Learning from Class-Imbalanced Data in Wireless Sensor Networks

Predrag Radivojac*, Uttara Korad†, Krishna M. Sivalingam† and Zoran Obradovic*
*Center for Information Science and Technology, Temple University, Philadelphia, PA 19122
†Department of CSEE, University of Maryland, Baltimore County (UMBC), Baltimore, MD 21250

***Abstract*— In this paper, we study wireless sensor networks used for detection of rare events (e.g. intrusion). The task of the sensor node is to collect data points (examples) at regular time intervals and communicate them to the central base station (BS) using wireless links. Since sensor nodes have limited battery power, it is necessary to minimize their energy consumption. One way is to reduce the amount of sensor data packets transmitted. In this paper, we incorporate machine learning strategies to intelligently reduce the amount of transmitted data, in order to increase life-span of the sensors and thus profitability of the system. In our proposed approach, after a short initialization period, the sensors obtain a classification model from the BS based upon which they detect interesting (positive) data points. Positive examples are, together with selected negative examples, then reported to the BS. In time, BS will have stored an abundant number of negatives and a limited number of positives causing what is termed as a *class-imbalance* problem in learning. In order to understand the impact of network architecture on learning performance, two different architectures are studied: cluster-based (LEACH) and tiered (UNPF). With the aid of experiments using generated data sets, the paper analyzes the tradeoffs between prediction success, learning cost, packets transmitted and energy consumed. The results show that the proposed learning mechanism significantly reduces energy consumption compared to the baseline system.**

I. Introduction

Recent advances in MEMS systems technology have led to the development of small inexpensive sensor devices equipped with one or more sensors, an embedded processor, low-power radio and battery. Several hundred such nodes can be interconnected to form a wireless sensor network [1], [2]. Networking these sensors provides several valuable capabilities: the system can cover a wider area of operation and provide desired properties such as redundancy, intelligent sensing, and improved accuracy. However, an important limitation is that the sensors have significantly limited computation, memory storage, communication, and battery power capabilities [3].

In this paper, we consider a wireless sensor network, composed of nodes distributed within a given geographic area, that is used for rare event (e.g. intrusion) detection purposes. The task of the sensor node is to detect specific rare events and communicate them to the central base station (BS) using wireless links. The BS is assumed to have unlimited battery and computational resources while the sensors are taken to be simplest possible devices running on batteries.

One way to reduce the amount of energy consumed and thus increase nodes' and network lifetimes is to intelligently reduce the amount of transmitted sensor data. This paper proposed the use of learning protocols to achieve this goal. In the baseline protocol, the sensors report all collected data to the BS. In the proposed model, the sensors use learned classification models to detect interesting events [4]–[6] and report all positively classified examples and a subset of negatively classified examples. In time, the BS, that has the ability to label all new examples, will have stored an abundant number of negatives and a limited number of positives causing a class imbalance problem from machine-learning point of view [7], [8]. The BS continuously retrains its classification models based on the received sensor data and periodically communicates updated model information to the sensors. The goal is to design a system that maximizes the number of detected rare events despite the data imbalance, maximizes the lifetime of the sensors (through reduced false alarms and lower energy consumption) and also overall network lifetime.

The two protocols are implemented in two different network organization architectures: a clustering based architecture called *Low-Energy Adaptive Clustering Hierarchy* (LEACH) [9] and a multi-hop tree-based architecture called MINA/UNPF [10]–[12]. The performance of the two learning protocols in both network architectures is compared using discrete-event simulation in terms of prediction accuracy, system cost, and energy consumption. The proposed learning protocol is shown to significantly reduce the number of transmitted packets and thereby energy consumed, while achieving desirable prediction accuracy.

II. Background

Wireless sensor networks have been actively considered in recent research and a summary of the key research issues are available in [1], [2]. Sensors have been studied for a wide variety of applications ranging from military, environmental monitoring to bio-medical applications [13], [14]. The key networking related issues that have been studied include medium access control and routing protocol design, data dissemination models, data aggregation, localization, and synchronization [1]. However, there is very little work to study the interaction between data mining algorithms and network protocols and architectures. This paper attempts to explore this area.

