Correlation-based Feature Partitioning for Rare Event Detection in Wireless Sensor Networks*

Sitaram Asur and Srinivasan Parthasarathy†
Department of Computer Science, The Ohio State University

ABSTRACT
Rare event detection is an important problem for wireless sensor networks in surveillance or monitoring applications. Events such as intrusions are typically rare and need to be handled efficiently. Traditionally, sensor nodes report their observations to a centralized unit for processing. However, global centralized classification models are known to suffer in the presence of imbalanced data. They also consume a large amount of energy due to transmissions. Since sensor nodes are limited in battery power, it is important to reduce communication costs by distributing the processing among the sensor nodes.

In this work, we propose a correlation-based scheme to partition the features observed by the sensor nodes into disjoint mutually uncorrelated feature subsets. An ensemble of local classifiers can then be trained on these subsets. We implement our model on a cluster-based sensor network architecture (LEACH). We also provide an energy efficient routing scheme for the above model. Our experimental results on real and synthetic data show that the proposed technique provides benefits both in terms of accuracy of detection and energy savings of the network.

1. INTRODUCTION
Wireless sensor networks are becoming ubiquitous in their use in security, defense, monitoring and tracking applications. They are frequently deployed to perform tasks such as detection, tracking and classification of events or targets within the sensor field. Sensor nodes perform acquisition of multi-modal data from their surroundings. This sensed data can either be processed locally in the network in a distributed fashion or can be transmitted to a base station for processing in a centralized manner [3, 4]. In the distributed approach [5, 9, 25], sensors perform data aggregation and processing locally and send the processed data to the base station, thereby limiting the amount of communication. The main motivation for distributed processing is the limited battery power of sensors and the high energy costs associated with communication of data to the base stations.

Given the nature of the problem there is a clear need for energy conserving, distributed data preprocessing and analysis algorithms in order for sensor networks to last longer. Various routing schemes have been suggested to provide energy-efficient communication to facilitate such algorithms. In this paper, we consider LEACH [13, 14], a cluster-based routing architecture, which has been developed for energy-efficient routing in sensor networks. The basic idea is as follows. The sensor network is divided into clusters of sensor nodes. In each cluster, nodes report their observations to a node designated as the cluster-head and it is the task of the cluster-head to assimilate the information from all the nodes in the cluster and communicate with the base-station. This routing scheme has been shown to yield energy savings due to the fact that individual nodes do not need to communicate directly with the base station. However, every node has to transmit its data to its cluster-head. The nodes in a cluster merely transmit their information and it is up to the cluster-head to perform any processing that is required and also transmit data periodically to the base-station.

In this work, we focus on the task of rare event detection in sensor networks. Of particular interest are strategies that can reduce the energy consumption significantly with little or no loss in accuracy. Intrusion detection is an important rare event detection problem in sensor networks. Most intrusions are typically rare and need to be identified efficiently when they occur. Apart from the need to predict accurately, the number of false alarms also need to be reduced, since mis-predictions can translate to unnecessary communication and energy costs in resource-constrained systems like sensor networks. For this, suitable energy-efficient models are required to increase the precision and recall of intrusion detections.

The difficulty of detecting rare events arises from the fact that the training data in such applications is imbalanced [6, 16, 17, 18, 22]. A dataset is imbalanced if the frequency of each class in it is not approximately equal. Since, intrusions are rare, most of the data samples observed by sensor nodes are negative. This makes it hard to train a model to detect positive intrusions. Previous approaches have dealt almost

*This work is supported in part by NSF grants #CAREER-IIS-0347662, #RI-CNS-0403342, and #NGS-CNS-0406386
†Contact Email : srini@cse.ohio-state.edu

entirely with sampling-based schemes to balance training data at a centralized location.

In this work, we propose a scheme that is based on partitioning the attributes sensed by the sensor nodes into subsets, each of which are used to build local classification models. We propose the use of a correlation-based method to partition the attributes effectively into disjoint subsets, each containing mutually uncorrelated attributes. These local models can be constructed in a few 'classifier' nodes in each cluster. These classifier nodes together form an ensemble classifier, each with a model based on different attributes. Earlier approaches on handling imbalanced data [2, 8, 7, 11] have suggested that using ensemble learning can help improve the precision and recall of predictions. Since each of these classifiers are independently trained, their mis-predictions will be independent. Accordingly, the local decisions made by the nodes in each cluster can be merged together at the corresponding cluster-heads. Another important advantage with using an ensemble of classifier nodes is that it improves the fault tolerance of the sensor network. Since sensor nodes are fragile and prone to failure, the presence of multiple classifier nodes ensures that node failures do not affect the overall event detection process. This also ensures robustness against noisy or missing data.

We provide a routing scheme based on the above model that is more energy efficient than the baseline LEACH model. We show with our experiments on real and synthetic datasets that our scheme improves both the classification performance as well as the energy savings of the network. We also show how the proposed technique is robust to node failure and missing data.

To summarize, the key points of our paper include:

- A novel vertical partitioning of the attributes sensed by clusters of sensor nodes that are fed to local classifiers.
- The use of an ensemble of localized classification models to handle imbalanced data and to detect rare events resulting in significant accuracy gains
- An energy-efficient routing algorithm for the above classification model that results in twice the savings in energy.

2. BACKGROUND AND RELATED WORK

In this section, we will provide details of the LEACH routing algorithm and discuss related work on existing partitioning schemes and rare class prediction from the data mining and machine learning perspective.

2.1 LEACH

LEACH (Low-Energy Adaptive Clustering Hierarchy) [13, 14] was introduced in 2000 as an energy-efficient communication protocol. LEACH works by constructing clusters of sensor nodes in the network. Clusters are created by having nodes select themselves as cluster-heads randomly. The cluster-heads transmit messages to all the other nodes in the network. Nodes choose the cluster-head that is closest to them and join the corresponding cluster. Once the clusters have been setup, the cluster-head sets up a schedule for each node to transmit. Nodes transmit their readings to their respective cluster-head which performs aggregation and transmits to the base station. Thus, individual nodes need not communicate with the base station directly. To extend the lifetime of the sensor network, the cluster-heads are re-elected and the process of associating a node with a cluster head is repeated.

