WCCI 2012 IEEE World Congress on Computational Intelligence
June, 10-15, 2012 - Brisbane, Australia IJCNN

Similarity Clustering for Data Fusion in Wireless Sensor Networks using k-means

Afonso D. Ribas*†, Juan G. Colonna †, Carlos M. S. Figueiredo*, Eduardo F. Nakamura*†
*Computer Science Lab, Research and Technological Innovation Center (FUCAPI)
Danilo Areosa de Matos, 381, 69075-351, Manaus, AM, Brazil
{juancolonna, nakamura}@icomp.ufam.edu.br
†Institute of Computing, Federal University of Amazonas
Rodrigo Octavio Jordao, 3000, Setor Norte, 69077-000, Manaus, AM, Brazil
{afonso.degmar, mauricio.figueiredo, eduardo.nakamura}@fucapi.br

***Abstract*—Wireless Sensor Networks consist of a powerful technology for monitoring the physical world. Particularly, in-network data fusion techniques are very important to applications such as target classification and tracking to reduce the communication burden in these constrained networks. However, the efficiency of the solution can be affected by the data correlation among several sensor nodes. Thus, the application of value fusion (for clusters of nodes with correlated measurements) and decision fusion (combining the local decisions of the clusters) is a common strategy. In this work, we propose an algorithm for properly selecting the groups of nodes with correlated measurements. Experiments show that our algorithm is 30% better than a solution that considers only the spatial coherence regions.**

I. Introduction

Wireless sensor networks (WSNs) are ad hoc wireless networks formed by several low cost nodes which can sense the environment [1]. The nodes are computationaly constrained and battery powered posing new research challenges and opportunities. WSNs can be used for monitoring, classification and tracking of targets [2]. In these applications, sending all the gathered data to a base station for centralized processing can be very expansive and is discouraged. Thus, distributed algorithms [3] with in-networking data fusion [4] arise as an alternative [5]–[8].

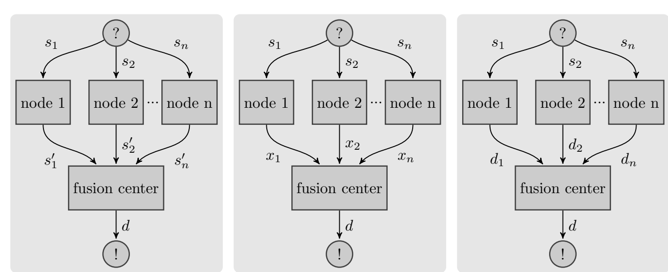
Event or target classification is a relevant class of WSN applications [5], [9], [10]. These solutions can improve classification results (accuracy, precision, recall) and extend the network lifetime by reducing the communication burden.

In this work, we use the k-means algorithm to physically cluster the WSN nodes according to data similarity. Then, a data fusion scheme which combines value fusion and decision fusion is applied. We evaluate our algorithm in an application of anurans (frogs and toads) classification based on their calls. Our experiments show that, in some scenarios, the proposed algorithm is 30% better than an algorithm from the state-of-the-art.

The remainder of this paper is organized as follows: Section II presents the related work; we describe the proposed solution in Section III; our main contribution is presented in Section III-C; the experimental results are discussed in Section IV; and finally, Section V presents the conclusions and feature work.

II. Related Work

Nakamura et al. [4] have classified data fusion into three levels depending on the abstraction of the combined data as shown in Fig. 1. In value fusion (Fig. 1(a)), a fusion center (also called master node or cluster head) combines a pre-processed version of the measured values or signals s'_i and make a global decision. We have feature fusion (Fig. 1(b)) when each node exchanges their features locally extracted x_i from the collected data s_i. In decision fusion (Fig. 1(c)), each node takes its independent decision d_i and sends it to the fusion center that combines through a decision fusion rule. In general, value and feature fusion rules take the average value, whereas decision fusion rule applies the majority vote or other that considers the local sensor statistics [11].



(a) Value Fusion. (b) Feature Fusion. (c) Decision Fusion.

Fig. 1. Three levels of data fusion.

