Distributed and Efficient Classifiers for Wireless Audio-Sensor Networks

Technical Report TR08-02. Department of Computing Science. University of Alberta

Baljeet Malhotra
University of Alberta, Canada
Email: baljeet@cs.ualberta.ca

Ioanis Nikolaidis
University of Alberta, Canada
Email: yannis@cs.ualberta.ca

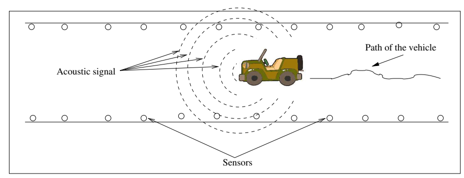
Mario A. Nascimento
University of Alberta, Canada
Email: mn@cs.ualberta.ca

***Abstract*—We consider the problem of vehicle classification using acoustic signals captured within a sensor network. The sensors perform collaborative decision and/or data fusion in a distributed and energy efficient manner. We present a distributed cluster-based algorithm, where sensors form clusters on-demand for the sake of running the classification task. We aim at minimizing the energy costs incurred due to the transmission of the feature vectors among collaborating sensors within a cluster. To this end, we present schemes to generate effective feature vectors of low dimension. An experimental study has been conducted using real acoustic signals of military vehicles recorded during DARPA's Sensit/IXOs project. The features generated through our proposed schemes are evaluated using K-Nearest Neighbor (k-NN) and Maximum Likelihood (ML) classifiers. Performance results indicate that the proposed schemes are effective in terms of classification accuracy, and can even outperform previously proposed approaches, but, in addition, they are also efficient in terms of communication overhead.**

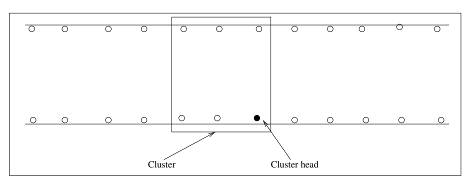
I. Introduction

Vehicle tracking on acoustic data is based on the fact that different vehicles produce distinctly different acoustic signals because their engine and propulsion mechanisms are unique [12]. The problem of vehicle detection using the acoustic signature has existed for years and many solutions have been proposed in the literature [2], [12], [15]. Recently, target classification based on acoustic signals in wireless sensor networks has been addressed in [3], [7], [9]. Sensor networks provide redundancy in terms of sensing and processing units. Hence, they can operate to detect and report the presence of a target vehicle, possibly refining the tracking and classification quality as the target is moving. In this paper we restrict our attention to the classification task. Classification is necessary because sensors may be required to report on specific types of vehicles once sensors recognize them which are of some interest, e.g., belong to a specific class. We further assume that traffic is constrained to paths, e.g., roads, where sensors are deployed in an organized manner, for instance, on light poles as suggested in Fig. 1(a).

The inherently distributed environment of a sensor network provides further challenges to achieve good classification results. Moving vehicles can be detected, and detection measurements can be recorded at multiple sensors. These measurements can be exploited in a number of ways. Two basic approaches that are popular in the literature are data fusion (DAF) and decision fusion (DEF). In a data fusion approach, individual measurements from multiple sensors can be collected at a central location to perform classification. The choice of a central location may depend on various constraints such as number of sensor nodes that detect the same event at approximately the same time, and communication range of the sensors. Inspired by clustering approaches similar to the ones presented in [1], [4], we use data fusion based classification techniques in which a cluster–head becomes the central location. It may use multiple feature vectors collected from the cluster members to predict the class of the unknown vehicle. In the second approach of decision fusion, sensors in a cluster can individually perform a classification technique to make a decision. Individual decisions can be collected by the cluster head to predict, e.g., by voting, the class of an unknown vehicle. More discussion on the data and decision fusion based approaches can also be found in [3].



(a) Vehicle detection by the sensors.



(b) A cluster and its cluster–head.

Fig. 1. Example of vehicle detection and cluster formation in a sensor network along a narrow one–way street.

Implementing decision or data fusion based approaches requires sensors to collaborate efficiently with each other to perform classification. We use a sensor network to perform the classification task within a limited and non-fixed subset of the network, which we call a *cluster*. Clusters can be formed *on-demand* when a signal suggesting the presence of a vehicle is detected by a minimum number of sensors. The purpose of on-demand based clustering is that a static logical structure is not needed to be maintained until an actual classification operation is to be performed. Maintaining a static logical structure requires sensors to send periodic communication messages, which are considered expensive in sensor networks [13]. Moreover, it may not be useful to add sensors to the classification task that did not receive an audio signal of appreciable strength or quality. Finally, the logical structures used for distributed computation can be dissolved after a sufficient amount of time has elapsed, and there is no more presence of a signal suggesting vehicular activity.