The application scenario considered in this paper is intrusion detection, where there are far more negative instances than positive instances. This is referred to as *class-imbalanced data*. In machine learning, this problem has to be carefully approached due to a possibility of different misclassification

0-7803-7954-3/03/$17.00 ©2003 IEEE.

costs for examples of each class [15] and a significantly degraded performance when the class distribution in the training data is heavily skewed [7], [8]. There are two major groups of techniques designed to address class imbalance. The first group consists of supervised techniques that usually include three approaches [16]: (1) methods in which the minority population is kept intact, while the majority population is under-sampled, (2) methods in which the minority examples are over-sampled so that the desired class distribution is obtained in the training set, and (3) methods that use a recognition based instead of discrimination-based inductive scheme [6]. The second large class of techniques for detecting rare events involves an unsupervised framework, i.e. outlier detection. Initially, positive examples are completely ignored (if available) and a model is trained using all examples from the negative class. Then, the outliers are detected as the data points with low probability of occurrence, small number of neighboring examples etc [17], [18], where positive examples are typically used for threshold tuning.

III. Network Architectures

The network consists of the sensor nodes and the base station. Each sensor node is equipped with either a half duplex or a full-duplex wireless transceiver, which can transmit and receive data within a local area range (typically ranging from 10m to 200m). Each node has a unique hard-coded ID to identify itself to its neighbors and to the base station. The embedded computational device (processor) in the sensor node does the necessary signal and network processing. The base-station node is the information gathering point for the network and may also act as the interface between the wireless network and a wired network infrastructure, if available. The base station is assumed to have a large transmission range to cover the whole network and hence uses a single *broadcast* transmission to reach all the nodes in the network.

In this paper, two different network architectures are considered: a clustering based architecture called *Low-Energy Adaptive Clustering Hierarchy* (LEACH) [9] and a multi-hop tree-based architecture called MINA/UNPF [10]–[12].

In LEACH, sensor nodes which are close to each other group into a cluster. The nodes in the cluster send their data to a local *cluster-head*, which then takes the responsibility of sending these data to the base station. To extend the network system's lifetime, LEACH also utilizes randomized rotation of the clustering head nodes among all the nodes in the network to evenly distribute the energy load. A simple TDMA-based medium access control (MAC) protocol is used to coordinate transmissions within a cluster.

UNPF uses a layered (or tiered) multi-hop network architecture where the network nodes that have the same hop-count to the base station are grouped into a layer. Fig. 1 shows an example network consisting of a base station, 10 sensor nodes, and 3 mobile nodes. The number of layers and the number of nodes in each layer is determined by the geographical distribution of the nodes and the location of the BS. For instance, when 100-nodes are randomly placed in a

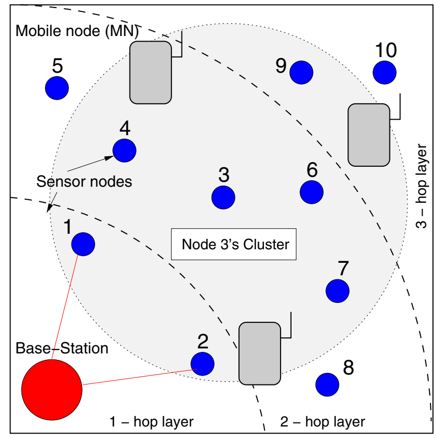

Fig. 1. Multi-hop infrastructure wireless sensor network architecture.

$250\ m \times 250\ m$ field, we have 4 to 5 layers if the BS is in the center of the field and 8 to 9 layers if the BS is in a corner. The Medium Access Control (MAC) protocol used is based on Time Division CDMA (TCDMA). Packets are routed from one layer to the next inward layer until the packet reaches the BS.

The LEACH architecture tends to provide lower delay since data travels in at most 2 hops from a sensor node to the BS. However, the cluster-heads may be far away from the BS resulting in longer-range transmissions leading to higher energy consumption. In UNPF, the delay and energy consumed per packet depends on the layer where the sensor node is situated. However, since shorter hops are used, the energy consumption tends to be lower, thus leading to long lifetimes. Detailed comparisons of these approaches can be found in [11].

IV. Learning Protocols

We assume that the data points (examples) $\mathbf{x}$ collected by the sensors are k-dimensional feature vectors from a set $X = \mathbf{R}$. As previously mentioned, data points are measured by each sensor at regular time instants. Each vector can be labeled at the base station as a member of one of the two classes, positive (interesting) and negative (not-interesting). We denote class label of the i-th example $\mathbf{x}_i$ by y_i, where $y_i \in Y = \{0, 1\}$. Here, we use a convention that the negative examples are labeled by zeros and positive examples by ones. In addition to the baseline protocol in which all sensors send all the data, we consider the following learning protocol. The system is initialized with no data points and sensors send all they collect until the first model from the base station is received. Once the base station collects a minimum number of positive examples, it trains a model and sends it to all the sensors. From that point, the sensors start sending only examples of the positive class together with randomly chosen negative examples in order to diversify training samples. As necessary, the base station retrains the classifier so that the sensors can update their models.

