Evaluation of Learning Algorithms for Optimal Policy Representation in Sensor-Network Based Human Health Monitoring Systems

Shuping Liu and Mi Zhang
Ming Hsieh Department of Electrical Engineering
University of Southern California
Los Angeles, CA, USA
{lius,mizhang}@usc.edu

***Abstract*— The paper suggests the methods for learning compact representation of the optimal decision policies in a Markov Decision Process (MDP) framework for sensor-network based human health monitoring systems. The learning of a small decision policy is key to deploying the model in small sensor nodes with limited memory. The decision process enables distributed sensor nodes to adapt their sampling rates in response to changing event criticality and the availability of resources (energy) at each node. The globally optimal policy is first calculated offline using an MDP and deployed onto each node. However, the space complexity of the representation is exponential in the number of the sensor nodes and discretization grain of the problem. In this paper, we compare the capability of the compact representation of the optimal decision policy by using different base supervised learners. The results show that unpruned decision trees and high confidence pruned decision trees provide the lowest error rate while the required node number of the decision tree is enough small to be stored in the sensors. Ensembles of lower-confidence trees are capable of perfect representation with only an order of magnitude increase in classifier size compared to individual pruned trees.**

***Keywords* -- Human Health Monitoring, Body Sensor Network, Markov Decision Process, Energy Efficiency, Policy representation, Supervised Learning**

I. INTRODUCTION

With the availability of inexpensive sensors, many new applications are possible with a network of embedded sensors. Our work is motivated by a human health monitoring application being developed at the Children's Hospital Los Angeles and the University of Southern California [1,2]. In this application, a patient's vital signs are continuously measured by using multiple physiological and metabolic sensors attached to the patient's body (also called a "body sensor network"). The sensors include body temperature, heart rate, blood oxygenation and interstitial fluid (ISF) alcohol level sensors. The energy consumed at each sensor node depends on the sampling rate at that sensor. For example, ISF alcohol sensor requires a pump to draw out ISF before a measurement can be made. Higher sampling rate also increases the radio transmission and data-processing in the sensor. This is an energy-intensive process and hence reduces the lifetime of that node [2,3]. However, lower sampling rate reduces the accuracy of the measurements.

A multi-agent Markov Decision Process (MDP) framework has been developed which represents the relative utilities of energy efficiency, system life time, and accuracy [3,4]. Note that in [2] authors only discuss single agent case. Value or policy iteration is used offline to calculate the optimal policy for the MDP before it is stored in the sensors, and the policy is represented in the sensor node as a lookup table.

However, the space complexity of the MDP policy is exponential in the number of sensors in the network and discretization grain of the problem. For health monitoring systems the latter will be more important. Little academic effort has been spent towards representing the large policy generated by the MDP in the sensors with the limited memory. In this paper, we are using supervised learning techniques to provide a more compact representation of the decision policy, which would allow the policy to be stored in individual sensors. We will study various learning mechanisms and evaluate them based on their solution to the memory problem. As far as we know, this is the first effort to represent a large policy in the limited-memory sensors by learning a compact pattern using supervised learning techniques.

This paper is organized as follows. In the second section we will describe architecture for MDP controller based health monitoring systems. Section 3 introduces the related works. Supervised learning techniques are presented in section 4. In section 5, we describe the numerical and simulation results. Finally we conclude the paper in section 6.

II. ARCHITECTURE FOR MDP CONTROLLER BASED HEALTH MONITORING SYSTEM

In the context of health monitoring systems, several sensor based architectures have been proposed [5,6]. Both systems record signal, such as heart rate and blood pressure. In [5], measured data will be displayed on the system and requires manual entering of data into computer for further analysis. In [6], the patient should be near the monitoring system for the system to take readings. This makes these systems inconvenient and inflexible. It would be more helpful for automatic transfer of measured data to computers via wireless communication. In this work, we are developing a system for health monitoring with such capacities. The goal is to continuously monitor vital signs of a patient and transfer the

Supported in part by NSF grant No. 0615132

measured sensor data to desktop computers via wireless communication. Since the system will have limited battery power, types of measurements and their frequencies have to be controlled to extend the life of the system. We are using a MDP controller based two-tier architecture, shown in Fig. 1, for monitoring local processing and communicating data to remote computers.