In this context, we describe in Section II a clustering algorithm, which forms the basis of our distributed classification framework. Using the same we can evaluate the benefits of DAF and DEF in various scenarios. We evaluate these approaches and scenarios using K-Nearest Neighbor (*k-NN*) and Maximum Likelihood (*ML*) classifiers, discussed in Section III. These classifiers operate upon feature vectors, i.e., from each audio sample a time series data analysis is performed upon the obtained audio sample. Selection of feature vectors is thus critical for achieving good classification results [6]. However, in order to keep the communication costs low, particularly in the data fusion based approaches, it is highly desirable to have low dimensional feature vectors which faithfully represent the acoustic signals of vehicles. Towards that goal we describe in Section III simple feature extraction schemes that yield low dimensional, yet representative, feature vectors of the captured acoustic signals. In Section IV, we present performance evaluation results in terms of classification accuracy and energy expenditure trade-offs. Finally, Section V summarizes the findings of the paper and outlines our future research goals.

II. On-Demand Based Clustering

We consider a network of static sensor nodes, and a subset of the network that forms a cluster. The classification task is to be performed collectively by the nodes within a cluster. We denote by T the time during which a vehicle is classified. Further, we assume that no more than one vehicle is within the network during T, all sensors have a synchronized global clock, and all sensors gather (sense) data every W_t time units ($W_t < T$).

The problem at hand is then to organize the sensors in the form of a cluster that work as a single unit to classify a vehicle that is moving in the network. A scenario of vehicle detection, and cluster formation is depicted in Fig. 1. Sensors are placed along a path to classify a moving vehicle as illustrated in Fig. 1(a). As a vehicle enters the network region sensors can detect its presence by measuring its acoustic energy using a detection algorithm such as the one proposed in [5]. Our focus here is not on detection mechanisms, but using a detection mechanism to form the clusters. Our proposed on-demand clustering algorithm simply relies on existing methods for detection. Upon detection some sensors communicate with each other to form a cluster as suggested in Fig. 1(b).

A. Cluster Head Selection & Cluster Formation

Cluster formation is triggered when a number of sensors confirm the detection of an acoustic signal of an appreciable strength, suggesting the presence of a vehicle in the vicinity of sensors. A sensor that detects the vehicle broadcasts a *detection message* to its neighbors. To distinguish the sensors that have detected a vehicle from the sensors that have not detected the vehicle, we call the detecting sensors as *active nodes*. The detection message sent by the active nodes consists of the signal strength of the acoustic energy they have received. All active nodes keep track of only the most recent *detection messages* they have received from their neighboring active nodes.

An active node that finds its signal strength greater than its neighboring active nodes chooses to become a *cluster head* if it has received at least N_t detection messages from its neighboring active nodes. Otherwise, it waits for W_t amount of time before measuring the signal strength again, and, if appropriate, sending a detection message. This addresses the case where there is not a minimum set of active nodes to form a sufficiently large cluster. (As we will see, the size of the cluster matters in the resulting classification accuracy.) The cluster–head broadcasts a *cluster-head message*, so that all other neighboring active nodes know the presence of an active node that has greater signal strength, and therefore, they do not broadcast a cluster-head message. Since the cluster–head is aware of its neighbor active nodes it can select all of these to form a cluster or it may choose only those that have reported a good signal strength. In our proposed approach we assume the cluster–head selects all reporting active nodes.

The cluster–head message consists of a *membership list* of the active nodes that have been selected to form the cluster. All active nodes, which receive the cluster–head message, and found themselves on the membership list reply back with a *confirmation message* to form a cluster. It is possible that an active node may receive multiple cluster–head messages. In that case we assume it selects a cluster–head that has greater signal strength. Once part of a cluster, an active node does not send further detection messages and also ignores the detection messages from other active nodes until the classification time, T, expires.

Once a cluster is formed, the classification process starts and it involves two parts: (i) assigning tasks to the individual active nodes, and (ii) collecting the classification results. A cluster–head coordinates with the active nodes in its cluster to perform this process. Recall that at this point in time all active nodes in a cluster have their own feature vectors ready to be used in the classification process. A cluster–head orchestrates the classification process as follows:

1) A cluster–head prepares a *schedule*, and broadcasts it in the cluster. A schedule consists of *task assignments* for all (active) nodes in a cluster. A typical task for an active node is to compute the similarity measure of an

unknown sample with respect to the training samples as specified in the schedule.

2) After performing their assigned task, sensors report back to the cluster–head with their individual results (e.g., a decision or distance measurement computed in response to the first step). After collecting results the cluster–head makes a decision on the class of the unknown vehicle.

It is conceivable that multiple clusters be formed as the vehicle moves along its path, and these clusters may cooperate in the classification process. For the time being however, we assume that the classification task is performed by single clusters (perhaps several times by different clusters as the vehicle moves) using either data fusion or decision fusion. We make this assumption as multiple clusters may not easily reach a consensual decision about a detected vehicle, which may have not exhibited a certain audible characteristic consistently along its path, e.g., due to environmental noise. Thus, we consider the classification task within a single cluster using either data fusion or decision fusion.