Assuming that cost of sending a message from a sensor to the base station is constant, the total cost of the baseline system can be expressed as $C_B = n_T c_T$, where n_T is the number of transmitted data points and c_T is the cost of sending one data point. To design a profitable machine-learning based system we need to express its total cost. Considering the bi-directional communication of this system and the fact that sending a false positive can trigger an unwanted action and that sometimes not sending false negative data can be damaging, the cost of our system is measured as

$$C_P = n_T c_T + n_{FP} c_{FP} + n_{FN} c_{FN} + n_M c_M,$$

where n_{FP} and n_{FN} are numbers of false positives and false negatives respectively; c_{FP} and c_{FN} are their corresponding costs per data point; n_M is the number of models (predictors) sent by the base station to the sensors and c_M is the cost incurred by such communication. The first goal of our work is characterizing the situations for which C_P is lower than C_B; the second goal is to increase the lifetime of a system as a result of the energy savings from reducing the number of transmitted data packets.

From the perspective of model learning, a typical task of a Bayesian optimal classifier in a cost-based model is to minimize the average cost of classifying a new, unlabeled data point randomly drawn from the same underlying distribution as the labeled data points. This cost C is given by

$$C = \sum_{i \in Y} \sum_{j \in Y} p(i|j) p(j) c(i, j)$$

where indices i and j denote the predicted and actual class of an unlabeled query example, $p(j)$ is the a priori probability of class j, and $p(i|j)$ is the conditional probability that predicted class is i given that the true class is j.

The penalties of classifying a query example into the class i when the actual class is j are represented by a 2×2 matrix with elements $c(i, j)$. Minimization of the average cost requires precise knowledge of the a priori class probabilities as well as of the penalties $c(i, j)$. The model is trained for a given penalty matrix, while the class probabilities are estimated in the base station according to the already transmitted data. The penalty matrix consists of four numbers: $c(0, 0)$ denotes the penalty for not sending a negative data point, $c(0, 1)$ the penalty for not sending a positive example, i.e. penalty for a false negative, $c(1, 0)$ the penalty for sending a negative example, i.e. penalty for a false positive, and $c(1, 1)$ the penalty for sending a true positive data point. We assume that $c(0, 0) = 0$ since no data is transmitted and $c(1, 1) = c_T$. Penalties $c(0, 1) = c_{FN}$ and $c(1, 0) = c_T + c_{FP}$ are varied and the range in which the system is profitable is reported in Section V. Cost c_M is ignored in the training process since the classifiers are constructed rarely relatively to the number of transmitted data points.

The training process is based on common approaches of over-sampling the smaller class and under-sampling the larger class [8], [15], [16]. Assuming that conditional probabilities $p(0|0) \approx p(1|1)$, the class ratio in each training set is based on the relative ratio of $c(0, 1)$ and $c(1, 0)$ and estimated a priori class distribution.

The base station trains a neural network classifier and sends its weights to the sensors as needed. We use a configuration with 3 hidden neurons and one output neuron, all with logsig activation function [19]. The training algorithm used by the base station is resilient propagation [20] with the maximum number of epochs set to 500.

V. Performance evaluation

We studied the relative performance of described architectures in terms of prediction costs and energy consumed. A discrete-event simulation model was developed for the network architecture. The network simulation parameters were: channel bandwidth of 1 Mbps; packet sizes of 32 bytes (sensor data) and 140 bytes (BS learning model); 100 nodes in a field of $250m \times 250m$ with the BS in the center and initial node energy of 2 J. Energy calculations are similar to those used in [12].

To create a dataset suitable for this problem we employed the *PRTools* data generators [21] and obtained examples using *gendatd* and *gendatb* routines. The dataset contained 1,000,000 5-dimensional examples. Furthermore, we used a skewed class distribution where the probability of a positive example was 0.02 and the probability of a negative example was 0.98.