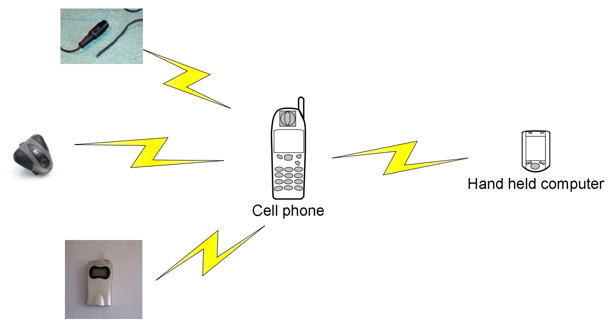

Figure 1. Two-tier architecture

In Fig. 1, it is possible to use the handheld computer as a centralized controller to regulate the sensing frequencies of all sensors. With multiple sensors, there will be high energy cost for sending control data from the handheld computer to sensors. Further, there can be failures in the communication and the sensors may not operate optimally. For these reasons, we use distributed control with each sensor storing control information about global policy computed offline. In our previous work, we implemented a MDP controller within each sensor so that it can make its own decision about how to choose the sensing frequencies based on the current situation. The global optimal policy (mapping from the current global states to actions) is calculated offline in advance with the assumption of full observation, which is loaded to MDP policy table in each sensor. But during execution, the world is partially observable, i.e., one sensor does not know the local status (consumed energy) of other sensors. Hence each sensor has a SM (Stochastic Model) model, which is used to predict the locally consumed energy of other sensors. From these estimated global status, the sensor will choose the action from the loaded MDP policy table, which is globally optimal. Whenever the sensors communicate their status to one another, then each sensor knows the true global status. However we want to reduce such communications to save energy. Therefore we developed a value of information (VI) based communication scheme. For each global status, we calculate its value of information. The sensor triggers a communication only when VI is greater than a given threshold, θ. The HM (Health Monitor) model monitors the patient regularly and the data sensed is sent to the network node via radio links. Fig. 2 shows the components for each sensor.

How to formulate our health monitoring system as a multi-agent MDP problem is detailed in [3,4]. We omit it here. In this paper, we address the problem how to represent the global policy table (potentially big) in memory-limited sensors by using machine learning techniques.

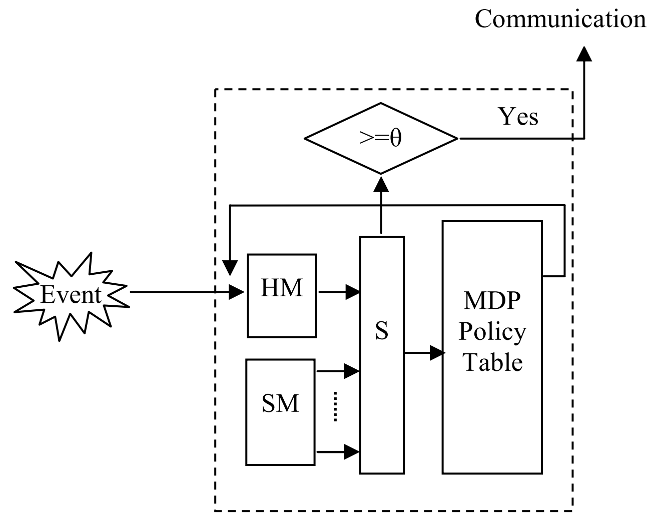

Figure 2. The components of one sensors

III. RELATED WORKS

Due to limited battery resources, energy efficiency in communication and processing is important in sensor networks to extend its lifetime. There has been extensive research in the past decade, ranging from hardware platforms to operating system, network protocols and application software [7-17] about energy efficiency. Because active state consumes significant amount of energy, sleep techniques are widely studied for energy savings [11, 18]. All these studies either focus on individual sensor or just save energy sub-optimally without considering the energy issue from the whole system point of view.