When DAF is used, all sensors in a cluster send their feature vectors to their cluster–head. The cluster–head combines all received feature vectors (including one from itself) and executes the classification task using, e.g., k-NN or Maximum-Likelihood classifier. When DEF is employed, all sensors in a cluster execute the classification task using their own feature vectors, and make their own decision on the class of the unknown vehicle. The sensors then provide the cluster–head with their decision, which, after receiving all decisions (including one from itself), predicts the class of the unknown vehicle based on the majority of decisions.

III. Classification Techniques and Features Selection

Techniques such as K-Nearest Neighbor (k-NN) and Maximum Likelihood (ML) can be used for classification. k-NN is one of the simplest, yet accurate, classification method. It is based on the idea that similar objects are close to each other in a multidimensional feature space. Consider a set U consisting of n samples, $\{u_1,\ u_2,\ ......u_n\}$, such that for all $u_i \in U$ class labels are known in advance. To predict the class of an unknown sample, x, the k-NN method finds k "closest" samples from the set U, and classifies x as the majority class of the k retrieved samples. Closeness can be computed, for instance, using the Euclidean distance if the samples are feature vectors in the Euclidean space. In our case the set U consists of n acoustic samples. Some well known methods such as fast Fourier transform (FFT) and power spectral density (PSD) can be used to extract features from the acoustic samples.

If we assume d to be the length of the feature vectors, and l to be the number of training samples in each of the c classes then the number of computations performed by a sensor to classify an unknown sample is proportional to $d \times l \times c$. If the total number of training samples are large, and there are many classes then the time required to classify an unknown sample using k-NN method may be quite high. The dimensionality of the selected feature vectors makes the k-NN method costly for most real time application. Therefore, feature vectors of low dimension are important to k-NN method, especially when sensors are assumed to have limited computing resources.

On the other hand in the ML classifier an unknown sample, x, is classified to be in class j if

$$q_j p_j(x) = max\{q_i p_i(x)\ \forall i\} \tag{1}$$

where q_i and p_i are, respectively, the known a priory probability and the probability density function of a class i. Assuming an equal a priori probability for each class Eq. 1 becomes:

$$p_j(x) = max\{p_i(x')\ \forall i\} \tag{2}$$

The probability density function, $p_i(x')$ is given by the following known equation.

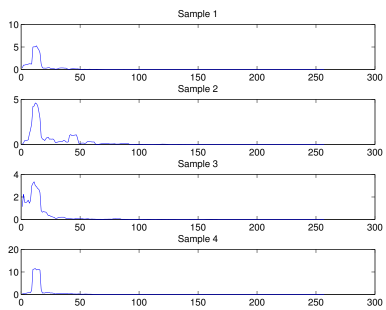
$$p_i(x') = \frac{1}{(2\pi)^{\frac{d}{2}}|\Sigma_i|^{\frac{1}{2}}} \exp\left[\frac{-1}{2}(x'-\mu_i)^T {\Sigma_i}^{-1}(x'-\mu_i)\right], \tag{3}$$

where Σ_i, of size d^2, is the covariance matrix of class i, μ_i is the mean of samples in the class i, and $|\Sigma_i|$ and ${\Sigma_i}^{-1}$ are, respectively, the determinant and inverse of the covariance matrix of the the class i. In order to classify an unknown sample, x', a sensor must determine the values for μ_i, $|\Sigma_i|$, and ${\Sigma_i}^{-1}$, which can be computed *off-line*. In Eq. 3 variable x' is unknown until the sample is made available. Therefore, a sensor must compute $(x'-\mu_i)^T {\Sigma_i}^{-1}(x'-\mu_i)$ *on-line* in order to classify the unknown sample, x'. It is worth noting that the size of the ${\Sigma_i}^{-1}$ matrix is $d \times d$, and the number of operations performed by a sensor to compute $(x'-\mu_i)^T {\Sigma_i}^{-1}(x'-\mu_i)$ for all classes is proportional to d^2. Therefore, dimensionality of the feature vector is also important to the ML classifier.

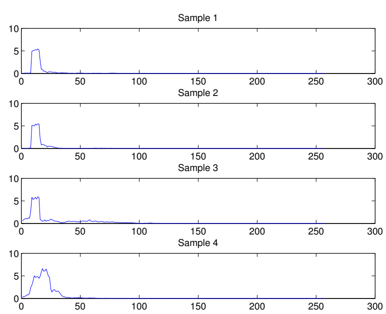
Selection of d is thus critical as we would like to reduce the computational load from sensors, of-course, without compromising on the quality of solution. In addition, the size of a feature vector is also crucial for communication costs in sensor networks. That is, a larger d causes a higher rate of energy consumption for the communication among the sensors.