A design concern on data fusion systems is when to use value, feature, or decision fusion. Brooks et al. [12] provide a collaborative signal processing (CSP) framework for target classification and tracking in sensor networks. They consider various scenarios of value and decision fusion using single and multiple sensors and modalities (e.g. acoustic and seismic data). They suggest that "decision fusion is preferable due to lower communication and computational burden", and it also requires less amount of training examples, whereas value fusion can potentially yield better performance. Indeed, measured values have high dimensionality and can be very redundant. The extracted features present lower dimensions but still higher than decisions which are only scalar numbers.

Clouqueur et al. [13] provide two collaborative target detection algorithms that are tolerant to sensor faults based on value and decision fusion. They showed that decision fusion yields better performance then value fusion when the number of faulty sensors increases. Malhotra et al. [14] performed experiments with real acoustic data and found similar performance for feature and decision fusion using k-NN and Maximum Likelihood classifiers.

D'Costa and Sayeed [15] showed that when measurements are perfectly correlated, data averaging yields optimal solution. Also, decision fusion is preferred when sensors collect independent measurements. In general, measurements from different nodes exhibit a mixture of correlated and independent (uncorrelated) measurements. So, an optimal classifier should combine value or feature fusion with decision fusion. Based on these results, D'Costa et al. [16] modeled events as Gaussian space-time sources and proposed to divide the region into uniform *spatial coherence regions* (SCRs), where the measurements are strongly correlated. Feature fusion should be performed among nodes inside the SCRs, which reduces the noise variance and decision fusion across different SCRs to reduce the inherent signal variability.

But in practical scenarios, the effect of noise, such as other signal sources, can lead to non-regular SCRs and it impact in the classifier performance. Thus, efficient algorithms for node clustering based on data similarity are required for a better performance, as shown in this work.

III. Distributed Classification System

In this work, we consider the problem of classifying different species of anurans based on their calls (emitted acoustic signal) in a distributed manner. We model the environment as a grid of $N \times N$ nodes separated l meters from each other with an additional random variation following a standard deviation σ_p. An event is characterized by an acoustic signal s (a vector of energy amplitudes) emitted from a random point near the center of the grid. Each node receives a different version of s. Next, we describe the acoustic signal model for sensor measurements.

A. Signal Model

The signals captured by different nodes are different due to attenuation, temporal delay and effects of noise. We are assuming the same attenuation model of Malhotra et al. [8]. The attenuated and delayed signal s_i^{at} at the sensor i can be calculated by the function $atn(\cdot)$ as follows

$$s_i^{at} = atn(s(t), i) = \frac{s(t - d_i/v)}{10^{\frac{\alpha d_i}{20}}}, \tag{1}$$

where $s(t)$ is the original audio vector at time t, v is the speed of propagation of the sound, α is a attenuator factor, d_i is the distance between node i and the source of event.

We consider two sources of noise: *anurans noise* and *white noise*. The *anurans noise* are three other anuran calls randomly positioned in the grid. These calls represent the common case where several signals from different individuals can interfere. We limited the amplitude of these additional calls by

$$s^{an} = \delta \frac{s_j}{\max(abs(s_j))}, \tag{2}$$

where s_j is a different signal from the main event. In this case, δ represents the intensity or volume of the additional calls. The *white noise* is an additive white Gaussian noise $n_i \sim \mathcal{N}(0, \sigma)$.

Then, the signal captured by node i is

$$s_i = atn(s, i) + \sum_{j=1}^{3} atn(s_j^{an}, i) + n_i. \tag{3}$$

B. Methodology

The problem is to classify in a distributed manner a given anuran call s based on the signals s_i measured by the sensor nodes. The focus of this work is on the classification task. Thus, we consider that a threshold-based detection algorithm can tell us that an unknown anuran call was emitted in the region based on its acoustic energy measurements. Sophisticated detection algorithms were proposed by Ding et al. [17] and Li et al. [2] and can be used for this purpose. In addition, the detection algorithm indicates those m nodes with highest signal power levels, namely *active nodes*. These are the nodes that participate on the distributed classification task. The node with highest signal power levels is called *master node* and coordinates the *active nodes*.

The pseudo-code of the proposed algorithm is presented in Algorithm 1. We propose to divide the region into non-regular subregions and apply value fusion for nodes inside a subregion, and decision fusion for the decisions from different subregions. This follows the same idea presented by D'Costa et al. [16] but differs on the way that the region is partitioned. Instead of using the stationary SCRs, we propose to use the clustering-based partition algorithm described in details in Section III-C. Further, we consider hard decision fusion (output class represented by integer values), instead of soft decision fusion (output class is represented by likelihood values).