We developed a multi-agent MDP framework to optimally coordinate action taken (sampling rate), energy efficiency, and system life time through each sensor operates intelligently [3,4]. However, very little academic work focuses on how to represent a large policy in the sensors with limited memory size. J. L. De Coe et. al. [19] try to integrate all policies in one framework to solve policy confliction. In [20], J. Marecki et. al. suggest an algorithm called FANS to represent the policy for POMDP by employing finite state machine. Both works represent the policy in a short and compatible way. But the policy size is still huge when the number of agents in the networks increases, which is not specified for the sensors whose memory capacity is limited. In this paper, we are using supervised learning to learn a compact pattern of the policy and only store the pattern in the sensors, instead of policy table, which would allow the policy to be stored in individual sensors with limited memory.

IV. SUPERVISED LEARNING TECHNIQUES

Supervised learning is a machine learning technique for learning a function from training data with true labels. The training data consist of both inputs and outputs (labels). The learning task is to learn a function (hypothesis) that can predict the outputs for any valid inputs. To achieve this, the learner has to generalize from the presented data to unseen situations in a "reasonable" way. The output of the function can be continuous values, which are called regression, or can predict discrete values, which are called classification. The basic difference between machine learning and human learning is that we assume that task is stationary in machine learning while it changes typically in human learning. The typical base supervised learners are as follows (take two classes as example),

A. Perceptron

Perceptron is a classifier for learning Linear Threshold Units (LTU),

$$h(\mathbf{X}) = \begin{cases} +1 & if\ \mathbf{W}\cdot\mathbf{X} \geq W_0 \\ -1 & otherwise \end{cases}$$

The target is to learn the weight $\mathbf{W}$ and W_0. It can be solved by Gradient Descent or Ascent search.

B. Logistic Regression

Logistic Regression is also to learn LTU. But it uses the ratio of class-conditional densities, $\frac{p(x|\mathbf{W}^1, W_0^1)}{p(x|\mathbf{W}^1, W_0^1)}$. The algorithm assumes that log likelihood ratio is linear, i.e.,

$$log\frac{p(\boldsymbol{x}|\mathbf{W}^1, W_0^1)}{p(\boldsymbol{x}|\mathbf{W}^1, W_0^1)} = \mathbf{W}\cdot\mathbf{X} + W_0$$

Then we will get,

$$y = \widehat{P}((\mathbf{W}^1, W_0^1|\boldsymbol{x}) = \frac{1}{1+\exp[-(\mathbf{W}\cdot\mathbf{X} + W_0)]}$$

and $\widehat{P}((\mathbf{W}^2, W_0^2|\boldsymbol{x}) = \frac{\exp[-(\mathbf{W}\cdot\mathbf{X}+W_0)]}{1+\exp[-(\mathbf{W}\cdot\mathbf{X}+W_0)]}$, which can be solved by Gradient Descent or Ascent search.

C. Decision Tree

A decision tree is a hierarchical model for supervised learning, which is composed of internal decision nodes and terminal leaves. Each decision nodes m implements a test function $f_m(\boldsymbol{x})$ with discrete outcomes labeling the branches. Given an input, at each node, a test is applied and one of the branches is taken depending on the outcomes. This process starts at the root and is repeated recursively until a leaf node is hit, at which point the value written in the leaf constitutes the output [21].

D. Nearest Neighbor (NN)

NN classifies a new data $\boldsymbol{x}$ by finding the training example $(\boldsymbol{x_i}, y_i)$ that is nearest to $\boldsymbol{x}$ according to some distance metric, such as,

$$||\boldsymbol{x} - \boldsymbol{x_i}|| = \sqrt{\sum_j(\boldsymbol{x_i} - \boldsymbol{x_{ij}})^2}.$$

k-NN is to find k nearest neighbors and vote the results by all these k neighbors. NN usually does not explicitly compute decision boundaries.