Many researches have used various techniques to extract feature vectors from the acoustic signatures, Duarte *et. al.* in [7], first choose 100 FFT points from the fast Fourier transform of 512 data points sampled at a rate of 4.960 kHZ. Then, they average the 100 FFT points by pairing consecutive points to get a 50 dimensional feature vector. Brooks *et. al.* [3] also used a 50-dimensional FFT feature vectors extracted from the time series data. Wang *et. al.* [14] used PCA to choose the 15 largest eigenvalues to form the eigenspace for their training and test data. Unfortunately, selecting the first few principal components provides only a measure of statistical significance without guaranteeing to yield the best subset of features that can discriminate between the classes. The reason is that PCA finds feature combinations that model the variance of a data set, but these may not be the same features that separate the classes [10].

These pre–existing methods are either simply not meeting the demands of energy conservation in sensor networks [7], or they are generic in nature (*e.g.* PCA) and computationally expensive, while not even yielding the best results. More importantly, however, they do not address the issue of relating dimensionality of the feature vectors issue with the objective of energy conservation. What is required is to meet the two demands that compete with each other *i.e.* creating feature vectors that are low on dimensions and that can still produce good classification results. Towards that goal we present two schemes for feature extraction from the acoustic signatures of vehicles.



(a) PSD of four randomly chosen samples from the Assault Amphibian Vehicle (AAV) set.



(b) PSD of four randomly chosen samples from the Dragon Wagon (DW) set.

Fig. 2. Example PSDs for samples from two different classes of vehicle extracted from the SensIT dataset [7].

A. Features Selection Schemes

A vehicle sound is a stochastic signal. In practice, the sound of a moving vehicle observed over a period of time can be treated as a stationary signal [15]. In our case we use the signal's duration to be 51.6 ms, *i.e.*, 256 data points sampled at a frequency of 4.960 kHz. In our study we considered power spectral density (PSD) based features. This feature is generated by taking PSD estimates of 256 data points yielding a linear vector of 128 PSD points with a resolution of 38.75 Hz. In the rest of the discussion a PSD point is also called a band of frequencies, or simply a dimension, because a PSD point represents a collection of consecutive frequencies.

Our proposed schemes start by considering all 128 dimensions, and subsequently pruning many of them through a number of steps described next. The basis of our proposed schemes, and the first pruning criteria is that most of the power in a vehicle's sound lies in the lower frequencies. As shown in Fig. 2 for two classes of vehicles, AAV and DW taken from the Sensit dataset [7], the power is mostly concentrated in the lower frequencies, *i.e.*, between 200 Hz and 1000 Hz. To create a feature vector our schemes start by choosing only those dimensions that correspond to frequencies that have the maximum power as reported by the samples of the corresponding training class.

Specifically, let f_i^j be a frequency band that has the maximum power as reported by the sample j in the class i. Let S_i be the set of all f_i^j's reported by all samples j in the class i. Note that $|S_i| \leq l$, i.e., some samples in class i may report on a common dimension. This particular situation is favorable for producing feature vectors that are low on dimension, and yet be effective. Our intuition is that a dimension that has been reported by a large number of samples in a class is more suitable to characterize that particular class than a dimension which has not. Following this intuition, we rearrange S_i. First, we count the number of times each unique dimension has appeared in S_i to obtain their rank. Then, we place each unique dimension in S_i in an decreasing order based on their rank. After rearranging set, S_i, we further prune some more dimensions by selecting a percentage, ρ, of top ranked dimensions from S_i to constitute another set, S_i^ρ. By this pruning criteria, we eliminate those dimensions that are less frequent in the training class. This process is repeated, in order to derive the S_i^ρ set for each class i = 1,2...*c*.

We propose two schemes to select elements (dimensions) from sets S_i^ρ to create the feature vectors, which will be stored in the sensor nodes.

In the first approach the sets S_i^ρ are combined to get a final set,

$$S = \bigcup S_i^\rho \quad \forall i. \tag{4}$$

which selects feature vectors with dimensions those that are present in S. We name this approach an independent feature selection (IFS) scheme. In the second approach, features are selected by considering only those dimensions from sets S_i^ρ that are common to all classes.

$$S = \bigcap S_i^\rho \; \forall i. \tag{5}$$

We name this latter approach global feature selection (GFS), as dimensions are common to all training classes. A potential problem that might occur in the GFS scheme is that the final set S may remain empty if there is no common dimension among the sets S_i^ρ. Since we can control the size S_i^ρ by setting an appropriate value for ρ, we can handle this exception by increasing the value of ρ. If the set S remains empty even for $\rho = 1.0$, the first element from all sets S_i^ρ is chosen to be inserted into the final set, S.

IFS/GFS feature vectors can be obtained in advance from a training set. The training samples, and their corresponding feature vectors are then uploaded to the sensors before their deployment. After deployment , sensors can extract 128 PSD points from the time series data of unknown vehicles, and directly fetch IFS/GFS feature vectors (from local storage) for the known vehicle using the selected dimensions learned from the training phase.