Algorithm 1 Classification Algorithm.

Require: set of signal vectors $S_a = \{s_1, \ldots, s_m\}$ measured by the *active nodes*.
1: $C \leftarrow$ PARTITIONREGION(S_a)
2: **for all** $c_i \in C$ **do**
3: $d_i \leftarrow$VALUEFUSION(S_{c_i})
4: **end for**
5: **return** $d \leftarrow$ MAJORITYVOTE($\bigcup d_i$)

The Algorithm 1 is called after an event detection. The elected *master node* finds the clusters or groups of nodes with correlated measurements by performing the Algorithm 3 described later. The *master node* then reports to *active nodes* their cluster membership. So, each cluster c_i obtain a decision d_i by applying value fusion over the set S_{c_i} of signals measured by nodes of the cluster c_i (line 3). At last, the *master node* combines the local decisions and reports the result to base station.

The value fusion algorithm is described in Algorithm 2. The selected features are the first twelve Mel-Frequency Cepstral Coefficients (MFCCs) [18] of 22 mel filter channels. The classifier is based on a decision tree [19] with prune which is very suitable for WSNs once nodes need to keep only the generated rules implemented by **if** statements.

Algorithm 2 Value Fusion Algorithm.

```
Require: set of signal vectors S_v = {s_1, ..., x_M}.
function VALUEFUSION(S_v)
    s_0 ← MEAN(S_v)
    x_0 ← EXTRACTFEATURES(s_0)
    return d ← CLASSIFY(x_0)
end function
```

Fig. 2 illustrates the scenario with a grid of 10 x 10 nodes. The asterisk represents the signal that we want to classify, and the other points are sensor nodes. The square, rectangle, and triangle are fifteen *active nodes* where different shapes represent different clusters.

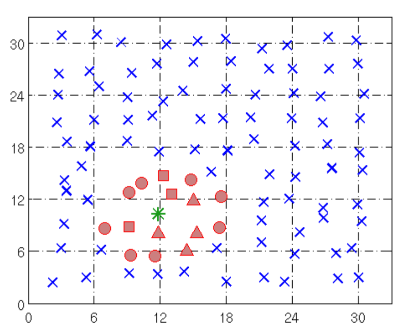


Fig. 2. Example of a deployment and the clusters of nodes.

C. Partitioning the region with k-means

In this section, we introduce the algorithm based on k-means that partitions the region of interest into non-regular subregions where the measurements are correlated. The k-means is an unsupervised machine learning algorithm which groups unlabeled instances by similarity [19]. The k indicates the number of groups or clusters. We propose to use the k-means to form clusters of nodes based on the clusters of instances returned. The pseudo-code is presented in Algorithm 3.

The *active nodes* send a fraction ϕ of their measured values s_i to the *master node* to perform the k-means algorithm. Due to constrained capabilities of sensor nodes and for energy savings, we prefer to use a fraction of s_i instead of the whole signal. The output is K clusters of similar data instances that are mapped to form physical network clusters. This mapping is depicted in Fig. 3, where each active node n_i collects its signal s_i in the physical space. In the data space, each point is a signal s_i. For the sake of simplicity, we illustrate only 2 dimensions, but in real applications, we consider any N_s-dimension space where N_s is the length of vector s_i. So, the k-means runs over the samples in the data space producing data clusters with similar samples. Now we map the data cluster in the data space to network clusters in the physical space through the index i. We expect that nodes with similar measurements (in the data space) will be clustered together (now in the physical space), and nodes with uncorrelated measurements will remain separated.

Algorithm 3 Clustering-Based Partition Algorithm

```
Require: set of signal vectors S = {s_1, ..., s_m}, a percentage value φ, and the number of clusters K.
Ensure: set of K clusters C = {c_1, ..., c_K} where c_i is a set of nodes in cluster i.
function PARTITIONREGION(S, φ, K)
    for all s_i ∈ S do
        z_i ← FRACTION(s_i, φ)
    end for
    return C ← k-MEANS(⋃ z_i, K)
end function
```

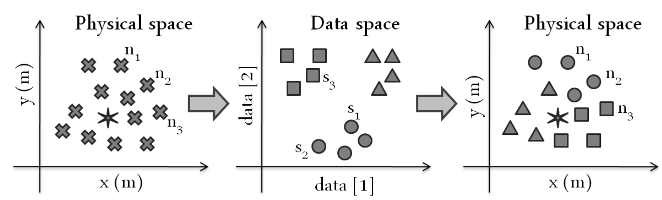


Fig. 3. Mapping from physical space to data space to physical space. Only active nodes are represented.