E. Neural Networks

Neural Networks are layered networks of units connected by directed links. Each unit i first computes a weighted sum of its inputs,

$$in_i = \mathbf{W}\cdot\mathbf{X} - W_0$$

Then it applies an activation function g to this sum to derive the output [22],

$$output_i = \boldsymbol{g}(in_i) = \boldsymbol{g}(\mathbf{W}\cdot\mathbf{X} - W_0).$$

Neural Networks can represent the complex boundaries. It can be proved that any bounded continuous function can be approximated to arbitrary accuracy with enough hidden units [23,24]. And any function can be approximated to arbitrary accuracy with two hidden layers of sigmoid units and a linear output unit [25]. Neural Networks can be solve by Backpropagation Algorithm.

F. Bayesian Learning

Bayesian Learning is based on Bayesian Networks, which is a directed graph. Each node X_i has a conditional probability distribution $P(X_i|Parents(X_i))$ that quantifies the effect of the parents on the node [22]. Considering hypothesis space H and training data D, the hypothesis is chosen based on the following Maximum a Posteriori (MAP) rule,

$$\begin{aligned} h_{MAP} &= argmax_{h\in H}P(h|D) \\ &= argmax_{h\in H}\frac{P(D|h)P(h)}{P(D)} \\ &= argmax_{h\in H}P(D|h)P(h) \end{aligned}$$

Naïve Bayesian learning assumes that X_i is conditionally independent given the parents. This assumption is often violated in real world. But it works surprisingly well.

G. Support Vector Machine (SVM)

SVM is also to learn the linear discriminant. But it tries to maximize the margin and combines three important ideas,

- Apply optimization algorithms from Operations Research (Linear Programming)
- Implicit feature transformation using kernels
- Control of overfitting by maximizing the margin

V. NUMERIAL RESULTS

We first calculate the optimal policy offline by using multi-agent MDP model in section 4. When the policy is available, we next learn it by different supervised learning techniques so that we can obtain a more compact pattern of the policy to be stored in the sensor whose memory is limited. Both training data set and test data set are set to be the whole policy table.

For simplicity, we take 2-agent system as example in this section. More than 2-agent case has the similar results. The parameters used are as follows,

T=20	E=10
H=10	A=10

We compare different base learners in term of error rate, shown in Fig. 3. The error rate is calculated on the average of two classes on the training set, which we are using for both training and testing. This is because we have a gold-standard, which is the output of the optimal decision policy calculated using the MDP.

We can see that the best three base learners are Decision Tree (DT), K-Nearest Neighbor (NNbr) and Neural Network (NN). Fig. 4 shows the error rate for different k-values in the learned K-Nearest Neighbor classifier. The error rate for k=1 is zero,and the average error rate is very low, but the learned classifiers require us to store the whole policy in the sensor.

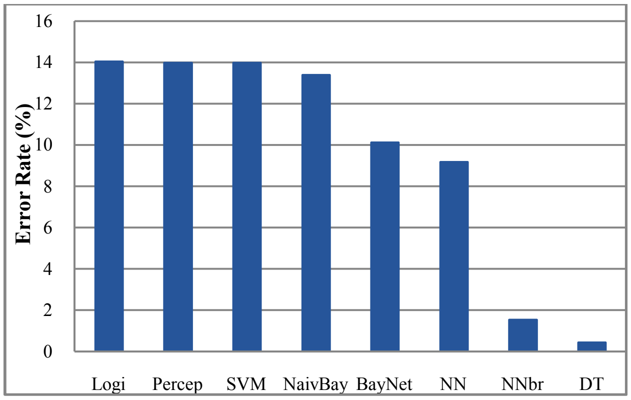

Fig. 3: Comparison of different supervised learners

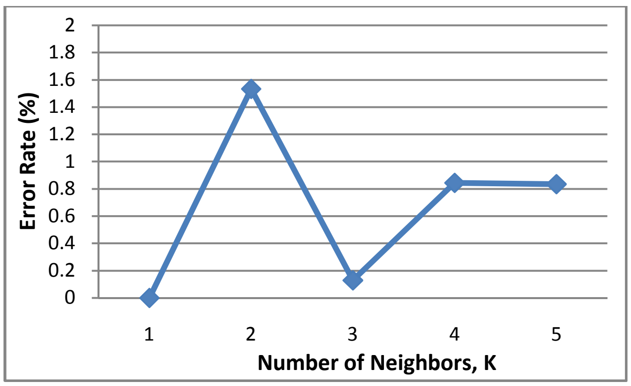

Fig. 4: The effect of neighbor number on error rate (NNbr)