IV. Experimental Study

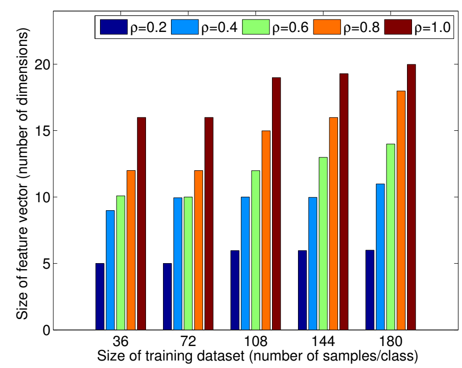
In this section we present the results of our experimental study where we evaluate the performance of our distributed classification schemes as well as merits of feature vectors generated through our proposed feature extraction schemes. We are mainly interested in comparing accuracy results with the results of already existing studies. With that in mind, we chose an acoustic dataset that has been used elsewhere for similar studies. The dataset we consider was generated during the third SensIT situational experiment (SITEX02), organized by DARPA/IXOs SensIT program. In the rest of the paper we refer to this dataset as Sensit dataset[1] It consists of acoustic samples of Assault Amphibian Vehicle (AAV), Dragon Wagon (DW) recorded at 29 Palms, California in Nov. 2001. These samples are organized by the run numbers, from 2 to 12, and the sensors that recorded the acoustics. There are total of 23 sensors that are numbered 1-6, 41-42, 46-56, and 58-61. There are total of 9 runs for the AAV class (AAV3-AAV11) and 11 runs for the DW class (DW2-DW12). After rearranging the original Sensit dataset we had 180 and 209 samples, respectively, from AAV and DW class of vehicles. We standardized our dataset to remove any shifting and scaling factors by using the *normal form* [8] of the original time series data.

One of the challenges in our experimental study was to simulate a distributed environment of a sensor network. The signal captured of an unknown vehicle that is captured by a sensor may be different from the signal from the same vehicle but captured by another sensor at approximately the same time. This is due to the placement of the sensors. In order to create multiple copies of an acoustic signal for different sensors we adopted the following procedure. We selected an acoustic signal from our dataset, and created multiple copies of the same, attenuating the original signal based on the distance of the sensors from the moving vehicle. Then, we introduced time difference of arrival lags for the sensors based on their relative position with respect to the moving target. We also added white noise for each of the sensor's signal. Finally, we also standardize the (attenuated and noised) signal that is assumed to be received at each sensor, by applying the normal form of the synthetic time series created as described above. This procedure is repeated for every testing signal in our dataset. For the sake of simplicity we assume the environment to be such that effects such a reverberation and Doppler effect can be safely ignored as negligible.

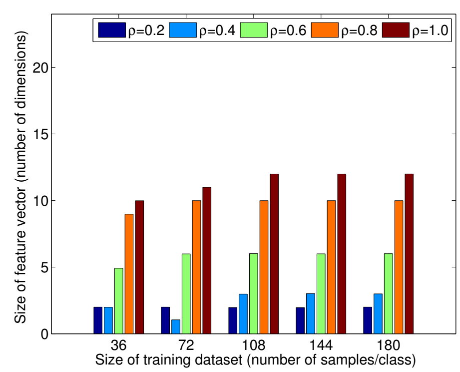
[1]http://www.ece.wisc.edu/∼sensit

A. Performance Metrics

We consider two performance metrics: (i) classification accuracy, and (ii) energy expenditure. Classification accuracy is computed based on the number of unknown samples that are correctly predicted by the classifiers. For constituting training classes we choose an equal number of samples in both the classes from our dataset. The training samples are chosen at random. The rest of the samples become testing samples. Naturally, the class label of the testing sample is unknown to the classifier. A sample is considered correctly classified if the true class is predicted. In the k-NN method similarity between any two samples is computed using $L1$ distance metric. Energy expenditure is computed based on the number of bits transmitted by a sensor. We assume the same radio model as in [13], according to which a sensor spends $50 + .1 \times R^3\ nJ/bit$ of energy to send one bit at R distance. The cost of assembling the cluster is the same for both data and decision fusion approaches, and is therefore irrelevant to discriminate which one is more efficient



(a) IFS



(b) GFS

Fig. 3. IFS & GFS feature vector size.

B. Size of IFS/GFS Feature Vectors vs. Training Classes

Before implementing the proposed classification schemes we studied the relationship between the size of the training classes and the size of the feature vectors generated through our proposed IFS/GFS schemes. Fig. 3 summarizes our findings on this relationship. The results shown in Fig. 3 are obtained by varying the size of the training classes. Equal number of training samples are chosen for both the classes to disfavor any particular class. Since two classes have different sizes, we chose the smaller class, *i.e.* AAV, to be the maximum size of a training class. That means if we chose to select 20% of samples from AAV class, *i.e.*, 36 out of 180 samples, to constitute a AAV training class, then we also chose 36 samples to constitute the DW training class. Samples from the respective classes are chosen randomly. With this policy we chose 20, 30, 40, 60, 80, and 100 percent of total samples in AAV class, which gave us the set 36, 72, ..., 180 that we used as the number of samples to constitute our training classes for AAV and DW. Also, the number of top ranked dimensions selected from sets S_i^ρ were varied by setting a ρ value from the set 0.2, 0.4, ..., 1.0.