Compared with the SCRs of D'Costa et al. [16], this technique can lead to better results in practical scenarios since it can detect non-regular subregions where the signals are correlated.

D. Complexity

The time complexity of the Algorithm 3 is given by the complexity of the k-means algorithm which is $O(mN_sKT)$ [20], in which m is the number of instances that is equal to the number of active nodes m, N_s is the dimension of the input instance, K is the number of clusters and T is the number of iterations to converge. The most complex operation in Algorithm 2 is the FFT computation, applied to extract the MFCC from the signals, and it runs in $O(N_s \log N_s)$. This operation is performed for each cluster which gives $O(KN_s \log N_s)$. The complexity of the majority vote can be neglected. Then, the time complexity of our proposed distributed algorithm is $O(mN_sKT) + O(KN_s \log N_s)$.

If c is the cost of sending a real number over the RF link, then the communication cost of the partition region algorithm

C_1 is

$$C_1 = \phi N_s(m-1)c + (m-1)c, \quad (4)$$

that is, it requires each active node to send a fraction of its measured values and later, the master node reports the corresponding cluster of every m-1 nodes. The communication cost of the lines 2 to 4 of Algorithm 1 is

$$C_{24} = (m-K)N_s c, \quad (5)$$

that is, the active nodes send their measured values s_i to the local cluster head to perform the value fusion. At least, each cluster head sends to the master node its local decision resulting in $C_5 = Kc$. The total communication cost is

$$C = C_1 + C_{24} + C_5 = N_s(\phi m - \phi + m - K)c + (m + K - 1)c. \quad (6)$$

IV. Evaluation

In this section, we evaluate the proposed solution through simulation experiments implemented on Matlab, and compare it to other two schemes discussed in detail later. We are using the dataset collected by Colonna et al. [21]. The original audio was recorded at 8820Hz and divided into small segments of calls, called syllables, of fixed length of 1147 energy samples. In total we have 1953 syllables distributed over nine species of anurans according to Table I.

TABLE I
Distribution of samples among the classes.

Class	Specie name	# syllables	%
1	Adenomera Andre	313	16.03
2	Ameerega Trivittata	160	8.19
3	Leptodactylus Hylaedactylus	47	2.41
4	Hyla Minuta	227	11.62
5	Hypsiboas Cinerascens	223	11.42
6	Leptodactylus Fuscus	52	2.66
7	Osteocephalus Oophagus	420	21.51
8	Rhinella Granulosa	61	3.12
9	Scinax Ruber	450	23.04
Total	———	1953	100.0

The analysis was conducted over a 10-fold stratified cross validation methodology. The considered metrics are accuracy (classification rate) and macro F1 score [19]. For each new instance to be classified, we repeat the process 30 times with different noise anurans' calls (see Equation 3) and different positions. Only the mean values for the ten folds are plotted on the graphs for each analysis. In addition, we perform the paired t-test with 95% of confidence level to inquire about the superiority of our technique, but only the t-test results for F1 measure are presented.

In all cases, the total number of nodes was 10×10 nodes separated 3 m with 0.5 of standard deviation σ_p. The attenuator factor α was 0.1053 dB/m[1] that is the same value of Malhotra et al. [8]. The number m of active nodes was set to 25. Table II summarizes the default configuration parameters used on the experiments.

[1]The attenuator factor can be calculated using the Atmospheric Sound Absorption Calculator (http://www.csgnetwork.com/atmossndabsorbcalc.html).

TABLE II
Default configuration parameters.

Meaning	Symbol	Value
Number of nodes in one dimension of the grid	N	10
Distance between nodes on the grid	l	3 m
Standard deviation of the node positioning	σ_p	0.5 m
Length of the signal vector	N_s	1147
Sampling frequency of the audio signal	f_s	8820 Hz
Acoustic attenuator factor	α	0.1053 dB/m
Number of active nodes	m	25
k-means input size (fraction of the signal)	ϕ	1
Normalization factor of the anurans noise	δ	0.2
Standard deviation of the white noise	σ	0.15
Number of clusters	K	3
Maximum number of k-means iterations	T_{max}	30

For comparison we also implemented a scheme which uses only decision fusion and an adaption of the SCRs scheme of D'Costa et al. [16]. For the *only decision fusion* scheme (*only DF*, for abbreviation), we consider the majority vote of decisions of the 25 active nodes with no clustering or partitioning. This is a common approach usually adopted in distributed classification systems.