The size of the classifier is just as large as the lookup table, and thus not a good solution to the problem. For the Neural Network algorithm, error rate decreases with the number of hidden nodes, shown in Fig. 5. Given enough internal nodes, the NN learned using backpropagation will approach zero error rate. However, its error rate for 6 internal nodes is still much higher than Decision Tree. To get a enough accuracy NN model, the computation time will be huge. Unpruned Decision tree and Pruned Decision Tree with higher confidence (C=1) give the best error rate, shown in Fig. 6. The higher the confidence factor is, the less likely the algorithm is to prune a node, thus when the confidence factor is lowered, the number of nodes is lessened, and the error is increased. The tree size is about 500 nodes, which is small enough to be stored in the sensors. Fig. 7 compares the memory size of the whole policy, unpurned decision tree and pruned decision tree with C=1. The results show that the memory size required for the optimal policy reduces from 222KB to 20KB when we employ the decision tree to learn the optimal policy table. The compression ratio is up to 11.1.

It is possible to achieve zero error rate by using ensemble learning methods on pruned decision trees with low confidence (C=0.00001) without increasing the classifier size more than about one order of magnitude. The dramatic reduction in error rate is expected when using boosting meta-learners like AdaBoost, since decision trees have fairly low bias and fairly high variance. Boosting reduces the amount of

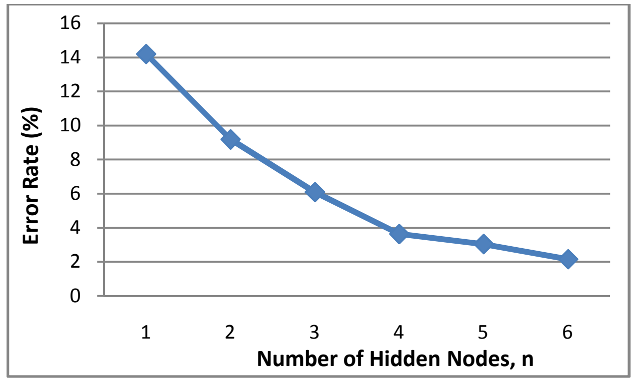

Fig. 5: The effect of number of hidden nodes on error rate (NN)

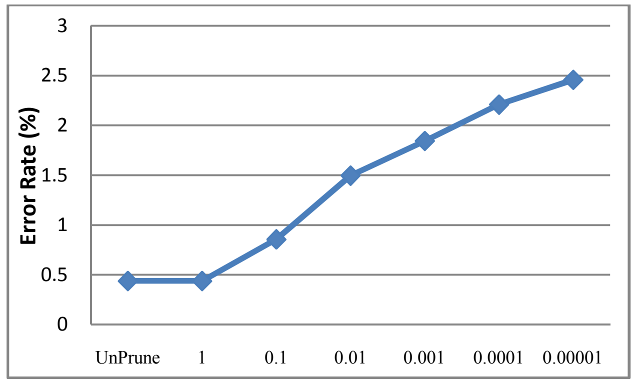

Fig. 6: The effect of confidence factor on error rate (decision tree)

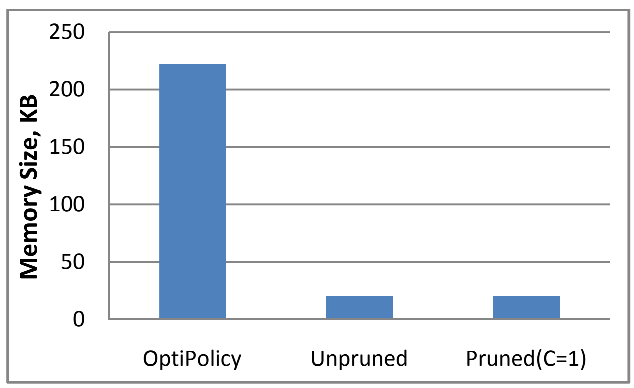

Fig. 7: The comparison of memory size for the optimal policy table,unpruend and pruned (C=1) decision tree

error due to variance, thus decreasing the overall error. Fig. 8 shows the memory size for the optimal policy table, AdaBoost with low confidence pruned decision tree (C=0.00001), high confidence pruned decision tree (C=1) and unpruend decision tree.