Since it is widely known that communication costs constitute a very large fraction of the energy losses of sensor nodes, the authors provide an energy model, consisting of energy for transmission and energy for reception.

The energy for transmission in a LEACH network is given by:

$$Energy - trans(k,d) = E_{elec} * k + \epsilon_{amp} * k * d^2 \qquad (1)$$

where E_{elec} represents the energy to run the circuitry of the radio and ϵ_{amp} denotes the energy required to transmit k bits over a distance of d.

The energy for reception in a LEACH network is given by:

$$Energy - rec(k) = E_{elec} * k \qquad (2)$$

Thus, reception is also an expensive operation for sensor nodes.

The Total Energy per round of transmission in a network with n nodes :

$$Energy = \sum_{i=1}^{n} Energy - trans_i + \sum_{i=1}^{n} Energy - rec_i \qquad (3)$$

The authors use $E_{elec} = 50$ nJ/bit and ϵ_{amp}=$100 pJ/bit/m^2$. They compare LEACH with two approaches - direct communication and Minimum Transmission Energy(MTE) routing. Their experiments showed that LEACH provides up to 8 times the communication energy gains of the two approaches. LEACH also doubled the system lifetime compared with the other approaches.

2.2 Related Work

Distributed classification in sensor networks has been studied by several earlier works. Wang et al [9] propose a fault-tolerant decision fusion algorithm for distrubuted multiclass classification. Zhang and Varshney [25] also discuss distributed classification using hieararchical decision fusion. Kargupta [5] proposed Collective Data Mining, a framework for creating and aggregating local data models with minimal communication among sensor nodes. Meesookho and colleagues [4] describe collaborative approaches for data fusion for target classification in sensor networks.

McConnell et al [21] discuss distributed prediction in sensor networks by building local models at each sensor based on its input data. This approach does not partition attributes globally. Instead, each sensor builds a model on its own set. The main disadvantage with this approach is that observations made at neighboring sensor nodes are not considered for prediction. Each sensor's readings is assumed to be independent. This is not likely to be true, since neighboring sensors typically measure the same event. Also, each node learns a local model and transmits its decision. This represents an extreme case of distribution with each node having to construct a local model and perform classifications, placing enormous burdens for computation on the sensor nodes. This approach is likely to result in poorly fit local models which will not be able to handle imbalanced data. In our case, we build local models for groups of uncorrelated attributes. In addition, we ensure that observations from other sensors are considered while training these models. Using these partitions, we are also able to reduce the transmissions required for efficient classification.

The problem of learning from imbalanced data has been studied extensively in the data mining and machine learning literature [8, 7, 11, 2]. Japkowicz [16] claims that there are three main ways to balance training data - oversample the positive samples, under-sample the negative samples or implement a recognition-based induction scheme. Joshi et al [18] proposed PNRule, a rule-based classifier designed to handle skewed class distributions. PNRule works by discovering positive rules that cover the target class and negative rules on the negative class. A test sample is classified positive only if it is found to satisfy a positive rule and no negative rules. Other approaches such as SMOTE [6], SMOTE-Boost [7], DataBoost [10] and DataBoost-IM [11] perform data generation along with ensemble boosting to improve the accuracy of classification algorithms. Batista et al [2] performed a study of several balancing methods and found that oversampling methods provide more accurate results than undersampling methods.

A sampling based approach to balance data in sensor networks has been studied by Radivojac [22]. This is the only work to discuss classification with imbalanced data in sensor networks, to the best of our knowledge. They propose an approach where the base station trains a classification model and sends it to each sensor. The sensors classify samples based on the model and transmit all the positive samples and selected negative samples to the base station. The base station uses these samples to construct new models which are transmitted back to the sensor nodes. This technique uses a sampling-based scheme to handle imbalanced data. However, this approach does require a lot of communication between nodes and the base-station. Even employing a routing protocol like LEACH within the substrate of this approach is bound to incur great costs. Instead, in our approach, the classifier nodes themselves train and re-train models based on a subset of the attributes. This reduces communication with the base station. Also, as we demonstrate in our experiments on real data, the local models based on uncorrelated attributes are sufficient to handle imbalanced data.

Another problem with their approach is that they construct a global model at the base station, albeit with balanced data. This is bound to be problematic since a one-model-fits-all approach will not work well in the context. Since they produce a single model which is transmitted to all the nodes, the model is not going to be as accurate as local models. In their evaluations, they focus more on communication cost rather than accuracy. We show that our approach provide good accuracy along with good energy savings.

3. RARE EVENT DETECTION

Before we describe the details of our approach we would like to describe a key assumption in this work. We assume that each sensor node in a cluster observes the same m-dimensional feature vectors from a set $X = \mathbb{R}$. We would like to note that this is a fairly common assumption in the literature [22, 23], since nodes in the same cluster are typically close and hence, typically, do observe the same set of attributes, although their values may differ. Nodes across clusters may be heterogeneous in nature, i.e., there are no restrictions for attributes sensed by nodes in different clusters in our work.

In this work, we concern ourselves with detecting rare but important events occurring within a cluster. We formally define this as follows.

Definition 1: An (α, β)-event is an event that is observed by α nodes in a cluster and occurs with frequency β in that cluster.

Definition 2: An (α, β)-event is a rare but important event if $\alpha \geq min\_quorum$ and $\beta < max\_support$.

Here, min_quorum represents the minimum number of nodes that need to observe an event for it to be important and $max\_support$ indicates the frequency threshold for an event to be rare. Since the nodes in a cluster are typically close to each other, an event is deemed important only if it is detected by a certain percentage of sensor nodes represented by min_quorum.

Since our intended application is intrusion detection, we consider only binary classification in this paper. Throughout this paper the positive class refers to detected intrusions and the negative class refers to situations where there is no intrusion.

In the next two subsections, we describe our partitioning scheme and energy-efficient routing technique respectively.