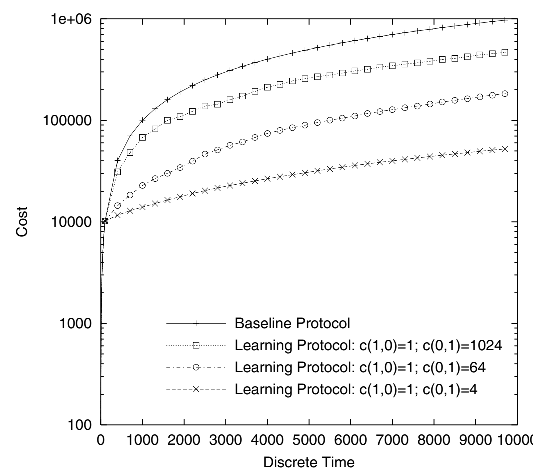

Fig. 2. Cost in time for the learning protocol for different penalties and baseline protocol.

$c(1,0)$	$c(0,1)$	n_{FP}	n_{FN}	C_P
1	1	1,236	8,872	33,091
1	4	1,931	6,435	53,590
1	16	2,952	4,326	101,234
1	64	7,008	2,320	190,616
1	256	17,413	1,127	352,651
1	1,024	47,901	354	488,384
1	4,096	46,653	580	2,498,764

TABLE I

Total costs of learning protocol for different penalties assuming baseline cost of 1,000,000.

A. Evaluation of Learning Protocols

To determine the range of the classification penalties for which the machine learning based system outperforms the baseline system, we simulated learning and networking protocols for the following values of classification penalties: $c(1,0) \in \{1,4,16,64\}$ and $c(0,1) \in \{1,4,16,64,256,1024,4096\}$. In Fig. 2, we show the total cost of the machine-learning based system as a function of time as compared to that of the baseline system. The increase of the false negative penalty beyond 1024 resulted in a non-profitable system. However, note that the range of parameters for which the system is profitable depends on the difficulty of the classification problem and noise in the dataset. In Table I we show simulation statistics for the fixed false positive cost $c(1,0) = c_T = 1$ and increasing false negative cost. The cost of the baseline protocol was in all cases $C_B = 1,000,000$ indicating that there is a significant gain in the first 6 cases. An interesting phenomenon occurs in the case when $c(0,1) = 4096$ for which the quality of the trained model (in terms of the false positives and false negatives) is essentially worse even though it was optimized for a specific penalty ratio. Since the class ratio in the training set is determined according to the estimated class ratio in the transmitted data and classification penalties $c(i,j)$ the class distribution in the training set becomes more and more skewed thus causing a decrease in prediction accuracy [8]. To allow for the further increase of the false negative penalty, our future work will concentrate on avoiding this effect through calculation of ROC curves (at the base station) from which the optimal class distribution for a particular learning task can be determined.

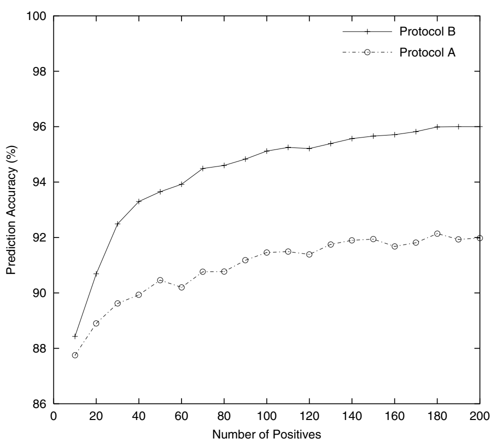

Fig. 3. Prediction accuracy for learning protocol and achievable accuracy by the baseline protocol, as functions of the number of received positive examples by BS.

In previous experiments with learning protocols all data points were transmitted until at least 200 positive examples were received. This was necessary since the base station needs not only to train an accurate system, but also to estimate the a priori class distribution. In Fig. 3 we compare the prediction accuracy of the system that builds the first model after only 10 received positives and subsequently retrains it every 10 positives (protocol A) to the system that uses all available data points for model construction (protocol B). Prediction accuracy was calculated as $(p(1|1) + p(0|0))/2$, while the cost parameters were set to $c(1,0) = 1$ and $c(0,1) = 64$, providing nearly balanced training sets. Therefore, due to a slow increase in performance accuracy, in situations where false negative penalty is high, a reasonable solution is to transmit all data points until a sufficiently accurate model can be constructed which in our experiments was achieved when about 200 positives were observed.