In the SCRs scheme, the region is partitioned into disjoint subregions of uniform size. Fig. 4 shows the SCRs scheme applied for the same instance presented in Fig. 2. The subregions are rectangles represented by the dashdot lines. We are considering SCRs of 6×6 m which can hold around four sensor nodes. A SCR with at least one active node is an active SCR. In the example, the event generates six active SCRs. Then, the scheme performs the lines two to five of Algorithm 1 for the active SCRs. Actually, the original algorithm proposes to use feature fusion instead of value fusion, but preliminar tests reveal poor performance, even though the required communication cost is much lower in this case.

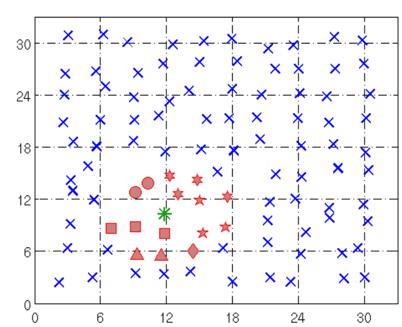


Fig. 4. Example of SCRs scheme for partitioning the region.

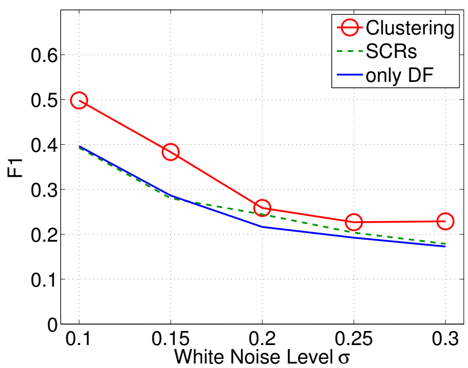
A. Noise sensitivity

We first analyze the impact of noise in our algorithm. As described in Section III-A, we have two kinds of noise: white noise and anurans noise parameterized by σ and δ, respectively. The simulation was configured with the default parameters of Table II except for the parameter that we are analyzing.

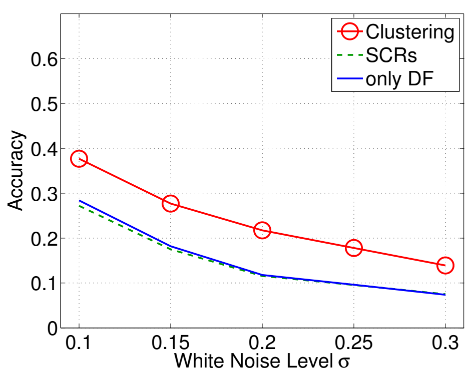
Fig. 5 shows the F1 score and accuracy when the white noise increases. The performance of the only DF and SCRs schemes was very similar, but inferior to our proposed algorithm called, from this point on, *Clustering*. The difference between the curves is high for the initial values and decreases at the end. This occurs because when the noise level increases, all algorithms converge to a point where every unknown instance is classified as the closest class to noise signals.

Table III presents the results of the t-test for the F1 score which confirms the results above. The second and third lines represent the confidence interval returned by the test. It indicates, for example, that with $\sigma = 0.15$, the F1 score of the Clustering algorithm ranges from 5.9 to 14.7 percentage points (pp) better than the SCRs scheme. The "mean F1" is the center point of the confidence interval in pp. The fourth line represents the percentage difference (in %) between the mean values of the Clustering and SCR schemes.

Except for the $\sigma = 0.20$ and 0.25, our algorithm was at least 26% better than SCRs. Considering only the accuracy, this difference is even greater.



(a) F1 score.



(b) Accuracy.

Fig. 5. White noise sensitivity evaluation.