3.1 Correlation-Based Partitioning

Our technique to detect rare events involves constructing local models based on disjoint subsets of attributes sensed by the nodes in each cluster. Previous research on the importance of feature selection for classification performance [12, 19] have hypothesized that a good feature subset should contain attributes that are uncorrelated with each other and correlated with the class. This is ideal since it removes redundancy in the classifier, enabling the discovery of more compact models, while also improving the predictive capability. Hence, if a given feature's predictive ability is covered by another then it can safely be removed [12]. Hogarth [15] notes that, when adding features to a subset, low inter-correlation with the already selected features may well predominate high correlation with the class, as an important criterion.

In our work, instead of feature selection, we use the correlation between attributes to distribute them into partitions such that each subset contains mutually uncorrelated attributes. The idea is to build an ensemble classifier from each of the local classifiers with the intent to improve accuracy.

The correlation between two attributes can be calculated as:

$$Corr(i,j) = \frac{Cov(i,j)}{\sqrt{Var(i) * Var(j)}} \tag{4}$$

where $Cov(i,j)$ is the Covariance between i and j and $Var(i)$ is the Variance of i. We are interested in attributes with low correlation with each other. Hence, we define the distance between the attributes in terms of their correlation as:

$$Dist(i,j) = 1 - \|Corr(i,j)\| \tag{5}$$

We are interested in obtaining subsets in which the attributes have high distance values from each other. Hence, we use Hierarchical clustering to group the attributes that are distant from each other into subsets. This operation may need to be performed periodically, since the data distribution in a sensor network is likely to change over time. However, since the number of attributes is generally not too high, performing these computations periodically will not be expensive. In this work, we confine ourselves to Hierarchical clustering, although in general, any clustering method can be employed.

To find the optimal number of clusters, we try different values of k and pick the value that provides the most balance in the partitions. We would like our partitions to be reasonably balanced so that there is no skew in the prediction of local models i.e. the number of attributes do not differ significantly in the partitions. Since we are using an ensemble of these local models and we are treating each model the same, we would like each of them to contain a suitable number of attributes to yield good accuracy. Hence, we ensure that each partition contain more than one attribute.

Our goal is to train a classifier on each of these partitions separately. The classifiers constructed on the partitions together form an ensemble of local classifiers, each housed in a sensor node. Ensemble approaches to learning with imbalanced data have been suggested in several works [8, 7, 11, 2]. In our work, we assume that the number of local classifiers will be fewer than the number of nodes – as a result a subset of the nodes in each cluster will need to be selected to be 'classifier' nodes and form the ensemble.

Each classifier will then build a local model for the rare class based on the values of a subset of the attributes. For each round of observations made by the sensor nodes in a cluster, the classifier nodes will detect if a rare event has occurred. In the context of intrusion detection, a positive prediction would suggest a possible intrusion. Each classifier node will confirm an event to be positive only if at least min_quorum nodes report intrusions. After each classifier node makes a decision, the cluster-head will perform decision fusion[1] to obtain the joint decision of the cluster. Once the observations made by all the nodes in the cluster are classified, the cluster-head will report to the base station only if the resultant prediction is positive.

To demonstrate the effectiveness of this technique in handling imbalanced data, we present some results on real imbalanced datasets obtained from the UCI Machine Language Repository.

[1]Essentially a majority vote in this work but can be extended to other options in the future.

Dataset	Attributes	Skew (Min:Maj)
Satimage	36	0.09:0.91
Breast-Wisconsin	9	0.34:0.66
Glass	9	0.13:0.87
Yeast	8	0.04:0.96
Phoneme	5	0.29:0.71

Table 1: Real Imbalanced Datasets

Table 1 provides information regarding the imbalanced datasets that we consider, the number of attributes they contain and the corresponding skew in the class values. Some of these datasets contain more than 2 classes. We pick the class with a small number of samples to be the positive class. This is consistent to experiments on these datasets in earlier works [11, 7, 2]

To perform an efficient comparison, we obtain the corresponding performance values of two algorithms proposed to handle imbalanced data - SMOTEBoost [7] and DataBoost [10]. One thing to note in this context is that both SMOTEBoost and DataBoost generate new synthetic positive samples. They use boosting, an ensemble learning technique to improve performance. The resulting models are much more complex than our local models, involving a lot more computation and communication, which is unlikely to be practical in a sensor network setting. The reason we choose these algorithms to compare with, rather than a classifier such as PNRule, is that we wish to test our partitioning technique irrespective of the base classifier. SMOTEBoost and DataBoost are techniques to balance the training data, on which any simple classifier can be learnt. We believe that our partitioning technique will improve the performance of any classifier on imbalanced data. In our implementation, we use two simple classifiers - J48 decision trees and 5-nearest neighbor, from the Weka [24] suite of algorithms.

To express the accuracy of predicting the rare class, we use the F-measure value of the minority class. The F-measure value is a function of the Precision and Recall of predictions and it gives a good indication of the accuracy of predicting the rare class.

$$Precision = \frac{TruePositives}{TruePositives + FalsePositives} \tag{6}$$

$$Recall = \frac{TruePositives}{TruePositives + FalseNegatives} \tag{7}$$

$$F - measure = \frac{(1+\beta^2) * Precision * Recall}{\beta * (Precision + Recall)} \tag{8}$$

The value of β is generally set to 1.

We present the comparative analysis of our technique Correlation-based Partitioning (CBP) with the two other approaches in Table 2. The centralized C4.5 F-measure value is provided as the baseline case.

We find from the results that our method is competitive with other existing approaches for rare class prediction. In 3 out of the 5 cases, we obtained a better F-measure value for the rare class than SMOTEBoost and DataBoost. The

Dataset	C4.5	CBP		SMOTE	Data
		J48	5NN	Boost	Boost
Satimage	56.4	60.1	63.4	64.5	**66.3**
Breast-Wisconsin	92.4	93	**94.6**	94.1	93.7
Glass	78.5	**87.1**	81.6	84	86.2
Yeast	9	0	**79.1**	57.6	54.1
Phoneme	77.2	68.6	70.3	77.37	**81.8**

Table 2: Comparative Results on Imbalanced Datasets

only relatively poor result for our method is for the Phoneme dataset. This can be explained by the fact that the Phoneme dataset contains only 5 attributes, so partitioning it might not be effective. These results show that our method, although simple, is extremely effective in classifying imbalanced data.