VI. CONCLSION AND FUTURE WORKS

A multi-agent MDP framework for sensor-network health monitoring of human patients provides an optimal policy when learned using policy or value iteration. However, the complexity of the optimal policy representation is too great to be practical when the number of sensors and discretization grain of the problem increases. We are using supervised

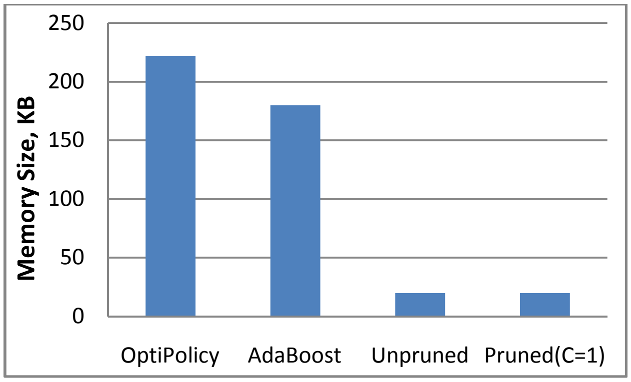

Fig. 8: The comparison of memory size for the optimal policy table, AdaBoost with low confidence pruned decision tree (C=0.00001, error rate=0), high confidence pruned (C=1) decision tree (error rate =0.45%) and unpruend decision tree (error rate =0.45%)

learning techniques to learn a compact representation of the optimal policy table so that it can be stored in the sensors whose memory is limited. We compare different base supervised learners. The results suggest that high-confidence pruned decision trees and unpruned decision trees are best for deploying in a limited-memory system, while still achieving very low error rates. It is possible to achieve zero error rate by using ensemble learning methods on pruned decision trees with low confidence (c=.00001) without increasing the classifier size more than about one order of magnitude. The dramatic reduction in error rate is expected when using boosting meta-learners like AdaBoost, since decision trees have fairly low bias and fairly high variance. Boosting reduces the amount of error due to variance, thus decreasing the overall error.

Neural networks may also be an ideal classifier for this problem, as they are relatively small. For instance, 6 neurons (and thus around 40 weights) is all that is needed to represent a classifier that achieves about 2% error. Based on the data in figure 12, it appears to be converging to zero fairly rapidly, it could be that just 12 internal neurons would be needed to get very nearly zero error. And since the number of weights is linear in the number of internal neurons (in one hidden layer), this means the classifier would still be very small.

These results suggest further study of neural networks and ensembles of other base learners. It may be possible to achieve zero error using a suitably large (yet still quite small) neural network.

References

[1] M. Venugopal, K. E. Feuvrel, D. Mongin, S. Bambot, M. Faupel, A. Panangadan, A. Talukder, and R. Pidva, "Clinical Evaluation of a Novel Interstitial Fluid Sensor System for Remote Continuous Alcohol Monitoring," *IEEE Sensors Journal,* vol. 8, pp. 71-80, 2008.

[2] A. Panangadan, S. M. Ali, and A. Talukder, "Markov Decision Processes for Control of a Sensor Network-based Health Monitoring System," in *Proceedings of the Seventeenth Innovative Applications of Artificial Intelligence Conference* Pittsburgh: AAAI Press, Menlo Park, California, 2005, pp. 1529-1534.

[3] S. Liu, A. Panangadan, A. Talukder and C. S. Raghavendra, "MDP Framework for Sensor Network Coordination," The 8th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN) Poster Session, San Francisco, CA, US, 2009.

[4] S. Liu, A. Panangadan, A. Talukder and C. S. Raghavendra, "Evaluation of a Markov Decision Process-based Coordinated Sampling Method," Workshop on Sensor Networks for Earth and Space Science Application, The 8th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN), San Francisco, CA, US, 2009.