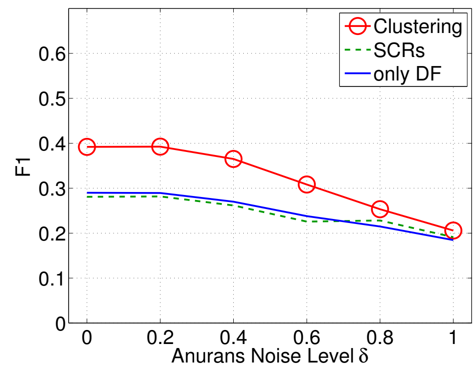
Following the same format used to analyze the white noise, Fig. 6 and Table IV show the impact of the anurans noise on the classification system. The Clustering scheme presented a significant better performance until $\delta = 0.6$, with F1 score at least 36% greater than SCRs. High values of δ (close to one) indicate that there are other anurans on the region calling

TABLE III

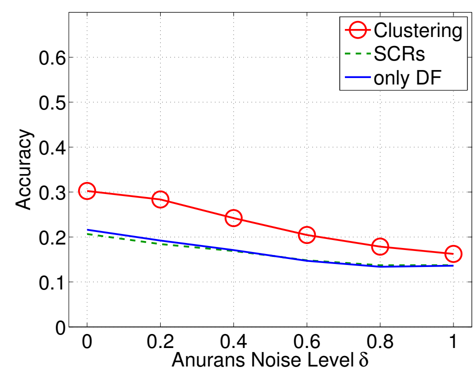
RESULTS OF THE T-TEST FOR F1 SCORE BETWEEN CLUSTERING AND SCRS SCHEMES WITH 95% CONFIDENCE LEVEL IN THE WHITE NOISE ANALYSIS.

σ	**0.10**	**0.15**	**0.20**	**0.25**	**0.30**
min	3.2	5.9	-2.0	-1.8	-0.9
max	17.9	14.7	4.9	6.4	10.9
mean	10.5	10.3	1.4	2.3	5.0
%	26.8	36.8	5.8	11.4	28.0

with almost the same intensity (volume) of the main calling. In these cases, none of the schemes presents good results, because the noises become events that should be also classified. This is a multitarget classification problem and it is out of the scope of this work.



(a) F1 score.



(b) Accuracy.

Fig. 6. Anurans noise sensitivity evaluation.

TABLE IV

RESULTS OF THE T-TEST FOR F1 SCORE BETWEEN CLUSTERING AND SCRS SCHEMES WITH 95% CONFIDENCE LEVEL IN THE ANURANS NOISE ANALYSIS.

δ	**0.0**	**0.2**	**0.4**	**0.6**	**0.8**	**1.0**
min	4.8	6.3	5.6	4.8	-0.1	-0.2
max	17.4	15.9	15.0	11.8	5.1	3.1
mean	11.1	11.1	10.3	8.3	2.5	1.5
%	39.5	39.4	39.5	36.6	11.0	7.7

Once again, the results of the baseline schemes were very similar, indicating that the SCRs do not capture the non-Gaussian regularity behavior of the signal.

B. Impact of number of clusters

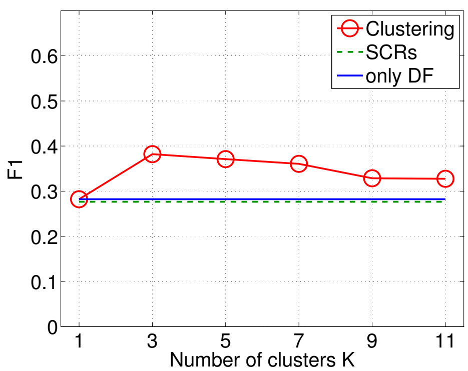
As follows we investigate the impact of the number of clusters K. Fig. 7 and Table V show the results for different values of K. We plot the two baseline algorithms as reference for comparison, but they do not vary the number of clusters. In SCRs case, even though the number of active clusters could change for each new event, there is no parameter to be configured.

The Clustering algorithm presents better performance when the number of clusters is three, and it gradually decreases for more clusters. Thus, $k = 3$ is the optimal choice. Higher Ks leads to fewer nodes per cluster, which impacts on the local classification accuracy due to a lower SNR. In other words, the signal average from more nodes leads to higher noise amortization. The number of clusters is important because it has an impact on the time complexity and communication cost as discussed on Section III-D.