3.2 Energy-efficient Routing

We now describe our routing model designed to use the partitioning scheme described in the previous subsection. Our goal is to reduce the energy consumption of the network and increase the lifetime of the sensor nodes.

3.2.1 Node Clustering:

As we mentioned earlier, LEACH begins by grouping the nodes in the network, based on the distance between them, into clusters. A cluster-head is chosen and the nodes in each cluster communicate with the cluster-head. In the original LEACH architecture, the cluster-head is chosen randomly and the role is periodically re-distributed among the nodes in a cluster. This is in accordance with the notion of re-distributing the energy required to perform the duties of the cluster-head. These duties include constant reception of observations from sensors, aggregation and periodic communication with the base station. In our work, we introduce an intermediate level between the sensor nodes and the cluster-head - the classifier nodes.

As a result, the cluster-head, in our case, is more limited in responsibilities. The only role of the cluster-head is to combine the decisions of the classifier nodes. Instead of reception of observations from all nodes in the cluster (which can be quite a few), the cluster-head only needs to receive a binary decision from the k ($k << 20$) classifier nodes. The only aggregation to be performed is the majority vote. Also, the cluster-head needs to communicate with the base station only if there is an intrusion, which is a very rare event. Due to this reduced energy consumption, our observation is that in our work we do not need to re-distribute the role of the cluster-head as frequently. Hence, the overhead involved in computing new clusters in traditional LEACH can be significantly reduced. Also, another issue with the LEACH model of choosing cluster-heads is that the process is random. Any node that has sufficient energy can be chosen. Hence, the cluster-head might well end up being the outlier in a cluster, which makes transmission expensive, since all nodes will have to transmit over the length of the cluster.

3.2.2 Classifier Node Selection:

Once we obtain the partitioning of attributes, we need to select the nodes needed to perform the classification. The number of nodes required cannot be more than the number of attributes sensed in the cluster. Typically, the number should be as small as possible to provide greatest benefit. A prerequisite for selecting the classifier nodes is that they must be easily reachable from all other nodes in the cluster. From the LEACH energy equations(1,2,3), we know that the energy spent is directly proportional to the square of the distance of communication. Hence, it is important that the classifier nodes are well-situated in the cluster. They must also have sufficient energy to perform the classification and related communication. For each cluster, we find all nodes that have energy greater than a threshold. These are the potential classifier nodes. Next, for each of these nodes, we compute the sum of the distances from all other nodes. We choose the k nodes that have the least sum of distances to build the k classifiers. This is done independently in each cluster, since the number of partitions is dependent on the distribution of the data, which may not be the same for different clusters. Redistribution is done periodically once the classifier nodes are below a energy threshold but it will need not be done as frequently as in the LEACH architecture.

The pseudo code is provided below.

Algorithm 1 Classifier Node Selection(k)

```
for each cluster c in the network do
  for each potential classifier node i in cluster c do
    for each node j in cluster c do
      dist(i)=dist(i)+distance(i,j)
    end for
  end for
  Classifier-nodes(c)= k nodes with minimum distance
end for
```

Once the classifier nodes are chosen and they build local models, classification can begin. The local models can be periodically re-learnt based on the data.

3.2.3 Routing Scheme

Each sensor node in a cluster reads an m-dimensional vector. It partitions the attributes based on the vertical partitions set up. It transmits only the relevant attributes of its observations to each of the k classifier nodes that have been set up. The classifier nodes obtain the values for the attributes they model, from all the other nodes in the cluster. They then proceed to use their local model to classify each of the subsamples.

Once they have classified the readings of all the nodes, they can then decide if a rare event has occurred using the min_quorum value for that specific cluster. They transmit only their decisions to the cluster-head. The cluster-head performs the merging and reports to the base station only if a rare event has been detected. The cluster-head communicates the result of the decision only to classifier nodes that provided incorrect predictions for that event. The classifier nodes can refine their models using this feedback. This can be done by weighting the samples that were incorrectly predicted more than others. Depending on the computational capacity of the sensor nodes, more complex schemes can be used.

The roles of individual nodes, classifier nodes and cluster-

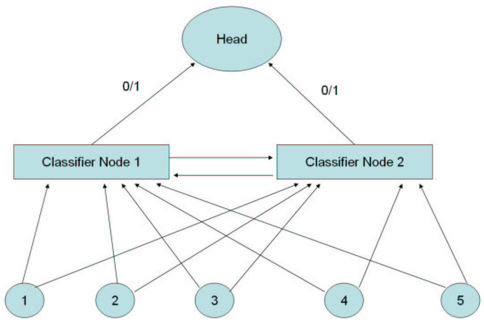

Figure 1: Each node transmits subsets of its readings to each classifier node. The classifier nodes in turn transmit their decisions to the cluster-head

heads are illustrated in Figure 1. The individual sensor nodes transmit subsets of their attributes to the classifier nodes, which in turn transmit their decisions to the cluster-heads.

Since, each node transmits only a partial set of attribute values, the amount of data transmitted is reduced. The classifier nodes effectively perform aggregation over the readings they obtain and transmit only their decisions. Since the classifier nodes are chosen such that they are close to the center of the cluster, the distance each message has to travel is not significant, when compared to the original LEACH model. There is no need for any other node to directly communicate with the cluster-head. In some senses, we are providing an extra level of hierarchy while also distributing processing. If there are n nodes in a cluster and k classifier nodes, for every n readings made by the nodes, the cluster-head needs to receive only one decision message from the k classifier nodes. Typically, since $k << n$, the number of messages sent out to the cluster-head will be low.

4. EXPERIMENTS ON SENSOR DATA

In this section, we present experimental results for our classification and routing schemes on sensor data. Since we could not obtain a real sensor dataset with node location information and suitable number of attributes, we are currently unable to evaluate both our classification and routing schemes on the same dataset. Instead, we divide our evaluation into two parts. In the first part, we evaluate our energy-efficient CBP routing scheme on a real as well as a synthetic sensor network. In the second part, we evaluate our CBP Classification Scheme on rare event detection on a sensor dataset.