[5] I. Korhonen, T. Iivainen, R. Lappalainen, T. Tuomisto, T. Koobi, V. Pentikainen, M. Tuomisto, and V. Turjanmaa. "TERVA: system for long-term monitoring of wellness at home," *Telemedicine Journal and e-Health* 7(1), 2001:61–72.

[6] M. Ogawa, T. Tamura and T. Togawa. "Automated acquisition system for routine, noninvasive monitoring of physiological data," *Telemedicine Journal* 4(2), 1998:177–185.

[7] L. Peshkin, N. Merleau, K. E. Kim, and L. Kaelbling. "Learning to cooperate via policy search," In UAI, 2000.

[8] I. Korhonen, T. Iivainen, R. Lappalainen, T. Tuomisto, T. Koobi, V. Pentikainen, M. Tuomisto, and V. Turjanmaa. "TERVA: system for long-term monitoring of wellness at home," *Telemedicine Journal and e-Health* 7(1), 2001:61–72.

[9] M. Ogawa, T. Tamura and T. Togawa. "Automated acquisition system for routine, noninvasive monitoring of physiological data," *Telemedicine Journal* 4(2), 1998:177–185.

[10] L. Benini and G. DeMicheli, "*Dynamic Power Management: Design Techniques & CAD Tools*," Norwell, MA: Kluwer Academic Publishers, 1997.

[11] A. P. Chandrakasan and R. W. Brodersen, "*Low Power CMOS Digital Design,*" Norwell, MA: Kluwer Academic Publishers, 1996.

[12] P. Lettieri, C. Fragouli, and M. Srivastava, "Low power error control for wireless links," in *Proceedings of Mobicom*, 1997, pp. 139-150.

[13] T. A. Pering, T. D. Burd, and R. W. Brodersen, "The simulation and evaluation of dynamic voltage scaling algorithms," in *Proceedings of ISLPED*, 1998, pp. 76-81.

[14] J. Rabaey, M. J. Ammer, J. L. d. Silva, D. Patel, and S. Roundy, "PicoRadio supports ad hoc ultra low power wireless networking," *IEEE Computer Magazine,* vol. 33, pp. 42 - 48, 2000-07 2000.

[15] C. S. Raghavendra, K. M. Sivalingam, and T. Znati, "Wireless Sensor Networks," Springer, 2006, pp. 3-107.

[16] V. Raghunathan, P. Spanos, and M. Srivastava, "Adaptive power-fidelity in energy aware wireless embedded systems," in *IEEE Real Time Systems Symposium*, 2001.

[17] C. Schurgers, O. Aberthorne, and M. Srivastava, "Modulation scaling for energy aware communication systems," in *Proceedings of ISLPED*, 2001.

[18] A. Sinha, A. Wang, and A. P. Chandrakasan, "Algorithmic transforms for efficient enrgy scalable computation," in *Proceedings of ISLPED*, 2000.

[19] J. L. De Coe, P. Karger, D. Olmedilla and S. Zerr, "Policy Representation & Reasoning,", presentation slide, Leibniz Hannover University, April 2008.

[20] J. Marecki, T. Gupta, P. Varakantham and M. Tambe, "Not All Agents Are Equal: Scaling up Distributed POMDPs for Agent Networks," In *Proceedings of AAMAS*, May 2008, Estoril Portugal.

[21] E. Alpaydin, "Introduction to Machine Learning," T*he MIT Press*, Cambridge, Massachusetts, London, Enland.

[22] S. J. Russell and P. Norvig, "Artificial Intelligence: A Modern Approach," Second Edition, 2003.

[23] G. Cybenko, "Approximation by Super Positions of a Sigmoidal Function," *Math Control, Signals and Systems*, Volume 2, 1989:304--314.

[24] K. Hornik, M. Stinchcombe and H. White, "Multilayer Feedforward Networks are Universal Approximators," *Neural Networks*, Volume 2, Issue 5, 1989: 359--366.

[25] G. Cybenko, "Continuous Value Neural Networks with Two Hidden Layers are Sufficient,", *Technical Report*, Tufts University, 1988.