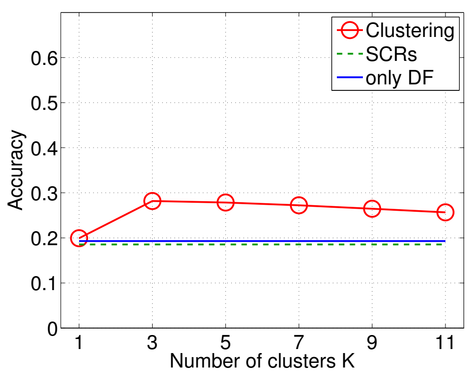
C. Impact of k-means input size

We also evaluate the size of the input for k-means which is a fraction or percentage ϕ of the measured signal s_i. Fig. 8 and Table VI show the results. Again, this parameter does not affect the baseline algorithms, but the results were presented for comparison.

This is the parameter that has the greatest impact on the overall system, including time complexity, communication cost, and classification rates. Small values yield low time complexity and low communication cost but poor classification performance. On the other hand, large values produce an expensive but efficient classification. Thus, a good choice is the value of $\phi = 0.6$ which produces a classifier with 34% of improvement with respect to the SCR scheme. With higher ϕ values, we have lower improvements on F1 and accuracy, but these cases can lead to higher communication costs and complexity.

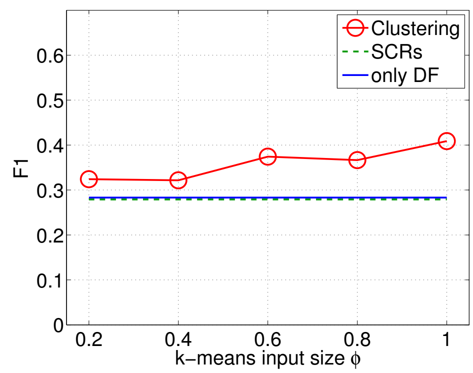


(a) F1 score.

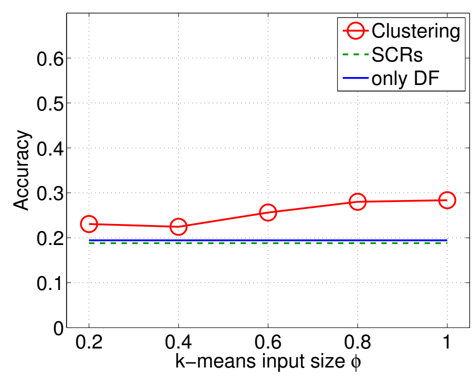


(b) Accuracy.

Fig. 7. Impact of number of clusters K.



(a) F1 score.



(b) Accuracy.

Fig. 8. Impact of k-means input size ϕ.

TABLE V
RESULTS OF THE T-TEST FOR F1 SCORE BETWEEN CLUSTERING AND SCRS SCHEMES WITH 95% CONFIDENCE LEVEL.

K	1	3	5	7	9	11
min	-2.9	4.1	4.1	2.6	-1.4	-1.7
max	4.1	17.0	14.8	14.2	11.8	11.94
mean	0.6	10.5	9.4	8.4	5.2	5.1
%	2.1	38.1	34.1	30.4	18.8	18.4

TABLE VI
RESULTS OF THE T-TEST FOR F1 SCORE BETWEEN CLUSTERING AND SCRS SCHEMES WITH 95% CONFIDENCE LEVEL.

ϕ	0.2	0.4	0.6	0.8	1.0
min	0.1	0.5	3.4	2.3	6.3
max	8.9	9.0	15.7	15.3	19.7
mean	4.5	4.3	9.5	8.8	13
%	16.1	15.3	34.2	31.4	46.5

V. Conclusion

We proposed an algorithm to improve classification rates when schemes of data fusion are used. We consider a noisy scenario in which an event observation is a combination of correlated and uncorrelated measurements. In these cases, a combination of value fusion and decision fusion is required. Our solution is based on k-means algorithm for clustering nodes with correlated measurements. Data averaging is performed in measurements inside a cluster and local decisions are combined by voting across different clusters.

The experiments show the efficiency of our solution for an application to classify anurans based on acoustic measurements. Our algorithm presented around 30% better performance than the SCR algorithm.

In the future, we intend to evaluate the impact of the number of k-means iterations, the effects of faults on the sensors, and the case of multitarget classification.

Acknowledgement

This work is partially financed by the National Council for Scientific and Technological Development (CNPq) and the Amazon State Research Foundation (FAPEAM), through the grants 575808/2008-0 (Revelar Project - CNPq) and 2210.UNI175.3532.03022011 (Anura Project - PRONEX 023/2009).