4.1 Energy Experiments

4.1.1 Intel Lab Data

The Intel Lab Data represents data collected from 54 sensors deployed in the Intel Berkeley Research lab between February 28th and April 5th, 2004. The sensor nodes were arranged as shown in Figure 2.

The Intel Lab data does not correspond to a rare event detection dataset. Hence, we use only the location information of the sensors to test the energy savings of our routing scheme against the conventional LEACH scheme. The data that they monitor consists of only 4 attributes. For our use, we consider a case where the sensor nodes are observing 10 attributes. The reason we choose 10 is that it can afford different partitioning combinations. We used the NS2 network simulator [20] along with the LEACH implementation [1] to implement the network and compute clusters.

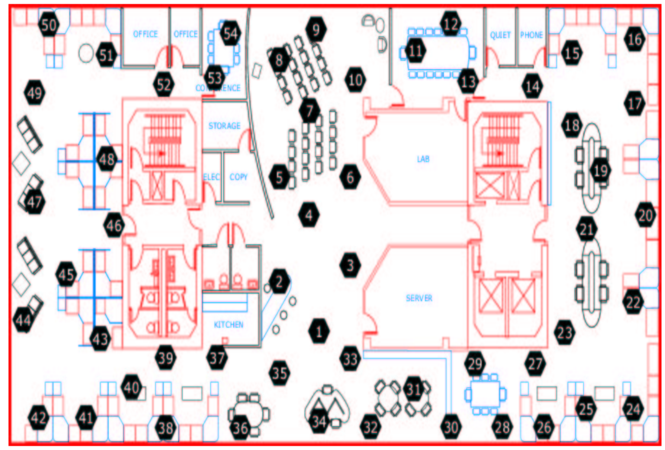

Figure 2: Intel Lab Sensor Nodes

We consider two schemes apart from the baseline LEACH routing scheme. The first one is the scheme we outlined in detail in the previous section. An additional optimization that can be done to reduce the amount of transmission involves each node transmitting only to k-x classifier nodes instead of k. This does not affect the classification too much, since each classifier node bases its predictions on the readings obtained from all the sensor nodes. Even if it does not hear from a few, it will still have sufficient information to predict accurately. Also, an additional level of security is provided by the majority vote in the ensemble classification. However, this reduction in transmission makes a big difference to the energy conservation of the network, as we show. In our experiments, we use two values for x, 1 and 2.

We used the LEACH energy equations to compute the energy consumed in all three cases. We assume the initial energy of each node in the network is uniform(2 Joules). We use the default LEACH parameters. Since the approaches do not differ in the cluster setup and communication with the base station, we consider only the energy consumed by communications within the clusters. We vary the number of clusters between 1 and 4 and the partitioning of attributes over the possible 4 combinations - (2 2 2 2 2), (3 3 2 2), (3 3 4) and (5 5).

The results are provided in Figures 3 and 4. In each case, we perform 10 trials and average the values. CBP-all is the conventional Correlation-based Partitioning using all the readings to predict. CBP-Opt(1) and CBP-Opt(2) represent the k-1 and k-2 Optimized schemes respectively. We observe

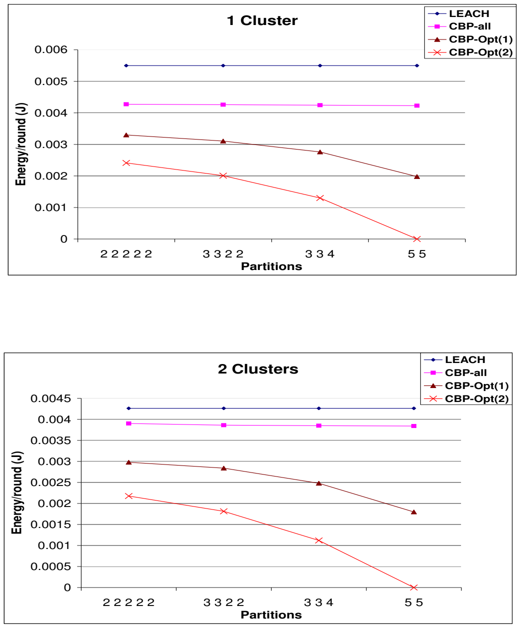

Figure 3: Energy Results for Intel Data with a) 1 cluster and b) 2 clusters

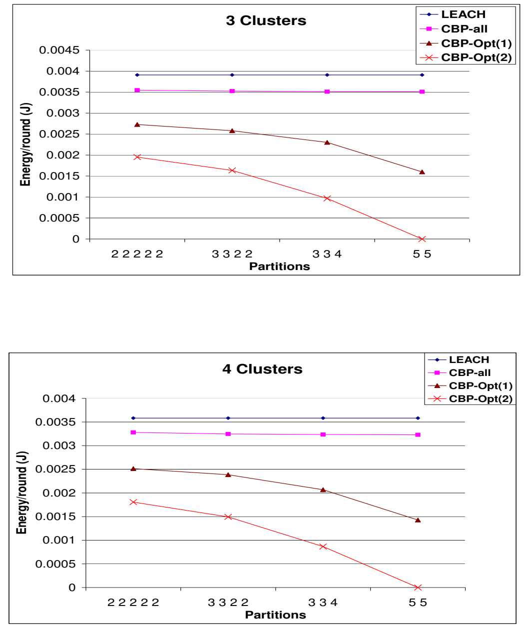

Figure 4: Energy Results for Intel Data with a) 3 clusters and b) 4 clusters

from our results that our routing schemes perform consistently better than the LEACH scheme in all cases. The optimized k-2 scheme does the best in all cases. It is important to note that we used x as 1 and 2 in the optimized schemes for this experiment. Further savings can be obtained using higher values for x, as the number of transmissions will reduce. The k-1 optimized scheme provides energy savings of more than twice that of standard LEACH in the best case, which is when the partitions are 2. The k-2 scheme reduces the energy even further, up to 3 times as much as LEACH, in the best case. Note that the CBP-Opt(2) values are 0 for (5 5) since there are only two classifier nodes(k=2) for this case.

