two classes (1.5-1.6V or 1.7-3V)

Classification Algorithms	**2-meter distance dataset**		**5-meter distance dataset**	
	Accuracy (%)	Time (sec)	Accuracy (%)	Time (sec)
SVM-RBK	89.11%	1.5670	86.08%	2.1382
SVM-PK	NA	> 1 minute	NA	> 1 minute
GMM	67.08%	0.4543	64.14%	0.4907
RFT	83.19%	0.1664	79.19%	0.2405
KNN	81.10%	0.0109	76.95%	0.0110
LogR+RBM	87.44%	2.8912	87.41%	4.8276
LogR	92.44%	0.1741	84.65%	0.1659
LR	24.75%	0.0218	26.03%	0.0220
RC	50.00%		50.00%	

Table 5: Prediction algorithms evaluation. Input data come from the output of ADWIN in change-detection mode. RSSI values are classified to one of two classes (1.5-1.6V or 1.7-3V)

Classification Algorithms	**2-meter distance dataset**		**5-meter distance dataset**	
	Accuracy (%)	Time (sec)	Accuracy (%)	Time (sec)
SVM-RBK	86.51%	0.0004	85.35%	0.0009
SVM-PK	92.13%	0.3010	78.34%	10.8300
GMM	68.53%	0.0137	73.88%	0.0206
RFT	88.76%	0.0040	85.35%	0.0040
KNN	88.76%	0.0006	84.71%	0.0005
LogR+RBM	91.01%	0.0387	85.98%	0.0629
LogR	89.88%	0.0019	85.98%	0.0032
LR	29.21%	0.0003	31.84%	0.0003
RC	50.00%		50.00%	

data-points into either the 1.5V-1.6V class or the 1.7V-3.0V class. That classification can practically infer if the battery of the remote sensor has maximum 0.2V before it is drained. Table 4 presents an accuracy of tested classifiers up to 92.4% for the 2-meter distance dataset and up to 87.4% for the 5-meter distance dataset. The benefit of using ADWIN in change-detection mode is shown in Table 5 as the accuracy or execution time of many algorithms is considerably improved compared to Table 4.

6 Conclusions and Future Work

We found through our experimentation that the nature of the RSSI values does not allow for an early and accurate prediction of the stationary node's current voltage level. On the contrary, the cut-off point (1.6V and 1.5V), most of the time detectable by majority of the classification algorithms.

In the course of the experiment, we discovered that providing the classification algorithm with raw RSSI values reduces the accuracy of the prediction of the algorithms. The reason for this being large of number of outliers.
Furthermore, we found that it is not possible to classify whether the battery level is good, average, or bad with single RSSI value. Therefore, we needed to maintain a window that buffers the most recent RSSI values.
In addition to this, since classification algorithms are computational intensive process it should be triggered only when the average sequence of RSSI values in the windows exceeds the sensitivity set by the users.
To cater these demanding needs, we found ADWIN algorithm to be best suited to reduce the time and computing cost of the machine learning algorithms.

References

1. N. Baccour, A. Koubaa, L. Mottola, M. A. Zuniga, H. Youssef, C. A. Boano, and M. Alves. Radio link quality estimation in wireless sensor networks. *ACM Transactions on Sensor Networks*, 8(4):1–33, Sept. 2012.
2. Y. Bengio. Learning deep architectures for ai. *Found. Trends Mach. Learn.*, 2(1):1–127, Jan. 2009.
3. A. Bifet, G. Holmes, R. Kirkby, and B. Pfahringer. *Data stream mining.* Number May. 2011.
4. C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics).* Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.
5. L. Breiman. Random forests. *Mach. Learn.*, 45(1):5–32, Oct. 2001.
6. C. Cortes and V. Vapnik. Support-vector networks. *Mach. Learn.*, 20(3):273–297, Sept. 1995.
7. A. Forster, D. Puccinelli, and S. Giordano. Sensor node lifetime: An experimental study. In *2011 IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*, pages 202–207. IEEE, Mar. 2011.
8. A. Liotta. The cognitive NET is coming. *IEEE Spectrum*, 50(8):26–31, Aug. 2013.
9. L. Mottola, G. P. Picco, M. Ceriotti, S. Guna, and A. L. Murphy. Not all wireless sensor networks are created equal. *ACM Transactions on Sensor Networks*, 7(2):1–33, Aug. 2010.
10. F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
11. K. Srinivasan, P. Dutta, A. Tavakoli, and P. Levis. An empirical study of low-power wireless. *ACM Transactions on Sensor Networks*, 6(2):1–49, Feb. 2010.
12. I. H. Yano, V. C. Oliveira, E. Alberto, D. M. Fagotto, A. D. A. Mota, L. Toledo, and M. Mota. Predicting battery charge deplition in Wireless Sensor Networks using received signal strength indicator. *Journal of Computer Science*, 9(7):821–826, July 2013.
13. G. Zhou, T. He, S. Krishnamurthy, and J. A. Stankovic. Models and solutions for radio irregularity in wireless sensor networks. *ACM Transactions on Sensor Networks*, 2(2):221–262, May 2006.