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cyirci, "Wireless sensor networks: A survey," *Computer Networks*, vol. 38, no. 4, pp. 393–422, March 2002.

[2] D. Li, K. Wong, Y. H. Hu, and A. Sayeed, "Detection, classification, and tracking of targets." *IEEE Signal Processing Magazine.*, vol. 19 (2), pp. 17–29, 2002.

[3] N. A. Lynch, *Distributed Algorithms*. San Francisco, CA: Morgan Kaufmann, 1997.

[4] E. F. Nakamura, A. A. F. Loureiro, and A. C. Frery, "Information fusion for wireless sensor networks: Methods, models, and classifications," *ACM Computing Surveys*, vol. 39(3), pp. 1–55, 2007.

[5] M. Duarte and Y.-h. Hu, "Vehicle classification in distributed sensor networks," *Journal of Parallel and Distributed Computing*, vol. 64(7), pp. 826–838, 2004.

[6] E. Kokiopoulou and P. Frossard, "Distributed classification of multiple observation sets by consensus," *IEEE Transactions on Signal Processing.*, vol. 59(1), pp. 104–114, 2011.

[7] J. Kotecha, V. Ramachandran, and A. Sayeed, "Distributed multitarget classification in wireless sensor networks," *IEEE Journal on Selected Areas in Communications.*, vol. 23 (4), pp. 703–713, 2005.

[8] B. Malhotra, I. Nikolaidis, and J. Harms, "Distributed classification of acoustic targets in wireless audio-sensor networks," *Computer Networks*, vol. 52, pp. 2582–2593, 2008.

[9] N. Katenka, E. Levina, and G. Michailidis, "Local vote decision fusion for target detection in wireless sensor networks," *IEEE Transactions on Signal Processing.*, vol. 56(1), pp. 329–338, 2008.

[10] T.-Y. Wang, Y. Han, B. Chen, and P. Varshney, "A combined decision fusion and channel coding scheme for distributed fault-tolerant classification in wireless sensor networks," *IEEE Transactions on Wireless Communications*, vol. 5(7), pp. 1695–1705, 2006.

[11] Z. Chair and P. Varshney, "Optimal data fusion in multiple sensor detection systems," *IEEE Transactions on Aerospace and Electronic Systems,*, vol. AES-22, no. 1, pp. 98–101, 1986.

[12] R. R. Brooks, P. Ramanathan, and A. M. Sayeed, "Distributed target classification and tracking in sensor networks," *Proceedings of the IEEE*, vol. 91(8), pp. 1163–1171, 2003.

[13] T. Clouqueur, K. K. Saluja, and P. Ramanathan, "Fault tolerance in collaborative sensor networks for target detection," *IEEE Transactions on Computers*, vol. 53, no. 3, pp. 320–333, 2004.

[14] B. Malhotra, I. Nikolaidis, and M. A. Nascimento, "Distributed and efficient classifiers for wireless audio-sensor networks," in *Proceedings of the 5th International Conference on Networked Sensing Systems (INSS'08).*, 2008, pp. 203–206.

[15] A. D'Costa and A. M. Sayeed, "Data versus decision fusion for distributed classification in sensor networks," in *Proceedings of the IEEE Military Communications Conference (MILCOM 2003)*, 2003, pp. 585–590 Vol.1.

[16] A. D'Costa, V. Ramachandran, and A. M. Sayeed, "Distributed classification of Gaussian space-time sources in wireless sensor networks," *IEEE Journal on Selected Areas in Communications*, vol. 22(6), pp. 1026–1036, 2004.

[17] J. Ding, S.-Y. Cheung, C.-W. Tan, and P. Varaiya, "Signal processing of sensor node data for vehicle detection," in *Proceedings of the 7th International IEEE Conference on Intelligent Transportation Systems*, 2004, pp. 70–75.

[18] S. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 28, no. 4, pp. 357–366, 1980.

[19] T. Mitchell, *Machine Learning*. McGraw-Hill Education, 1997.

[20] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. New York: Wiley, 2001.

[21] J. Colonna, A. Ribas, E. dos Santos, and E. Nakamura, "Feature subset selection for automatically classifying anuran calls using sensor networks," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, 2012.