Several observations can be made from the results shown in Fig. 3. In Figures 3(a) and 3(b) values along the y-axis (feature vector size) indicate the consensus among the training samples on common bands of frequencies (dimensions) that have the maximum power. A smaller feature vector size indicates a wider consensus among the training samples on common dimensions, in contrast to a larger size of feature vector indicates that fewer samples agree on common dimensions. With a particular training class size, when we allow to select more top ranked dimensions (by increasing ρ from 0.2 to 1.0), the size of feature vector naturally increases for both of the schemes as shown in Figures 3(a) and 3(b). This trend continues for all sizes of training class, i.e., 36-180 samples/class. Another more noticeable trend is that as the training class size increases from 108 samples/class to 180 samples/class, the size of the feature vector does not change much. This behavior can be explained as follows: when the size of the training class is sufficient the consensus among the samples of the training classes is high, hence, adding more samples into the training classes does not affect the consensus much. As we see in Figures 3(a) and 3(b) when the training class size changes from 108 to 180 samples/class the feature vector size does not go beyond 18 and 12, respectively, in GFS and IFS schemes. As expected, with a similar setting of parameters for IFS and GFS schemes, GFS scheme produced feature vectors of smaller size. This trend can be seen by comparing the results in Figures 3(a) and 3(b).

C. *Classification Accuracy*

We used both IFS and GFS schemes to generate feature vectors for the k-NN and ML classifiers. k-NN and ML classifiers obtained different classification accuracies with various settings of IFS and GFS schemes. The best classification accuracies are reported here and compared with the previously achieved best accuracies on the same dataset. In particular the results reported in this section were obtained with a training class of 45 samples/class and ρ value of 0.3 for GFS scheme in the k-NN classifier, and 63 samples/class and ρ value of 0.5 for IFS scheme in the ML classifier. With this setting, the average size of the feature vectors was found to be 8, which is almost 1/6 and 1/2 the size of the features vectors used in [7] and [3], respectively. In general, selecting 30-50% of the top ranked dimensions (i.e. a ρ value in the range [0.3 0.5]) produced the best results.

	Our accuracies	**Brooks**[3]	**Duarte**[7]
k-NN	77.89	−−	69.36
ML	89.46	77.90	68.95

TABLE I
DEF CLASSIFICATION ACCURACY.

	Our accuracies	**Brooks**[3]	**Wang**[14]
k-NN	77.63	−−	84.68
ML	89.20	81.30	−−

TABLE II
DAF CLASSIFICATION ACCURACY.

	Accuracy (%)			
	k-NN		**ML**	
Study	DEF	DAF	DEF	DAF
Our's	**77.89**	77.63	**89.46**	**89.20**
Brooks[3]	–	–	77.90	81.30
Duarte[7]	69.36	–	68.95	–
Wang[14]	–	**84.68**	–	–

TABLE III
COMPARISION OF CLASSIFICATION ACCURACIES.

	Accuracy (%)				**Cost** (μJ/sensor)			
	k-NN		**ML**		**k-NN**		**ML**	
Cluster Size	DEF	DAF	DEF	DAF	DEF	DAF	DEF	DAF
3	70	68	88	83	17	21	17	21
5	70	71	89	89	19	25	19	24
10	72	73	88	89	29	38	26	37
20	74	77	89	89	67	83	67	82
40	78	76	89	88	209	240	210	239

TABLE IV
CLASSIFICATION ACCURACY AND COMMUNICATION COST FOR VARIOUS CLUSTER SIZE.

Tables I and II summarize and compare our classification results with the results from the previous studies on the same dataset using various decision and data fusion approaches. In the study of Brooks *et. al.*, the authors considered various scenarios of data fusion and decision fusion using single and multiple sensors. The main scenarios that were considered are (i) decision and data fusion of multiple modalities at a single sensor, (ii) decision and data fusion of single modality from multiple sensors, (iii) decision fusion of multiple modalities from multiple sensors, (iv) data fusion of multiple modalities at a single sensor with decision fusion of multiple sensors, and (v) data fusion of multiple modalities from multiple sensors.

Comparison with all these approaches was not in the scope of our study. We were mainly interested in evaluating the impact of feature vectors in various settings of data and decision fusion approaches using the well known classifiers and a cluster logical structure.

In their study Brooks *et. al.* considered 2 modalities and 3 sensors. Their compared results presented in Table I are from the acoustic modality with decision fusion from multiple sensors, and the compared results in Table II are again using the acoustic modality with data fusion from multiple sensors. The results from the study of Wang *et. al.* compared in Table II are based on a data fusion approach for which they modified the multi-resolution integration (MRI) algorithm originally proposed by Prasad *et. al* in [11]. The idea here is to construct a simple overlap function for input acoustic data from multiple sensors. The overlap function is resolved at successive finer scales of resolution to select the highest and widest peaks of the fused data. The authors considered data of a single as well as three sensors from the various combinations of sensor numbers 1, 3, 5, and 11 from the Sensit dataset. Duarte *et. al.* used acoustic as well as seismic modality in their study. Their results presented in Table I are based on the local classification with decision fusion using the acoustic modality.