B. Energy Consumption Evaluation

Penalties		*UNPF (Joules)*	*LEACH (Joules)*
Baseline			
-	-	78.72	131.08
Learning			
$c(1,0)$	$c(0,1)$		
1	1	1.82	48.24
1	4	1.96	48.42
1	16	2.30	48.76
1	64	2.77	49.24
1	256	3.68	50.27
1	1,024	6.14	52.93
1	4,096	6.04	52.82

TABLE II

TOTAL ENERGY CONSUMPTION FOR LEARNING AND BASELINE SYSTEMS.

Table II lists the total energy consumption (in Joules) for various classification penalties for UNPF and LEACH network architectures. As the penalty of sending false negative increases, the training models try to achieve desired gain by increasing the number of false positives and thus minimizing the actual penalty incurred. This results in increased network traffic and thus energy consumption for both network architectures. The networking cost incurred for LEACH is seen to be substantially higher than that of UNPF. This is due to the higher energy spent in the cluster formation and cluster-head selection process (which involves several network-wide broadcast beacon packets) that is repeated periodically. In the simulations, the cluster formation phase of LEACH is carried out every 60 time units. In UNPF, each node can select a forwarding node in the next layer based on its neighbors' power levels thereby allowing more balanced per-node energy consumption in each layer. This saves considerable amount of energy and thus the majority of the energy consumed by UNPF is due to transmitting network traffic.

Fig. 4(a) shows the energy performance of UNPF for different misclassification costs. In the baseline protocol, the packet generation rate is high as the sensor nodes send all the data points to the BS in absence of any classification scheme. Hence energy consumption was high as compared to the learning protocol for UNPF. In machine-learning based protocols, the network traffic is remarkably reduced over a long period of time. This helps in conserving energy in the network and thus extending the network lifetime. With UNPF, the learning protocol achieved savings in energy ranging from 92% to 97% over the baseline protocol depending upon the mis-classification penalties. This indicates that the network lifetime can be extended by a similar amount. As seen in Fig. 4(b), LEACH also achieves energy reduction but the energy savings range from 60% to 63% due to higher network organization overhead.

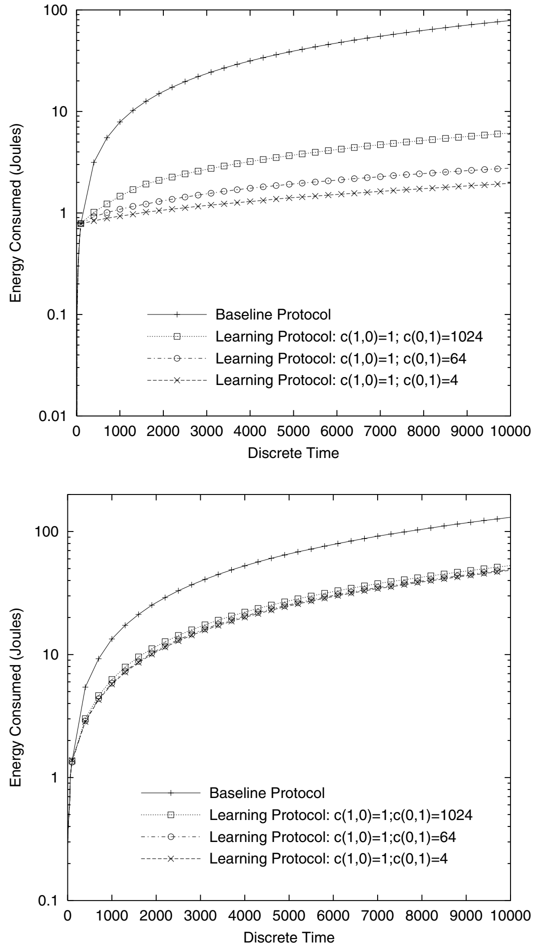

Fig. 4. Energy consumption analysis of UNPF and LEACH network architectures, varying penalty costs: (a) UNPF and (b) LEACH.

VI. Conclusions

We incorporated a machine-learning based cost-reduction technique into a wireless sensor network system designed for detection of rare events. Using a dataset with a highly skewed class ratio, we designed a learning/communication protocol and simulated network traffic with 100 sensors and a base station. We showed that significant decrease in overall cost and network traffic can be achieved using our system. The learning protocol was designed such that all sensors send all collected data points until a sufficiently accurate model can be constructed at the base station. The classifier is then broadcast to the sensors such that only data points of interest are further sent. The base station later retrains the model and sends it to the sensors as necessary. For a given difficulty of the classification problem, we characterized the range of false positive and false negative penalties (relative to the energy cost of one message) for which the system is profitable. From the networking perspective, we simulated two architectures, LEACH and MINA/UNPF and compared the relative reduction in energy consumption. The UNPF architecture was shown to have substantially lower energy consumption compared to LEACH since the latter architecture spent considerable energy in frequent re-clustering.