In terms of partitioning, smaller number of partitions perform better, which can be expected, since the required communication will be less. Also, when the number of partitions is large, most of the nodes in a cluster will function as classifier nodes, which will degrade performance. A centralized scheme will be more useful in such situations.

In terms of clustering, when the number of clusters increase, we notice that the gap between LEACH and the other two schemes reduce. This can be explained by the fact that the Intel dataset contains only 54 nodes. When the number of clusters increases, there will be fewer nodes in a cluster. Most of the nodes in each cluster will function as classifier nodes. This will cause increased communication and energy consumption. In a typical sensor network, cluster sizes range from 20-50 [14], which is equivalent to our 1-2 cluster case, for which our schemes will provide good efficiency.

4.1.2 *Synthetic Data*

To get a good idea of our performance, we evaluated our system over a larger random network. The 100 node random network is shown in Figure 5. We used the same parameters as in the original LEACH paper [13]. The authors claim that the optimal number of clusters for this setup is $1 < cl_{opt} < 6$. Hence we varied the value of cl from 2 to 5 in this experiment. We evaluated the same three approaches as in the previous experiment. The results we obtained were similar to the earlier case. The resulting graph is shown in Figure 6. Once again we use k-1 and k-2 for the optimized technique. We find that our proposed methods perform better than the baseline method, with the CBP-Opt schemes using up less than half(in the case of CBP-Opt(1)) and 1/3(in the case of CBP-Opt(2)) the energy of LEACH, when the partitions are less. In terms of partitions, the same pattern is seen, with reduced energy costs when the number of partitions reduce.

From both these experiments, we find that our approach performs consistently better than the centralized approach in terms of energy consumption. It is important to note that the above experiments presented the energy consumption per round of transmission. In a typical network intrusion detection scenario, there can be thousands to millions of samples to be transmitted and processed by sensors with small batteries. In this context, our optimizations can provide huge energy savings and increase the lifetime of the network.

4.2 Experiments on Rare Event Detection

To test the performance of our CBP scheme on rare event detection on sensor data, we used a dataset obtained from the Physiological Modeling Contest[2] held at the International Conference on Machine Learning, 2004. It contains data obtained over a few months period, from 32 people who wore BodyMedia body sensors. Each set of body sensors recorded 9 attributes such as body temperature, heat flux, skin response, skin temperature etc. The goal was to predict the

[2] http://www.cs.utexas.edu/users/sherstov/pdmc/

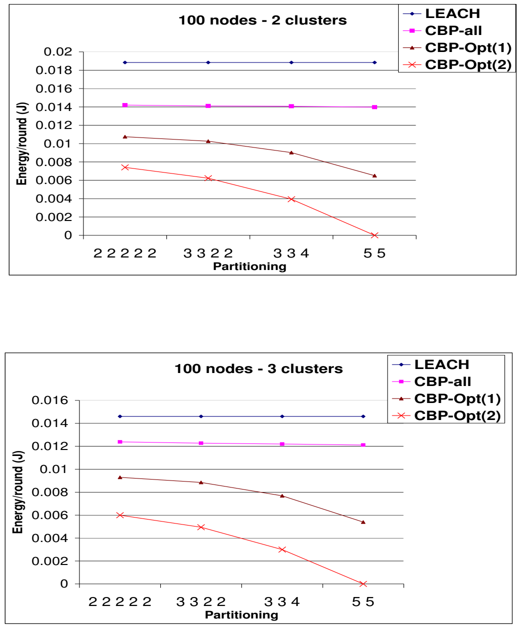

Figure 5: Energy Results for 100 Random Nodes with a) 2 clusters and b) 3 clusters

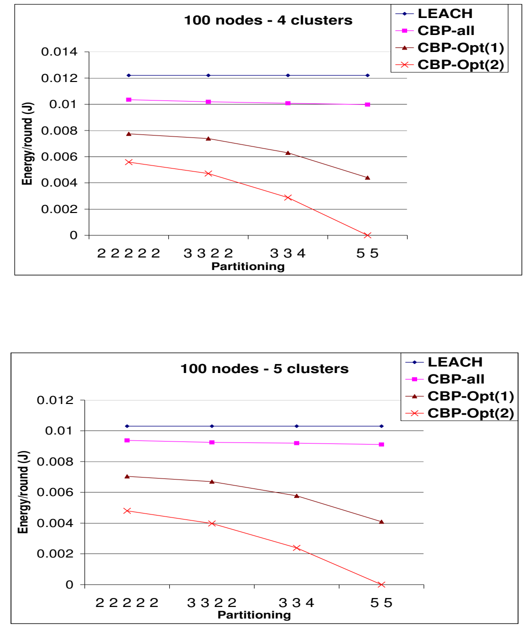

Figure 6: Energy Results for 100 Random Nodes with a) 4 clusters and b) 5 clusters

activity of the person based on the readings. The original dataset consisted of 720790 samples. We chose to use 11416 samples, which represented 400 timestamps for 32 people (Some people did not have readings at some timestamps). We considered each person as a node sensing 9 attributes. We chose one activity - sleeping as the positive class and the rest of the samples were considered negative. There were 1083 positive samples and 10333 negative samples resulting in a skew in the data of around 0.10:0.90. Since we had only 32 people, we considered them to be part of one cluster. We sorted the entire dataset by timestamp to simulate a sensor network.

Note, that there was no location information provided in the dataset. Hence, we use this dataset only to evaluate our rare event detection scheme.