To the best of our ability, through the two Tables I and II presented above, we have compared the superior results from the studies mentioned previously with the best of our results obtained with various settings of data and decision fusion schemes proposed here. In particular our decision fusion approach, DEF, using GFS feature vectors produced the best classification accuracy of 89.46% as compared to all other results presented for the ML classifier.

D. Impact of Clustering : Data Fusion vs. Decision Fusion

Clustering has significant impact on the results obtained through data and decision fusion approaches. Figures 4 and 5 summarize our findings in the k-NN classifier. As shown in Fig. 4, a cluster with only three sensors achieved an accuracy of nearly 69% using DEF, while DAF achieved a slightly less accuracy. Accuracy for both of the approaches is improved as we increased the cluster size by adding more sensors to the cluster. The reason for improved accuracy is that increasing the number of sensors in the cluster increases the probability of making a correct prediction. However, after a sufficient number of sensors are available within the cluster, adding more sensors did not improve the accuracy much. It can be seen in Fig. 4 that accuracy improved from 69% to 73% as the number of sensors in the cluster increased from 3 to 20. Adding additional 20 sensors did not improve the accuracy much.

Clustering has much more impact on the energy expenditure than on the accuracy. Results for k-NN classifier are summarized in Fig. 5. As shown in Fig. 5, when the number of sensors in a single cluster increases from 3 to 40 sensors the energy expenditures increase. The reason is that the increased number of sensors causes more communication exchanges between the cluster members. Data fusion approach, DAF, incurred more cost due to the transmission of feature vectors by the sensors.

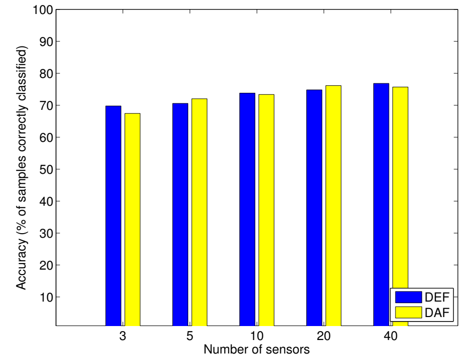


Fig. 4. k-NN classifier classification accuracy for varying cluster size.

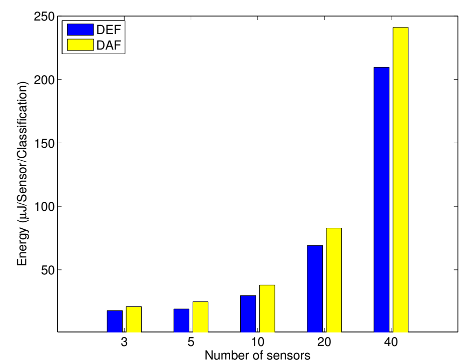


Fig. 5. k-NN classifier communication cost for varying cluster size.

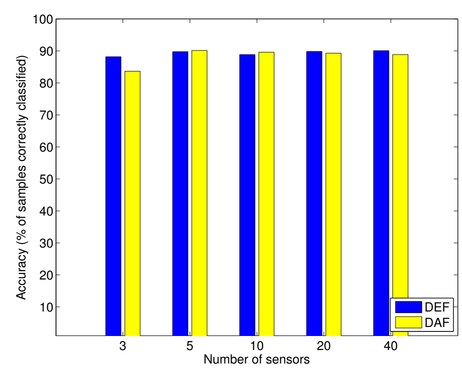


Fig. 6. ML classifier classification accuracy for varying cluster size.

Figures 6 and 7 summarize the results on accuracy and cost, respectively, in the ML classifier. As shown in Fig. 6 when cluster size changed from 3 to 40 sensors accuracy improved slightly. Overall the DEF approach performed better than DAF for reasons similar to those for the case of k-NN classifier.

Also, similar trends of energy expenditures can be noted in ML classifier as observed in k-NN classifier. Results for energy expenditures in ML classifier are summarized in Fig. 7.

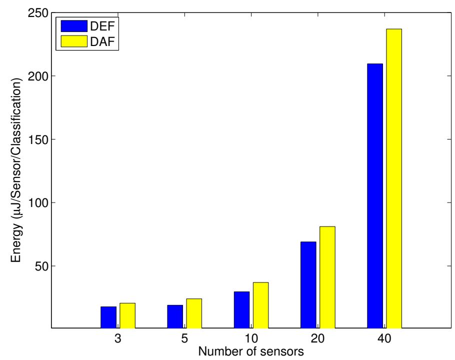


Fig. 7. ML classifier communication cost for varying cluster size.

A single large cluster incurs heavy costs without giving much benefit in terms of classification accuracy.