Acknowledgments

Part of the research was supported by Air Force Office of Scientific Research grants F-49620-97-1-0471 and F-49620-99-1-0125; NSF grants No. CCR-0209211, IIS-0196237 and ITR-0219736 and by Intel Corporation. The authors are pleased to acknowledge the efforts of Ms. Jin Ding, graduate student at Washington State University, for assistance with the network simulator.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, pp. 102–114, Aug. 2002.

[2] G. J. Pottie and W. J. Kaiser, "Wireless integrated network sensors," *Communications of the ACM*, vol. 43, no. 5, pp. 51–58, 2000.

[3] S. Lindsey, K. M. Sivalingam, and C. S. Raghavendra, "Data gathering algorithms in sensor networks using energy metrics," *IEEE Transactions on Parallel and Distributed Systems*, vol. 13, pp. 924–935, Sept. 2002.

[4] E. M. Knorr and R. T. Ng, "Algorithms for mining distance-based outliers in large data sets," in *Proc. of VLDB conference*, 1998.

[5] C. C. Aggarwal and P. Yu, "Outlier detection for high dimensional data," in *Proc. of ACM SIGMOD conference*, 2001.

[6] S. Vucetic, D. Pokrajac, H. Xie, and Z. Obradovic, "Detection of underrepresented biological sequences using class-conditional distribution models," in *Proc. Third SIAM Intl. Conf. on Data Mining*, (San Francisco, CA), May 2003.

[7] M. Kubat, R. Holte, and S. Matwin, "Machine learning for the detection of oil spills in satellite radar images," *Machine Learning*, vol. 30, pp. 195–215, 1998.

[8] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: Synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.

[9] W. Heinzelman, "Application-specific protocol architectures for wireless networks," *PhD Thesis, Massachusetts Institute of Technology*, June 2000.

[10] R. Kashyapa, "Medium access control and routing protocols for data gathering using wireless sensor networks: Design and analysis," Master's thesis, Washington State University, Pullman, Aug. 2001.

[11] J. Ding, "Design and analysis of an integrated MAC and routing protocol framework for large-scale multi-hop wireless sensor networks," Master's thesis, Washington State University, Pullman, Aug. 2002.

[12] J. Ding, K. M. Sivalingam, R. Kashyapa, and L. J. Chuan, "A multi-layered architecture and protocols for large-scale wireless sensor networks," in *Proc. IEEE Semiannual Vehicular Technology Conference – Fall*, (Orlando, FL), Oct. 2003.

[13] C.-C. Shen, C. Srisathapornphat, and C. Jaikaeo, "Sensor information networking architecture and applications," *IEEE Personal Communications*, vol. 8, pp. 52–59, Aug. 2001.

[14] A. Mainwaring, J. Polastre, R. Szewczyk, and D. Culler, "Wireless sensor networks for habitat monitoring," in *First ACM International Workshop on Wireless Sensor Networks and Applications*, (Atlanta, GA), Sept. 2002.

[15] P. Domingos, "MetaCost: a general method for making classifiers cost-sensitive," in *In Proc. of the Fifth International Conference on Knowledge Discovery and Data Mining*, (San Diego, CA), pp. 155–164, 1999.

[16] N. Japkowicz, "The class imbalance problem: significance and strategies," in *In Proc. of the International Conference on Artificial Intelligence: Special Track on Inductive Learning*, (Las Vegas, NV), 2000.

[17] V. Barnett and T. Lewis, *Outliers in statistical data*. John Wiley and Sons, 3 ed., 1994.

[18] E. Knorr, R. Ng, and V. Tucakov, "Distance-based outliers: algorithms and applications.," *The VLDB Journal*, vol. 8, pp. 237–253, 2000.

[19] S. Haykin, *Neural networks: a comprehensive foundation*. Upper Saddle River, NJ: Prentice Hall, 2 ed., 1999.

[20] M. Riedmiller and H. Braun, "A direct adaptive method for faster backpropogation learning: the rprop algorithm," in *In Proc. IEEE International Conference on Neural Networks*, pp. 586 – 591, 1993.

[21] "PRTools, a Matlab Toolbox for Pattern Recognition, version 3.0." http://www.ph.tn.tudelft.nl/prtools, 2002.