4.2.1 Prediction of the Minority Class

Our first experiment was to evaluate the performance of our approach in identifying the minority class in the imbalanced sensor dataset. The baseline case we used was centralized classification(CENT). We used the J48 decision tree as our classifier. We also considered the k-1 Optimization case in this experiment. We implement this by randomly using k-1 classifiers to classify each sample. This is equivalent to the case where k-1 subsets are sent out by each node. To provide a good comparison, we also implemented the completely distributed approach suggested by McConnell et al [21] where a separate model is trained for each attribute and the decisions of each of them are merged using majority vote decision fusion. We call this approach DIST. The classification results are summarized in Figure 7. We once again use the F-measure of the minority class to evaluate the approaches. From the results, we find that both our approaches perform better than the baseline centralized classification. Our non-optimized algorithm performs the best overall. The optimized algorithm, despite the approximation involved, fares better than the centralized version. For DIST, we found that using majority vote(ie $\geq 5/9$), we got no positive predictions. When we relaxed the majority constraint, we got some positive predictions, which we have illustrated in the figure. However, these values are still significantly smaller than the centralized version. This indicates that this approach is not very suitable for imbalanced data classification.

4.2.2 Rare Event Detection

We have seen that our approach can handle imbalanced data effectively. Our next experiment is to test it in a rare event detection scenario. We use the same activity as in the previous experiment. To find a rare event, we choose a $max\_support$ value of 30%. We examined the frequencies of the positive class over different values of min_quorum and picked the one that produced a skew. Accordingly, we chose the value of min_quorum as 4/26(15%) which gave a support of the event as 112/400 or 0.28. This represents a (15%,28%) event, which is rare and important according to our definition, since at least 15%(min_quorum) nodes observe it and the support is less than the $max\_support$(30%). We used the same approaches for classification as for the previous experiment. CENT is the centralized baseline scheme and DIST is the completely distributed approach. We also implemented the k-2 optimization for this experiment. We report the F-measure along with the accuracies, both overall and for the rare event(min). The experimental results are provided in Figure 8. We can observe from the figure that our approaches perform the best with CBP having the highest F-Measure(80%) and Accuracy values. The Optimized approaches performs better than the Centralized and Distributed schemes and surprisingly CBP-Opt(1) has the best accuracy for predicting the minority class. Its overall accuracy is, low possibly because of a high number of

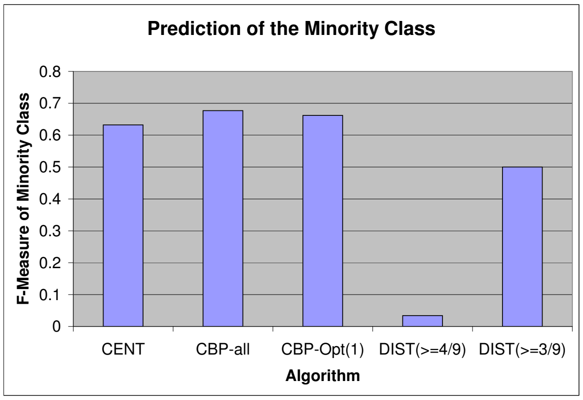

Figure 7: Body Sensor Data - Prediction of the Minority Class

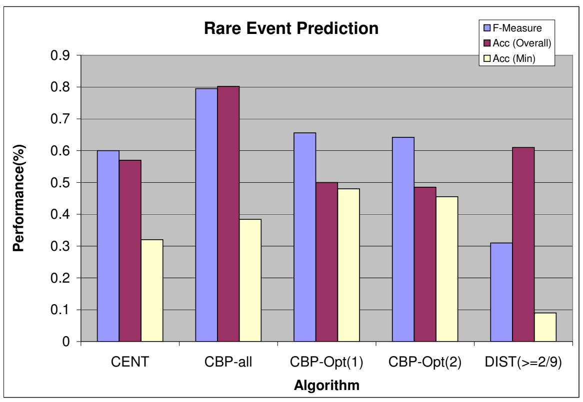

Figure 8: Body Sensor Data - Rare Event Prediction

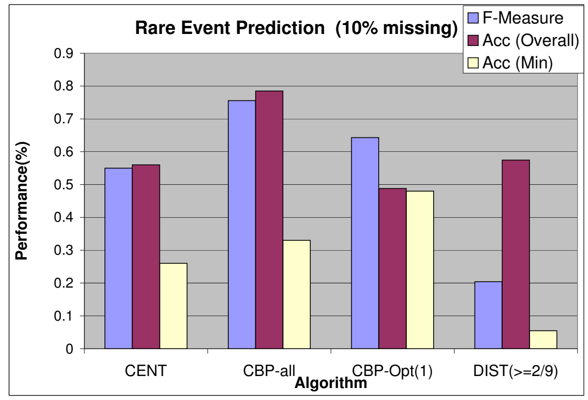

Figure 9: Rare Event Prediction with 10% Missing Data

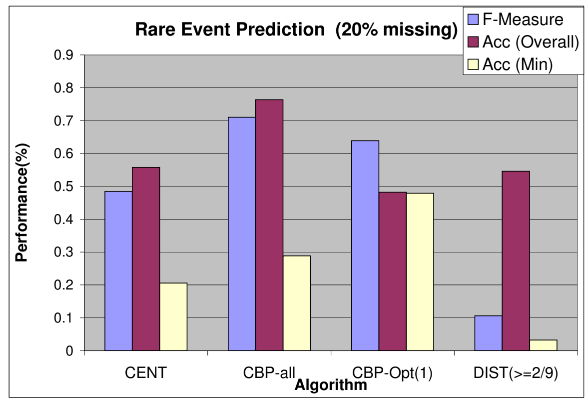

Figure 10: Rare Event Prediction with 20% Missing Data

false positives. Although the overall accuracy of the DIST is better than the centralized approach, its accuracy in identifying the rare event is extremely poor. Once again, DIST with majority vote ($\geq 5/9$, $\geq 4/9$ and $\geq 3/9$) did not yield any positive predictions. Hence, we present the results for the ($\geq 2/9$) case.