V. Conclusions and Future Directions

Classifying audio signals is an important application in wireless sensor networks where features extracted from acoustic signatures form the basis for classification. Efficient implementation of classification depends on whether necessary operations, such as clustering, data and decision fusion, can be performed efficiently in a distributed fashion, achieving high classification accuracy at reasonable energy cost. To address this problem we proposed a distributed clustering algorithm by which sensors can form clusters on-demand without using a static logical structure. We proposed two distributed classification schemes, which take into account the inherently distributed nature of the problem leading to good classification results. We also proposed schemes to extract low-dimensional, yet meaningful, feature vectors from the acoustics signals. Using feature vectors generated through our proposed schemes the two classifiers we used, ML and k-NN, yielded better or competitive classification results when compared to other existing approaches.

Our proposed feature extraction schemes are generic, and may find applications in other areas where feature selection is a difficult task due to high dimensionality. One limitation of our proposed schemes is finding the right size of training classes, and setting an appropriate value for ρ. However, these parameters can be learned during the training phase. We are currently investigating the proposed feature extraction schemes to improve their efficiency even further.

In the context of our application domain, another promising venue for further work is to allow the classification process to be a continuous one along the vehicle's path. For that the classification will inevitably be performed at several different clusters, which should ideally communicate among themselves in order to improve the accuracy of the result. Issues such as how and what to communicate among different clusters are the key ones that we are currently investigating.

Acknowledgment

This research is partially supported by NSERC Canada.

References

[1] S. Basagni, *Distributed Clustering for Ad Hoc Networks*, Proc. of the Int. Symp. on Parallel Architectures, Algorithms and Networks, pp. 310-315, June 1999.

[2] R. Braunlin, R. M. Jensen, and M. A. Gallo, *Acoustic Target Detection, Tracking, Classification, and Location in a Multiple-Target Environment*, Proc. of the SPIE Conf. on Peace and Wartime Applications and Technical Issues for Unattended Ground Sensors, vol. 3081, pp. 57-66, July 1997.

[3] R. R. Brooks, P. Ramanathan, and A. M. Sayeed, *Distributed Target Classification and Tracking in Sensor Networks*, Proc. of the IEEE, vol. 91, no. 8, pp. 1163-1171, August 2003.

[4] W. -P. Chen, J. C. Hou, and L. Sha, *Dynamic Clustering for Acoustic Target Tracking in Wireless Sensor Networks*, IEEE Transactions on Mobile Computing, vol. 3, no. 3, pp. 258-271, July 2004.

[5] J. Ding, S. -Y. Cheung, C. -W. Tan, P. Varaiya, *Signal Processing of Sensor Node Data for Vehicle Detection*, Proc. of the 7th Int. IEEE Conf. on Intelligent Transportation Systems, pp. 70-75, October 2004.

[6] R. Duda, P. Hart, and D. Stork, *Pattern Classification*, 2nd ed. New York: Wiley, p. 11, 2000.

[7] M. Duarte and Y. -H. Hu, *Vehicle Classification in Distributed Sensor Networks*, Journal of Parallel and Distributed Computing, vol. 64 no. 7, pp. 826-838, July 2004.

[8] D. Goldin and P. Kanellakis, *On Similarity Queries for Time-Series Data: Constraint Specification and Implementations*, Proc. of the 1st Int. Conf. on the Principles and Practice of Constraint Programming, pp. 137-153, September 1995.

[9] D. Li, K. D. Wong, Y. H. Hu, and A. M. Sayeed, *Detection, Classification, and Tracking of Targets in Distributed Sensor Networks*, IEEE Signal Processing Magazine, vol. 19, no. 2, pp. 17-30, March 2002.

[10] M. Nadler and E. Smith, *Pattern Recognition Engineering*, Wiley and Sons, p. 209, 1993.

[11] L. Prasad, S. S. Iyengar, and R. L. Rao, *Fault-Tolerant Sensor Integration Using Multiresolution Decomposition*, Physical Review E (Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics), vol. 49, no. 4, pp. 3452-3461, April 1994.

[12] G. Succi and T. K. Pedersen, *Acoustic Target Tracking and Target Identification - Recent Results*, Proc. of the SPIE Conf. on Unattended Ground Sensor Technologies and Applications, vol. 3713, pp. 10-21, July 1999.

[13] W. H. Heinzelman, A. Chandrakasan, and H. Balakrishnan. *Energy-Efficient Communication Protocol for Wireless Microsensor Networks*, Proc. of the 33rd Annual Hawaii Int. Conf. on System Sciences, vol. 2, pp. 1-10, January 2000.

[14] X. Wang, H. Qi, *Acoustic Target Classification Using Distributed Sensor Arrays*, Proc. of the IEEE Int. Conf. on Acoustics, Speech, and Signal Processing, vol. 4, pp. 4186-4189, May 2002.

[15] H. Wu, M. Siegel and P. Khosla, *Vehicle Sound Signature Recognition by Frequency Vector Principal Component Analysis*, IEEE Transactions on Instrumentation and Measurement, vol. 48, no. 5, pp. 1005-1009, October 1999.