4.2.3 Experiments with Missing Data

To test the robustness of our approach in the presence of missing data, we removed 10% and 20% of the transmissions at random. We used the CBP, CBP-Opt(1), CENT and DIST approaches on this data. The results are shown in Figure 9 and 10. We find that our techniques perform well even with 20% of the transmissions missing. Athough all the techniques have lower performance when some data is missing, it is interesting to note that the CBP approaches do better with up to 20% of the data missing than the Centralized approach with full data. It is surprising to see that the CBP-Opt(1) suffers the least in terms of performance in the presence of missing data. This can be explained by the fact that it produces a lot of false positives, as we observed in the previous case, so in spite of having missing data, it can still provide a high recall for the minority class. Since, we are using only k-1 classifiers to predict, the presence of missing data does not affect the performance too much.

From our experimental results, we can conclude that our Correlation-based Partitioning approach provides good performance gains in terms of accuracy and energy efficiency. Although the absence of a combined evaluation is a minor limitation of this work, from our experimental results, we strongly believe that our approach will indeed provide enormous gains in both accuracy and energy savings on real sensor network datasets.

Another important observation from our results is the good performance we obtain when we use k-1 and k-2 classifier nodes in the Optimized scheme. Although this approach is approximate, it provides better performance than the Centralized and the totally distributed scenarios. This justifies our decision to have multiple classifier nodes to improve fault tolerance. Our partitioning scheme ensures that each classifier node gets a subset of the readings of each individual node. Hence, the presence of node failures, noise or missing data which might disrupt classification performance in the centralized case, does not affect our performance.

5. CONCLUSIONS AND FUTURE WORK

In this paper, we have proposed a correlation-based partitioning technique for rare event detection in sensor networks. From our detailed experimental evaluations, we have found that our technique not only improves the accuracy of detection but also provides significant energy savings to the sensor network. One limitation of our work is the fact that we have not been able to obtain a suitable sensor dataset to test our scheme in its entirety. In the future, we would like to perform this evaluation. We strongly believe that our technique will yield the same significant improvements it has shown in the experiments we have performed. Apart from the improvements in accuracy and energy savings, our partitioning technique also makes the sensor network more robust to node failures and noisy or missing data.

In this work, we have confined ourselves to studying rare event detection within a cluster. In the future, we would like to extend our scheme to detect rare events on heterogeneous sites spread over multiple clusters.

6. REFERENCES

[1] The mit uamps leach ns code extensions. *http://www-mtl.mit.edu/researchgroups/icsystems/uamps/research/leach/.*

[2] G.E.A.P.A. Batista, R.C. Prati, and M.C. Monard. A study of the behavior of several methods for balancing machine learning training data. *SIGKDD Explorations*, 6(1):20–29, 2004.

[3] M. F. Duarte and Y. H. Hu. Vehicle classification in distributed sensor networks. *J. Parallel Distrib. Comput.*, 64(7):826–838, 2004.

[4] C. Meesookho et al. Collaborative classification applications in sensor networks. *In 2nd IEEE Sensor Array and Multichannel Signal Processing Workshop*, 2002.

[5] H. Kargupta et al. Collective data mining: A new perspective toward distributed data mining. *Advances in Distributed Data Mining, Eds: Hillol Kargupta and Philip Chan, AAAI/MIT Press*, 1999.

[6] N. V. Chawla et al. Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence and Research*, 16:321–357, 2002.

[7] N. V. Chawla et al. Smoteboost: Improving prediction of the minority class in boosting. In *PKDD*, pages 107–119, 2003.

[8] R. Barandela et al. New applications of ensembles of classifiers. *Pattern Analysis and Applications*, 6(3):245–256, 2003.

[9] T. Y. Wang et al. Distributed fault-tolerant classification in wireless sensor networks. *IEEE Journal On Selected Areas In Communications*, 23(4), 2005.

[10] H. Guo and H.L. Viktor. Boosting with data generation: Improving the classification of hard to learn examples. *IEA/AIE*, 2004.

[11] H. Guo and H.L. Viktor. Learning from imbalanced data sets with boosting and data generation: the databoost-im approach. *SIGKDD Explorations*, 6(1):30–39, 2004.

[12] M. Hall. Correlation-based feature selection for machine learning. *Ph.D diss. Hamilton, NZ: Waikato University, Department of Computer Science.*, 1998.

[13] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In *HICSS*, 2000.

[14] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan. An application-specific protocol architecture for wireless microsensor networks. *IEEE Transactions on Wireless Communications*, 1(4), 2002.

[15] R. M. Hogarth. Methods for aggregating opinions. *In H. Jungermann and G. de Zeeuw, editors, Decision Making and Change in Human Affairs*, 1977.

[16] N. Japkowicz. The class imbalance problem: Significance and strategies. In *Proceedings of the 2000 International Conference on Artificial Intelligence (IC-AI'2000)*, volume 1, pages 111–117, 2000.

[17] N. Japkowicz. Concept-learning in the presence of between-class and within-class imbalances. pages 67–77. Springer-Verlag, 2001.

[18] M. V. Joshi, R. C. Agarwal, and V. Kumar. Mining needle in a haystack: classifying rare classes via two-phase rule induction. *SIGMOD Record (ACM Special Interest Group on Management of Data)*, 30(2):91–102, 2001.

[19] P. Langley and S. Sage. Induction of selective bayesian classifiers. *In Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence*, pages 399–406, 1994.

[20] S. McCanne and S. Floyd. The ns-2 network simulator. *http://www.isi.edu/nsnam/ns/.*

[21] S. M. McConnell and D. B. Skillicorn. A distributed approach for prediction in sensor networks. *SIAM Workshop on Data Mining in Sensor Networks*, 2005.

[22] P. Radivojac, U. Korad, K.M. Sivalingam, and Z. Obradovic. Learning from class-imbalanced data in wireless sensor networks. *Vehicular Technology Conference(VTCÓ3-Fall)*, pages 3030–3034, 2003.

[23] C. Rosenberg V. Mhatre. Homogeneous vs heterogeneous clustered sensor networks: A comparative study. *IEEE International Conference on Communications*, 2004.

[24] I. H. Witten and E. Frank. Data mining: Practical machine learning tools and techniques. *2nd Edition Morgan Kaufmann*, 2005.

[25] Q. Zhang and P. K. Varshney. Decentralized m-ary detection via hierarchical binary decision fusion. *Information Fusion*, 2(1):3–16, 2001.